



(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)



1-0027405

(51)⁷ G10L 19/16

(13) B

(21) 1-2015-01315

(22) 15/01/2014

(86) PCT/US2014/011672 15/01/2014

(87) WO 2014/113465 A1 24/07/2014

(30) 61/754,882 21/01/2013 US; 61/824,010 16/05/2013 US

(45) 25/02/2021 395

(43) 26/10/2015 331A

(73) DOLBY LABORATORIES LICENSING CORPORATION (US)

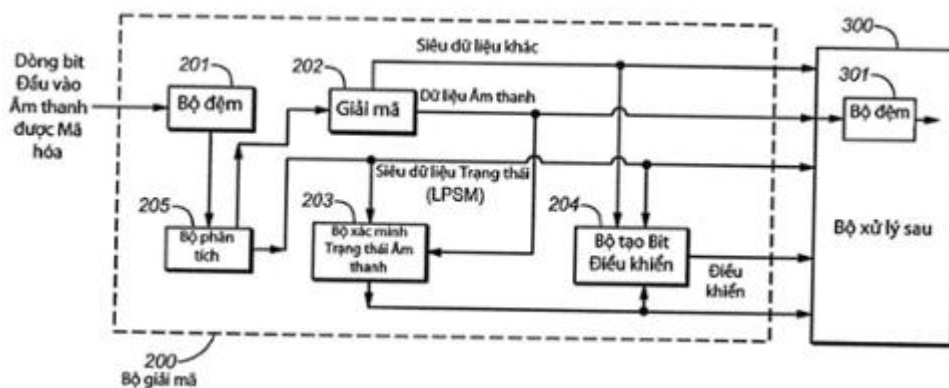
1275 Market Street, San Francisco, California 94103, United States of America.

(72) GRANT, Michael (AU); NORCROSS, Scott Gregory (CA); RIEDMILLER, Jeffrey (US); WARD, Michael (GB).

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

(54) ĐƠN VỊ XỬ LÝ ÂM THANH VÀ PHƯƠNG PHÁP GIẢI MÃ DÒNG BIT ÂM THANH ĐƯỢC MÃ HÓA

(57) Sáng chế đề cập đến đơn vị xử lý âm thanh bao gồm bộ nhớ đệm để lưu trữ ít nhất một khung của dòng bit âm thanh được mã hóa; bộ giải mã âm thanh được nối với bộ nhớ đệm để giải mã dữ liệu âm thanh; và bộ phân tích được ghép hoặc được tích hợp với bộ giải mã âm thanh để phân tích dòng bit âm thanh được mã hóa. Sáng chế cũng đề cập đến phương pháp giải mã dòng bit âm thanh được mã hóa.



Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến việc xử lý tín hiệu âm thanh, và cụ thể hơn đến việc mã hóa và giải mã các dòng bit dữ liệu âm thanh với siêu dữ liệu chỉ báo trạng thái xử lý âm lượng của nội dung âm thanh và vị trí của các biên chương trình âm thanh được chỉ báo bởi các dòng bit. Một số phương án theo sáng chế tạo ra hoặc giải mã dữ liệu âm thanh ở một trong các định dạng đã biết như AC-3, AC-3 Nâng cao hay E-AC-3, hoặc Dolby E.

Tình trạng kỹ thuật của sáng chế

Dolby, Dolby Digital, Dolby Digital Plus, và Dolby E là các nhãn hiệu của Dolby Laboratories Licensing Corporation. Dolby Laboratories cung cấp các phương án thực hiện độc quyền của AC-3 và E-AC-3 được biết đến lần lượt là Dolby Digital và Dolby Digital Plus.

Các đơn vị xử lý dữ liệu âm thanh thông thường vận hành theo phương thức mù và không chú ý đến lịch sử xử lý dữ liệu âm thanh mà xảy ra trước khi dữ liệu được nhận. Phương thức này có thể hoạt động trong khuôn khổ xử lý trong đó một thực thể đơn nhất thực hiện tất cả việc xử lý và mã hóa dữ liệu âm thanh cho nhiều loại thiết bị kết xuất đa phương tiện đích trong khi thiết bị kết xuất đa phương tiện đích thực hiện tất cả việc giải mã và kết xuất dữ liệu âm thanh đã mã hóa. Tuy nhiên, việc xử lý mù này không hoạt động tốt (hoặc hoàn toàn không hoạt động) trong các trường hợp mà nhiều đơn vị xử lý âm thanh nằm phân tán trên khắp một mạng phức hoặc được bố trí trước sau (chẳng hạn, theo chuỗi) và được mong đợi sẽ thực hiện một cách tối ưu các dạng xử lý âm thanh tương ứng của chúng. Ví dụ, một số dữ liệu âm thanh có thể được mã hóa cho các hệ thống đa phương tiện hiệu suất cao và có thể phải được chuyển đổi về dạng rút gọn phù hợp với thiết bị di động theo chuỗi xử lý phương tiện. Theo đó, đơn vị xử lý âm thanh có thể không cần thiết thực hiện một dạng xử lý đối với dữ liệu âm thanh mà đã được thực hiện. Chẳng hạn, đơn vị cân chỉnh âm lượng có thể thực hiện việc xử lý trên một đoạn âm thanh đầu vào, bất kể việc cân chỉnh âm lượng giống hoặc tương tự trước đó đã được thực hiện trên đoạn âm thanh đầu vào này hay chưa. Kết quả là, đơn vị cân chỉnh âm lượng có thể thực hiện việc cân chỉnh ngay cả khi việc đó không cần thiết. Việc xử lý

không cần thiết này cũng có thể gây ra sự suy giảm và/hoặc sự loại bỏ các đặc tính cụ thể trong khi kết xuất nội dung của dữ liệu âm thanh.

Dòng dữ liệu âm thanh thông thường bao gồm cả nội dung âm thanh (ví dụ, một hoặc nhiều kênh nội dung âm thanh) và siêu dữ liệu chỉ báo ít nhất một đặc tính của nội dung âm thanh. Ví dụ, trong dòng bit AC-3 có một số tham số siêu dữ liệu âm thanh được dành riêng để sử dụng trong việc thay đổi âm thanh của chương trình được phát đến môi trường nghe. Một trong các tham số siêu dữ liệu là tham số DIALNORM, dùng để chỉ báo mức thoại trung bình trong chương trình âm thanh, và được sử dụng để xác định mức tín hiệu phát lại âm thanh.

Trong khi phát lại dòng bit bao gồm trình tự các đoạn chương trình âm thanh khác nhau (mỗi đoạn có tham số DIALNORM khác nhau), bộ giải mã AC-3 sử dụng tham số DIALNORM của mỗi đoạn để thực hiện một kiểu xử lý âm lượng mà ở đó nó điều biến mức phát lại hoặc âm lượng sao cho âm lượng thoại nghe thấy của trình tự các đoạn là ở một mức độ nhất quán. Mỗi đoạn âm thanh được mã hóa (mục) theo trình tự của các mục âm thanh được mã hóa sẽ (nói chung) có tham số DIALNORM khác nhau, và bộ giải mã sẽ đo mức độ của mỗi mục dữ liệu sao cho mức phát lại hoặc âm lượng thoại của mỗi mục là giống nhau hoặc rất tương tự nhau, mặc dù điều này có thể đòi hỏi phải áp dụng các lượng khuếch đại khác nhau cho một số mục khác nhau trong số các mục trong khi phát lại.

DIALNORM thường được thiết lập bởi người dùng, và không được tạo ra tự động, mặc dù có trị số DIALNORM mặc định nếu không có trị số nào được thiết lập bởi người dùng. Ví dụ, bộ tạo nội dung có thể thực hiện các phép đo âm lượng bằng thiết bị ở ngoài bộ mã hóa AC-3 và sau đó truyền kết quả (chỉ báo âm lượng thoại dạng tiếng nói của chương trình âm thanh) đến bộ mã hóa để thiết lập trị số DIALNORM. Do đó, có sự dựa vào bộ tạo nội dung để thiết lập chính xác tham số DIALNORM.

Có một số lý do khác nhau giải thích tại sao tham số DIALNORM trong dòng bit AC-3 có thể không chính xác. Thứ nhất, mỗi bộ mã hóa AC-3 có một trị số DIALNORM mặc định được sử dụng trong khi tạo ra dòng bit nếu trị số DIALNORM không được thiết lập bởi bộ tạo nội dung. Trị số mặc định này có thể khác đáng kể so với mức âm lượng thoại thực của âm thanh. Thứ hai, ngay cả khi bộ tạo nội dung đo âm lượng và thiết lập trị

số DIALNORM cho phù hợp, thì có thể đã sử dụng một thuật toán đo hoặc thước đo âm lượng mà không phù hợp với phương pháp đo âm lượng AC-3 được đề xuất, kết quả là cho ra trị số DIALNORM không chính xác. Thứ ba, ngay cả khi dòng bit AC-3 đã được tạo ra với trị số DIALNORM được đo và thiết lập chính xác bởi bộ tạo nội dung, thì nó có thể đã bị thay đổi thành một trị số không đúng trong quá trình truyền và/hoặc lưu trữ dòng bit. Ví dụ, trong các ứng dụng phát sóng truyền hình, các dòng bit AC-3 thường được giải mã, điều biến và sau đó mã hóa lại bằng cách sử dụng thông tin siêu dữ liệu DIALNORM không chính xác. Do đó, trị số DIALNORM có trong dòng bit AC-3 có thể không đúng hoặc không chính xác và do đó có thể gây tác động tiêu cực đến chất lượng của trải nghiệm nghe.

Hơn nữa, tham số DIALNORM không chỉ báo trạng thái xử lý âm lượng của dữ liệu âm thanh tương ứng (ví dụ (các) kiểu xử lý âm lượng nào đã được thực hiện trên dữ liệu âm thanh). Cho đến trước khi có sáng chế này, dòng bit âm thanh không bao gồm siêu dữ liệu, chỉ báo trạng thái xử lý âm lượng (ví dụ, (các) kiểu xử lý âm lượng đã được áp dụng), nội dung âm thanh của dòng bit âm thanh hoặc trạng thái xử lý âm lượng và âm lượng của nội dung âm thanh của dòng bit, ở định dạng của kiểu đã được mô tả trong phần bộc lộ này. Siêu dữ liệu trạng thái xử lý âm lượng ở định dạng như vậy là hữu dụng để tạo điều kiện thuận lợi cho việc xử lý âm lượng thích ứng của dòng bit âm thanh và/hoặc xác minh hiệu lực của trạng thái xử lý âm lượng và âm lượng của nội dung âm thanh theo cách đặc biệt hiệu quả.

Mặc dù sáng chế không bị giới hạn ở việc sử dụng với dòng bit AC-3, dòng bit E-AC-3, hoặc dòng bit Dolby E, nhưng để thuận tiện thì sáng chế sẽ được mô tả theo các phương án mà ở đó nó tạo ra, giải mã, hoặc theo cách khác xử lý dòng bit này mà bao gồm siêu dữ liệu trạng thái xử lý âm lượng.

Dòng bit được mã hóa AC-3 bao gồm siêu dữ liệu và một đến sáu kênh nội dung âm thanh. Nội dung âm thanh là dữ liệu âm thanh đã được nén nhờ sử dụng kỹ thuật mã hóa âm thanh cảm quan. Siêu dữ liệu bao gồm một số tham số siêu dữ liệu âm thanh được dự định dùng để thay đổi âm thanh của chương trình được phát đến môi trường nghe.

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

ATSC Standard A52/A: Digital Audio Compression Standard (AC-3), Revision A, Advanced Television Systems Committee, ngày 20 tháng 08 năm 2001; và

Các Patent Mỹ số 5,583,962; 5,632,005; 5,633,981; 5,727,119 và 6,021,386, tất cả đều được đưa vào đây bằng cách viện dẫn toàn bộ.

Chi tiết về việc mã hóa Dolby Digital Plus (E-AC-3) được trình bày trong "Introduction to Dolby Digital Plus, an Enhancement to the Dolby Digital Coding System," AES Convention Paper 6196, Hội nghị AES lần thứ 117, ngày 28 tháng 10 năm 2004.

Chi tiết về việc mã hóa Dolby E được trình bày trong "Efficient Bit Allocation, Quantization, and Coding in an Audio Distribution System", AES Preprint 5068, Hội nghị AES lần thứ 107, tháng 08 năm 1999 và "Professional Audio Coder Optimized for Use with Video", AES Preprint 5033, Hội nghị AES lần thứ 107, tháng 08 năm 1999.

Mỗi khung dòng bit âm thanh được mã hóa AC-3 chứa nội dung âm thanh và siêu dữ liệu cho 1536 mẫu âm thanh kỹ thuật số. Đối với tỉ lệ lấy mẫu là 48 kHz, điều này thể hiện 32 mili giây âm thanh kỹ thuật số hoặc tỷ lệ 31,25 khung trên một giây âm thanh.

Mỗi khung dòng bit âm thanh được mã hoá E-AC-3 chứa nội dung âm thanh và siêu dữ liệu cho 256, 512, 768 hoặc 1536 mẫu âm thanh kỹ thuật số, tùy thuộc vào việc xem khung chứa lần lượt một, hai, ba hay sáu khối dữ liệu âm thanh. Với tỉ lệ lấy mẫu là 48 kHz, điều này thể hiện lần lượt 5,333, 10,667, 16 hoặc 32 mili giây âm thanh kỹ thuật số hoặc tỷ lệ lần lượt là 189,9, 93,75, 62,5 hoặc 31,25 khung trên một giây âm thanh.

Như được thể hiện trên Fig.4, mỗi khung AC-3 được chia thành các phần (các đoạn), bao gồm: phần Thông tin Đồng bộ hoá (Synchronization Information - SI) chứa (như được thể hiện trên Fig.5) lệnh đồng bộ hoá (synchronization word - SW) và lệnh thứ nhất trong hai lệnh sửa lỗi (CRC1); phần Thông tin Dòng bit (Bitstream Information - BSI) chứa hầu hết siêu dữ liệu; sáu Khối Âm thanh (từ AB0 đến AB5) chứa nội dung âm thanh được nén dữ liệu (và cũng có thể bao gồm siêu dữ liệu); đoạn bit thừa (W) chứa bit không sử dụng bất kỳ còn lại sau khi nội dung âm thanh được nén; phần thông tin Phụ trợ

(Auxiliary - AUX) có thể chứa nhiều siêu dữ liệu hơn; và lệnh thứ hai trong số hai lệnh sửa lỗi (CRC2). Đoạn bit thừa (W) cũng có thể được gọi là “trường bỏ qua”.

Như được thể hiện trên Fig.7, mỗi khung E-AC-3 được chia thành các phần (các đoạn), bao gồm: phần Thông tin Đồng bộ hoá (Synchronization Information - SI) chứa (như được thể hiện trên Fig.5) lệnh đồng bộ hoá (synchronization word - SW); phần Thông tin Dòng bit (Bitstream Information - BSI) chứa hầu hết siêu dữ liệu; từ một đến sáu Khối Âm thanh (từ AB0 đến AB5) chứa nội dung âm thanh được nén dữ liệu (và cũng có thể bao gồm siêu dữ liệu); đoạn bit thừa (W) chứa bit không sử dụng bất kỳ còn lại sau khi nội dung âm thanh được nén (mặc dù chỉ một đoạn bit thừa được thể hiện, nhưng một đoạn bit thừa khác thường theo sau mỗi khối âm thanh); phần thông tin Phụ trợ (Auxiliary - AUX) có thể chứa nhiều siêu dữ liệu hơn; và một lệnh sửa lỗi (CRC). Đoạn bit thừa (W) cũng có thể được gọi là “trường bỏ qua”.

Trong dòng bit AC-3 (hoặc E-AC-3) có một số tham số siêu dữ liệu âm thanh mà đặc biệt chỉ dùng để thay đổi âm thanh của chương trình được phát đến môi trường nghe. Một trong các tham số siêu dữ liệu là tham số DIALNORM, mà được bao gồm trong đoạn BSI.

Như được thể hiện trên Fig.6, đoạn BSI của khung AC-3 bao gồm tham số năm bit (“DIALNORM”) chỉ báo trị số DIALNORM của chương trình. Một tham số năm bit (“DIALNORM2”) chỉ báo trị số DIALNORM của chương trình âm thanh thứ hai được mang trong cùng một khung AC-3 được bao gồm nếu chế độ mã hoá âm thanh (“acmod”) của khung AC-3 là “0”, chỉ báo rằng cấu hình kênh kép-đơn hoặc “1+1” đang được sử dụng.

Đoạn BSI còn bao gồm cờ (“addbsie”) chỉ báo sự có mặt (hoặc vắng mặt) của thông tin dòng bit bổ sung theo sau bit “addbsie”, tham số (“addbsil”) chỉ báo độ dài của thông tin dòng bit bổ sung bất kỳ theo sau trị số “addbsil”, và lên đến 64 bit của thông tin dòng bit bổ sung (“addbsi”) theo sau trị số “addbsil”.

Đoạn BSI bao gồm các trị số siêu dữ liệu khác không được thể hiện cụ thể trên Fig.6.

Bản chất kỹ thuật của sáng chế

Theo một nhóm các phương án, sáng chế đề cập đến đơn vị xử lý âm thanh bao gồm bộ nhớ đệm, bộ giải mã âm thanh, và bộ phân tích. Bộ nhớ đệm lưu trữ ít nhất một khung của dòng bit âm thanh được mã hoá. Dòng bit âm thanh được mã hoá bao gồm dữ liệu âm thanh và bộ chứa siêu dữ liệu. Bộ chứa siêu dữ liệu bao gồm phần đầu, một hoặc nhiều tải trọng siêu dữ liệu, và dữ liệu bảo vệ. Phần đầu bao gồm một từ đồng bộ xác định sự bắt đầu của bộ chứa. Một hoặc nhiều tải trọng siêu dữ liệu mô tả chương trình âm thanh kết hợp với dữ liệu âm thanh. Dữ liệu bảo vệ được đặt sau một hoặc nhiều tải trọng siêu dữ liệu. Dữ liệu bảo vệ cũng có thể được sử dụng để xác minh tính nhất quán của bộ chứa siêu dữ liệu và một hoặc nhiều tải trọng bên trong bộ chứa siêu dữ liệu. Bộ giải mã âm thanh được nối với bộ nhớ đệm và có khả năng giải mã dữ liệu âm thanh. Bộ phân tích được nối hoặc được tích hợp với bộ giải mã âm thanh và có khả năng phân tích bộ chứa siêu dữ liệu.

Theo các phương án thông thường, phương pháp bao gồm bước nhận dòng bit âm thanh được mã hoá trong đó dòng bit âm thanh được mã hoá được phân đoạn thành một hoặc nhiều khung. Dữ liệu âm thanh được trích ra từ dòng bit âm thanh được mã hoá, cùng với bộ chứa siêu dữ liệu. Bộ chứa siêu dữ liệu bao gồm phần đầu theo sau bởi một hoặc nhiều tải trọng siêu dữ liệu được theo sau bởi dữ liệu bảo vệ. Cuối cùng, tính nhất quán của bộ chứa và một hoặc nhiều tải trọng siêu dữ liệu được xác minh thông qua việc sử dụng dữ liệu bảo vệ. Một hoặc nhiều tải trọng siêu dữ liệu có thể bao gồm tải trọng âm lượng chương trình mà chứa dữ liệu chỉ báo âm lượng đo được của chương trình âm thanh kết hợp với dữ liệu âm thanh.

Tải trọng của siêu dữ liệu về âm lượng chương trình, còn gọi là siêu dữ liệu về trạng thái xử lý âm lượng (loudness processing state metadata- “LPSM”), được nhúng vào dòng bit âm thanh theo các phương án điển hình của sáng chế có thể được xác nhận và phê duyệt, ví dụ, để cho phép các thực thể điều chỉnh âm lượng xác minh xem âm lượng của một chương trình cụ thể đã nằm trong phạm vi quy định hay chưa và rằng chính dữ liệu âm thanh tương ứng chưa được điều biến (qua đó đảm bảo tuân thủ các quy định áp dụng). Trị số âm lượng được bao gồm trong khối dữ liệu chứa siêu dữ liệu về trạng thái xử lý âm lượng có thể được đọc ra để xác minh điều này, thay cho việc tính toán âm lượng một lần nữa. Để đáp lại LPSM, cục điều chỉnh có thể xác định rằng nội dung âm thanh tương ứng là tuân thủ (như được chỉ báo bởi LPSM) theo các yêu cầu điều

chính và/hoặc luật định về âm lượng (ví dụ, các quy định ban hành theo Luật giảm âm lượng quảng cáo thương mại, còn được gọi là Luật “CALM”) mà không cần tính toán âm lượng của nội dung âm thanh.

Các phép đo âm lượng mà được yêu cầu để tuân thủ đúng theo các yêu cầu điều chỉnh và/hoặc luật định về âm lượng (ví dụ, các quy định ban hành theo Luật CALM) là dựa trên âm lượng chương trình được tích hợp. Âm lượng chương trình được tích hợp đòi hỏi rằng việc đo âm lượng, hoặc của mức thoại hoặc mức phối âm toàn bộ, được thực hiện trên toàn bộ chương trình âm thanh. Do đó, để thực hiện các phép đo âm lượng chương trình (ví dụ, ở các giai đoạn khác nhau trong chuỗi phát sóng) nhằm xác minh sự tuân thủ theo các yêu cầu pháp lý điển hình, các phép đo cần được thực hiện với sự nhận biết dữ liệu (và siêu dữ liệu) âm thanh nào xác định toàn bộ chương trình âm thanh, và điều này thường đòi hỏi sự nhận biết về vị trí của điểm bắt đầu và kết thúc của chương trình (ví dụ, trong quá trình xử lý dòng bit chỉ báo trình tự của các chương trình âm thanh).

Theo các phương án điển hình của sáng chế, dòng bit âm thanh được mã hoá chỉ báo ít nhất một chương trình âm thanh (ví dụ, trình tự của các chương trình âm thanh), và siêu dữ liệu biên chương trình và LPSM được bao gồm trong dòng bit cho phép thiết lập lại phép đo âm lượng chương trình ở cuối chương trình và qua đó cho phép đo âm lượng chương trình được tích hợp một cách tự động. Các phương án điển hình của sáng chế bao gồm siêu dữ liệu biên chương trình trong dòng bit âm thanh được mã hóa một cách hiệu quả, cho phép xác định chính xác và mạnh mẽ ít nhất một biên giữa các chương trình âm thanh liên tiếp được chỉ báo bởi dòng bit. Các phương án điển hình cho phép xác định chính xác và mạnh mẽ biên chương trình có nghĩa là chúng cho phép xác định biên chương trình chính xác ngay cả trong trường hợp các dòng bit chỉ báo các chương trình khác nhau được ghép lại với nhau (để tạo ra dòng bit theo sáng chế) theo cách cắt một hoặc cả hai dòng bit được ghép (và do đó loại bỏ siêu dữ liệu biên chương trình mà được bao gồm trong ít nhất một trong số các dòng bit đã được ghép trước đó).

Theo các phương án điển hình, siêu dữ liệu biên chương trình trong khung của dòng bit theo sáng chế là cờ biên chương trình chỉ báo số lượng khung. Thông thường, cờ chỉ báo số lượng khung giữa khung hiện thời (khung mà bao gồm cờ) và biên chương trình (điểm bắt đầu hoặc kết thúc của chương trình âm thanh hiện thời). Theo một số

phương án được ưu tiên, các cờ biên chương trình được chèn vào theo cách đối xứng, hiệu quả ở điểm bắt đầu và kết thúc của mỗi đoạn dòng bit mà chỉ báo một chương trình duy nhất (tức là, ở các khung xuất hiện trong một số khung đã định trước sau điểm bắt đầu của đoạn, và ở các khung xuất hiện trong một số khung đã định trước trước điểm kết thúc của đoạn), sao cho khi hai đoạn dòng bit này được ghép nối vào nhau (để chỉ báo trình tự của hai chương trình), thì siêu dữ liệu biên chương trình có thể có mặt (ví dụ, một cách đối xứng) trên cả hai phía của biên giữa hai chương trình.

Đề hạn chế sự gia tăng tốc độ dữ liệu là kết quả của việc đưa siêu dữ liệu biên chương trình vào trong dòng bit âm thanh được mã hóa (mà có thể chỉ báo một chương trình âm thanh hoặc trình tự của chương trình âm thanh), theo các phương án điển hình các cờ biên chương trình chỉ được chèn vào một tập hợp con các khung của dòng bit. Thông thường, tốc độ chèn cờ biên là một hàm số không tăng của khoảng cách tăng của mỗi khung trong số các khung của dòng bit (trong đó cờ được chèn) từ biên chương trình gần nhất với mỗi khung nêu trên, trong đó “tốc độ chèn cờ biên” thể hiện tỷ lệ trung bình của số lượng khung (chỉ báo chương trình) mà bao gồm cờ biên chương trình với số lượng khung (chỉ báo chương trình) mà không bao gồm cờ biên chương trình, trong đó giá trị trung bình là giá trị trung bình động trên một số (ví dụ, số tương đối nhỏ) khung liên tiếp của dòng bit âm thanh được mã hóa. Theo một nhóm các phương án, tốc độ chèn cờ biên là một hàm số giảm theo lôga của khoảng cách ngày càng tăng (của mỗi vị trí chèn cờ) từ biên chương trình gần nhất, và đối với mỗi khung chứa cờ mà bao gồm một trong số các cờ, độ lớn của cờ trong khung chứa cờ nêu trên bằng hoặc lớn hơn độ lớn của mỗi cờ trong khung nằm gần với biên chương trình gần nhất so với khung chứa cờ nêu trên (tức là, độ lớn của cờ biên chương trình trong mỗi khung chứa cờ là một hàm số không giảm của khoảng cách tăng của khung chứa cờ nêu trên từ biên chương trình gần nhất).

