



(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)



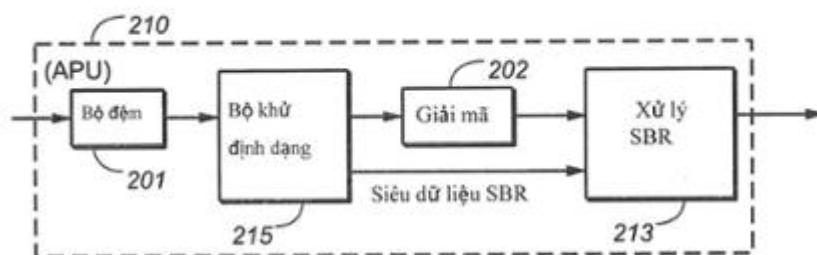
1-0041321

(51)<sup>2020.01</sup> G10L 19/02; H04N 19/00; G10L 19/16; (13) B  
G10L 19/00

- (21) 1-2020-04316 (22) 28/01/2019  
(86) PCT/US2019/015442 28/01/2019 (87) WO 2019/148112 A1 01/08/2019  
(30) 62/622,205 26/01/2018 US  
(45) 25/10/2024 439 (43) 26/10/2020 391A1  
(73) DOLBY INTERNATIONAL AB (NL)  
Apollo Building, 3E, Herikerbergweg 1-35, 1101 CN Amsterdam Zuidoost,  
Netherlands  
(72) KJOERLING, Kristofer (SE); VILLEMOES, Lars (SE); PURNHAGEN, Heiko  
(DE); EKSTRAND, Per (SE).  
(74) Công ty TNHH Quốc tế D & N (D&N INTERNATIONAL CO.,LTD.)

(54) PHƯƠNG PHÁP VÀ THIẾT BỊ XỬ LÝ ÂM THANH ĐỂ THỰC HIỆN TÁI TẠO  
TẦN SỐ CAO TÍN HIỆU ÂM THANH, PHƯƠNG TIỆN BẮT BIẾN ĐỌC ĐƯỢC  
BẰNG MÁY TÍNH

(57) Sáng chế đề cập đến phương pháp giải mã dòng bit âm thanh được mã hóa. Phương pháp bao gồm bước nhận dòng bit âm thanh được mã hóa và giải mã dữ liệu âm thanh để tạo ra tín hiệu âm thanh băng thấp giải mã. Phương pháp còn bao gồm bước trích siêu dữ liệu tái tạo tần số cao và lọc tín hiệu âm thanh băng thấp đã giải mã bằng giàn bộ lọc phân tích để tạo ra tín hiệu âm thanh băng thấp đã lọc. Phương pháp còn bao gồm bước trích còi chỉ báo xem quy trình dịch phổ hoặc quy trình chuyển vị sóng hài có được thực hiện trên dữ liệu âm thanh và tái tạo phần băng cao của tín hiệu âm thanh bằng cách sử dụng tín hiệu âm thanh băng thấp đã lọc và siêu dữ liệu tái tạo tần số cao theo còi không. Sáng chế còn đề cập đến phương pháp và thiết bị xử lý âm thanh để thực hiện tái cấu trúc tần số cao tín hiệu âm thanh, và vật ghi bắt biến đọc được bằng máy tính.



### **Lĩnh vực kỹ thuật được đề cập**

Các phương án của sáng chế đề cập đến việc xử lý tín hiệu âm thanh, và cụ thể hơn, đến việc mã hóa, giải mã, hoặc chuyển mã các dòng bit âm thanh với dữ liệu điều khiển chỉ ra rằng hoặc dạng tái tạo tần số cao (“high frequency reconstruction - HFR”) cơ sở hay dạng HFR nâng cao sẽ được thực hiện trên dữ liệu âm thanh.

### **Tình trạng kỹ thuật của sáng chế**

Dòng bit âm thanh thông thường bao gồm cả dữ liệu âm thanh (ví dụ, dữ liệu âm thanh được mã hóa) chỉ báo một hoặc nhiều kênh nội dung âm thanh, và siêu dữ liệu chỉ báo ít nhất một đặc tính của dữ liệu âm thanh hoặc nội dung âm thanh. Một định dạng được biết rộng rãi để tạo ra dòng bit âm thanh được mã hóa là định dạng mã hóa âm thanh tiên tiến (Advanced Audio Coding - AAC) MPEG-4, được mô tả trong chuẩn MPEG ISO/IEC 14496-3:2009. Trong chuẩn MPEG-4, AAC có nghĩa là “mã hóa âm thanh tiên tiến” và HE-AAC có nghĩa là “mã hóa âm thanh tiên tiến hiệu suất cao.”

Chuẩn MPEG-4 AAC định nghĩa một vài biên dạng âm thanh, biên dạng này xác định các đối tượng và công cụ mã hóa nào có mặt trong bộ mã hóa hoặc bộ giải mã phù hợp. Ba trong số các biên dạng âm thanh này là (1) biên dạng AAC, (