Một khía cạnh khác của sáng chế là đơn vị xử lý âm thanh (audio processing unit - APU) được tạo cấu hình để thực hiện phương án bất kỳ của phương pháp theo sáng chế. Theo một nhóm các phương án khác, sáng chế đề xuất APU bao gồm bộ nhớ đệm (bộ đệm) mà lưu trữ (ví dụ, theo cách không chuyển tiếp) ít nhất một khung của dòng bit âm thanh được mã hóa mà đã được tạo ra bởi phương án bất kỳ của phương pháp theo sáng chế. Các ví dụ của APU bao gồm, nhưng không giới hạn ở bộ mã hóa (ví dụ, bộ chuyển

mã), bộ giải mã, bộ mã hóa-giải mã, hệ thống xử lý trước (bộ xử lý trước), hệ thống xử lý sau (bộ xử lý sau), hệ thống xử lý dòng bit âm thanh, và sự kết hợp các thành phần đó.

Theo một nhóm các phương án khác, sáng chế đề xuất đơn vị xử lý âm thanh (APU) được tạo cấu hình để tạo ra dòng bit âm thanh được mã hóa chứa các đoạn dữ liệu âm thanh và đoạn siêu dữ liệu, trong đó các đoạn dữ liệu âm thanh chỉ báo dữ liệu âm thanh, và mỗi đoạn trong số ít nhất các đoạn siêu dữ liệu bao gồm siêu dữ liệu trạng thái xử lý âm lượng (loudness processing state metadata - LPSM) và cũng có thể bao gồm siêu dữ liệu biên chương trình. Thông thường, ít nhất một đoạn siêu dữ liệu này trong khung của dòng bit bao gồm ít nhất một đoạn LPSM chỉ báo kiểu xử lý âm lượng thứ nhất đã được thực hiện trên dữ liệu âm thanh của khung hay chưa (tức là, dữ liệu âm thanh trong ít nhất một đoạn dữ liệu âm thanh của khung), và ít nhất một đoạn khác của LPSM chỉ báo âm lượng của ít nhất một số dữ liệu âm thanh của khung (ví dụ, âm lượng thoại của ít nhất một số dữ liệu âm thanh của khung mà chỉ báo thoại). Theo một phương án thuộc nhóm này, APU là bộ mã hóa được tạo cấu hình để mã hóa âm thanh đầu vào nhằm tạo ra âm thanh được mã hóa, và các đoạn dữ liệu âm thanh bao gồm âm thanh được mã hóa. Theo các phương án điển hình thuộc nhóm này, mỗi đoạn siêu dữ liệu có một định dạng ưu tiên sẽ được mô tả dưới đây.

Theo một số phương án, mỗi đoạn siêu dữ liệu của dòng bit được mã hóa (dòng bit AC-3 hoặc dòng bit E-AC-3 theo một số phương án) mà bao gồm LPSM (ví dụ, LPSM và siêu dữ liệu biên chương trình) được bao gồm trong bit thừa của đoạn trường bỏ qua của khung của dòng bit (ví dụ, đoạn bit thừa W theo kiểu được thể hiện trên Fig.4 hoặc Fig.7). Theo các phương án khác, mỗi đoạn siêu dữ liệu của dòng bit được mã hóa (dòng bit AC-3 hoặc dòng bit E-AC-3 theo một số phương án) mà bao gồm LPSM (ví dụ, LPSM và siêu dữ liệu biên chương trình) được bao gồm dưới dạng thông tin dòng bit bổ sung trong trường “addbsi” của đoạn Thông tin Dòng bit (“BSI”) của khung của dòng bit hoặc trong trường auxdata (ví dụ, đoạn AUX theo kiểu được thể hiện trên Fig.4 hoặc Fig.7) ở cuối khung của dòng bit. Mỗi đoạn siêu dữ liệu bao gồm LPSM có thể có định dạng được nêu ở đây có tham chiếu đến các Bảng 1 và 2 dưới đây (tức là, nó bao gồm các bộ phận chính được nêu trong Bảng 1 hoặc biến đổi trên đó, theo sau bởi ID tải trọng (xác định siêu dữ liệu là LPSM) và các trị số độ lớn tải trọng, theo sau bởi tải trọng (dữ liệu LPSM mà có định dạng như được biểu thị trong Bảng 2, hoặc định dạng như được

biểu thị trong sự biến đổi trên Bảng 2 được mô tả dưới đây). Theo một số phương án, khung có thể bao gồm một hoặc hai đoạn siêu dữ liệu, mỗi đoạn bao gồm LPSM, và nếu khung bao gồm hai đoạn siêu dữ liệu, một đoạn có thể có mặt trong trường addbsi của khung và đoạn còn lại có mặt trong trường AUX của khung.

Theo một nhóm các phương án, sáng chế đề xuất phương pháp bao gồm các bước mã hóa dữ liệu âm thanh để tạo ra dòng bit âm thanh được mã hóa AC-3 hoặc E-AC-3, bằng cách đưa vào trong một đoạn siêu dữ liệu (của ít nhất một khung của dòng bit) LPSM và siêu dữ liệu biên chương trình và cũng tùy ý các siêu dữ liệu khác cho chương trình âm thanh mà khung thuộc vào đó. Theo một số phương án, mỗi đoạn siêu dữ liệu này được bao gồm trong trường addbsi của khung, hoặc trường auxdata của khung. Theo các phương án khác, mỗi đoạn siêu dữ liệu này được bao gồm trong đoạn bit thừa của khung. Theo một số phương án, mỗi đoạn siêu dữ liệu mà chứa LPSM và siêu dữ liệu biên chương trình chứa phần đầu chính (và cũng có thể bao gồm các phần tử chính bổ sung khác), và sau phần đầu chính (hoặc phần đầu chính và các phần tử chính khác) đoạn tải trọng LPSM (hoặc bộ chứa) có định dạng sau đây:

phần đầu, thường bao gồm ít nhất một trị số xác định (ví dụ, phiên bản định dạng LPSM, độ dài, chu trình, số lượng, và các trị số liên kết dòng phụ, như được thể hiện trong Bảng 2 được đưa ra trong bảng mô tả này), và

sau phần đầu, là LPSM và siêu dữ liệu biên chương trình. Siêu dữ liệu biên chương trình có thể bao gồm số lượng khung biên chương trình, và trị số mã (ví dụ, trị số “offset_exist” (“tồn tại_xê dịch”)) chỉ báo liệu khung chỉ bao gồm số lượng đếm khung biên chương trình hoặc bao gồm cả số lượng đếm khung biên chương trình và (trong một số trường hợp) trị số xê dịch. LPSM có thể bao gồm:

ít nhất một trị số chỉ báo thoại chỉ báo dữ liệu âm thanh tương ứng có chỉ báo thoại hay không chỉ báo thoại (ví dụ, kênh dữ liệu âm thanh tương ứng nào chỉ báo thoại). (Các) trị số chỉ báo thoại có thể chỉ báo thoại có mặt hay không dưới dạng kết hợp bất kỳ của, hoặc tất cả, các kênh dữ liệu âm thanh tương ứng;

ít nhất một trị số tuân thủ quy định về âm lượng chỉ báo liệu dữ liệu âm thanh tương ứng có tuân thủ theo một nhóm các quy định về âm lượng được chỉ định hay không;

ít nhất một trị số xử lý âm lượng chỉ báo ít nhất một kiểu xử lý âm lượng đã được thực hiện trên dữ liệu âm thanh tương ứng; và

ít nhất một trị số âm lượng chỉ báo ít nhất một đặc tính âm lượng (ví dụ, âm lượng đỉnh hoặc trung bình) của dữ liệu âm thanh tương ứng.

Theo các phương án khác, dòng bit được mã hóa là dòng bit không phải là dòng bit AC-3 hoặc dòng bit E-AC-3, và mỗi đoạn siêu dữ liệu mà bao gồm LPSM (và cũng có thể bao gồm siêu dữ liệu biên chương trình) được bao gồm trong một đoạn (hoặc trường hoặc khe) của dòng bit được dự trữ để lưu trữ dữ liệu bổ sung. Mỗi đoạn siêu dữ liệu bao gồm LPSM có thể có định dạng tương tự hoặc giống với loại được chỉ rõ ở đây có tham chiếu đến các Bảng 1 và 2 dưới đây (tức là, nó bao gồm các phần tử chính tương tự hoặc giống với các phần tử được chỉ rõ trong Bảng 1, theo sau bởi ID tải trọng (xác định siêu dữ liệu là LPSM) và các trị số độ lớn tải trọng, theo sau bởi tải trọng (dữ liệu LPSM mà có định dạng tương tự hoặc giống với định dạng được thể hiện trong Bảng 2 hoặc biến thể trên Bảng 2 được mô tả ở đây).

Theo một số phương án, dòng bit được mã hóa bao gồm trình tự của các khung, mỗi khung bao gồm đoạn Thông tin Dòng bit (“BSI”) bao gồm trường “addbsi” (đôi khi còn được gọi là đoạn hoặc khe), và trường dữ liệu auxdata hoặc khe (ví dụ, dòng bit được mã hóa là dòng bit AC-3 hoặc dòng bit E-AC-3), và bao gồm các đoạn dữ liệu âm thanh (ví dụ, các đoạn từ AB0 đến AB5 của khung được thể hiện trên Fig.4) và các đoạn siêu dữ liệu, trong đó các đoạn dữ liệu âm thanh chỉ báo dữ liệu âm thanh, và ít nhất mỗi đoạn trong số các đoạn siêu dữ liệu bao gồm siêu dữ liệu trạng thái xử lý âm lượng (LPSM) và cũng có thể bao gồm siêu dữ liệu biên chương trình. LPSM có mặt trong dòng bit dưới định dạng sau. Mỗi đoạn siêu dữ liệu mà bao gồm LPSM được bao gồm trong trường “addbsi” của đoạn BSI của khung của dòng bit, hoặc trong trường auxdata của khung của dòng bit, hoặc trong đoạn bit thừa của khung của dòng bit. Mỗi đoạn siêu dữ liệu bao gồm LPSM bao gồm đoạn tải trọng LPSM (hoặc bộ chứa) có định dạng sau:

phần đầu (thường bao gồm ít nhất một trị số xác định, ví dụ, phiên bản định dạng LPSM, độ dài, chu trình, đếm, và các trị số liên kết dòng phụ được thể hiện trong Bảng 2 dưới đây); và

sau phần đầu, là LPSM và cũng có thể là siêu dữ liệu biên chương trình. Siêu dữ liệu biên chương trình có thể bao gồm số lượng đếm khung biên chương trình, và trị số mã (ví dụ, trị số “tồn tại_xê dịch”) chỉ báo liệu khung chỉ bao gồm số lượng đếm khung biên chương trình hay cả số lượng đếm khung biên chương trình và (trong một số trường hợp) trị số xê dịch. LPSM có thể bao gồm:

ít nhất một trị số chỉ báo thoại (ví dụ, tham số “(các) kênh thoại” trên Bảng 2) thể hiện dữ liệu âm thanh tương ứng chỉ báo thoại hoặc không chỉ báo thoại (ví dụ, kênh dữ liệu âm thanh tương ứng nào chỉ báo thoại). (Các) trị số chỉ báo thoại có thể chỉ báo thoại có mặt hay không dưới dạng kết hợp bất kỳ của, hoặc hoặc tất cả, các kênh dữ liệu âm thanh tương ứng;

ít nhất một trị số tuân thủ quy định âm lượng (ví dụ, tham số “Loại Quy định về Âm lượng” trên Bảng 2) chỉ báo liệu dữ liệu âm thanh tương ứng có tuân thủ theo nhóm các quy định về âm lượng được chỉ định hay không;

ít nhất một trị số xử lý âm lượng (ví dụ, một hoặc nhiều tham số “cờ hiệu chỉnh âm lượng có cổng thoại,” “Kiểu hiệu chỉnh âm lượng,” trên Bảng 2) chỉ báo ít nhất một kiểu xử lý âm lượng đã được thực hiện trên dữ liệu âm thanh tương ứng; và

ít nhất một trị số âm lượng (ví dụ, một hoặc nhiều tham số “Âm lượng có cổng theo ITU,” “Âm lượng có cổng tiếng nói ITU,” “Âm lượng ngắn 3 giây ITU (EBU 3341),” và “Mức thật của Đỉnh” trong Bảng 2) chỉ báo ít nhất một đặc tính âm lượng (ví dụ, âm lượng đỉnh hoặc trung bình) của dữ liệu âm thanh tương ứng.

Theo phương án bất kỳ của sáng chế mà dự định, sử dụng, hoặc tạo ra ít nhất một trị số âm lượng chỉ báo dữ liệu âm thanh tương ứng, (các) trị số âm lượng có thể chỉ báo ít nhất một đặc tính đo âm lượng được sử dụng để xử lý âm lượng và/hoặc dải động của dữ liệu âm thanh.

Theo một số phương án thực hiện, mỗi đoạn siêu dữ liệu trong trường “addbsi”, hoặc trường auxdata, hoặc đoạn bit thừa, của khung của dòng bit có định dạng sau:

phần đầu chính (thường bao gồm một lệnh đồng bộ hóa xác định điểm bắt đầu của một đoạn siêu dữ liệu, theo sau bởi các trị số xác định, ví dụ, phiên bản phần tử chính, độ dài, và chu trình, số phần tử mở rộng, và các trị số liên kết dòng phụ được thể hiện trong Bảng 1 dưới đây); và

sau phần đầu chính, ít nhất một trị số bảo vệ (ví dụ, các trị số Vân tay Âm thanh và giải thuật tiêu hóa HMAC (HMAC digest), trong đó giải thuật tiêu hóa HMAC có thể là giải thuật tiêu hóa HMAC 256 bit (sử dụng thuật toán SHA-2) được tính toán trên dữ liệu âm thanh, phần tử chính chính, và tất cả các phần tử mở rộng, của toàn bộ khung, như được chỉ báo trong Bảng 1) hữu dụng cho ít nhất một trong các việc giải mã, xác thực, hoặc xác minh ít nhất một trong số các siêu dữ liệu trạng thái xử lý âm lượng hoặc dữ liệu âm thanh tương ứng); và

cũng sau phần đầu chính, nếu đoạn siêu dữ liệu bao gồm LPSM, xác định tải trọng LPSM (“ID”) và các trị số độ lớn tải trọng LPSM mà xác định siêu dữ liệu theo sau là tải trọng LPSM và chỉ báo độ lớn của tải trọng LPSM. Đoạn tải trọng LPSM (tốt hơn là có định dạng được quy định như trên) theo sau ID tải trọng LPSM và các trị số độ lớn tải trọng LPSM.

Theo một số phương án của kiểu được mô tả trong đoạn trước, mỗi đoạn siêu dữ liệu trong trường auxdata (hoặc trường “addbsi” hoặc đoạn bit thừa) của khung có ba cấp độ cấu trúc:

cấu trúc cấp cao, bao gồm chỉ báo liệu trường auxdata (hoặc addbsi) bao gồm siêu dữ liệu, ít nhất một trị số ID chỉ báo (các) loại siêu dữ liệu gì có mặt, và thông thường còn có trị số chỉ báo bao nhiêu bit của siêu dữ liệu (ví dụ, của mỗi loại) có mặt (nếu siêu dữ liệu có mặt). Một loại siêu dữ liệu mà có thể có mặt là LSPM, loại siêu dữ liệu khác mà có thể có mặt là siêu dữ liệu biên chương trình, và loại siêu dữ liệu khác nữa mà có thể có mặt là siêu dữ liệu nghiên cứu đa phương tiện;

cấu trúc cấp trung, bao gồm một phần tử chính cho mỗi loại siêu dữ liệu được xác định (ví dụ, phần đầu chính, các trị số bảo vệ, và ID tải trọng và các trị số độ lớn tải trọng, ví dụ, của loại được nêu trên, cho mỗi loại siêu dữ liệu được xác định); và

cấu trúc cấp thấp, bao gồm mỗi tải trọng cho một phần tử chính (ví dụ, tải trọng LPSM, nếu nó được xác định bởi phần tử chính là có mặt, và/hoặc một tải trọng siêu dữ liệu của loại khác, nếu nó được xác định bởi phần tử chính là có mặt).

Các trị số dữ liệu trong ba cấp cấu trúc này có thể được lồng vào nhau. Ví dụ, (các) trị số bảo vệ cho tải trọng LPSM và/hoặc tải trọng siêu dữ liệu khác được xác định bởi một phần tử chính có thể được bao gồm sau mỗi tải trọng được xác định bởi phần tử

chính (và do đó sau phần đầu chính của phần tử chính). Trong một ví dụ, phần đầu chính có thể xác định tải trọng LPSM và tải trọng siêu dữ liệu khác, ID tải trọng và các trị số độ lớn tải trọng cho tải trọng thứ nhất (ví dụ, tải trọng LPSM) có thể theo sau phần đầu chính, tải trọng thứ nhất có thể tự theo sau các trị số ID và độ lớn, ID tải trọng và trị số độ lớn tải trọng cho tải trọng thứ hai có thể theo sau tải trọng thứ nhất, tải trọng thứ hai có thể tự theo sau các trị số ID và độ lớn này, và (các) trị số bảo vệ cho một trong hai hoặc cả hai tải trọng (hoặc cho các trị số phần tử chính và một trong hai hoặc cả hai tải trọng) có thể theo sau tải trọng cuối cùng.

Theo một số phương án, phần tử chính của đoạn siêu dữ liệu trong trường auxdata (hoặc trường “addbsi” hoặc đoạn bit thừa) của khung bao gồm phần đầu chính (thường bao gồm các trị số xác định, ví dụ, phiên bản phần tử chính), và sau phần đầu chính: các trị số chỉ báo dữ liệu vân tay có được bao gồm cho siêu dữ liệu của đoạn siêu dữ liệu hay không, các trị số chỉ báo dữ liệu ngoài (liên quan đến dữ liệu âm thanh tương ứng với siêu dữ liệu của đoạn siêu dữ liệu) có tồn tại không, ID tải trọng và các trị số độ lớn tải trọng cho mỗi loại siêu dữ liệu (ví dụ, LPSM, và/hoặc siêu dữ liệu của loại khác ngoài LPSM) được xác định bởi phần tử chính, và các trị số bảo vệ cho ít nhất một loại siêu dữ liệu được xác định bởi phần tử chính. (Các) tải trọng siêu dữ liệu của đoạn siêu dữ liệu theo sau phần đầu chính, và được (trong một số trường hợp) lồng với các trị số của phần tử chính.

Trong định dạng ưu tiên khác, dòng bit được mã hóa là dòng bit Dolby E, và mỗi đoạn siêu dữ liệu mà bao gồm LPSM (và cũng có thể bao gồm siêu dữ liệu biên chương trình) được bao gồm trong N vị trí mẫu đầu tiên của khoảng dải bảo vệ (guard band interval) Dolby E.

Theo một nhóm các phương án khác, sáng chế đề xuất APU (ví dụ, bộ giải mã) được lắp và tạo cấu hình để thu dòng bit âm thanh được mã hóa bao gồm các đoạn dữ liệu âm thanh và đoạn siêu dữ liệu, trong đó các đoạn dữ liệu âm thanh chỉ báo dữ liệu âm thanh, và ít nhất mỗi đoạn trong số các đoạn siêu dữ liệu bao gồm siêu dữ liệu trạng thái xử lý âm lượng (LPSM) và cũng có thể bao gồm siêu dữ liệu biên chương trình, và để trích xuất LPSM từ dòng bit, để tạo ra dữ liệu âm thanh được giải mã để đáp lại dữ liệu âm thanh và để thực hiện ít nhất một hoạt động xử lý âm lượng thích ứng trên dữ liệu âm thanh bằng cách sử dụng LPSM. Một số phương án thuộc nhóm này cũng bao gồm bộ

xử lý sau được lắp vào APU, trong đó bộ xử lý sau được lắp và tạo cấu hình để thực hiện ít nhất một hoạt động xử lý âm lượng thích ứng trên dữ liệu âm thanh bằng cách sử dụng LPSM.

Theo một nhóm các phương án khác, sáng chế đề xuất đơn vị xử lý âm thanh (APU) bao gồm bộ nhớ đệm (bộ đệm) và hệ thống xử lý phụ được nối với bộ đệm, trong đó APU được ghép để thu dòng bit âm thanh được mã hóa bao gồm các đoạn dữ liệu âm thanh và đoạn siêu dữ liệu, trong đó các đoạn dữ liệu âm thanh chỉ báo dữ liệu âm thanh, và ít nhất mỗi đoạn trong số các đoạn siêu dữ liệu bao gồm siêu dữ liệu trạng thái xử lý âm lượng (LPSM) và cũng có thể bao gồm siêu dữ liệu biên chương trình, bộ đệm lưu trữ (ví dụ, theo cách không chuyển tiếp) ít nhất một khung của dòng bit âm thanh được mã hóa, và hệ thống xử lý phụ được tạo cấu hình để trích xuất LPSM từ dòng bit và để thực hiện ít nhất một hoạt động xử lý âm lượng thích ứng trên dữ liệu âm thanh bằng cách sử dụng LPSM. Theo các phương án điển hình thuộc nhóm này, APU là một trong số bộ mã hóa, bộ giải mã, và bộ xử lý sau.

Theo một số phương án thực hiện của phương pháp theo sáng chế, dòng bit âm thanh được tạo ra là một trong số dòng bit được mã hóa AC-3, dòng bit E-AC-3, hoặc dòng bit Dolby E, bao gồm siêu dữ liệu trạng thái xử lý âm lượng, cũng như các siêu dữ liệu khác (ví dụ, tham số siêu dữ liệu DIALNORM, tham số siêu dữ liệu điều khiển dải động, và các tham số siêu dữ liệu khác). Theo một số phương án thực hiện khác của phương pháp này, dòng bit âm thanh được tạo ra là dòng bit được mã hóa thuộc loại khác.

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

(và/hoặc được cấu hình theo cách khác) để thực hiện một phương án của phương pháp theo sáng chế (hoặc các bước của nó) để đáp lại dữ liệu được giữ nguyên ở đó.

Mô tả vắn tắt các hình vẽ

Fig.1 là sơ đồ khối thể hiện một phương án của hệ thống mà có thể được tạo cấu hình để thực hiện một phương án của phương pháp theo sáng chế.

Fig.2 là sơ đồ khối thể hiện bộ mã hóa mà là một phương án của đơn vị xử lý âm thanh theo sáng chế.

Fig.3 là sơ đồ khối thể hiện bộ giải mã mà là một phương án của đơn vị xử lý âm thanh theo sáng chế, và bộ xử lý sau được nối với nó theo một phương án khác của đơn vị xử lý âm thanh theo sáng chế.

Fig.4 là sơ đồ thể hiện khung AC-3, bao gồm các đoạn mà nó được chia thành.

Fig.5 là sơ đồ thể hiện đoạn Thông tin Đồng bộ hóa (Synchronization Information - SI) của khung AC-3, bao gồm các đoạn mà nó được chia thành.

Fig.6 là sơ đồ thể hiện đoạn Thông tin Dòng bit (Bitstream Information - BSI) của khung AC-3, bao gồm các đoạn mà nó được chia thành.

Fig.7 là sơ đồ thể hiện khung E-AC-3, bao gồm các đoạn mà nó được chia thành.

Fig.8 là sơ đồ thể hiện các khung của dòng bit âm thanh được mã hóa mà bao gồm siêu dữ liệu biên chương trình, định dạng của nó theo một phương án của sáng chế.

Fig.9 là sơ đồ thể hiện các khung khác của dòng bit âm thanh được mã hóa trên Fig.9. Một số các khung này bao gồm siêu dữ liệu biên chương trình có định dạng theo một phương án của sáng chế.

Fig.10 là sơ đồ thể hiện hai dòng bit âm thanh được mã hóa: dòng bit (IEB) trong đó biên chương trình (được gắn nhãn “Biên”) được căn chỉnh với sự chuyển tiếp giữa hai khung của dòng bit, và dòng bit khác (TB) trong đó biên chương trình (được gắn nhãn “Biên Thực”) được xê dịch bởi 512 mẫu từ sự chuyển tiếp giữa hai khung của dòng bit.

Fig.11 là một tập hợp các sơ đồ thể hiện bốn dòng bit âm thanh được mã hóa. Dòng bit ở đầu Fig.11 (được gắn nhãn “Trường hợp 1”) chỉ báo chương trình âm thanh thứ nhất (P1) bao gồm siêu dữ liệu biên chương trình theo sau bởi chương trình âm thanh

thứ hai (P2) mà cũng bao gồm siêu dữ liệu biên chương trình; dòng bit thứ hai (được gắn nhãn “Trường hợp 2”) chỉ báo chương trình âm thanh thứ nhất (P1) mà bao gồm siêu dữ liệu biên chương trình theo sau bởi chương trình âm thanh thứ hai (P2) mà không bao gồm siêu dữ liệu biên chương trình; dòng bit thứ ba (được gắn nhãn “Trường hợp 3”) chỉ báo chương trình âm thanh thứ nhất bị cắt (P1) mà bao gồm siêu dữ liệu biên chương trình, và đã được nối với toàn bộ chương trình âm thanh thứ hai (P2) mà bao gồm siêu dữ liệu biên chương trình; và dòng bit thứ tư (được gắn nhãn “Trường hợp 4”) chỉ báo chương trình âm thanh thứ nhất bị cắt (P1) mà bao gồm siêu dữ liệu biên chương trình, và chương trình âm thanh thứ hai bị cắt (P2) mà bao gồm siêu dữ liệu biên chương trình và đã được nối với một phần của chương trình âm thanh thứ nhất.

Mô tả chi tiết các phương án thực hiện sáng chế

Kí hiệu và thuật ngữ

Trong suốt phần bộc lộ này, bao gồm cả các yêu cầu bảo hộ, thuật ngữ thực hiện các hoạt động “trên” tín hiệu hoặc dữ liệu (ví dụ, lọc, đo, chuyển đổi, hoặc áp dụng độ khuếch đại, cho tín hiệu hoặc dữ liệu) được sử dụng theo nghĩa rộng để chỉ việc thực hiện các hoạt động trực tiếp trên tín hiệu hoặc dữ liệu, hoặc trên một phiên bản đã được xử lý của tín hiệu hoặc dữ liệu (ví dụ, trên phiên bản của tín hiệu mà đã trải qua quá trình lọc sơ bộ hoặc xử lý trước trước khi thực hiện các hoạt động trên đó).

Trong suốt phần bộc lộ này bao gồm cả các yêu cầu bảo hộ, thuật ngữ “hệ thống” được sử dụng theo nghĩa rộng để chỉ thiết bị, hệ thống, hoặc hệ thống con. Ví dụ, hệ thống con mà thực hiện việc giải mã có thể được gọi là hệ thống giải mã, và hệ thống bao gồm hệ thống con này (ví dụ, hệ thống mà tạo ra các tín hiệu đầu ra X để đáp lại nhiều tín hiệu đầu vào, trong đó hệ thống con tạo ra M tín hiệu đầu ra và X - M tín hiệu đầu vào khác thu được từ một nguồn bên ngoài) cũng có thể được gọi là hệ thống giải mã.

Trong suốt phần bộc lộ này bao gồm cả các yêu cầu bảo hộ, thuật ngữ “bộ xử lý” được sử dụng theo nghĩa rộng để chỉ hệ thống hoặc thiết bị có thể lập trình được hoặc theo cách khác tạo cấu hình được (ví dụ, với phần mềm hoặc phần sụn) để thực hiện các hoạt động trên dữ liệu (ví dụ, âm thanh, hoặc video hoặc dữ liệu hình ảnh khác). Các ví dụ của bộ xử lý bao gồm mảng công lập trình được dạng trường (hoặc mạch hoặc bộ vi mạch tích hợp tạo cấu hình được khác), bộ xử lý tín hiệu kỹ thuật số được lập trình

và/hoặc theo cách khác được tạo cấu hình để thực hiện xử lý đường ống (pipeline) trên âm thanh hoặc dữ liệu âm thanh khác, bộ xử lý mục đích chung có thể lập trình được hoặc máy tính, và chip hoặc bộ chip của bộ vi xử lý lập trình được.

Trong suốt phần bộc lộ này bao gồm cả các yêu cầu bảo hộ, các thuật ngữ “bộ xử lý âm thanh” và “đơn vị xử lý âm thanh” được sử dụng theo cách có thể thay thế cho nhau, và theo nghĩa rộng, để chỉ hệ thống được tạo cấu hình để xử lý dữ liệu âm thanh. Ví dụ của các đơn vị xử lý âm thanh bao gồm, nhưng không giới hạn ở bộ mã hóa (ví dụ, bộ chuyển mã), bộ giải mã, bộ mã hóa-giải mã, hệ thống xử lý trước, hệ thống xử lý sau, và hệ thống xử lý dòng bit (đôi khi được gọi là các công cụ xử lý dòng bit).

Trong suốt phần bộc lộ này bao gồm cả các yêu cầu bảo hộ, thuật ngữ “siêu dữ liệu trạng thái xử lý” (ví dụ, như trong thuật ngữ “siêu dữ liệu trạng thái xử lý âm lượng”) đề cập đến dữ liệu khác và tách biệt khỏi dữ liệu âm thanh tương ứng (nội dung âm thanh của dòng dữ liệu âm thanh mà cũng bao gồm siêu dữ liệu trạng thái xử lý). Siêu dữ liệu trạng thái xử lý kết hợp với dữ liệu âm thanh, chỉ báo trạng thái xử lý âm lượng của dữ liệu âm thanh tương ứng (ví dụ, (các) kiểu xử lý nào đã được thực hiện trên dữ liệu âm thanh), và cũng thường chỉ báo ít nhất một đặc điểm hoặc đặc tính của dữ liệu âm thanh. Việc kết hợp của siêu dữ liệu trạng thái xử lý với dữ liệu âm thanh là đồng bộ theo thời gian. Do đó, siêu dữ liệu trạng thái xử lý hiện thời (được thu hoặc cập nhật mới nhất) chỉ báo rằng dữ liệu âm thanh tương ứng đồng thời bao gồm các kết quả của (các) kiểu xử lý dữ liệu âm thanh được chỉ báo. Trong một số trường hợp, siêu dữ liệu trạng thái xử lý có thể bao gồm lịch sử xử lý và/hoặc một số hoặc tất cả các tham số mà được sử dụng trong và/hoặc có nguồn gốc từ các kiểu xử lý được chỉ báo. Ngoài ra, siêu dữ liệu trạng thái xử lý có thể bao gồm ít nhất một đặc điểm hoặc đặc tính của dữ liệu âm thanh tương ứng, mà đã được tính toán hoặc trích xuất từ dữ liệu âm thanh. Siêu dữ liệu trạng thái xử lý có thể cũng bao gồm siêu dữ liệu khác mà không liên quan đến hoặc có nguồn gốc từ kiểu xử lý bất kỳ của dữ liệu âm thanh tương ứng. Ví dụ, dữ liệu bên thứ ba, thông tin theo dõi, ký hiệu nhân dạng, thông tin về quyền sở hữu hoặc thông tin chuẩn, dữ liệu chú giải của người dùng, dữ liệu sở thích của người dùng, v.v. có thể được thêm vào bởi một đơn vị xử lý âm thanh cụ thể theo sáng chế để chuyển đến các đơn vị xử lý âm thanh khác theo sáng chế.

Trong suốt phần bộc lộ này bao gồm cả các yêu cầu bảo hộ, thuật ngữ “siêu dữ

liệu trạng thái xử lý âm lượng” (hay “LPSM”) chỉ siêu dữ liệu trạng thái xử lý chỉ báo trạng thái xử lý âm lượng của dữ liệu âm thanh tương ứng (ví dụ, (các) kiểu xử lý âm lượng đã được thực hiện trên dữ liệu âm thanh) và cũng thường chỉ báo ít nhất một đặc điểm hoặc đặc tính (ví dụ, âm lượng) của dữ liệu âm thanh tương ứng. Siêu dữ liệu trạng thái xử lý âm lượng có thể bao gồm dữ liệu (ví dụ, siêu dữ liệu khác) mà không phải là (tức là, khi nó được xét riêng) siêu dữ liệu trạng thái xử lý âm lượng.

Trong suốt phần bộc lộ này bao gồm cả các yêu cầu bảo hộ, thuật ngữ “kênh” (hoặc “kênh âm thanh”) chỉ tín hiệu âm thanh đơn âm.

Trong suốt phần bộc lộ này bao gồm cả các yêu cầu bảo hộ, thuật ngữ “chương trình âm thanh” chỉ tập hợp gồm một hoặc nhiều kênh âm thanh và cũng có thể bao gồm siêu dữ liệu được kết hợp (ví dụ, siêu dữ liệu mà mô tả sự biểu diễn âm thanh không gian mong muốn, và/hoặc LPSM, và/hoặc siêu dữ liệu biên chương trình).

Trong suốt phần bộc lộ này bao gồm cả các yêu cầu bảo hộ, thuật ngữ “siêu dữ liệu biên chương trình” chỉ siêu dữ liệu của dòng bit âm thanh được mã hóa, trong đó dòng bit âm thanh được mã hóa chỉ báo ít nhất một chương trình âm thanh (ví dụ, hai hoặc nhiều chương trình âm thanh), và siêu dữ liệu biên chương trình chỉ báo vị trí trong dòng bit của ít nhất một biên (đầu và/hoặc cuối) của ít nhất một chương trình âm thanh nêu trên. Ví dụ, siêu dữ liệu biên chương trình (của dòng bit âm thanh được mã hóa chỉ báo chương trình âm thanh) có thể bao gồm siêu dữ liệu chỉ báo vị trí (ví dụ, điểm bắt đầu của khung thứ “N” của dòng bit, hoặc vị trí mẫu thứ “M” của khung thứ “N” của dòng bit) của phần đầu chương trình, và siêu dữ liệu bổ sung chỉ báo vị trí (ví dụ, điểm bắt đầu của khung thứ “J” của dòng bit, hoặc vị trí mẫu thứ “K” của khung thứ “J” của dòng bit) của phần cuối chương trình.

Trong suốt phần bộc lộ này bao gồm cả các yêu cầu bảo hộ, thuật ngữ “ghép” hoặc “được ghép” được sử dụng để chỉ việc liên kết trực tiếp hoặc gián tiếp. Do đó, nếu thiết bị thứ nhất ghép với thiết bị thứ hai, sự liên kết đó có thể là thông qua một liên kết trực tiếp, hoặc thông qua một liên kết gián tiếp bằng các thiết bị và liên kết khác.

Tương ứng với các phương án điển hình của sáng chế, tải trọng của siêu dữ liệu âm lượng chương trình, còn gọi là siêu dữ liệu trạng thái xử lý âm lượng (“LPSM”) và cũng có thể bao gồm siêu dữ liệu biên chương trình được nhúng vào một hoặc nhiều

trường dự trữ (hoặc các khe) của đoạn siêu dữ liệu của dòng bit âm thanh mà còn bao gồm dữ liệu âm thanh trong các đoạn khác (các đoạn dữ liệu âm thanh). Thông thường, ít nhất một đoạn của mỗi khung của dòng bit bao gồm LPSM, và ít nhất một đoạn khác của khung bao gồm dữ liệu âm thanh tương ứng (tức là, dữ liệu âm thanh mà trạng thái xử lý âm lượng và âm lượng của nó được chỉ báo bởi LPSM). Theo một số phương án, khối lượng dữ liệu của LPSM có thể đủ nhỏ để được mang mà không làm ảnh hưởng đến tốc độ bit được phân bổ để mang dữ liệu âm thanh.

Việc truyền siêu dữ liệu trạng thái xử lý âm lượng trong một chuỗi xử lý dữ liệu âm thanh là đặc biệt hữu dụng khi hai hoặc nhiều đơn vị xử lý âm thanh cần phải làm việc trước sau với một thiết bị khác trong suốt chuỗi xử lý (hoặc chu trình nội dung). Nếu không đưa siêu dữ liệu trạng thái xử lý âm lượng vào trong dòng bit âm thanh, các vấn đề xử lý đa phương tiện nghiêm trọng chẳng hạn như suy giảm chất lượng, mức độ và không gian có thể xảy ra, ví dụ, khi hai hoặc nhiều bộ mã hóa-giải mã âm thanh được sử dụng trong chuỗi và quá trình cân chỉnh âm lượng ở một đầu được áp dụng nhiều hơn một lần khi dòng bit di chuyển đến thiết bị tiêu thụ đa phương tiện (hoặc điểm kết xuất của nội dung âm thanh của dòng bit).

Fig.1 là sơ đồ khối thể hiện chuỗi xử lý âm thanh ví dụ (hệ thống xử lý dữ liệu âm thanh), trong đó một hoặc nhiều cấu phần của hệ thống có thể được tạo cấu hình theo một phương án của sáng chế. Hệ thống bao gồm các cấu phần sau đây, được nối với nhau như thể hiện trên hình vẽ: đơn vị xử lý trước, bộ mã hóa, đơn vị phân tích tín hiệu và hiệu chỉnh siêu dữ liệu, bộ chuyển mã, bộ giải mã, và đơn vị xử lý sau. Trong các biến thể của hệ thống như được thể hiện trên hình vẽ, một hoặc nhiều bộ phận được lược bỏ, hoặc các đơn vị xử lý dữ liệu âm thanh bổ sung được bao gồm.

Theo một số phương án thực hiện, đơn vị xử lý trước trên Fig.1 được tạo cấu hình để chấp nhận các mẫu PCM (miền thời gian) bao gồm nội dung âm thanh làm đầu vào, và để xuất ra các mẫu PCM đã được xử lý. Bộ mã hóa có thể được tạo cấu hình để chấp nhận các mẫu PCM làm đầu vào và để xuất ra dòng bit âm thanh được mã hóa (ví dụ, được nén) chỉ báo nội dung âm thanh. Dữ liệu của dòng bit mà chỉ báo nội dung âm thanh trong bản mô tả này đôi khi được gọi là “dữ liệu âm thanh.” Nếu bộ mã hóa được tạo cấu hình tương ứng với một phương án điển hình của sáng chế, thì đầu ra dòng bit âm thanh từ bộ mã hóa bao gồm siêu dữ liệu trạng thái xử lý âm lượng (và cũng thường bao

gồm siêu dữ liệu khác, có thể bao gồm siêu dữ liệu biên chương trình) cũng như dữ liệu âm thanh.

Đơn vị phân tích tín hiệu và hiệu chỉnh siêu dữ liệu trên Fig.1 có thể chấp nhận một hoặc nhiều dòng bit âm thanh được mã hóa làm đầu vào và xác định (ví dụ, xác minh) siêu dữ liệu trạng thái xử lý trong mỗi dòng bit âm thanh được mã hóa có chính xác hay không, bằng cách thực hiện phân tích tín hiệu (ví dụ, sử dụng siêu dữ liệu biên chương trình trong dòng bit âm thanh được mã hóa). Nếu đơn vị phân tích tín hiệu và hiệu chỉnh siêu dữ liệu nhận thấy siêu dữ liệu được bao gồm là không hợp lệ, nó thường thay thế (các) trị số không chính xác bằng (các) trị số chính xác thu được từ việc phân tích tín hiệu. Do đó, mỗi đầu ra dòng bit âm thanh được mã hóa từ đơn vị phân tích tín hiệu và hiệu chỉnh siêu dữ liệu có thể bao gồm siêu dữ liệu trạng thái xử lý cũng như dữ liệu âm thanh được mã hóa đã được hiệu chỉnh (hoặc không được hiệu chỉnh).

Bộ chuyển mã trên Fig.1 có thể chấp nhận các dòng bit âm thanh được mã hóa làm các dòng bit âm thanh được điều biến (ví dụ, được mã hóa theo cách khác) đầu vào, và đầu ra để đáp lại (ví dụ, bằng cách giải mã một dòng đầu vào và mã hóa lại dòng đã được giải mã theo định dạng mã hóa khác). Nếu bộ chuyển mã được tạo cấu hình theo một phương án điển hình của sáng chế, đầu ra dòng bit âm thanh từ bộ chuyển mã bao gồm siêu dữ liệu trạng thái xử lý âm lượng (và cũng thường bao gồm siêu dữ liệu khác) cũng như dữ liệu âm thanh được mã hóa. Siêu dữ liệu có thể được bao gồm trong dòng bit.

Bộ giải mã trên Fig.1 có thể chấp nhận các dòng bit âm thanh được mã hóa (ví dụ, được nén) làm các dòng đầu vào, và đầu ra (đáp lại) các mẫu âm thanh PCM được giải mã. Nếu bộ giải mã được tạo cấu hình tương ứng với một phương án điển hình của sáng chế, thì đầu ra của bộ giải mã trong trạng thái hoạt động điển hình là hoặc bao gồm bất kỳ loại sau đây:

dòng mẫu âm thanh, và dòng siêu dữ liệu trạng thái xử lý âm lượng tương ứng (và thông thường còn có siêu dữ liệu khác) được trích xuất từ dòng bit được mã hóa đầu vào; hoặc

dòng mẫu âm thanh, và dòng bit điều khiển tương ứng được xác định từ siêu dữ liệu trạng thái xử lý âm lượng (và thông thường còn có siêu dữ liệu khác) được trích xuất từ dòng bit được mã hóa đầu vào; hoặc

dòng mẫu âm thanh, mà không có dòng siêu dữ liệu trạng thái xử lý hoặc dòng bit điều khiển tương ứng được xác định từ siêu dữ liệu trạng thái xử lý. Trong trường hợp cuối cùng này, bộ giải mã có thể trích xuất siêu dữ liệu trạng thái xử lý âm lượng (và/hoặc siêu dữ liệu khác) từ dòng bit được mã hóa đầu vào và thực hiện ít nhất một thao tác trên siêu dữ liệu được trích xuất (ví dụ, xác minh), mặc dù nó không xuất ra siêu dữ liệu được trích xuất hoặc các bit điều khiển được xác định từ đó.

Bằng cách tạo cấu hình đơn vị xử lý sau trên Fig.1 tương ứng với một phương án điển hình của sáng chế, đơn vị xử lý sau được tạo cấu hình để chấp nhận dòng các mẫu âm thanh PCM được giải mã, và để thực hiện việc xử lý sau trên đó (ví dụ, cân chỉnh âm lượng của nội dung âm thanh) bằng cách sử dụng siêu dữ liệu trạng thái xử lý âm lượng (và thông thường còn có cả siêu dữ liệu khác) thu được với các mẫu, hoặc các bit điều khiển (được xác định bởi bộ giải mã từ siêu dữ liệu trạng thái xử lý âm lượng và thông thường còn bởi siêu dữ liệu khác) thu được với các mẫu. Đơn vị xử lý sau cũng thường được tạo cấu hình để kết xuất nội dung âm thanh đã được xử lý sau để phát lại bằng một hoặc nhiều loa.

Các phương án điển hình của sáng chế đề xuất chuỗi xử lý âm thanh nâng cao trong đó các đơn vị xử lý âm thanh (ví dụ, bộ mã hóa, bộ giải mã, bộ chuyển mã, và đơn vị xử lý trước và sau) điều hợp việc xử lý tương ứng của chúng để áp dụng vào dữ liệu âm thanh theo trạng thái hiện thời của dữ liệu đa phương tiện như được chỉ báo bởi siêu dữ liệu trạng thái xử lý âm lượng lần lượt được thu bởi các đơn vị xử lý âm thanh.

Đầu vào dữ liệu âm thanh vào đơn vị xử lý âm thanh bất kỳ của hệ thống trên Fig.1 (ví dụ, bộ mã hóa hoặc bộ chuyển mã trên Fig.1) có thể bao gồm siêu dữ liệu trạng thái xử lý âm lượng (và cũng có thể bao gồm siêu dữ liệu khác) cũng như dữ liệu âm thanh (ví dụ, dữ liệu âm thanh được mã hóa). Siêu dữ liệu này có thể được bao gồm trong âm thanh đầu vào bởi phần tử khác của hệ thống trên Fig.1 (hoặc nguồn khác, không được thể hiện trên Fig.1) theo một phương án của sáng chế. Đơn vị xử lý mà thu âm thanh đầu vào (cùng với siêu dữ liệu) có thể được tạo cấu hình để thực hiện ít nhất một thao tác trên siêu dữ liệu (ví dụ, xác minh) hoặc để đáp lại siêu dữ liệu (ví dụ, xử lý điều hợp âm thanh đầu vào), và cũng thường để bao gồm trong âm thanh đầu ra của nó siêu dữ liệu, phiên bản đã được xử lý của siêu dữ liệu, hoặc các bit điều khiển được xác định từ siêu dữ liệu.

Một phương án điển hình của đơn vị xử lý âm thanh theo sáng chế (hoặc bộ xử lý âm thanh) được tạo cấu hình để thực hiện việc xử lý điều hợp dữ liệu âm thanh dựa trên trạng thái của dữ liệu âm thanh như được chỉ báo bởi siêu dữ liệu trạng thái xử lý âm lượng tương ứng với dữ liệu âm thanh. Theo một số phương án, xử lý điều hợp là (hoặc bao gồm) xử lý âm lượng (nếu siêu dữ liệu chỉ báo rằng việc xử lý âm lượng này, hoặc việc xử lý tương tự với nó, chưa được thực hiện trên dữ liệu âm thanh, nhưng không phải là (và không bao gồm) xử lý âm lượng (nếu siêu dữ liệu chỉ báo rằng việc xử lý âm lượng này, hoặc việc xử lý tương tự với nó, đã được thực hiện trên dữ liệu âm thanh). Theo một số phương án, xử lý điều hợp là hoặc bao gồm việc xác minh siêu dữ liệu (ví dụ, được thực hiện trong đơn vị con xác minh siêu dữ liệu) để đảm bảo đơn vị xử lý âm thanh thực hiện việc xử lý điều hợp khác của dữ liệu âm thanh dựa trên trạng thái của dữ liệu âm thanh như được chỉ báo bởi siêu dữ liệu trạng thái xử lý âm lượng. Theo một số phương án, quá trình xác minh xác định độ tin cậy của siêu dữ liệu trạng thái xử lý âm lượng được kết hợp (ví dụ, được bao gồm trong dòng bit) với dữ liệu âm thanh. Ví dụ, nếu siêu dữ liệu được xác minh là đáng tin cậy, thì các kết quả từ kiểu xử lý âm thanh đã được thực hiện trước đó có thể được sử dụng lại và có thể tránh việc thực hiện lại cùng một kiểu xử lý âm thanh. Mặt khác, nếu siêu dữ liệu được nhận thấy là đã bị thay đổi (hoặc theo cách khác không đáng tin cậy), thì kiểu xử lý đa phương tiện đã được thực hiện trước đó một cách công khai (như được chỉ báo bởi siêu dữ liệu không đáng tin cậy) có thể được lặp lại bởi đơn vị xử lý âm thanh, và/hoặc kiểu xử lý khác có thể được thực hiện bởi đơn vị xử lý âm thanh trên siêu dữ liệu và/hoặc dữ liệu âm thanh. Đơn vị xử lý âm thanh cũng có thể được tạo cấu hình để báo hiệu đến các đơn vị xử lý âm thanh phía sau khác theo sáng chế trong chuỗi xử lý đa phương tiện nâng cao mà siêu dữ liệu trạng thái xử lý âm lượng (ví dụ, có mặt trong dòng bit đa phương tiện) là hợp lệ, nếu đơn vị này xác định rằng siêu dữ liệu trạng thái xử lý là hợp lệ (ví dụ, dựa trên sự so khớp của trị số mật mã được trích xuất và trị số mật mã tham chiếu).

Fig.2 là sơ đồ khối của bộ mã hóa 100 mà là một phương án của đơn vị xử lý âm thanh theo sáng chế. Bất kỳ cấu phần hoặc phần tử nào của bộ mã hóa 100 có thể có thể được thực hiện như một hoặc nhiều quy trình và/hoặc một hoặc nhiều mạch (ví dụ, mạch ASIC, mạch FPGA, hoặc các mạch tích hợp khác), trong phần cứng, phần mềm, hoặc kết hợp phần cứng và phần mềm. Bộ mã hóa 100 bao gồm bộ đệm khung 110, bộ phân tích

111, bộ giải mã 101, bộ xác minh trạng thái âm thanh 102, tầng xử lý âm lượng 103, tầng chọn dòng âm thanh 104, bộ mã hóa 105, tầng bộ nhồi/bộ định dạng 107, tầng tạo siêu dữ liệu 106, hệ thống con đo âm lượng thoại 108, và bộ đệm khung 109, được nối như thể hiện trên hình vẽ. Bộ mã hóa 100 cũng thường bao gồm các bộ phận xử lý khác (không được thể hiện trên hình vẽ).

Bộ mã hóa 100 (mà là bộ chuyển mã) được tạo cấu hình để chuyển đổi dòng bit âm thanh đầu vào (mà, ví dụ, có thể là một dòng bit trong số dòng bit AC-3, dòng bit E-AC-3, hoặc dòng bit Dolby E) thành dòng bit âm thanh đầu ra được mã hóa (mà, ví dụ, có thể là một dòng bit khác trong số dòng bit AC-3, dòng bit E-AC-3, hoặc dòng bit Dolby E) bao gồm bằng cách thực hiện xử lý âm lượng tự động và điều hợp nhờ sử dụng siêu dữ liệu trạng thái xử lý âm lượng được bao gồm trong dòng bit đầu vào. Ví dụ, bộ mã hóa 100 có thể được tạo cấu hình để chuyển đổi dòng bit Dolby E đầu vào (một định dạng thường được sử dụng trong các phương tiện sản xuất và phát thanh chứ không dùng trong các thiết bị tiêu dùng mà nhận chương trình âm thanh được phát sóng đến đó) thành dòng bit âm thanh đầu ra được mã hóa (phù hợp để phát sóng cho các thiết bị tiêu dùng) ở định dạng AC-3 hoặc E-AC-3.

Hệ thống trên Fig.2 cũng bao gồm hệ thống con cấp phát âm thanh được mã hóa 150 (mà lưu trữ và/hoặc cấp phát các dòng bit đầu ra được mã hóa từ bộ mã hóa 100) và bộ giải mã 152. Đầu ra dòng bit âm thanh được mã hóa từ bộ mã hóa 100 có thể được lưu trữ bởi hệ thống con 150 (ví dụ, ở dạng đĩa DVD hoặc đĩa Blu ray), hoặc được truyền bởi hệ thống con 150 (mà có thể triển khai liên kết hoặc mạng lưới truyền), hoặc có thể được cả lưu trữ và truyền bởi hệ thống con 150. Bộ giải mã 152 được tạo cấu hình để giải mã dòng bit âm thanh được mã hóa (được tạo ra bởi bộ mã hóa 100) mà nó nhận thông qua hệ thống con 150, bao gồm bằng cách trích xuất siêu dữ liệu trạng thái xử lý âm lượng (LPSM) từ mỗi khung của dòng bit (và cũng có thể trích xuất siêu dữ liệu biên chương trình từ dòng bit), và tạo ra dữ liệu âm thanh được giải mã. Thông thường, bộ giải mã 152 được tạo cấu hình để thực hiện xử lý âm lượng thích ứng trên dữ liệu âm thanh được giải mã bằng cách sử dụng LPSM (và cũng có thể bao gồm siêu dữ liệu biên chương trình), và/hoặc để chuyển tiếp dữ liệu âm thanh được giải mã và LPSM đến bộ xử lý sau được tạo cấu hình để thực hiện xử lý âm lượng thích ứng trên dữ liệu âm thanh được giải mã bằng cách sử dụng LPSM (và cũng có thể bao gồm siêu dữ liệu biên chương trình).

Thông thường, bộ giải mã 152 bao gồm bộ đệm mà lưu trữ (ví dụ, theo cách không chuyển tiếp) dòng bit âm thanh được mã hóa thu được từ hệ thống con 150.

Nhiều phương án thực hiện khác nhau của bộ mã hóa 100 và bộ giải mã 152 được tạo cấu hình để thực hiện các phương án khác nhau của phương pháp theo sáng chế. Bộ đệm khung 110 là bộ nhớ đệm được lắp để thu dòng bit âm thanh đầu vào được mã hóa. Trong khi hoạt động, bộ đệm 110 lưu trữ (ví dụ, theo cách không chuyển tiếp) ít nhất một khung của dòng bit âm thanh được mã hóa, và trình tự của các khung của dòng bit âm thanh được mã hóa được gán từ bộ đệm 110 đến bộ phân tích 111.

Bộ phân tích 111 được lắp và tạo cấu hình để trích xuất siêu dữ liệu trạng thái xử lý âm lượng (LPSM), và một cách tùy ý còn trích xuất siêu dữ liệu biên chương trình (và/hoặc siêu dữ liệu khác) từ mỗi khung của âm thanh đầu vào được mã hóa trong đó siêu dữ liệu này được bao gồm, để gán ít nhất LPSM (và một cách tùy ý cả siêu dữ liệu biên chương trình và/hoặc siêu dữ liệu khác) đến bộ xác minh trạng thái âm thanh 102, tầng xử lý âm lượng 103, tầng 106 và hệ thống con 108, để trích xuất dữ liệu âm thanh từ âm thanh đầu vào được mã hóa, và để gán dữ liệu âm thanh đến bộ giải mã 101. Bộ giải mã 101 của bộ mã hóa 100 được tạo cấu hình để giải mã dữ liệu âm thanh để tạo ra dữ liệu âm thanh được giải mã, và để gán dữ liệu âm thanh được giải mã đến tầng xử lý âm lượng 103, tầng chọn dòng âm thanh 104, hệ thống con 108, và thông thường còn đến bộ xác minh trạng thái 102.

Bộ xác minh trạng thái 102 được tạo cấu hình để xác thực và xác minh LPSM (và có thể cả siêu dữ liệu khác) được gán đến đó. Theo một số phương án, LPSM là (hoặc được bao gồm trong) khối dữ liệu mà đã được bao gồm trong dòng bit đầu vào (ví dụ, theo một phương án của sáng chế). Khối này có thể bao gồm băm mật mã (mã xác thực thông điệp dựa trên hàm băm hay “HMAC”) để xử lý LPSM (và cũng có thể bao gồm siêu dữ liệu khác) và/hoặc dữ liệu âm thanh nền (được cấp từ bộ giải mã 101 đến bộ xác minh 102). Khối dữ liệu có thể được đánh dấu kỹ thuật số trong các phương án này, sao cho đơn vị xử lý âm thanh phía sau có thể xác thực và xác minh siêu dữ liệu trạng thái xử lý một cách tương đối dễ dàng.

Ví dụ, HMAC được sử dụng để tạo ra một giải thuật tiêu hóa (digest), và (các) trị số bảo vệ được bao gồm trong dòng bit theo sáng chế có thể bao gồm giải thuật tiêu hóa. Giải thuật tiêu hóa có thể được tạo ra như sau cho một khung AC-3:

1. Sau khi dữ liệu AC-3 và LPSM được mã hóa, các byte dữ liệu khung (dữ liệu_khung #1 và dữ liệu_khung #2 được ghép nối) và các byte dữ liệu LPSM được sử dụng làm đầu vào cho hàm băm HMAC. Dữ liệu khác, mà có thể có mặt trong trường auxdata, không được xét đến để tính toán giải thuật tiêu hóa. Dữ liệu khác này có thể là các byte không thuộc về dữ liệu AC-3 hay dữ liệu LPSM. Các bit bảo vệ được bao gồm trong LPSM có thể không được xét đến để tính toán giải thuật tiêu hóa HMAC.
2. Sau khi giải thuật tiêu hóa được tính toán, nó được viết thành dòng bit trong một trường được dự trữ cho các bit bảo vệ.
3. Bước cuối cùng trong việc tạo ra khung AC-3 hoàn chỉnh là tính toán phép thử CRC. Điều này được viết ở đầu cuối cùng của khung và tất cả dữ liệu thuộc về khung này được đưa vào xem xét, bao gồm các bit LPSM.

Các phương pháp mật mã khác bao gồm nhưng không giới hạn ở phương pháp bất kỳ trong số một hoặc nhiều phương pháp mật mã khác HMAC có thể được sử dụng cho việc xác minh LPSM (ví dụ, trong bộ xác minh 102) để đảm bảo việc truyền và nhận an toàn của LPSM và/hoặc dữ liệu âm thanh nền. Ví dụ, việc xác minh (sử dụng phương pháp mật mã này) có thể được thực hiện trong mỗi đơn vị xử lý âm thanh mà thu nhận dòng bit âm thanh theo một phương án của sáng chế để xác định siêu dữ liệu trạng thái xử lý âm lượng và dữ liệu âm thanh tương ứng có được bao gồm trong dòng bit đã trải qua (và/hoặc là kết quả của) quá trình xử lý âm lượng cụ thể (như được chỉ báo bởi siêu dữ liệu) và chưa được điều biến sau khi thực hiện quá trình xử lý âm lượng cụ thể này.

Bộ xác minh trạng thái 102 gán dữ liệu điều khiển đến tầng chọn dòng âm thanh 104, bộ tạo siêu dữ liệu 106, và hệ thống con đo âm lượng thoại 108, để chỉ báo các kết quả của hoạt động xác minh. Để đáp lại dữ liệu điều khiển, tầng 104 có thể lựa chọn (và đi qua đến bộ mã hóa 105) hoặc:

đầu ra được xử lý điều hợp của tầng xử lý âm lượng 103 (ví dụ, khi LPSM chỉ báo rằng đầu ra dữ liệu âm thanh từ bộ giải mã 101 chưa trải qua kiểu xử lý âm lượng cụ thể, và các bit điều khiển từ bộ xác minh 102 chỉ báo rằng LPSM là hợp lệ); hoặc

đầu ra dữ liệu âm thanh từ bộ giải mã 101 (ví dụ, khi LPSM chỉ báo rằng đầu ra dữ liệu âm thanh từ bộ giải mã 101 đã trải qua kiểu xử lý âm lượng cụ thể mà được thực hiện bởi tầng 103, và các bit điều khiển từ bộ xác minh 102 chỉ báo rằng LPSM là hợp lệ).

Tầng 103 của bộ mã hóa 100 được tạo cấu hình để thực hiện xử lý âm lượng thích ứng trên đầu ra dữ liệu âm thanh được giải mã từ bộ giải mã 101, dựa trên một hoặc nhiều đặc tính dữ liệu âm thanh được chỉ báo bởi LPSM được trích xuất bởi bộ giải mã 101. Tầng 103 có thể là bộ xử lý âm lượng thời gian thực chuyển đổi miền điều hợp và kiểm soát dải động. Tầng 103 có thể nhận đầu vào từ người dùng (ví dụ, các trị số dải động/âm lượng đích của người dùng hoặc các trị số dialnorm), hoặc đầu vào siêu dữ liệu khác (ví dụ, một hoặc nhiều loại dữ liệu của bên thứ ba, thông tin theo dõi, ký hiệu nhân dạng, thông tin về quyền sở hữu hoặc thông tin chuẩn, dữ liệu chú giải của người dùng, dữ liệu ưu tiên của người dùng, v.v.) và/hoặc các đầu vào khác (ví dụ, từ quá trình lấy dấu vân tay), và sử dụng các đầu vào này để xử lý đầu ra dữ liệu âm thanh được giải mã từ bộ giải mã 101. Tầng 103 có thể thực hiện xử lý âm lượng thích ứng trên dữ liệu âm thanh được giải mã (đầu ra từ bộ giải mã 101) chỉ báo một chương trình âm thanh đơn nhất (như được chỉ báo bởi siêu dữ liệu biên chương trình được trích xuất bởi bộ phân tích 111), và có thể thiết lập lại quá trình xử lý âm lượng để đáp lại việc nhận dữ liệu âm thanh được giải mã (đầu ra từ bộ giải mã 101) chỉ báo một chương trình âm thanh khác như được chỉ báo bởi siêu dữ liệu biên chương trình được trích xuất bởi bộ phân tích 111.

Hệ thống con đo âm lượng thoại 108 có thể hoạt động để xác định âm lượng của các đoạn của âm thanh được giải mã (từ bộ giải mã 101) mà chỉ báo thoại (hoặc tiếng nói khác), ví dụ, bằng cách sử dụng LPSM (và/hoặc siêu dữ liệu khác) được trích xuất bởi bộ giải mã 101, khi các bit điều khiển từ bộ xác minh 102 chỉ báo rằng LPSM là hợp lệ. Hoạt động của hệ thống con đo âm lượng thoại 108 có thể bị vô hiệu hóa khi LPSM trước đó chỉ báo âm lượng được xác định của các đoạn thoại (hoặc các tiếng nói khác) của âm thanh được giải mã (từ bộ giải mã 101) khi các bit điều khiển từ bộ xác minh 102 chỉ báo rằng LPSM là hợp lệ. Hệ thống con 108 có thể thực hiện phép đo âm lượng trên dữ liệu âm thanh được giải mã chỉ báo một chương trình âm thanh duy nhất (như được chỉ báo bởi siêu dữ liệu biên chương trình được trích xuất bởi bộ phân tích 111), và có thể thiết lập lại phép đo để đáp lại việc nhận dữ liệu âm thanh được giải mã chỉ báo chương trình âm thanh khác như được chỉ báo bởi siêu dữ liệu biên chương trình.

Các công cụ hữu ích (ví dụ, thiết bị đo âm lượng Dolby LM100) sẵn có dùng để đo mức thoại trong nội dung âm thanh một cách thuận tiện và dễ dàng. Theo một số phương án của sáng chế, APU (ví dụ, tầng 108 của bộ mã hóa 100) được cài đặt để bao gồm (hoặc để thực hiện các chức năng của) công cụ này để đo âm lượng thoại trung bình của nội dung âm thanh của dòng bit âm thanh (ví dụ, dòng bit AC-3 được giải mã được gán đến tầng 108 từ bộ giải mã 101 của bộ mã hóa 100).

Nếu tầng 108 được cài đặt để đo âm lượng thoại trung bình thực của dữ liệu âm thanh, phép đo có thể bao gồm bước tách các đoạn của nội dung âm thanh mà chứa phần lớn tiếng nói. Các đoạn âm thanh mà phần lớn là tiếng nói sau đó được xử lý tương ứng với một thuật toán đo âm lượng. Đối với dữ liệu âm thanh được giải mã từ dòng bit AC-3, thuật toán này có thể là phép đo âm lượng trọng số K chuẩn (tương ứng với chuẩn quốc tế ITU-R BS.1770). Theo cách khác, các phép đo âm lượng khác có thể được sử dụng (ví dụ, các phép đo mà dựa trên các mô hình tâm thính học của âm lượng).

Việc tách các đoạn tiếng nói là không nhất thiết để đo âm lượng thoại trung bình của dữ liệu âm thanh. Tuy nhiên, nó nâng cao độ chính xác của phép đo và thường đem đến kết quả thỏa mãn hơn cho người nghe. Bởi vì không phải tất cả nội dung âm thanh đều chứa thoại (tiếng nói), phép đo âm lượng của toàn bộ nội dung âm thanh có thể đưa ra ước tính đầy đủ của mức thoại của âm thanh, với sự có mặt của tiếng nói.

Bộ tạo siêu dữ liệu 106 tạo ra (và/hoặc đi qua đến tầng 107) siêu dữ liệu để được bao gồm bởi tầng 107 trong dòng bit được mã hóa để làm đầu ra từ bộ mã hóa 100. Bộ tạo siêu dữ liệu 106 có thể chuyển tiếp LPSM đến tầng 107 (và cũng có thể cả siêu dữ liệu biên chương trình và/hoặc siêu dữ liệu khác) được trích xuất bởi bộ mã hóa 101 và/hoặc bộ phân tích 111 (ví dụ, khi các bit điều khiển từ bộ xác minh 102 chỉ báo rằng LPSM và/hoặc siêu dữ liệu khác là hợp lệ), hoặc tạo ra LPSM mới (và cũng có thể bao gồm siêu dữ liệu biên chương trình và/hoặc siêu dữ liệu khác) và gán siêu dữ liệu mới đến tầng 107 (ví dụ, khi các bit điều khiển từ bộ xác minh 102 chỉ báo rằng LPSM và/hoặc siêu dữ liệu khác được trích xuất bởi bộ giải mã 101 là không hợp lệ, hoặc nó có thể gán đến tầng 107 kết hợp của siêu dữ liệu được trích xuất bởi bộ giải mã 101 và/hoặc bộ phân tích 111 và siêu dữ liệu mới tạo ra. Bộ tạo siêu dữ liệu 106 có thể bao gồm dữ liệu âm lượng được tạo ra bởi hệ thống con 108, và ít nhất một trị số chỉ báo kiểu xử lý

âm lượng thực hiện bởi hệ thống con 108, trong LPSM nó gán đến tầng 107 để đưa vào trong dòng bit được mã hóa để làm đầu ra từ bộ mã hóa 100.

Bộ tạo siêu dữ liệu 106 có thể tạo ra các bit bảo vệ (mà có thể có hoặc bao gồm mã xác thực thông điệp dựa trên hàm băm hay “HMAC”) hữu dụng cho ít nhất một trong số các hoạt động giải mã, xác thực, hoặc xác minh LPSM (và cũng có thể là siêu dữ liệu khác) để được bao gồm trong dòng bit được mã hóa và/hoặc dữ liệu âm thanh nền để được bao gồm trong dòng bit được mã hóa. Bộ tạo siêu dữ liệu 106 có thể đưa các bit bảo vệ này đến tầng 107 để đưa vào trong dòng bit được mã hóa.

Trong trạng thái hoạt động bình thường, hệ thống con đo âm lượng thoại 108 xử lý đầu ra dữ liệu âm thanh từ bộ giải mã 101 để tạo ra các trị số âm lượng để đáp lại đầu ra này (ví dụ, các trị số âm lượng thoại có cổng và không cổng) và các trị số dải động. Để đáp lại các trị số này, bộ tạo siêu dữ liệu 106 có thể tạo ra siêu dữ liệu trạng thái xử lý âm lượng (LPSM) để đưa vào (bởi bộ nhồi/bộ định dạng 107) trong dòng bit được mã hóa để làm đầu ra từ bộ mã hóa 100.

Ngoài ra, một cách tùy ý, hoặc theo cách khác, các hệ thống con của 106 và/hoặc 108 của bộ mã hóa 100 có thể thực hiện phép phân tích thêm của dữ liệu âm thanh để tạo ra siêu dữ liệu chỉ báo ít nhất một đặc tính của dữ liệu âm thanh để đưa vào trong dòng bit được mã hóa để làm đầu ra từ tầng 107.

Bộ mã hóa 105 mã hóa (ví dụ, bằng cách thực hiện nén trên đó) đầu ra dữ liệu âm thanh từ tầng lựa chọn 104, và gán âm thanh được mã hóa đến tầng 107 để đưa vào trong dòng bit được mã hóa để làm đầu ra từ tầng 107.

Tầng 107 dồn kênh âm thanh được mã hóa từ bộ mã hóa 105 và siêu dữ liệu (bao gồm LPSM) từ bộ tạo 106 để tạo ra dòng bit được mã hóa để làm đầu ra từ tầng 107, tốt hơn là sao cho dòng bit được mã hóa có định dạng như được xác định bởi phương án được ưu tiên của sáng chế.

Bộ đệm khung 109 là bộ nhớ đệm mà lưu trữ (ví dụ, theo cách không chuyển tiếp) ít nhất một khung của đầu ra dòng bit âm thanh được mã hóa từ tầng 107, và trình tự của khung dòng bit âm thanh được mã hóa sau đó được gán từ bộ đệm 109 làm đầu ra từ bộ mã hóa 100 đến hệ thống cấp 150.

LPSM được tạo ra bởi bộ tạo siêu dữ liệu 106 và được bao gồm trong dòng bit được mã hóa bởi tầng 107 chỉ báo trạng thái xử lý âm lượng của dữ liệu âm thanh tương ứng (ví dụ, (các) kiểu xử lý âm lượng nào đã được thực hiện trên dữ liệu âm thanh) và âm lượng (ví dụ, âm lượng thoại đo được, âm lượng có cổng và/hoặc không cổng, và/hoặc dải động) của dữ liệu âm thanh tương ứng.

Ở đây, “việc tạo cổng” của phép đo âm lượng và/hoặc phép đo mức được thực hiện trên dữ liệu âm thanh chỉ một ngưỡng âm lượng hoặc mức cụ thể trong đó (các) trị số được tính toán được mà vượt quá ngưỡng được bao gồm trong phép đo cuối cùng (ví dụ, bỏ qua các trị số âm lượng ngắn hạn dưới -60 dBFS trong các trị số đo được cuối cùng). Việc tạo cổng trên trị số tuyệt đối chỉ một mức hoặc âm lượng cố định, trong khi đó việc tạo cổng trên trị số tương đối chỉ một trị số mà phụ thuộc vào trị số đo “không được tạo cổng” hiện thời.

Theo các phương án thực hiện của bộ mã hóa 100, dòng bit được mã hóa được đệm trong bộ nhớ 109 (và xuất đến hệ thống cấp 150) là dòng bit AC-3 hoặc dòng bit E-AC-3, và bao gồm các đoạn dữ liệu âm thanh (ví dụ, các đoạn từ AB0 đến AB5 của khung được thể hiện trên Fig.4) và các đoạn siêu dữ liệu, trong đó các đoạn dữ liệu âm thanh chỉ báo dữ liệu âm thanh, và ít nhất mỗi đoạn trong số các đoạn siêu dữ liệu bao gồm siêu dữ liệu trạng thái xử lý âm lượng (LPSM). Tầng 107 chèn LPSM (và cũng có thể là siêu dữ liệu biên chương trình) vào dòng bit dưới định dạng sau. Mỗi đoạn siêu dữ liệu mà bao gồm LPSM (và cũng có thể bao gồm siêu dữ liệu biên chương trình) được bao gồm trong đoạn bit thừa của dòng bit (ví dụ, đoạn bit thừa “W” như được thể hiện trên Fig.4 hoặc Fig.7), hoặc trường “addbsi” của đoạn Thông tin Dòng bit (BSI) của khung của dòng bit, hoặc trong trường auxdata (ví dụ, đoạn AUX được thể hiện trên Fig.4 hoặc Fig.7) ở cuối khung của dòng bit. Khung của dòng bit có thể bao gồm một hoặc hai đoạn siêu dữ liệu, mỗi đoạn bao gồm LPSM, và nếu khung bao gồm hai đoạn siêu dữ liệu, một đoạn có thể có mặt trong trường addbsi của khung và đoạn còn lại có mặt trong trường AUX của khung. Theo một số phương án, mỗi đoạn siêu dữ liệu bao gồm LPSM bao gồm đoạn tải trọng LPSM (hoặc bộ chứa) có định dạng sau:

phần đầu (thường bao gồm một lệnh đồng bộ hóa xác định điểm bắt đầu của tải trọng LPSM, theo sau bởi ít nhất một trị số xác định, ví dụ, phiên bản định dạng LPSM,

độ dài, chu trình, đếm, và các trị số liên kết dòng phụ được chỉ báo trong Bảng 2 dưới đây); và

sau phần đầu,

ít nhất một trị số chỉ báo thoại (ví dụ, tham số “(các) kênh thoại” trong Bảng 2) chỉ báo dữ liệu âm thanh tương ứng có chỉ báo thoại hay không chỉ báo thoại (ví dụ, các kênh dữ liệu âm thanh tương ứng nào chỉ báo thoại);

ít nhất một trị số tuân thủ quy định âm lượng (ví dụ, tham số “Loại Quy định về Âm lượng” trong Bảng 2) chỉ báo dữ liệu âm thanh tương ứng có tuân thủ đúng theo một nhóm các quy định về âm lượng được chỉ định hay không;

ít nhất một trị số xử lý âm lượng (ví dụ, một hoặc nhiều tham số “Cờ Hiệu chỉnh Âm lượng công Thoại,” “Kiểu hiệu chỉnh Âm lượng,” trong Bảng 2) chỉ báo ít nhất một kiểu xử lý âm lượng mà đã được thực hiện trên dữ liệu âm thanh tương ứng; và

ít nhất một trị số âm lượng (ví dụ, một hoặc nhiều tham số “Âm lượng có công liên quan đến ITU,” “Âm lượng có công tiếng nói ITU,” “Âm lượng Ngắn kì 3 giây ITU (EBU 3341),” và “Mức thật của Đỉnh” trong Bảng 2) chỉ báo ít nhất một đặc tính âm lượng (ví dụ, âm lượng đỉnh hoặc trung bình) của dữ liệu âm thanh tương ứng.

Theo một số phương án, mỗi đoạn siêu dữ liệu mà chứa LPSM và siêu dữ liệu biên chương trình chứa phần đầu chính (và cũng có thể bao gồm các phần tử chính bổ sung), và sau phần đầu chính (hoặc phần đầu chính và các phần tử chính khác) và đoạn tải trọng LPSM (hoặc bộ chứa) có định dạng sau:

phần đầu, thường bao gồm ít nhất một trị số xác định (ví dụ, phiên bản định dạng LPSM, độ dài, chu trình, đếm, và các trị số liên kết dòng phụ, như được thể hiện trong Bảng 2 được trình bày ở đây), và

sau phần đầu, LPSM và siêu dữ liệu biên chương trình. Siêu dữ liệu biên chương trình có thể bao gồm số lượng khung biên chương trình, và trị số mã (ví dụ, trị số “tồn tại_xê dịch”) chỉ báo khung chỉ bao gồm số lượng khung biên chương trình hay cả số lượng khung biên chương trình và trị số xê dịch), và (trong một số trường hợp) trị số xê dịch.

Theo một số phương án thực hiện, mỗi đoạn siêu dữ liệu được chèn bởi tầng 107 vào đoạn bit thừa hoặc trường “addbsi” hoặc trường auxdata của khung của dòng bit có định dạng sau:

phần đầu chính (thường bao gồm một lệnh đồng bộ hóa xác định điểm bắt đầu của đoạn siêu dữ liệu, theo sau bởi các trị số xác định, ví dụ, phiên bản phần tử chính, độ dài, và chu trình, đếm phần tử mở rộng, và các trị số liên kết dòng phụ được chỉ báo trong Bảng 1 dưới đây); và

sau phần đầu chính, ít nhất một trị số bảo vệ (ví dụ, thuật toán tiêu hóa HMAC và các trị số vân tay âm thanh ở Bảng 1) hữu dụng cho ít nhất một trong số các việc giải mã, xác thực, hoặc xác minh ít nhất một trong số siêu dữ liệu trạng thái xử lý âm lượng hoặc dữ liệu âm thanh tương ứng); và

cũng sau phần đầu chính, nếu đoạn siêu dữ liệu bao gồm LPSM, xác định tải trọng LPSM (“ID”) và các trị số độ lớn tải trọng LPSM mà xác định siêu dữ liệu theo sau là tải trọng LPSM và chỉ báo độ lớn của tải trọng LPSM.

Đoạn tải trọng (hoặc bộ chứa) LPSM (tốt hơn là có định dạng được quy định như trên) theo sau ID tải trọng LPSM và các trị số độ lớn tải trọng LPSM.

Theo một số phương án, mỗi đoạn siêu dữ liệu trong trường auxdata (hoặc trường “addbsi”) của khung có ba cấp độ cấu trúc:

cấu trúc cấp cao, bao gồm cờ chỉ báo trường auxdata (hoặc addbsi) có bao gồm siêu dữ liệu hay không, ít nhất một trị số ID chỉ báo (các) loại siêu dữ liệu nào có mặt, và cũng thường bao gồm một trị số chỉ báo bao nhiêu bit của siêu dữ liệu (ví dụ, của mỗi loại) có mặt (nếu siêu dữ liệu có mặt). Một loại siêu dữ liệu mà có thể có mặt là LSPM, loại siêu dữ liệu khác mà có thể có mặt là siêu dữ liệu biên chương trình, và loại siêu dữ liệu khác nữa mà có thể có mặt là siêu dữ liệu nghiên cứu đa phương tiện (ví dụ, siêu dữ liệu nghiên cứu đa phương tiện Nielsen);

cấu trúc cấp trung, bao gồm một phần tử chính cho mỗi loại siêu dữ liệu được xác định (ví dụ, phần đầu chính, các trị số bảo vệ, và ID tải trọng LPSM và các trị số độ lớn tải trọng LPSM, như nêu trên, cho mỗi loại siêu dữ liệu được xác định); và

cấu trúc cấp thấp, bao gồm mỗi tải trọng cho một phần tử chính (ví dụ, tải trọng LPSM, nếu một tải trọng được xác định bởi bộ phận chính là có mặt, và/hoặc một tải trọng siêu dữ liệu thuộc loại khác, nếu một tải trọng được xác định bởi phần tử chính là có mặt).

Các trị số dữ liệu trong cấu trúc ba cấp này có thể được lồng vào nhau. Ví dụ, (các) trị số bảo vệ cho tải trọng LPSM và/hoặc tải trọng siêu dữ liệu khác được xác định bởi một phần tử chính có thể được bao gồm sau mỗi tải trọng được xác định bởi phần tử chính (và do đó sau phần đầu chính của phần tử chính). Theo một ví dụ, phần đầu chính có thể xác định tải trọng LPSM và tải trọng siêu dữ liệu khác, các trị số ID tải trọng và độ lớn tải trọng cho tải trọng thứ nhất (ví dụ, tải trọng LPSM) có thể theo sau phần đầu chính, tải trọng thứ nhất có thể tự theo sau các trị số ID và độ lớn tải trọng, trị số độ lớn tải trọng và ID tải trọng cho tải trọng thứ hai có thể theo sau tải trọng thứ nhất, tải trọng thứ hai có thể tự theo sau các trị số ID và độ lớn này, và các bit bảo vệ cho cả hai tải trọng (hoặc cho các trị số phần tử chính và cả hai tải trọng) có thể theo sau tải trọng cuối cùng.

Theo một số phương án, nếu bộ giải mã 101 thu dòng bit âm thanh được tạo ra theo một phương án của sáng chế bằng băm mật mã, thì bộ giải mã được tạo cấu hình để phân tích và truy hồi băm mật mã từ một khối dữ liệu được xác định từ dòng bit, khối này bao gồm siêu dữ liệu trạng thái xử lý âm lượng (LPSM) và cũng có thể bao gồm siêu dữ liệu biên chương trình. Bộ xác minh 102 có thể sử dụng băm mật mã để xác minh dòng bit thu được và/hoặc siêu dữ liệu liên quan. Ví dụ, nếu bộ xác minh 102 nhận thấy LPSM là hợp lệ dựa trên sự ăn khớp giữa băm mật mã tham chiếu và băm mật mã được truy hồi từ khối dữ liệu, thì sau đó nó có thể vô hiệu hóa hoạt động của bộ xử lý 103 trên dữ liệu âm thanh tương ứng và khiến tầng lựa chọn 104 chuyển tiếp (không đổi) dữ liệu âm thanh. Ngoài ra, một cách tùy ý, hoặc theo cách khác, các loại kỹ thuật mật mã khác có thể được sử dụng thay cho phương pháp dựa trên băm mật mã.

Bộ mã hóa 100 trên Fig.2 có thể xác định (để đáp lại LPSM, và cũng có thể bao gồm siêu dữ liệu biên chương trình, được trích xuất bởi bộ giải mã 101) mà đơn vị xử lý trước/sau đã thực hiện một kiểu xử lý âm lượng trên dữ liệu âm thanh cần được mã hóa (trong các phần tử 105, 106, và 107) và do đó có thể tạo ra (trong bộ tạo 106) siêu dữ liệu trạng thái xử lý âm lượng mà bao gồm các tham số cụ thể được sử dụng trong và/hoặc có

nguồn gốc từ kiểu xử lý âm lượng được thực hiện trước đó. Theo một số phương án thực hiện, bộ mã hóa 100 có thể tạo ra (và bao gồm trong đầu ra dòng bit được mã hóa từ đó) siêu dữ liệu trạng thái xử lý chỉ báo lịch sử xử lý trên nội dung âm thanh miễn là bộ mã hóa nhận biết được kiểu xử lý mà đã được thực hiện trên nội dung âm thanh.

Fig.3 là sơ đồ khối của bộ giải mã (200) mà là một phương án của đơn vị xử lý âm thanh theo sáng chế, và đơn vị xử lý sau (300) được ghép vào đó. Bộ xử lý sau (300) cũng là một phương án của đơn vị xử lý âm thanh theo sáng chế. Bất kỳ cấu phần hoặc phần tử nào của bộ giải mã 200 và bộ xử lý sau 300 có thể được cài đặt dưới dạng một hoặc nhiều quy trình và/hoặc một hoặc nhiều mạch (ví dụ, mạch ASIC, mạch FPGA, hoặc các mạch tích hợp khác), trong phần cứng, phần mềm, hoặc sự kết hợp của phần cứng và phần mềm. Bộ giải mã 200 bao gồm bộ đệm khung 201, bộ phân tích 205, bộ giải mã âm thanh 202, tầng xác minh trạng thái âm thanh (bộ xác minh) 203, và tầng tạo bit điều khiển 204, liên kết như được thể hiện trên hình vẽ. Thông thường, bộ giải mã 200 cũng bao gồm các phần tử xử lý khác (không được thể hiện trên hình vẽ).

Bộ đệm khung 201 (bộ nhớ đệm) lưu trữ (ví dụ, theo cách không chuyển tiếp) ít nhất một khung dòng bit âm thanh được mã hóa được thu bởi bộ giải mã 200. Trình tự của các khung dòng bit âm thanh được mã hóa được gán từ bộ đệm 201 đến bộ phân tích 205.

Bộ phân tích 205 được ghép và tạo cấu hình để trích xuất siêu dữ liệu trạng thái xử lý âm lượng (LPSM) và cũng có thể bao gồm siêu dữ liệu biên chương trình, và siêu dữ liệu khác từ mỗi khung của âm thanh đầu vào được mã hóa, để gán ít nhất LPSM (và siêu dữ liệu biên chương trình nếu nó được trích xuất) đến bộ xác minh trạng thái âm thanh 203 và tầng 204, để gán LPSM (và cũng có thể bao gồm siêu dữ liệu biên chương trình) làm đầu ra (ví dụ, đến bộ xử lý sau 300), để trích xuất dữ liệu âm thanh từ âm thanh đầu vào được mã hóa, và để gán dữ liệu âm thanh được trích xuất đến bộ giải mã 202.

Đầu vào dòng bit âm thanh được mã hóa đến bộ giải mã 200 có thể là một trong số dòng bit AC-3, dòng bit E-AC-3, hoặc dòng bit Dolby E.

Hệ thống trên Fig.3 cũng bao gồm bộ xử lý sau 300. Bộ xử lý sau 300 bao gồm bộ đệm khung 301 và các phần tử xử lý khác (không được thể hiện trên hình vẽ) bao gồm ít nhất một phần tử xử lý được nối với bộ đệm 301. Bộ đệm khung 301 lưu trữ (ví dụ, theo

cách không chuyển tiếp) ít nhất một khung của dòng bit âm thanh được giải mã được thu bởi bộ xử lý sau 300 từ bộ giải mã 200. Các phần tử xử lý của bộ xử lý sau 300 được ghép và tạo cấu hình để nhận và xử lý điều hợp trình tự của các khung của đầu ra dòng bit âm thanh được giải mã từ bộ đệm 301, bằng cách sử dụng đầu ra siêu dữ liệu (bao gồm các trị số LPSM) từ bộ giải mã 202 và/hoặc đầu ra các bit điều khiển từ tầng 204 của bộ giải mã 200. Thông thường, bộ xử lý sau 300 được tạo cấu hình để thực hiện xử lý âm lượng thích ứng trên dữ liệu âm thanh được giải mã bằng cách sử dụng các trị số LPSM và cũng có thể bao gồm siêu dữ liệu biên chương trình (ví dụ, dựa trên trạng thái xử lý âm lượng, và/hoặc một hoặc nhiều đặc tính dữ liệu âm thanh, được chỉ báo bởi LPSM cho dữ liệu âm thanh chỉ báo một chương trình âm thanh duy nhất).

Nhiều phương án thực hiện khác nhau của bộ giải mã 200 và bộ xử lý sau 300 được tạo cấu hình để thực hiện các phương án khác của phương pháp theo sáng chế.

Bộ giải mã âm thanh 202 của bộ giải mã 200 được tạo cấu hình để giải mã dữ liệu âm thanh được trích xuất bởi bộ phân tích 205 để tạo ra dữ liệu âm thanh được giải mã, và để gán dữ liệu âm thanh được giải mã làm đầu ra (ví dụ, đến bộ xử lý sau 300).

Bộ xác minh trạng thái 203 được tạo cấu hình để xác thực và xác minh LPSM (và cũng có thể là siêu dữ liệu khác) được gán đến đó. Theo một số phương án, LPSM là (hoặc được bao gồm trong) khối dữ liệu mà đã được bao gồm trong dòng bit đầu vào (ví dụ, theo một phương án của sáng chế). Khối này có thể bao gồm băm mật mã (mã xác thực thông điệp dựa trên hàm băm hoặc “HMAC”) để xử lý LPSM (và cũng có thể bao gồm siêu dữ liệu khác) và/hoặc dữ liệu âm thanh nền (được cấp từ bộ phân tích 205 và/hoặc bộ giải mã 202 đến bộ xác minh 203). Khối dữ liệu có thể có chữ ký số theo các phương án này, sao cho đơn vị xử lý âm thanh phía sau có thể xác thực và xác minh siêu dữ liệu trạng thái xử lý một cách tương đối dễ dàng.

Các phương pháp mật mã khác bao gồm nhưng không giới hạn ở bất kỳ phương pháp nào trong số một hoặc nhiều các phương pháp mật mã khác HMAC có thể được sử dụng để xác minh LPSM (ví dụ, trong bộ xác minh 203) để đảm bảo việc truyền và nhận an toàn của LPSM và/hoặc dữ liệu âm thanh nền. Ví dụ, việc xác minh (bằng cách sử dụng phương pháp mật mã này) có thể được thực hiện trong mỗi đơn vị xử lý âm thanh mà thu dòng bit âm thanh theo phương án theo sáng chế để xác định siêu dữ liệu trạng

thái xử lý âm lượng và dữ liệu âm thanh tương ứng có được bao gồm trong dòng bit đã trải qua (và/hoặc là kết quả của) quá trình xử lý âm lượng cụ thể hay không (như được chỉ báo bởi siêu dữ liệu) và chưa được điều biến sau khi thực hiện quá trình xử lý âm lượng đặc biệt này.

Bộ xác minh trạng thái 203 gán dữ liệu điều khiển đến bộ tạo bit điều khiển 204, và/hoặc gán dữ liệu điều khiển làm đầu ra (ví dụ, đến bộ xử lý sau 300), để chỉ báo các kết quả của hoạt động xác minh. Để đáp lại dữ liệu điều khiển (và cũng có thể bao gồm siêu dữ liệu khác được trích xuất từ dòng bit đầu vào), tầng 204 có thể tạo ra (và gán đến đơn vị xử lý sau 300) hoặc:

các bit điều khiển chỉ báo rằng đầu ra dữ liệu âm thanh được giải mã từ bộ giải mã 202 đã trải qua kiểu xử lý âm lượng cụ thể (khi LPSM chỉ báo rằng đầu ra dữ liệu âm thanh từ bộ giải mã 202 đã trải qua kiểu xử lý âm lượng cụ thể, và các bit điều khiển từ bộ xác minh 203 chỉ báo rằng LPSM là hợp lệ); hoặc

các bit điều khiển chỉ báo rằng đầu ra dữ liệu âm thanh được giải mã từ bộ giải mã 202 cần trải qua kiểu xử lý âm lượng cụ thể (ví dụ, khi LPSM chỉ báo rằng đầu ra dữ liệu âm thanh từ bộ giải mã 202 chưa trải qua kiểu xử lý âm lượng cụ thể, hoặc khi LPSM chỉ báo rằng đầu ra dữ liệu âm thanh từ bộ giải mã 202 đã trải qua kiểu xử lý âm lượng cụ thể nhưng các bit điều khiển từ bộ xác minh 203 chỉ báo rằng LPSM là không hợp lệ).

Theo cách khác, bộ giải mã 200 gán siêu dữ liệu được trích xuất bởi bộ giải mã 202 từ dòng bit đầu vào, và LPSM (và cũng có thể là siêu dữ liệu biên chương trình) được trích xuất bởi bộ phân tích 205 từ dòng bit đầu vào đến bộ xử lý sau 300, và bộ xử lý sau 300 thực hiện xử lý âm lượng trên dữ liệu âm thanh được giải mã bằng cách sử dụng LPSM (và cũng có thể là siêu dữ liệu biên chương trình), hoặc thực hiện xác minh LPSM và sau đó thực hiện xử lý âm lượng trên dữ liệu âm thanh được giải mã bằng cách sử dụng LPSM (và cũng có thể là siêu dữ liệu biên chương trình) nếu việc xác minh chỉ báo rằng LPSM là hợp lệ.

Theo một số phương án, nếu bộ giải mã 200 thu dòng bit âm thanh được tạo ra theo một phương án của sáng chế bằng băm mật mã, bộ giải mã được tạo cấu hình để phân tích và truy hồi băm mật mã từ một khối dữ liệu được xác định từ dòng bit, khối này bao gồm siêu dữ liệu trạng thái xử lý âm lượng (LPSM). Bộ xác minh 203 có thể sử dụng

băm mật mã để xác minh dòng bit thu được và/hoặc siêu dữ liệu liên quan. Ví dụ, nếu bộ xác minh 203 nhận thấy LPSM là hợp lệ dựa trên sự ăn khớp giữa băm mật mã tham chiếu và băm mật mã được truy hồi từ khối dữ liệu, thì nó có thể báo hiệu cho đơn vị xử lý âm thanh phía sau (ví dụ, bộ xử lý sau 300, mà có thể là hoặc bao gồm đơn vị cân chỉnh âm lượng) để chuyển tiếp (không đổi) dữ liệu âm thanh của dòng bit. Ngoài ra, một cách tùy ý, hoặc theo cách khác, các loại kỹ thuật mật mã khác có thể được sử dụng thay cho phương pháp dựa trên băm mật mã.

Theo một số phương án thực hiện của bộ giải mã 200, dòng bit được mã hóa được thu (và được đệm trong bộ nhớ 201) là dòng bit AC-3 hoặc dòng bit E-AC-3, và bao gồm các đoạn dữ liệu âm thanh (ví dụ, các đoạn từ AB0 đến AB5 của khung được thể hiện trên Fig.4) và các đoạn siêu dữ liệu, trong đó các đoạn dữ liệu âm thanh chỉ báo dữ liệu âm thanh, và ít nhất mỗi đoạn trong số các đoạn siêu dữ liệu bao gồm siêu dữ liệu trạng thái xử lý âm lượng (LPSM) và cũng có thể bao gồm siêu dữ liệu biên chương trình. Tầng bộ giải mã 202 (và/hoặc bộ phân tích 205) được tạo cấu hình để trích xuất từ dòng bit LPSM (và cũng có thể bao gồm siêu dữ liệu biên chương trình) có định dạng sau. Mỗi đoạn siêu dữ liệu mà bao gồm LPSM (và cũng có thể bao gồm siêu dữ liệu biên chương trình) được bao gồm trong đoạn bit thừa của khung của dòng bit, hoặc trường “addbsi” của đoạn Thông tin Dòng bit (BSI) của khung của dòng bit, hoặc trong trường auxdata (ví dụ, đoạn AUX được thể hiện trên Fig.4) ở cuối của khung của dòng bit. Khung của dòng bit có thể bao gồm một hoặc hai đoạn siêu dữ liệu, mỗi đoạn có thể bao gồm LPSM, và nếu khung bao gồm hai đoạn siêu dữ liệu, thì một đoạn có thể có mặt trong trường addbsi của khung và đoạn kia có mặt trong trường AUX của khung. Theo một số phương án, mỗi đoạn siêu dữ liệu bao gồm LPSM bao gồm đoạn tải trọng LPSM (hoặc bộ chứa) có định dạng sau:

phần đầu (thường bao gồm một lệnh đồng bộ hóa xác định điểm bắt đầu của tải trọng LPSM, theo sau bởi các trị số xác định, ví dụ, phiên bản định dạng LPSM, độ dài, chu trình, đếm, và các trị số liên kết dòng phụ được chỉ báo trong Bảng 2 dưới đây); và

sau phần đầu, ít nhất một trị số chỉ báo thoại (ví dụ, tham số “(các) kênh thoại” trên Bảng 2) chỉ báo dữ liệu âm thanh tương ứng chỉ báo thoại hay không chỉ báo thoại (ví dụ, các kênh dữ liệu âm thanh tương ứng nào chỉ báo thoại);

ít nhất một trị số tuân thủ quy định âm lượng (ví dụ, tham số “Loại Quy định về Âm lượng” trên Bảng 2) chỉ báo dữ liệu âm thanh tương ứng có tuân thủ đúng theo một nhóm các quy định về âm lượng được chỉ định hay không;

ít nhất một trị số xử lý âm lượng (ví dụ, một hoặc nhiều tham số “cờ hiệu chỉnh âm lượng công thoát,” “Kiểu hiệu chỉnh âm lượng,” trên Bảng 2) chỉ báo ít nhất một kiểu xử lý âm lượng đã được thực hiện trên dữ liệu âm thanh tương ứng; và

ít nhất một trị số âm lượng (ví dụ, một hoặc nhiều tham số “Âm lượng có công theo ITU,” “Âm lượng có công tiếng nói ITU,” “Âm lượng ngắn kì 3 giây ITU (EBU 3341),” và “Mức thật của đỉnh” trên Bảng 2) chỉ báo ít nhất một đặc tính âm lượng (ví dụ, âm lượng đỉnh hoặc trung bình) của dữ liệu âm thanh tương ứng.

Theo một số phương án, mỗi đoạn siêu dữ liệu mà chứa LPSM và siêu dữ liệu biên chương trình chứa phần đầu chính (và cũng có thể chứa các phần tử chính bổ sung), và sau phần đầu chính (hoặc phần đầu chính và các phần tử chính khác) đoạn tải trọng LPSM (hoặc bộ chứa) có định dạng sau:

phần đầu, thường bao gồm ít nhất một trị số xác định (ví dụ, phiên bản định dạng LPSM, độ dài, chu trình, đếm, và các trị số liên kết dòng phụ, như được chỉ báo trong Bảng 2 dưới đây), và

sau phần đầu, LPSM và siêu dữ liệu biên chương trình. Siêu dữ liệu biên chương trình có thể bao gồm số lượng khung biên chương trình, và trị số mã (ví dụ, trị số “tồn tại_xê dịch”) chỉ báo khung chỉ bao gồm số lượng khung biên chương trình hay cả số lượng khung biên chương trình và trị số xê dịch), và (trong một số trường hợp) trị số xê dịch.

Theo một số phương án thực hiện, bộ phân tích 205 (và/hoặc tầng bộ giải mã 202) được tạo cấu hình để trích xuất từ đoạn bit thừa hoặc trường “addbsi” hoặc trường auxdata, của khung của dòng bit, mỗi đoạn siêu dữ liệu có định dạng sau:

phần đầu chính (thường bao gồm một lệnh đồng bộ hóa xác định sự bắt đầu của đoạn siêu dữ liệu, theo sau bởi ít nhất một trị số xác định, ví dụ, phiên bản phần tử chính, độ dài, và chu trình, số phần tử mở rộng, và các trị số liên kết dòng phụ được chỉ báo trong Bảng 1 dưới đây); và

sau phần đầu chính, ít nhất một trị số bảo vệ (ví dụ, thuật toán tiêu hóa HMAC và các trị số vân tay âm thanh trên Bảng 1) hữu dụng cho ít nhất một trong số các việc giải mã, xác thực, hoặc xác minh ít nhất một trong số các siêu dữ liệu trạng thái xử lý âm lượng hoặc dữ liệu âm thanh tương ứng); và

cũng sau phần đầu chính, nếu đoạn siêu dữ liệu bao gồm LPSM, xác định tải trọng LPSM (“ID”) và các trị số độ lớn tải trọng LPSM mà xác định siêu dữ liệu theo sau là tải trọng LPSM và chỉ báo độ lớn của tải trọng LPSM.

Đoạn tải trọng (hoặc bộ chứa) LPSM (tốt hơn là có định dạng nêu trên) theo sau các trị số ID tải trọng LPSM và độ lớn tải trọng LPSM.

Khái quát hơn, dòng bit âm thanh được mã hóa được tạo ra bởi các phương án được ưu tiên của sáng chế có cấu trúc nhằm mang lại cơ chế để dán nhãn các phần tử và phần tử phụ của siêu dữ liệu là các phần tử chính (bắt buộc) hoặc mở rộng (các bộ phận tùy chọn). Điều này cho phép tốc độ dữ liệu của dòng bit (bao gồm siêu dữ liệu của nó) thích hợp cho nhiều ứng dụng. Các phần tử chính (bắt buộc) có cú pháp dòng bit được ưu tiên cũng cần có khả năng báo hiệu rằng các phần tử mở rộng (tùy chọn) kết hợp với nội dung âm thanh có mặt (trong dải) và/hoặc tại một vị trí từ xa (ngoài dải).

(Các) bộ phận chính cần có mặt ở mọi khung của dòng bit. Một số phần tử phụ của các phần tử chính là tùy chọn và có thể có mặt dưới dạng kết hợp bất kỳ. Các phần tử mở rộng không cần thiết phải có mặt ở mọi khung (để hạn chế tốc độ bit quá cao). Do đó, các phần tử mở rộng có thể có mặt ở một số khung và không có mặt ở các khung khác. Một số phần tử phụ của phần tử mở rộng là tùy chọn và có thể có mặt dưới dạng kết hợp bất kỳ, trong khi một số phần tử phụ của phần tử mở rộng có thể là bắt buộc (tức là, nếu phần tử mở rộng có mặt trong khung của dòng bit).

Theo một nhóm các phương án, dòng bit âm thanh được mã hóa bao gồm trình tự của các đoạn dữ liệu âm thanh và các đoạn siêu dữ liệu được tạo ra (ví dụ, bởi đơn vị xử lý âm thanh mà thể hiện sáng chế). Các đoạn dữ liệu âm thanh chỉ báo dữ liệu âm thanh, ít nhất mỗi đoạn trong số các đoạn siêu dữ liệu bao gồm siêu dữ liệu trạng thái xử lý âm lượng (LPSM) và cũng có thể bao gồm siêu dữ liệu biên chương trình, và các đoạn dữ liệu âm thanh được dồn kênh phân thời với các đoạn siêu dữ liệu. Theo các phương án

được ưu tiên thuộc nhóm này, mỗi đoạn siêu dữ liệu có định dạng ưu tiên được mô tả trong bản mô tả này.

Theo một định dạng ưu tiên, dòng bit được mã hóa là dòng bit AC-3 hoặc dòng bit E-AC-3, và mỗi đoạn siêu dữ liệu mà bao gồm LPSM được bao gồm (ví dụ, bởi tầng 107 theo một phương án thực hiện ưu tiên của bộ mã hóa 100) như thông tin dòng bit bổ sung trong trường “addbsi” (được thể hiện trên Fig.6) của đoạn Thông tin Dòng bit (BSI) của khung của dòng bit, hoặc trong trường auxdata của khung của dòng bit, hoặc trong đoạn bit thừa của khung của dòng bit.

Ở định dạng ưu tiên, mỗi khung bao gồm một phần tử chính có định dạng được thể hiện trong Bảng 1 dưới đây, trong trường addbsi (hoặc đoạn bit thừa) của khung:

Bảng 1

Tham số	Mô tả	Bắt buộc/Tùy chọn
SYNC [ID]	Lệnh đồng bộ có thể là một nhóm trị số 16 bit đến trị số 0x5838	Bắt buộc
Phiên bản phần tử chính		Bắt buộc
Độ dài phần tử chính		Bắt buộc
Chu trình phần tử chính (xxx)		Bắt buộc
Số lượng phần tử mở rộng	Chỉ báo số lượng các phần tử siêu dữ liệu mở rộng kết hợp với phần tử chính. Trị số này có thể tăng/giảm khi dòng bit được chuyển từ khâu sản xuất đến phân phối và phát sóng	Bắt buộc

	cuối cùng.	
Kết hợp các dòng phụ	Mô tả (các) dòng phụ nào được liên kết với phần tử chính	Bắt buộc
Chữ ký (giải thuật tiêu hóa HMAC)	Giải thuật tiêu hóa HMAC 256 bit (bằng cách sử dụng thuật toán SHA-2) được tính toán trên dữ liệu âm thanh, phần tử chính, và tất cả các phần tử mở rộng, của toàn bộ khung.	Bắt buộc
Đếm ngược biên PGM	Trường chỉ xuất hiện ở một số khung ở đầu hoặc đuôi của tệp/dòng chương trình âm thanh. Do đó, thay đổi về phiên bản phần tử chính có thể được sử dụng để báo hiệu sự kết thúc của tham số này.	Tùy chọn
Vân tay âm thanh	Vân tay âm thanh được lấy từ các mẫu âm thanh PCM được thể hiện bởi trường chu trình phần tử chính.	Tùy chọn
Vân tay video	Vân tay video được lấy từ các mẫu video được nén (nếu có) được thể hiện bởi trường chu trình phần tử chính.	Tùy chọn
URL/UUID	Trường này được xác định để mang URL và/hoặc UUID (nó có thể không cần thiết cho vân tay) mà tham chiếu vị trí ngoài	Tùy chọn

	của nội dung (bản chất) chương trình bổ sung và/hoặc siêu dữ liệu kết hợp với dòng bit.	
--	---	--

Theo định dạng ưu tiên, mỗi trường addbsi (hoặc auxdata) hoặc đoạn bit thừa mà chứa LPSM chứa phần đầu chính (và cũng có thể bao gồm các phần tử chính bổ sung), và sau phần đầu chính (hoặc phần đầu chính và các phần tử chính khác), các trị số (các tham số) LPSM dưới đây:

ID tải trọng (xác định siêu dữ liệu là LPSM) theo sau các trị số phần tử chính (ví dụ, như được chỉ rõ trong Bảng 1);

độ lớn tải trọng (chỉ báo độ lớn của tải trọng LPSM) theo sau ID tải trọng; và

dữ liệu LPSM (theo sau ID tải trọng và trị số độ lớn tải trọng) có định dạng như được chỉ báo trong bảng dưới đây (Bảng 2):

Bảng 2

Tham số LPSM [Âm lượng Thông minh]	Mô tả	Số lượng các trạng thái đặc biệt	Bắt buộc/Tùy chọn	Tốc độ chèn (Chu trình cập nhật tham số)
Phiên bản LPSM			Bắt buộc	
Chu trình LPSM (xxx)	Chỉ áp dụng được với các trường xxx		Bắt buộc	

Số lượng LPSM			Bắt buộc	
Kết hợp dòng phụ LPSM			Bắt buộc	
(Các) kênh thoại	Chỉ báo dạng kết hợp nào của các kênh âm thanh L, C & R chứa tiếng nói trong 0,5 giây trước đó. Nếu tiếng nói không có mặt ở bất kỳ dạng kết hợp L, C hoặc R nào, thì tham số này sẽ chỉ báo “không có thoại”	8	Bắt buộc	~0,5 giây (thông thường)
Kiểu Quy định Âm	Chỉ báo ràng dòng	8	Bắt buộc	Khung

lượng	dữ liệu âm thanh được kết hợp tuân thủ đúng theo một nhóm các quy định cụ thể (ví dụ, ATSC A/85 hoặc EBU R128)			
Cờ hiệu hiệu chỉnh âm lượng có cổng thoại	Chỉ báo nếu dòng âm thanh được kết hợp đã được hiệu chỉnh dựa trên cổng thoại	2	Tùy chọn (chỉ có mặt nếu Kiểu_Quy định_Âm lượng chỉ báo rằng âm thanh tương ứng KHÔNG ĐƯỢC HIỆU CHỈNH)	Khung
Kiểu hiệu chỉnh Âm lượng	Chỉ báo nếu dòng âm thanh được kết hợp đã được hiệu chỉnh với tầm nhìn xa	2	Tùy chọn (chỉ có mặt nếu Kiểu_Quy định_Âm lượng chỉ báo rằng âm thanh tương ứng KHÔNG ĐƯỢC HIỆU CHỈNH)	Khung

	vô hạn (dựa trên tệp) hay với bộ điều khiển dải động và âm lượng thời gian thực (realtime - RT).			
Âm lượng có cổng theo ITU (INF)	Chỉ báo âm lượng được tích hợp ITU-R BS.1770-3 của dòng âm thanh được kết hợp mà không có siêu dữ liệu được áp dụng (ví dụ, 7 bit: -58 -> +5,5 LKFS các bước 0,5 LKFS)	128	Tùy chọn	1 giây
Âm lượng có cổng	Chỉ báo âm lượng được	128	Tùy chọn	1 giây

tiếng nói ITU (INF)	tích hợp ITU-R BS.1770- 1/3 của tiếng nói/thoại của dòng âm thanh được kết hợp mà không có siêu dữ liệu được áp dụng (ví dụ, 7 bit: -58 -> +5,5 LKFS các bước 0,5 LKFS)			
Âm lượng Ngắn kì 3 giây ITU (EBU 3341)	Chỉ báo âm lượng ITU (ITU- BS.1771-1) không có cổng 3 giây của dòng âm thanh được kết hợp mà không có	256	Tùy chọn	0,1 giây

	siêu dữ liệu được áp dụng (cửa sổ trượt) ở tốc độ chèn $\sim 10\text{Hz}$ (ví dụ, 8bit: $116 \rightarrow$ $+11,5$ LKFS các bước $0,5 \text{ LKFS}$)			
Trị số Mức thật của Đỉnh	Chỉ báo trị số Mức thật của Đỉnh ITU- R BS.1770- 3 Phụ lục 2 (dB TP) của dòng âm thanh được kết hợp mà không có siêu dữ liệu được áp dụng. (tức là, trị số lớn nhất trong chu trình khung	256	Tùy chọn	0,5 giây

	được báo hiệu trong trường chu trình phân tử) 116 -> +11,5 LKFS các bước 0,5 LKFS			
Xê dịch trộn giảm	Chỉ báo mức xê dịch âm lượng trộn giảm			
Biên Chương trình	Chỉ báo, trong các khung, khi biên chương trình sẽ hoặc đã xuất hiện. Khi biên chương trình không ở biên khung, xê dịch mẫu tùy chọn sẽ			

	chỉ báo biên chương trình thực tế xuất hiện bao xa trong khung			
--	---	--	--	--

Trong định dạng ưu tiên khác của dòng bit được mã hóa được tạo ra theo sáng chế, dòng bit là dòng bit AC-3 hoặc dòng bit E-AC-3, và mỗi đoạn siêu dữ liệu mà bao gồm LPSM (và cũng có thể bao gồm siêu dữ liệu biên chương trình) được bao gồm (ví dụ, bởi tầng 107 theo một phương án thực hiện ưu tiên của bộ mã hóa 100) trong bất kỳ: đoạn bit thừa của khung của dòng bit; hoặc trường “addbsi” (được thể hiện trên Fig.6) của đoạn Thông tin Dòng bit (BSI) của khung của dòng bit; hoặc trường auxdata (ví dụ, đoạn AUX được thể hiện trên Fig.4) ở cuối khung của dòng bit. Khung có thể bao gồm một hoặc hai đoạn siêu dữ liệu, mỗi đoạn siêu dữ liệu này bao gồm LPSM, và nếu khung bao gồm hai đoạn siêu dữ liệu, một đoạn có thể có mặt trong trường addbsi của khung và đoạn còn lại có mặt trong trường AUX của khung. Mỗi đoạn siêu dữ liệu bao gồm LPSM có định dạng được định rõ như trên có tham chiếu đến Bảng 1 và Bảng 2 trên đây (tức là, nó bao gồm các phần tử chính được định rõ trong Bảng 1, theo sau bởi ID tải trọng (xác định siêu dữ liệu là LPSM) và các trị số độ lớn tải trọng được chỉ rõ trên đây, theo sau bởi tải trọng (dữ liệu LPSM có định dạng như được chỉ báo trong Bảng 2).

Trong định dạng ưu tiên khác, dòng bit được mã hóa là dòng bit Dolby E, và mỗi đoạn siêu dữ liệu mà bao gồm LPSM (và cũng có thể bao gồm siêu dữ liệu biên chương trình) là N vị trí mẫu đầu tiên của khoảng dải bảo vệ Dolby E. Dòng bit Dolby E bao gồm một đoạn siêu dữ liệu như vậy bao gồm LPSM tốt hơn là bao gồm một trị số chỉ báo độ dài tải trọng LPSM được báo hiệu trong lệnh Pd của phần mở đầu SMPTE 337M (tốt hơn là, tốc độ lặp lệnh Pa SMPTE 337M vẫn giống với tốc độ khung video được kết hợp).

Trong một định dạng ưu tiên, trong đó dòng bit được mã hóa là dòng bit E-AC-3, mỗi đoạn siêu dữ liệu mà bao gồm LPSM (và cũng có thể bao gồm siêu dữ liệu biên chương trình) được bao gồm (ví dụ, bởi tầng 107 theo một phương án thực hiện ưu tiên của bộ mã hóa 100) làm thông tin dòng bit bổ sung trong đoạn bit thừa, hoặc trong trường "addbsi" của đoạn Thông tin Dòng bit (BSI), của khung của dòng bit. Tiếp theo, chúng tôi sẽ mô tả các khía cạnh bổ sung của việc mã hóa dòng bit E-AC-3 với LPSM trong định dạng ưu tiên này:

1. trong suốt quá trình tạo ra dòng bit E-AC-3, trong khi bộ mã hóa E-AC-3 (mà chèn các trị số LPSM vào dòng bit) "hoạt động," đối với mỗi khung (khung đồng bộ) được tạo ra, dòng bit cần bao gồm một khối siêu dữ liệu (bao gồm LPSM) được mang trong trường addbsi (hoặc đoạn bit thừa) của khung. Các bits cần thiết để mang khối siêu dữ liệu không nên làm tăng tốc độ bit của bộ mã hóa (độ dài khung);
2. Mỗi khối siêu dữ liệu (chứa LPSM) cần chứa các thông tin sau:

cờ_kiểu_hiệu chỉnh_âm lượng: trong đó '1' chỉ báo âm lượng của dữ liệu âm thanh tương ứng đã được hiệu chỉnh đứng trước bộ mã hóa, và '0' chỉ báo âm lượng đã được hiệu chỉnh bởi bộ hiệu chỉnh âm lượng được nhúng vào bộ mã hóa (ví dụ, bộ xử lý âm lượng 103 của bộ mã hóa 100 trên Fig.2);

kênh_tiếng nói: chỉ báo (các) kênh nguồn nào chứa tiếng nói (trong 0,5 giây trước đó). Nếu không có tiếng nói nào được phát hiện, thì điều này sẽ được chỉ báo như vậy;

âm lượng_tiếng nói: chỉ báo âm lượng tiếng nói được tích hợp của mỗi kênh âm thanh tương ứng mà chứa tiếng nói (trong 0,5 giây trước đó);

âm lượng_ITU: chỉ báo âm lượng ITU BS.1770-3 được tích hợp của mỗi kênh âm thanh tương ứng; và

độ khuếch đại: (các) độ khuếch đại hợp phần âm lượng để nghịch đảo trong bộ giải mã (để chứng minh tính khả nghịch);

3. Trong khi bộ mã hóa E-AC-3 (mà chèn các trị số LPSM vào dòng bit) "hoạt động" và đang nhận khung AC-3 với cờ 'tin cậy', bộ điều khiển âm lượng trong bộ mã hóa (ví dụ, bộ xử lý âm lượng 103 của bộ mã hóa 100 trên Fig.2) nên được bỏ qua. Các trị số DRC và dialnorm nguồn 'tin cậy' cần được chuyển tiếp (ví dụ, bởi bộ tạo 106 của bộ

- mã hóa 100) đến cấu phần bộ mã hóa E-AC-3 (ví dụ, tầng 107 của bộ mã hóa 100). Việc tạo khối LPSM tiếp tục và cờ_kiểu_hiệu chỉnh_âm lượng được đặt là '1'. Trình tự bỏ qua bộ điều khiển âm lượng phải được đồng bộ với sự bắt đầu của khung AC-3 được giải mã trong đó cờ 'tin cậy' xuất hiện. Trình tự bỏ qua bộ điều khiển âm lượng nên được thực hiện như sau: điều khiển lượng_cân chỉnh (leveler_amount control) giảm từ trị số 9 xuống trị số 0 trên 10 chu trình khối âm thanh (tức là 53,3 mili giây) và điều khiển bộ đếm_cân chỉnh_phía_sau (leveler_back_end_meter control) được đặt ở chế độ bỏ qua (hoạt động này sẽ dẫn đến sự chuyển tiếp liên tục). Thuật ngữ bỏ qua 'tin cậy' của bộ cân chỉnh có nghĩa là trị số dialnorm của dòng bit nguồn cũng được tái sử dụng ở đầu ra của bộ mã hóa. (ví dụ nếu dòng bit nguồn 'tin cậy' có trị số dialnorm là -30 thì đầu ra của bộ mã hóa cần sử dụng -30 cho trị số dialnorm xuất ra);
4. Trong khi bộ mã hóa E-AC-3 (mà chèn các trị số LPSM vào dòng bit) "hoạt động" và đang nhận khung AC-3 mà không có cờ 'tin cậy', bộ điều khiển âm lượng được nhúng vào bộ mã hóa (ví dụ, bộ xử lý âm lượng 103 của bộ mã hóa 100 trên Fig.2) cần hoạt động. Việc tạo ra khối LPSM tiếp tục và cờ_kiểu_hiệu chỉnh_âm lượng được đặt là '0'. Trình tự kích hoạt bộ điều khiển âm lượng cần được đồng bộ với sự bắt đầu của khung AC-3 được giải mã trong đó cờ 'tin cậy' biến mất. Trình tự kích hoạt bộ điều khiển âm lượng cần được thực hiện như sau: điều khiển lượng_cân chỉnh gia tăng từ trị số 0 xuống trị số 9 trên 1 chu trình khối âm thanh. (tức là 5,3 mili giây) và điều khiển bộ đếm_cân chỉnh_phía_sau được đặt ở chế độ 'hoạt động' (hoạt động này sẽ dẫn đến sự chuyển tiếp liên tục và bao gồm thiết lập lại sự tích hợp bộ đếm_phía_sau); và
 5. khi mã hóa, giao diện người dùng đồ họa (GUI) cần chỉ báo cho người dùng biết các tham số sau: "Chương trình Âm thanh Đầu vào: [Đáng tin cậy/Không đáng tin cậy]" – trạng thái của tham số này là dựa trên sự có mặt của cờ "tin cậy" trong tín hiệu đầu vào; và "Hiệu chỉnh Âm lượng Thời gian thực: [Cho phép/Không cho phép]" – tính trạng của tham số này dựa trên việc bộ điều khiển âm lượng này được nhúng vào bộ mã hóa có hoạt động hay không.

Khi giải mã dòng bit AC-3 hoặc E-AC-3 mà có LPSM (ở định dạng ưu tiên) được bao gồm trong đoạn bit thừa, hoặc trường "addbsi" của đoạn Thông tin Dòng bit (BSI), của mỗi khung của dòng bit, bộ giải mã cần phân tích dữ liệu khối LPSM (trong đoạn bit

thừa hoặc trường addbsi) và đưa tất cả các trị số LPSM được trích xuất đến giao diện người dùng đồ họa (GUI). Nhóm các trị số LPSM được trích xuất được làm mới ở mỗi khung.

Trong định dạng ưu tiên khác của dòng bit được mã hóa được tạo ra theo sáng chế, dòng bit được mã hóa là dòng bit AC-3 hoặc dòng bit E-AC-3, và mỗi đoạn siêu dữ liệu mà bao gồm LPSM được bao gồm (ví dụ, bởi tầng 107 theo một phương án thực hiện ưu tiên của bộ mã hóa 100) trong đoạn bit thừa, hoặc trong đoạn AUX, hoặc làm thông tin dòng bit bổ sung trong trường “addbsi” (được thể hiện trên Fig.6) của đoạn Thông tin Dòng bit (“BSI”), của khung của dòng bit. Trong định dạng này (mà là một biến thể của định dạng được mô tả như trên có tham chiếu đến các Bảng 1 và Bảng 2), mỗi trường addbsi (hoặc Aux hoặc bit thừa) mà chứa LPSM chứa các trị số LPSM sau đây:

các phần tử chính được chỉ rõ trong Bảng 1, theo sau bởi ID tải trọng (xác định siêu dữ liệu là LPSM) và các trị số độ lớn tải trọng, theo sau bởi tải trọng (dữ liệu LPSM) mà có định dạng sau (tương tự như các phần tử bắt buộc được biểu thị trong Bảng 2 nêu trên):

phiên bản của tải trọng LPSM: trường 2 bit mà chỉ báo phiên bản của tải trọng LPSM;

dialchan: trường 3 bit mà chỉ báo các kênh Trái, Phải và/hoặc Trung tâm của dữ liệu âm thanh tương ứng có chứa thoại tiếng nói hay không. Việc phân bổ bit của trường dialchan có thể là như sau: bit 0, mà chỉ báo sự có mặt của thoại trong kênh trái, được lưu trữ trong bit quan trọng nhất của trường dialchan; và bit 2, mà chỉ báo sự có mặt của thoại trong kênh trung tâm, được lưu trữ trong bit ít quan trọng nhất của trường dialchan.

Mỗi bit của trường dialchan được đặt là ‘1’ nếu kênh tương ứng chứa thoại tiếng nói trong 0,5 giây trước của chương trình;

loudregtyp: trường 4 bit mà chỉ báo chuẩn quy định âm lượng nào mà âm lượng chương trình tuân thủ. Đặt trường “loudregtyp” là ‘000’ chỉ báo rằng LPSM không chỉ báo tuân thủ quy định âm lượng. Ví dụ, một trị số của trường này (ví dụ, 0000) có thể chỉ báo rằng việc tuân thủ theo chuẩn quy định âm lượng là không được chỉ báo, trị số khác của trường này (ví dụ, 0001) có thể chỉ báo rằng dữ liệu âm thanh của chương trình tuân thủ đúng theo chuẩn ATSC A/85, và trị số khác của trường này (ví dụ, 0010) có thể chỉ

báo rằng dữ liệu âm thanh của chương trình tuân thủ đúng theo chuẩn EBU R128. Trong ví dụ, nếu trường được đặt trị số bất kỳ khác '0000', các trường loudcorrdialgat và loudcorrtyp cần theo sau trong tải trọng;

loudcorrdialgat: trường 1 bit mà chỉ báo việc hiệu chỉnh âm lượng công thoại có được áp dụng hay không. Nếu âm lượng của chương trình đã được hiệu chỉnh nhờ tạo công thoại, trị số của trường loudcorrdialgat được đặt là '1'. Nếu không thì nó được đặt là '0';

loudcorrtyp: trường 1 bit mà chỉ báo kiểu hiệu chỉnh âm lượng được áp dụng vào chương trình. Nếu âm lượng của chương trình đã được hiệu chỉnh bằng quá trình hiệu chỉnh âm lượng nhìn về phía trước vô hạn (dựa vào tệp), trị số của trường loudcorrtyp được đặt là '0'. Nếu âm lượng của chương trình đã được hiệu chỉnh bằng cách sử dụng kết hợp phép đo âm lượng thời gian thực và điều khiển dải động, thì trị số của trường này được đặt là '1';

loudrelgate: trường 1 bit mà chỉ báo dữ liệu âm lượng có cổng tương ứng (ITU) tồn tại hay không. Nếu trường loudrelgate được đặt là '1', trường ituloudrelgat 7 bit cần theo sau trong tải trọng;

loudrelgat: trường 7 bit mà chỉ báo âm lượng chương trình có cổng tương ứng (ITU). Trường này chỉ báo âm lượng được tích hợp của chương trình âm thanh, được đo theo ITU-R BS.1770-3 mà không có bất kỳ điều chỉnh độ khuếch đại nào do nén dải động và dialnorm đang được áp dụng. Các trị số từ 0 đến 127 được hiểu là từ -58 LKFS đến +5,5 LKFS, trong các bước 0,5 LKFS;

loudspchgate: trường 1 bit mà chỉ báo dữ liệu âm lượng có cổng tiếng nói (ITU) có tồn tại hay không. Nếu trường loudspchgate được đặt là '1', trường loudspchgat 7 bit cần theo sau trong tải trọng;

loudspchgat: trường 7 bit mà chỉ báo âm lượng chương trình có cổng tiếng nói. Trường này chỉ báo âm lượng được tích hợp của toàn bộ chương trình âm thanh tương ứng, được đo theo công thức (2) của ITU-R BS.1770-3 mà không có bất kỳ điều chỉnh độ khuếch đại nào do nén dải động và dialnorm đang được áp dụng. Các trị số từ 0 đến 127 được hiểu là từ -58 LKFS đến +5,5 LKFS, trong các bước 0,5 LKFS;

loudstrm3se: trường 1 bit mà chỉ báo dữ liệu âm lượng ngắn kì (3 giây) có tồn tại hay không. Nếu trường được đặt là '1', thì trường loudstrm3s 7-bit cần theo sau trong tải trọng;

loudstrm3s: trường 7 bit mà chỉ báo âm lượng không có công của 3 giây trước của chương trình âm thanh tương ứng, được đo theo ITU-R BS.1771-1 và không có bất kỳ điều chỉnh độ khuếch đại nào do nén dải động và dialnorm đang được áp dụng. Các trị số từ 0 đến 256 được hiểu là từ -116 LKFS đến +11,5 LKFS trong các bước 0,5 LKFS;

truepk: trường 1 bit mà chỉ báo mức thật của đỉnh dữ liệu âm lượng có tồn tại hay không. Nếu trường truepk được đặt là '1', thì trường truepk 8 bit cần theo sau trong tải trọng; và

truepk: trường 8 bit mà chỉ báo trị số mẫu mức thật của đỉnh của chương trình, được đo theo Phụ lục 2 của ITU-R BS.1770-3 và không có bất kỳ điều chỉnh độ khuếch đại nào do nén dải động và dialnorm đang được áp dụng. Các trị số từ 0 đến 256 được hiểu là từ -116 LKFS đến +11,5 LKFS trong các bước 0,5 LKFS.

Theo một số phương án, phần tử chính của một đoạn siêu dữ liệu trong đoạn bit thừa hoặc trong trường auxdata (hoặc trường "addbsi") của khung của dòng bit AC-3 hoặc dòng bit E-AC-3 bao gồm phần đầu chính (thường bao gồm các trị số xác định, ví dụ, phiên bản phần tử chính), và sau phần đầu chính: các trị số chỉ báo dữ liệu vãn tay (hoặc các trị số bảo vệ khác) có được bao gồm cho siêu dữ liệu của đoạn siêu dữ liệu hay không, các trị số chỉ báo dữ liệu ngoài (liên quan đến dữ liệu âm thanh tương ứng với siêu dữ liệu của đoạn siêu dữ liệu) có tồn tại hay không, ID tải trọng và các trị số độ lớn tải trọng cho mỗi loại siêu dữ liệu (ví dụ, LPSM, và/hoặc siêu dữ liệu thuộc loại khác ngoài LPSM) được xác định bởi phần tử chính, và các trị số bảo vệ cho ít nhất một loại siêu dữ liệu được xác định bởi phần tử chính. (Các) tải trọng siêu dữ liệu của đoạn siêu dữ liệu theo sau phần đầu chính, và được (trong một số trường hợp) lồng vào trong các trị số của phần tử chính.

Các phương án điển hình của sáng chế bao gồm siêu dữ liệu biên chương trình trong dòng bit âm thanh được mã hóa theo cách hiệu quả mà cho phép sự xác định chính xác và chắc chắn của ít nhất một biên giữa các chương trình âm thanh liên tiếp được chỉ báo bởi dòng bit. Các phương án điển hình cho phép sự xác định biên chương trình một

cách chính xác và chắc chắn sao cho chúng cho phép xác định chính xác biên chương trình ngay cả trong các trường hợp các dòng bit chỉ báo các chương trình khác nhau được ghép lại với nhau (để tạo ra dòng bit theo sáng chế) theo cách mà cắt một hoặc cả hai dòng bit được cắt (và do đó loại bỏ siêu dữ liệu biên chương trình mà đã được bao gồm trong ít nhất một trong số các dòng bit đã được cắt trước đó).

Theo các phương án điển hình, siêu dữ liệu biên chương trình trong khung dòng bit theo sáng chế là cờ biên chương trình chỉ báo số lượng khung. Thông thường, cờ chỉ báo số lượng khung ở giữa khung hiện thời (khung mà bao gồm cờ) và biên chương trình (đầu hoặc cuối của chương trình âm thanh hiện thời). Theo các phương án được ưu tiên, các cờ biên chương trình được chèn theo cách hiệu quả và đối xứng ở đầu và cuối mỗi đoạn dòng bit mà chỉ báo một chương trình duy nhất (tức là, trong các khung xuất hiện trong một số lượng khung định trước sau khi đoạn bắt đầu, và trong các khung xuất hiện trong một số lượng khung định trước trước khi đoạn kết thúc), sao cho khi hai đoạn dòng bit này được nối vào nhau (để chỉ báo trình tự của hai chương trình), siêu dữ liệu biên chương trình có thể có mặt (ví dụ, một cách đối xứng) ở cả hai phía của biên ở giữa hai chương trình.

Độ chắc chắn tối đa có thể đạt được bằng cách chèn cờ biên chương trình vào mỗi khung của dòng bit chỉ báo chương trình, nhưng điều này thường không khả thi do đi kèm với việc gia tăng tốc độ dữ liệu. Theo các phương án điển hình, các cờ biên chương trình chỉ được chèn vào một nhóm nhỏ của khung dòng bit âm thanh được mã hóa (mà có thể chỉ báo một chương trình âm thanh hoặc trình tự các chương trình âm thanh), và tốc độ chèn cờ biên là một hàm số không tăng của khoảng cách tăng của mỗi khung dòng bit (trong đó cờ được chèn) từ biên chương trình mà gần nhất với mỗi khung nêu trên, trong đó “tốc độ chèn cờ biên” thể hiện tỉ lệ trung bình giữa số lượng khung (chỉ báo chương trình) mà bao gồm cờ biên chương trình so với số lượng khung (chỉ báo chương trình) mà không bao gồm cờ biên chương trình, trong đó số trung bình là trung bình trên một số (ví dụ, số tương đối nhỏ) khung liên tiếp của dòng bit âm thanh được mã hóa.

Việc gia tăng tốc độ chèn cờ biên (ví dụ, ở các vị trí trong dòng bit gần với biên chương trình hơn) làm tăng tốc độ dữ liệu cần thiết để cấp dòng bit. Để bù đắp cho điều này, độ lớn (số bit) của mỗi cờ được chèn tốt hơn là được giảm trong khi tốc độ chèn cờ biên được tăng (ví dụ, sao cho độ lớn của cờ biên chương trình trong khung thứ “N” của

dòng bit, trong đó N là một số nguyên, là một hàm số không tăng của khoảng cách (số lượng khung) giữa khung thứ “ N ” và biên chương trình gần nhất). Theo một nhóm các phương án, tốc độ chen cờ biên là hàm giảm theo lôga của khoảng cách ngày càng tăng (của mỗi vị trí chen cờ) từ biên chương trình gần nhất, và cho mỗi khung chứa cờ mà bao gồm một trong số các cờ, độ lớn của cờ trong khung chứa cờ nêu trên bằng hoặc lớn hơn độ lớn của mỗi cờ trong khung nằm gần với biên chương trình gần nhất hơn so với khung chứa cờ nêu trên. Thông thường, độ lớn của mỗi cờ được xác định bởi một hàm tăng của số lượng khung từ vị trí chen cờ đến biên chương trình gần nhất.

Ví dụ, xem xét các phương án trên Fig.8 và Fig.9, trong đó mỗi cột được xác định bởi số khung (ở hàng đầu) chỉ báo khung dòng bit âm thanh được mã hóa. Dòng bit chỉ báo một chương trình âm thanh có biên chương trình thứ nhất (chỉ báo sự bắt đầu của chương trình) mà xuất hiện ngay lập tức ở bên trái cột được xác định bởi khung số “17” ở bên trái trên Fig.9, và biên chương trình thứ hai (chỉ báo sự kết thúc của chương trình) mà xuất hiện ngay lập tức ở bên phải cột được xác định bởi khung số “1” ở bên phải trên Fig.8. Các cờ biên chương trình được bao gồm trong khung được thể hiện trên Fig.8 đếm lùi số lượng khung giữa khung hiện thời và biên chương trình thứ hai. Các cờ biên chương trình được bao gồm trong khung được thể hiện trên Fig.9 đếm tăng số lượng khung giữa khung hiện thời và biên chương trình thứ nhất.

Theo phương án trên các hình vẽ Fig.8 và Fig.9, cờ biên chương trình chỉ được chen trong mỗi khung thứ “ 2^N ” của X khung đầu tiên của dòng bit được mã hóa sau điểm bắt đầu của chương trình âm thanh được chỉ báo bởi dòng bit, và trong mỗi khung thứ “ 2^N ” (của X khung cuối cùng của dòng bit) gần nhất với điểm cuối của chương trình được chỉ báo bởi dòng bit, trong đó chương trình bao gồm Y khung, X là một số nguyên nhỏ hơn hoặc bằng $Y/2$, và N là một số nguyên dương nằm trong khoảng từ 1 đến $\log_2(X)$. Do đó (như được chỉ báo trên các hình vẽ Fig.8 và Fig.9), cờ biên chương trình được chen vào khung thứ hai ($N = 1$) của dòng bit (khung chứa cờ gần nhất với điểm bắt đầu của chương trình), trong khung thứ tư ($N = 2$), trong khung thứ tám ($N = 3$), v.v., và trong khung thứ tám từ cuối dòng bit, trong khung thứ tư từ cuối dòng bit, và trong khung thứ hai từ cuối dòng bit (khung chứa cờ gần nhất với điểm kết thúc của chương trình). Trong ví dụ này, cờ biên chương trình trong khung thứ “ 2^N ” từ điểm bắt đầu (hoặc kết

thức) của chương trình bao gồm các bit nhị phân $\log_2(2^{N+2})$, như được chỉ báo trong các hình vẽ Fig.8 và Fig.9. Do đó, cờ biên chương trình trong khung thứ hai ($N = 1$) từ điểm bắt đầu (hoặc kết thúc) của chương trình bao gồm $\log_2(2^{N+2}) = \log_2(2^3) = 3$ bit nhị phân, và cờ trong khung thứ tư ($N = 2$) từ điểm bắt đầu (hoặc kết thúc) của chương trình bao gồm $\log_2(2^{N+2}) = \log_2(2^4) = 4$ bit nhị phân, v.v.

Trong ví dụ trên các hình vẽ Fig.8 và Fig.9, định dạng của mỗi cờ biên chương trình là như sau. Mỗi cờ biên chương trình bao gồm bit “1” dẫn đầu, trình tự của các bit “0” (hoặc không có bit “0” hoặc một hoặc nhiều bit “0” liên tiếp) sau bit dẫn đầu, và một mã vạch 2 bit. Mã vạch là “11” cho các cờ trong X khung cuối cùng của dòng bit (các khung gần với cuối chương trình nhất), như được chỉ báo trên Fig.8. Mã vạch là “10” cho các cờ trong X khung đầu tiên của dòng bit (các khung gần với đầu chương trình nhất), như được chỉ báo trên Fig.9. Do đó, để đọc (giải mã) mỗi cờ, số điểm zero giữa bit “1” dẫn đầu và mã vạch được đếm. Nếu mã vạch được xác định là “11”, cờ chỉ báo rằng có $(2^{Z+1} - 1)$ khung giữa khung hiện thời (khung mà bao gồm cờ) và phần cuối chương trình, trong đó Z là số lượng các điểm zero giữa bit “1” dẫn đầu của cờ và mã vạch. Bộ giải mã có thể được thực hiện một cách hiệu quả để bỏ qua bit đầu tiên và cuối cùng của mỗi cờ này, để xác định trình tự đảo ngược của các bit khác (trung gian) của cờ (ví dụ, nếu trình tự của các bit trung gian là “0001” với bit “1” là bit cuối cùng trong trình tự, trình tự đảo ngược của các bit trung gian là “1000” với bit “1” là bit đầu tiên trong trình tự đảo ngược), và để xác định trị số nhị phân của trình tự đảo ngược của các bit trung gian làm chỉ số của khung hiện thời (khung mà trong đó cờ được bao gồm) liên quan đến phần cuối chương trình. Ví dụ, nếu trình tự đảo ngược của các bit trung gian là “1000”, trình tự đảo ngược này có trị số nhị phân $2^4 = 16$, và khung được xác định là khung thứ 16 trước phần cuối của chương trình (như được chỉ báo trong cột trên Fig.8 trong đó mô tả khung “0”).

Nếu mã vạch được xác định là “10”, thì cờ chỉ báo rằng có $(2^{Z+1} - 1)$ khung giữa điểm bắt đầu của chương trình và khung hiện thời (khung mà bao gồm cờ), trong đó Z là số lượng các điểm zero giữa bit “1” dẫn đầu của cờ và mã vạch. Bộ giải mã có thể được thực hiện một cách hiệu quả để bỏ qua bit đầu tiên và cuối cùng của mỗi cờ này, để xác định trình tự đảo ngược của các bit trung gian của cờ (ví dụ, nếu trình tự của các bit trung gian là “0001” với bit “1” là bit cuối cùng trong trình tự, trình tự đảo ngược của các bit

trung gian là “1000” với bit “1” là bit đầu tiên trong trình tự đảo ngược), và để xác định trị số nhị phân của trình tự đảo ngược của các bit trung gian làm chỉ số của khung hiện thời (khung mà trong đó cờ được bao gồm) liên quan đến phần đầu chương trình. Ví dụ, nếu trình tự đảo ngược của các bit trung gian là “1000”, trình tự đảo ngược này có trị số nhị phân $2^4 = 16$, và khung được xác định là khung thứ 16 sau phần đầu chương trình (như được chỉ báo trong cột trên Fig.9 trong đó mô tả khung “32”).

Trong ví dụ trên các hình vẽ Fig.8 và Fig.9, cờ biên chương trình chỉ có mặt trong mỗi khung thứ “ 2^N ” của X khung đầu tiên của dòng bit được mã hóa sau phần đầu của một chương trình âm thanh được chỉ báo bởi dòng bit, và trong mỗi khung thứ “ 2^N ” (của X khung cuối cùng của dòng bit) gần nhất với phần cuối của chương trình được chỉ báo bởi dòng bit, trong đó chương trình bao gồm Y khung, X là một số nguyên nhỏ hơn hoặc bằng $Y/2$, và N là một số nguyên dương nằm trong khoảng từ 1 đến $\log_2(X)$. Việc bao gồm các cờ biên chương trình chỉ bổ sung tốc độ bit trung bình ;à 1,875 bit/khung vào tốc độ bit cần thiết để truyền dòng bit mà không có các cờ.

Trong một phương án thực hiện điển hình trên các hình vẽ Fig.8 và Fig.9 trong đó dòng bit là dòng bit âm thanh được mã hóa AC-3, mỗi khung chứa nội dung âm thanh và siêu dữ liệu cho 1536 mẫu âm thanh kỹ thuật số. Đối với tốc độ lấy mẫu 48 kHz, điều này thể hiện 32 mili giây âm thanh kỹ thuật số hoặc tốc độ 31,25 khung trên giây âm thanh. Do đó, theo một phương án, cờ biên chương trình trong khung được tách bởi một số khung (“X” khung) ra khỏi biên chương trình chỉ báo rằng biên xuất hiện 32X mili giây sau phần cuối của khung chứa cờ (hoặc 32X mili giây trước phần bắt đầu của khung chứa cờ).

Theo một phương án thực hiện điển hình trên các hình vẽ Fig.8 và Fig.9 trong đó dòng bit là dòng bit âm thanh được mã hóa E-AC-3, mỗi khung của dòng bit chứa nội dung âm thanh và siêu dữ liệu cho 256, 512, 768 hoặc 1536 mẫu âm thanh kỹ thuật số, tùy thuộc vào khung chứa lần lượt một, hai, ba hoặc sáu khối dữ liệu âm thanh. Đối với tốc độ lấy mẫu là 48 kHz, điều này thể hiện lần lượt 5,333, 10,667, 16 hoặc 32 mili giây âm thanh kỹ thuật số hoặc tốc độ lần lượt là 189,9, 93,75, 62,5 hoặc 31,25 khung trên giây âm thanh. Do đó, theo phương án này (giả định là mỗi khung chỉ báo 32 mili giây âm thanh kỹ thuật số), cờ biên chương trình trong khung được tách bởi một số khung

(“X” khung) ra khỏi biên chương trình chỉ báo rằng biên xuất hiện 32X mili giây sau khi kết thúc khung chứa cờ (hoặc 32X mili giây trước khi bắt đầu khung chứa cờ).

Theo một số phương án trong đó biên chương trình có thể xuất hiện trong khung của dòng bit âm thanh (tức là, không thẳng hàng với điểm đầu hoặc cuối của khung), siêu dữ liệu biên chương trình được bao gồm trong khung của dòng bit bao gồm số lượng khung biên chương trình (tức là, siêu dữ liệu chỉ báo số lượng khung hoàn chỉnh giữa điểm đầu và cuối của khung chứa số lượng khung và biên chương trình) và trị số xê dịch. Trị số xê dịch chỉ báo độ xê dịch (thường là một số mẫu) giữa điểm đầu và cuối của khung chứa biên chương trình, và vị trí thực của biên chương trình trong khung chứa biên chương trình.

Dòng bit âm thanh được mã hóa có thể chỉ báo trình tự của các chương trình (các rãnh âm thanh) của trình tự tương ứng của các chương trình video, và các biên của các chương trình âm thanh này có xu hướng xuất hiện ở các mép của các khung video hơn là ở các mép của các khung âm thanh. Ngoài ra, một số bộ mã hóa-giải mã âm thanh (ví dụ, bộ mã hóa-giải mã E-AC-3) sử dụng các độ lớn khung âm thanh mà không phù hợp với khung video. Hơn nữa, trong một số trường hợp dòng bit âm thanh được mã hóa ban đầu trải qua quá trình chuyển mã để tạo ra dòng bit được chuyển mã, và dòng bit được mã hóa ban đầu có độ lớn khung khác với dòng bit được chuyển mã sao cho biên chương trình (được xác định bởi dòng bit được mã hóa ban đầu) không được đảm bảo để xuất hiện ở biên khung của dòng bit được chuyển mã. Ví dụ, nếu dòng bit được mã hóa ban đầu (ví dụ, dòng bit “IEB” trên Fig.10) có độ lớn khung là 1536 mẫu trên khung, và dòng bit được chuyển mã (ví dụ, dòng bit “TB” trên Fig.10) có độ lớn khung là 1024 mẫu trên khung, quá trình chuyển mã có thể làm biên chương trình thực không xuất hiện ở biên khung của dòng bit được chuyển mã mà ở một nơi nào đó trong khung của nó (ví dụ, 512 mẫu thành khung của dòng bit được chuyển mã, như được chỉ báo trên Fig.10), do độ lớn khung khác nhau của các bộ mã hóa-giải mã khác nhau. Các phương án của sáng chế trong đó siêu dữ liệu biên chương trình được bao gồm trong khung dòng bit âm thanh được mã hóa bao gồm trị số xê dịch cũng như số lượng khung biên chương trình là hữu dụng trong ba trường hợp nêu trong đoạn này (cũng như trong các trường hợp khác).

Phương án được mô tả như trên có tham chiếu đến các hình vẽ Fig.8 và Fig.9 không bao gồm trị số xê dịch (ví dụ, trường xê dịch) trong bất kỳ khung nào của dòng bit

được mã hóa. Theo các biến thể của phương án này, trị số xê dịch được bao gồm trong mỗi khung của dòng bit âm thanh được mã hóa mà bao gồm cờ biên chương trình (ví dụ, trong các khung tương ứng với các khung được đánh số 0, 8, 12, và 14 trên Fig.8, và các khung được đánh số 18, 20, 24, và 32 trên Fig.9).

Theo một nhóm các phương án, cấu trúc dữ liệu (trong mỗi khung của dòng bit được mã hóa mà chứa siêu dữ liệu biên chương trình theo sáng chế) bao gồm trị số mã chỉ báo khung chỉ bao gồm số lượng khung biên chương trình, hay cả số lượng khung biên chương trình và trị số xê dịch. Ví dụ, trị số mã có thể là trị số của trường bit đơn (dưới đây được gọi là trường “tồn tại_xê dịch”), trị số “tồn tại_xê dịch” = 0 có thể chỉ báo rằng không có trị số xê dịch được bao gồm trong khung, và trị số “tồn tại_xê dịch” = 1 có thể chỉ báo rằng cả số lượng khung biên chương trình và trị số xê dịch đều được bao gồm trong khung.

Theo một số phương án, ít nhất một khung của dòng bit âm thanh được mã hóa AC-3 hoặc E-AC-3 bao gồm một đoạn siêu dữ liệu mà bao gồm LPSM và siêu dữ liệu biên chương trình (và cũng có thể bao gồm siêu dữ liệu khác) cho một chương trình âm thanh được xác định bởi dòng bit. Mỗi đoạn siêu dữ liệu này (mà có thể được bao gồm trong trường addbsi, hoặc trường auxdata, hoặc đoạn bit thừa của dòng bit) chứa phần đầu chính (và cũng có thể bao gồm các phần tử chính bổ sung), và sau phần đầu chính (hoặc phần đầu chính và các phần tử chính khác) đoạn tải trọng LPSM (hoặc bộ chứa) có định dạng sau:

phần đầu (thường bao gồm ít nhất một trị số xác định, ví dụ, phiên bản định dạng LPSM, độ dài, chu trình, đếm, và các trị số liên kết dòng phụ), và

sau phần đầu, siêu dữ liệu biên chương trình (mà có thể bao gồm số lượng khung biên chương trình, trị số mã (ví dụ, trị số “tồn tại_xê dịch”) chỉ báo khung chỉ bao gồm số lượng khung biên chương trình hay cả số lượng khung biên chương trình và trị số xê dịch, và trong một số trường hợp bao gồm trị số xê dịch) và LPSM. LPSM có thể bao gồm:

ít nhất một trị số chỉ báo thoại chỉ báo dữ liệu âm thanh tương ứng chỉ báo thoại hay không chỉ báo thoại (ví dụ, kênh dữ liệu âm thanh tương ứng nào chỉ báo thoại).

(Các) trị số chỉ báo thoại có thể chỉ báo thoại có mặt dưới dạng kết hợp bất kỳ của, hoặc tất cả, các kênh của dữ liệu âm thanh tương ứng;

ít nhất một trị số tuân thủ quy định âm lượng chỉ báo dữ liệu âm thanh tương ứng có tuân thủ đúng theo một nhóm các quy định về âm lượng được chỉ định hay không;

ít nhất một trị số xử lý âm lượng chỉ báo ít nhất một kiểu xử lý âm lượng đã được thực hiện trên dữ liệu âm thanh tương ứng; và

ít nhất một trị số âm lượng chỉ báo ít nhất một đặc tính âm lượng (ví dụ, âm lượng đỉnh hoặc trung bình) của dữ liệu âm thanh tương ứng.

Theo một số phương án, đoạn tải trọng LPSM bao gồm trị số mã (trị số “tồn tại_xê dịch”) chỉ báo khung chỉ bao gồm số lượng khung biên chương trình hay cả số lượng khung biên chương trình và trị số xê dịch. Ví dụ, theo phương án này, khi trị số mã này chỉ báo (ví dụ, khi “tồn tại_xê dịch” = 1) rằng khung bao gồm số lượng khung biên chương trình và trị số xê dịch, đoạn tải trọng LPSM có thể bao gồm trị số xê dịch mà là số nguyên không dấu 11 bit (tức là, có trị số từ 0 đến 2048) và chỉ báo số lượng mẫu âm thanh bổ sung giữa biên khung được báo hiệu (biên của khung mà bao gồm biên chương trình) và biên chương trình thực. Nếu số lượng khung biên chương trình chỉ báo số lượng khung (ở tốc độ khung hiện thời) đến khung chứa biên chương trình, vị trí chính xác (trong các đơn vị của số lượng mẫu) của biên chương trình (liên quan đến đầu hoặc cuối khung mà bao gồm đoạn tải trọng LPSM) sẽ được tính toán là:

$$S = (\text{bộ đếm_khung} * \text{độ lớn khung}) + \text{độ xê dịch},$$

trong đó S là số mẫu đến biên chương trình (từ đầu hoặc cuối khung mà bao gồm đoạn tải trọng LPSM), “bộ đếm_khung” là số lượng khung được chỉ báo bởi số lượng khung biên chương trình, “độ lớn khung” là số mẫu trên khung, và “độ xê dịch” là số mẫu được chỉ báo bởi trị số xê dịch.

Một số phương án trong đó tốc độ chèn của các cờ biên chương trình tăng gần biên chương trình thực thực hiện quy tắc là trị số xê dịch không bao giờ được bao gồm trong khung nếu khung ít hơn hoặc bằng một số (“Y”) của khung từ khung mà bao gồm biên chương trình. Thông thường, $Y = 32$. Đối với bộ mã hóa E-AC-3 mà thực hiện quy tắc này (với $Y = 32$), bộ mã hóa không bao giờ chèn trị số xê dịch vào giây cuối cùng của chương trình âm thanh. Trong trường hợp này, thiết bị thu có trách nhiệm duy trì bộ định

thời và do đó thực hiện tính toán độ xê dịch của chính nó (để đáp lại siêu dữ liệu biên chương trình, bao gồm trị số xê dịch, trong khung của dòng bit được mã hóa mà nhiều hơn Y khung từ khung chứa biên chương trình).

Đối với các chương trình mà chương trình âm thanh của chúng được biết là “khung được căn chỉnh” cho các khung video của các chương trình video tương ứng (ví dụ, các đường cấp đóng góp điển hình với âm thanh được mã hóa Dolby E), sẽ là thừa nếu bao gồm các trị số xê dịch trong các dòng bit được mã hóa chỉ báo chương trình âm thanh. Do đó, các trị số xê dịch thường sẽ không được bao gồm trong các dòng bit được mã hóa này.

Xem trên Fig.11, sau đây xem xét các trường hợp trong đó các dòng bit âm thanh được mã hóa được ghép lại với nhau để tạo thành dòng bit âm thanh theo một phương án của sáng chế.

Dòng bit ở trên cùng Fig.11 (được gắn nhãn “Trường hợp 1”) chỉ báo toàn bộ chương trình âm thanh thứ nhất (P1) bao gồm siêu dữ liệu biên chương trình (các cờ biên chương trình, F) theo sau bởi toàn bộ chương trình âm thanh thứ hai (P2) mà cũng bao gồm siêu dữ liệu biên chương trình (các cờ biên chương trình, F). Các cờ biên chương trình trong phần cuối của chương trình thứ nhất (một số được thể hiện trên Fig.11) là giống hoặc tương tự với các cờ đã được mô tả với sự tham chiếu đến Fig.8, và xác định vị trí của biên giữa hai chương trình (tức là, biên ở đầu chương trình thứ hai). Các cờ biên chương trình trong phần đầu của chương trình thứ hai (một số được thể hiện trên Fig.11) là giống hoặc tương tự với các cờ đã được mô tả với sự tham chiếu đến Fig.9, và chúng cũng xác định vị trí của biên. Theo các phương án điển hình, bộ mã hóa hoặc bộ giải mã thực hiện một bộ định thời (được điều chỉnh bởi các cờ trong chương trình thứ nhất) mà đếm lùi đến biên chương trình, và cùng bộ định thời đó (được điều chỉnh bởi các cờ trong chương trình thứ hai) đếm tiến từ cùng một biên chương trình. Như được chỉ báo bởi đồ thị bộ định thời biên trong Trường hợp 1 trên Fig.11, việc đếm ngược của bộ định thời này (được điều chỉnh bởi các cờ trong chương trình thứ nhất) đạt đến zero ở biên, và việc đếm tiến của bộ định thời (được điều chỉnh bởi các cờ trong chương trình thứ hai) chỉ báo cùng một vị trí của biên.

Dòng bit thứ hai từ trên xuống trên Fig.11 (được gắn nhãn “Trường hợp 2”) chỉ báo toàn bộ chương trình âm thanh thứ nhất (P1) bao gồm siêu dữ liệu biên chương trình (các cờ biên chương trình, F) theo sau bởi toàn bộ chương trình âm thanh thứ hai (P2) mà không bao gồm siêu dữ liệu biên chương trình. Các cờ biên chương trình trong phần cuối của chương trình thứ nhất (một số được thể hiện trên Fig.11) là giống hoặc tương tự với các cờ đã được mô tả với sự tham chiếu đến Fig.8, và xác định vị trí của biên giữa hai chương trình (tức là, biên ở đầu của chương trình thứ hai), cũng giống như trong Trường hợp 1. Theo các phương án điển hình, bộ mã hóa hoặc bộ giải mã thực hiện bộ định thời (được điều chỉnh bởi các cờ trong chương trình thứ nhất) mà đếm lùi đến biên chương trình, và cùng bộ định thời đó (mà không được điều chỉnh thêm) tiếp tục đếm tiến từ biên chương trình (như được chỉ báo bởi đồ thị bộ định thời biên trong Trường hợp 2 trên Fig.11).

Dòng bit thứ ba từ trên xuống trên Fig.11 (được gắn nhãn “Trường hợp 3”) chỉ báo chương trình âm thanh thứ nhất được cắt (P1) mà bao gồm siêu dữ liệu biên chương trình (các cờ biên chương trình, F), và mà đã được nối với toàn bộ chương trình âm thanh thứ hai (P2) mà cũng bao gồm siêu dữ liệu biên chương trình (các cờ biên chương trình, F). Việc ghép nối đã loại bỏ “N” khung cuối cùng của chương trình thứ nhất. Các cờ biên chương trình trong phần đầu của chương trình thứ hai (một số được thể hiện trên Fig.11) là giống hoặc tương tự với các cờ đã được mô tả với sự tham chiếu đến Fig.9, và chúng xác định vị trí của biên (chỗ ghép nối) giữa chương trình thứ nhất được cắt và toàn bộ chương trình thứ hai. Theo các phương án điển hình, bộ mã hóa hoặc bộ giải mã thực hiện bộ định thời (được điều chỉnh bởi các cờ trong chương trình thứ nhất) mà đếm lùi đến cuối chương trình thứ nhất không được cắt, và cùng bộ định thời đó (được điều chỉnh bởi các cờ trong chương trình thứ hai) đếm tiến từ đầu chương trình thứ hai. Đầu chương trình thứ hai là biên chương trình trong Trường hợp 3. Như được chỉ báo bởi đồ thị bộ định thời biên trong Trường hợp 3 trên Fig.11, việc đếm ngược của bộ định thời này (được điều chỉnh bởi siêu dữ liệu biên chương trình trong chương trình thứ nhất) được thiết lập lại (để đáp lại siêu dữ liệu biên chương trình trong chương trình thứ hai) trước khi nó đạt đến zero (để đáp lại siêu dữ liệu biên chương trình trong chương trình thứ nhất). Do đó, mặc dù việc cắt chương trình thứ nhất (bằng đoạn ghép nối) ngăn không cho bộ định thời xác định biên chương trình giữa chương trình thứ nhất được cắt và đầu

chương trình thứ hai để đáp lại (tức là, dưới sự căn chỉnh bởi) siêu dữ liệu biên chương trình trong một mình chương trình thứ nhất, siêu dữ liệu chương trình trong chương trình thứ hai thiết lập lại bộ định thời, sao cho bộ định thời được thiết lập lại chỉ báo chính xác (như vị trí tương ứng với lần đếm “zero” của bộ định thời được thiết lập lại) vị trí của biên chương trình giữa chương trình thứ nhất được cắt và đầu chương trình thứ hai.

Dòng bit thứ tư (được gắn nhãn “Trường hợp 4”) chỉ báo chương trình âm thanh thứ nhất được cắt (P1) mà bao gồm siêu dữ liệu biên chương trình (các cờ biên chương trình, F), và chương trình âm thanh thứ hai được cắt (P2) mà bao gồm siêu dữ liệu biên chương trình (các cờ biên chương trình, F) và đã được nối với một phần (phần không được cắt) của chương trình âm thanh thứ nhất. Các cờ biên chương trình trong phần đầu của toàn bộ chương trình thứ hai (trước quá trình cắt) (một số được thể hiện trên Fig.11) là giống hoặc tương tự với các cờ đã được mô tả với sự tham chiếu đến Fig.9, và các cờ biên chương trình trong phần cuối của toàn bộ chương trình thứ nhất (phần không được cắt) (một số được thể hiện trên Fig.11) là giống hoặc tương tự với các cờ đã được mô tả với sự tham chiếu đến Fig.8. Việc ghép nối đã loại bỏ “N” khung cuối cùng của chương trình thứ nhất (và do đó một số cờ biên chương trình mà đã được bao gồm trong đó trước khi tiến hành ghép nối) và “M” khung đầu tiên của chương trình thứ hai (và do đó một số cờ biên chương trình mà đã được bao gồm trong đó trước khi tiến hành ghép nối). Theo các phương án điển hình, bộ mã hóa hoặc bộ giải mã thực hiện bộ định thời (được điều chỉnh bởi các cờ trong chương trình thứ nhất được cắt) mà đếm lùi đến cuối chương trình thứ nhất không được cắt, và cùng bộ định thời đó (được điều chỉnh bởi các cờ trong chương trình thứ hai được cắt) đếm tiến từ đầu chương trình thứ hai không được cắt. Như được chỉ báo bởi đồ thị bộ định thời biên trong Trường hợp 4 trên Fig.11, việc đếm ngược của bộ định thời này (được điều chỉnh bởi siêu dữ liệu biên chương trình trong chương trình thứ nhất) được thiết lập lại (để đáp lại siêu dữ liệu biên chương trình trong chương trình thứ hai) trước khi nó đạt đến zero (để đáp lại siêu dữ liệu biên chương trình trong chương trình thứ nhất). Việc cắt chương trình thứ nhất (bằng đoạn ghép nối) ngăn không cho bộ định thời xác định biên chương trình giữa chương trình thứ nhất được cắt và đầu chương trình thứ hai được cắt) để đáp lại (tức là, dưới sự căn chỉnh của) siêu dữ liệu biên chương trình chỉ trong chương trình thứ nhất. Tuy nhiên, bộ định thời được thiết lập lại không chỉ báo đúng vị trí của biên chương trình giữa cuối chương trình thứ nhất

được cắt và đầu chương trình thứ hai được cắt. Do đó, việc cắt cả hai dòng bit được ghép có thể ngăn việc xác định chính xác biên ở giữa chúng.

Các phương án theo sáng chế có thể có thể được thực hiện trong phần cứng, phần sụn, hoặc phần mềm, hoặc kết hợp cả hai (ví dụ, như mảng logic có thể lập trình được). Trừ khi có quy định khác, các thuật toán hoặc quá trình được bao gồm như một phần của sáng chế vốn không liên quan đến bất kỳ máy tính hoặc thiết bị cụ thể nào. Cụ thể, nhiều máy công dụng chung có thể được sử dụng với các chương trình được viết theo nội dung bộc lộ ở đây, hoặc có thể thuận tiện hơn nếu tạo thiết bị chuyên dụng hơn (ví dụ, các mạch tích hợp) để thực hiện các bước của phương pháp được yêu cầu. Do đó, sáng chế có thể được triển khai trong một hoặc nhiều chương trình máy tính chạy trên một hoặc nhiều hệ thống máy tính có thể lập trình được (ví dụ, phương án thực hiện của các phần tử bất kỳ trên Fig.1, hoặc bộ mã hóa 100 trên Fig.2 (hoặc một phần tử của nó), hoặc bộ giải mã 200 trên Fig.3 (hoặc một phần tử của nó), hoặc bộ xử lý sau 300 trên Fig.3 (hoặc một phần tử của nó)) mỗi hệ thống này bao gồm ít nhất một bộ xử lý, ít nhất một hệ thống lưu trữ dữ liệu (bao gồm bộ nhớ khả biến và bất khả biến và/hoặc các phần tử lưu trữ), ít nhất một thiết bị hoặc cổng đầu vào, và ít nhất một thiết bị hoặc cổng đầu ra. Mã chương trình được áp dụng cho dữ liệu đầu vào để thực hiện các chức năng được mô tả ở đây và tạo ra thông tin đầu ra. Thông tin đầu ra được áp dụng cho một hoặc nhiều thiết bị đầu ra, theo cách đã biết.

Mỗi chương trình này có thể được cài đặt dưới ngôn ngữ máy tính mong muốn bất kỳ (bao gồm ngôn ngữ máy, hợp ngữ, hoặc ngôn ngữ thủ tục cấp cao, ngôn ngữ logic, hoặc ngôn ngữ lập trình định hướng đối tượng) để truyền thông với một hệ thống máy tính. Trong mọi trường hợp, ngôn ngữ này có thể là ngôn ngữ biên dịch hoặc diễn dịch.

Ví dụ, khi được triển khai bởi các trình tự chỉ lệnh phần mềm máy tính, các chức năng và bước khác nhau theo các phương án của sáng chế có thể được thực hiện bởi các trình tự chỉ lệnh phần mềm đa luồng chạy trong phần cứng xử lý tín hiệu kỹ thuật số phù hợp, trong trường hợp đó các thiết bị, các bước, và các chức năng khác nhau theo các phương án có thể tương ứng với các phần của các chỉ lệnh phần mềm.

Mỗi chương trình máy tính này tốt hơn là được lưu trữ trên hoặc được tải xuống một phương tiện hoặc thiết bị lưu trữ (ví dụ, bộ nhớ hoặc phương tiện nhớ thể rắn, hoặc

phương tiện nhớ từ hoặc quang) có thể đọc được bởi máy tính có thể lập trình được với mục đích chung hoặc mục đích đặc biệt, để tạo cấu hình và vận hành máy tính khi phương tiện hoặc thiết bị lưu trữ được đọc bởi hệ thống máy tính để thực hiện các quy trình được mô tả ở đây. Hệ thống theo sáng chế cũng có thể được triển khai như phương tiện lưu trữ có thể đọc được bởi máy tính, được tạo cấu hình với (tức là, lưu trữ) một chương trình máy tính, trong đó phương tiện lưu trữ được tạo cấu hình như vậy giúp hệ thống máy tính hoạt động theo cách được chỉ định và xác định trước để thực hiện các chức năng được mô tả ở đây.

Một số phương án của sáng chế đã được mô tả. Tuy nhiên, cần phải hiểu rằng nhiều cải biến khác nhau có thể được thực hiện mà không chệch ra khỏi bản chất và phạm vi của sáng chế. Nhiều cải biến và biến thể của sáng chế có thể được thực hiện với kỹ thuật nêu trên. Cần phải hiểu rằng trong phạm vi của bộ yêu cầu bảo hộ đính kèm, sáng chế có thể được thực hiện theo cách khác ngoài các cách được mô tả ở đây.

YÊU CẦU BẢO HỘ

1. Đơn vị xử lý âm thanh (100, 200) bao gồm:

bộ nhớ đệm (110, 201) để lưu trữ ít nhất một khung của dòng bit âm thanh được mã hóa, trong đó dòng bit âm thanh được mã hóa bao gồm dữ liệu âm thanh và đoạn siêu dữ liệu, trong đó đoạn siêu dữ liệu là dữ liệu tách riêng và khác với dữ liệu âm thanh và bao gồm phần đầu, một hoặc nhiều tải trọng siêu dữ liệu, và dữ liệu bảo vệ;

bộ giải mã âm thanh (101, 202) được nối với bộ nhớ đệm (110, 201) để giải mã dữ liệu âm thanh; và

bộ phân tích (111, 205) được nối hoặc tích hợp với bộ giải mã âm thanh (101, 202) để phân tích dòng bit âm thanh được mã hóa,

trong đó phần đầu bao gồm một lệnh đồng bộ hóa xác định điểm bắt đầu của đoạn siêu dữ liệu, một hoặc nhiều tải trọng siêu dữ liệu mô tả chương trình âm thanh được kết hợp với dữ liệu âm thanh, dữ liệu bảo vệ nằm ở sau một hoặc nhiều tải trọng siêu dữ liệu, và dữ liệu bảo vệ có khả năng được sử dụng để xác minh tính nhất quán của đoạn siêu dữ liệu và một hoặc nhiều tải trọng trong đoạn siêu dữ liệu.

2. Đơn vị xử lý âm thanh (100, 200) theo điểm 1 trong đó đoạn siêu dữ liệu được lưu trữ trong không gian dữ liệu dự trữ AC-3 hoặc E-AC-3 được chọn từ nhóm bao gồm trường bỏ qua, trường auxdata, trường addbsi, và kết hợp của chúng.

3. Đơn vị xử lý âm thanh (100, 200) theo điểm 1 hoặc 2 trong đó một hoặc nhiều tải trọng siêu dữ liệu bao gồm siêu dữ liệu chỉ báo ít nhất một biên giữa các chương trình âm thanh liên tiếp.

4. Đơn vị xử lý âm thanh (100, 200) theo điểm 1 hoặc 2 trong đó một hoặc nhiều tải trọng siêu dữ liệu bao gồm tải trọng âm lượng chương trình mà chứa dữ liệu chỉ báo âm lượng được đo của chương trình âm thanh.

5. Đơn vị xử lý âm thanh (100, 200) theo điểm 4 trong đó tải trọng âm lượng chương trình bao gồm: trường chỉ báo xem kênh âm thanh có chứa thoại tiếng nói hay không; hoặc trường chỉ báo phương pháp đo âm lượng mà đã được sử dụng để tạo ra dữ liệu âm lượng được chứa trong tải trọng âm lượng chương trình.

6. Đơn vị xử lý âm thanh (100, 200) theo điểm 4 trong đó tải trọng âm lượng chương trình bao gồm: trường chỉ báo xem âm lượng của chương trình âm thanh có được hiệu chỉnh bằng cách sử dụng kỹ thuật tạo cổng thoại hay không; hoặc trường chỉ báo xem âm lượng của chương trình âm thanh có được hiệu chỉnh bằng cách sử dụng quy trình hiệu chỉnh âm lượng nhìn về phía trước vô hạn hay dựa vào tệp hay không.
7. Đơn vị xử lý âm thanh (100, 200) theo điểm 4 trong đó tải trọng âm lượng chương trình bao gồm trường chỉ báo âm lượng được tích hợp của chương trình âm thanh mà không có bất kỳ sự điều chỉnh độ khuếch đại nào có thể được gây ra bởi việc nén dải động.
8. Đơn vị xử lý âm thanh (100, 200) theo điểm 4 trong đó tải trọng âm lượng chương trình bao gồm trường chỉ báo âm lượng được tích hợp của chương trình âm thanh mà không có bất kỳ sự điều chỉnh độ khuếch đại nào có thể được gây ra bởi việc chuẩn hóa thoại.
9. Đơn vị xử lý âm thanh (100, 200) theo điểm 4 trong đó đơn vị xử lý âm thanh (100, 200) được tạo cấu hình để thực hiện quá trình xử lý âm lượng thích ứng bằng cách sử dụng tải trọng âm lượng chương trình.
10. Đơn vị xử lý âm thanh (100, 200) theo điểm bất kỳ trong số các điểm từ 1 đến 9 trong đó dòng bit âm thanh được mã hóa là dòng bit AC-3 hoặc dòng bit E-AC-3.
11. Đơn vị xử lý âm thanh (100, 200) theo điểm bất kỳ trong số các điểm từ 4 đến 9 trong đó đơn vị xử lý âm thanh (100, 200) được tạo cấu hình để trích xuất tải trọng âm lượng chương trình ra khỏi dòng bit âm thanh được mã hóa và xác nhận hoặc phê duyệt tải trọng âm lượng chương trình.
12. Đơn vị xử lý âm thanh (100, 200) theo điểm bất kỳ trong số các điểm từ 1 đến 11 trong đó một hoặc nhiều tải trọng siêu dữ liệu, mỗi tải trọng bao gồm một ký hiệu nhận dạng tải trọng duy nhất, và ký hiệu nhận dạng tải trọng duy nhất này nằm ở đầu của mỗi tải trọng siêu dữ liệu.
13. Đơn vị xử lý âm thanh (100, 200) theo điểm bất kỳ trong số các điểm từ 1 đến 11 trong đó lệnh đồng bộ là lệnh đồng bộ 16 bit có trị số là 0x5838.
14. Phương pháp giải mã dòng bit âm thanh được mã hóa, phương pháp này bao gồm các bước:

nhận dòng bit âm thanh được mã hóa, dòng bit âm thanh được mã hóa này được phân đoạn thành một hoặc nhiều khung;

trích xuất dữ liệu âm thanh và đoạn siêu dữ liệu từ dòng bit âm thanh được mã hóa, đoạn siêu dữ liệu này là dữ liệu tách riêng và khác với dữ liệu âm thanh và bao gồm phần đầu theo sau bởi một hoặc nhiều tải trọng siêu dữ liệu theo sau bởi dữ liệu bảo vệ; và

xác minh tính nhất quán của đoạn và một hoặc nhiều tải trọng siêu dữ liệu thông qua việc sử dụng dữ liệu bảo vệ,

trong đó một hoặc nhiều tải trọng siêu dữ liệu bao gồm tải trọng âm lượng chương trình mà chứa dữ liệu chỉ báo âm lượng được đo của chương trình âm thanh kết hợp với dữ liệu âm thanh.

15. Phương pháp theo điểm 14 trong đó đoạn siêu dữ liệu được lưu trữ trong một hoặc nhiều trường bỏ qua hoặc các đoạn bit thừa của khung.

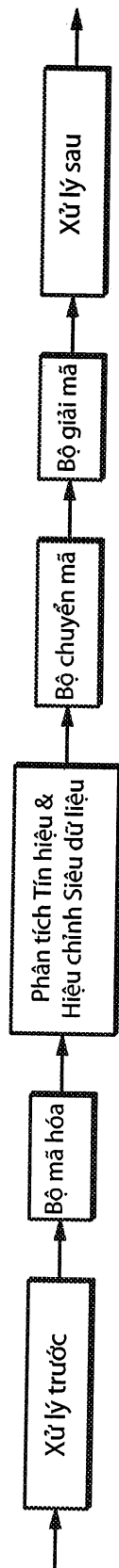


FIG. 1

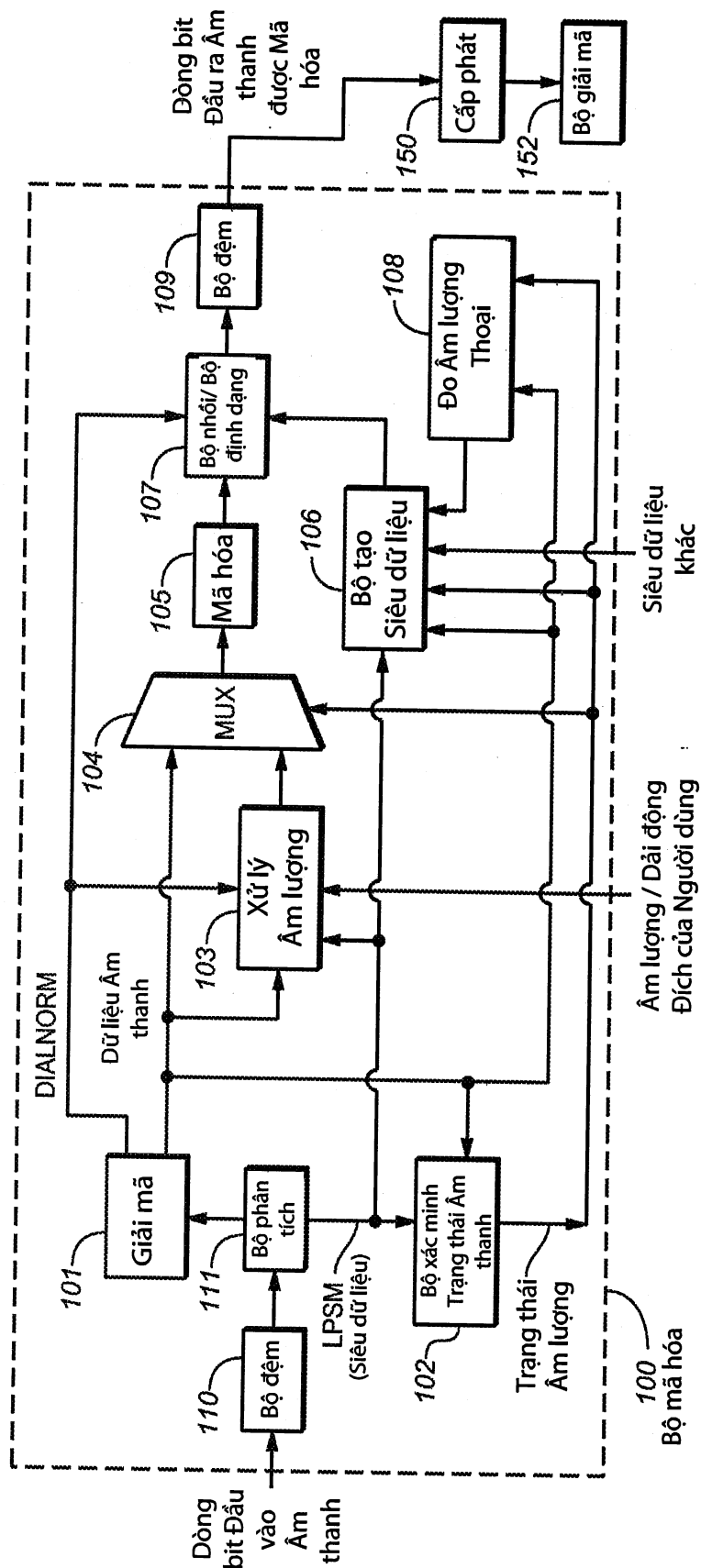


FIG. 2

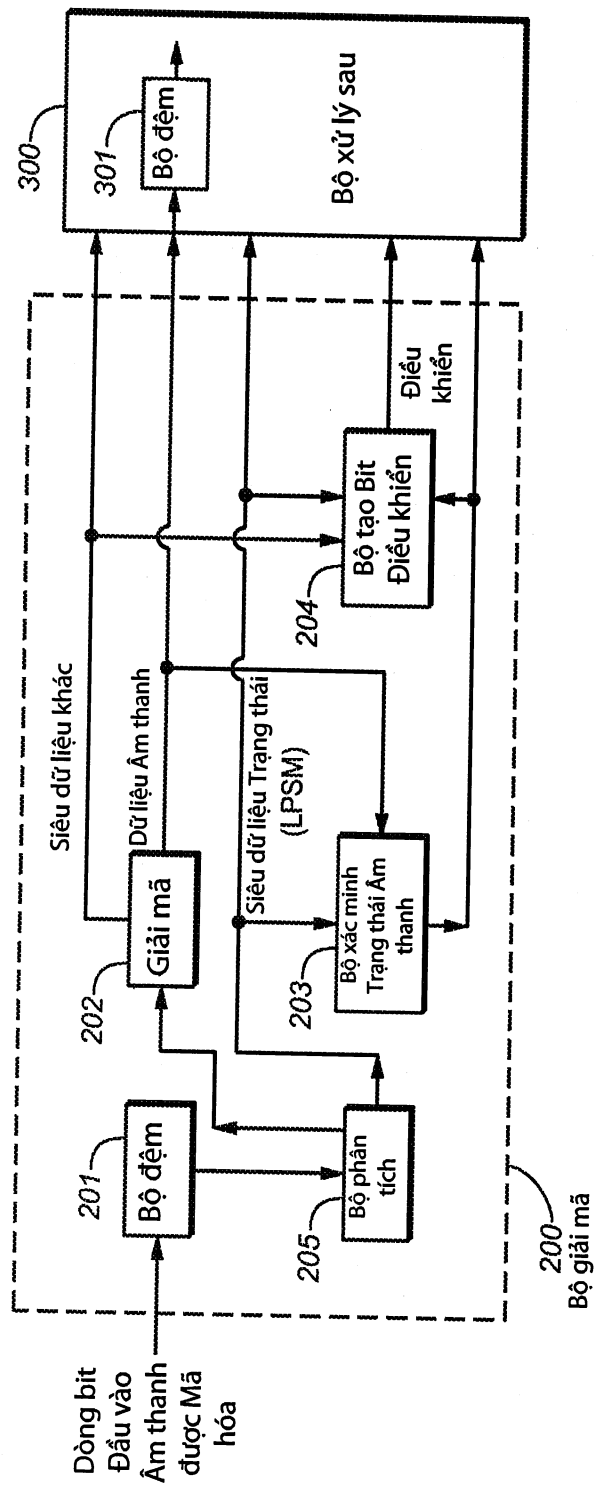


FIG. 3

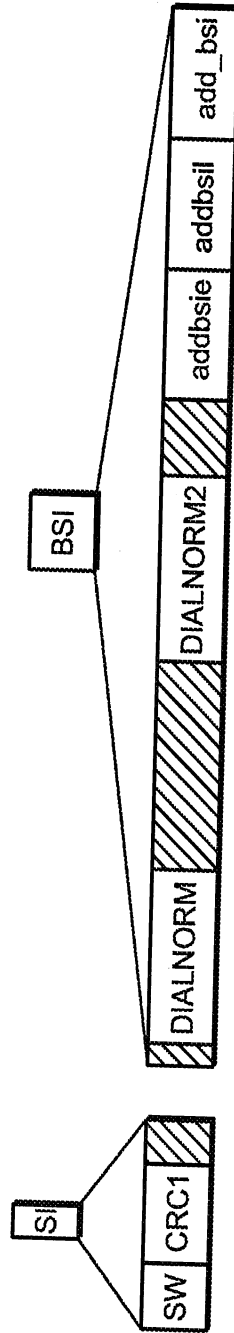
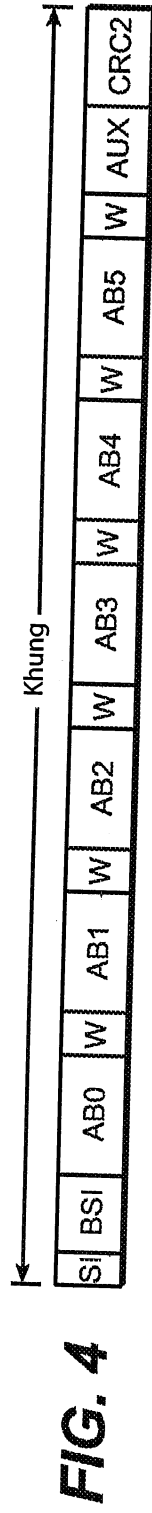


FIG. 5

FIG. 6



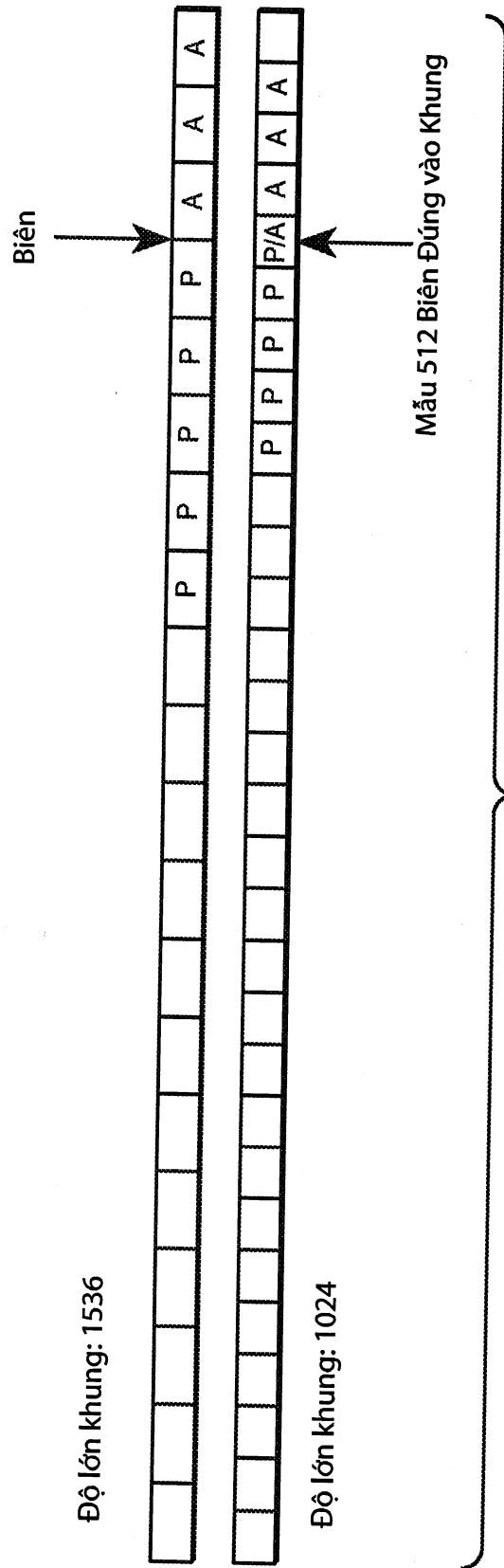
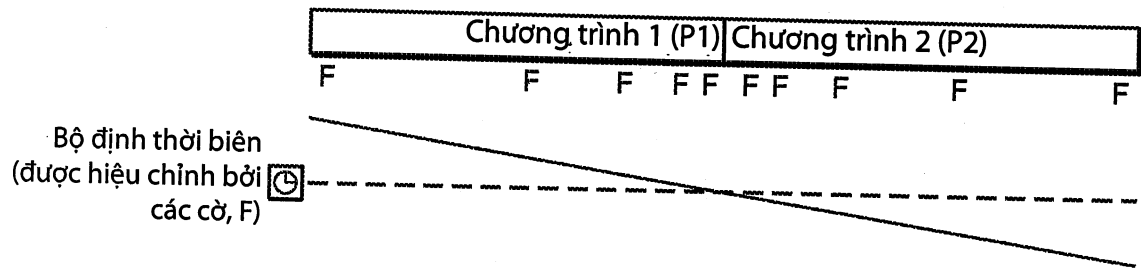
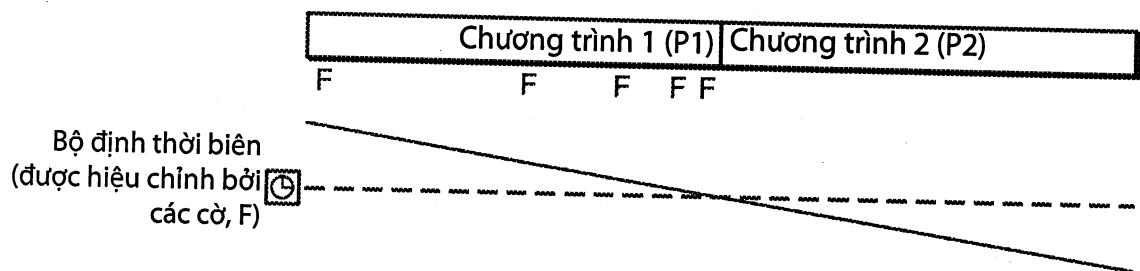


FIG. 10

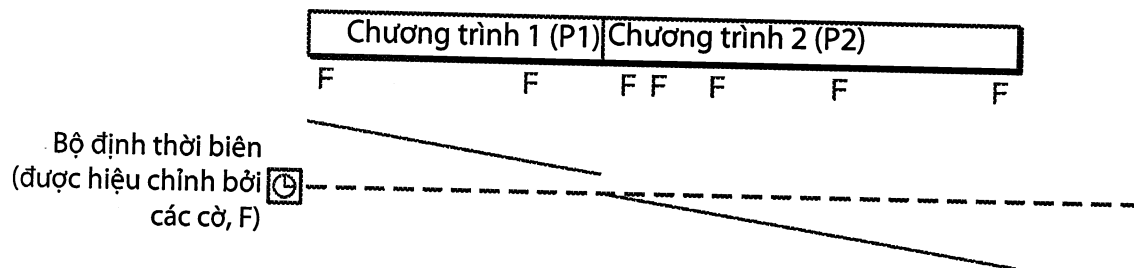
Trường hợp 1: Cả P1 và P2 đều có cờ



Trường hợp 2: Chỉ P1 có hiệu



Trường hợp 3: Cuối P1 được ghép



Trường hợp 3: Cuối P1, đầu P2 được ghép

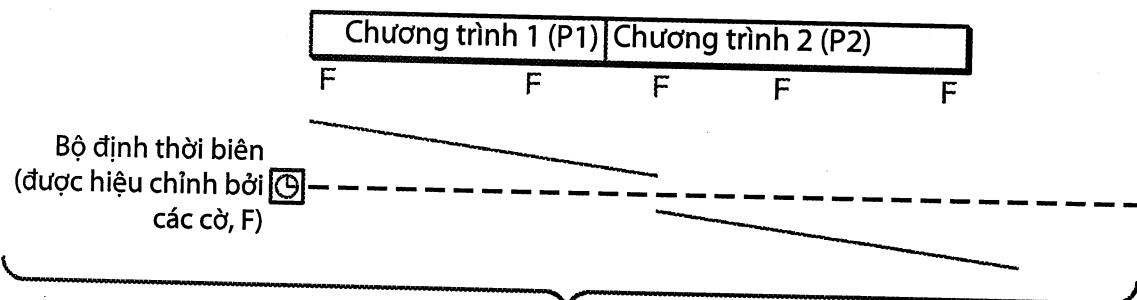


FIG. 11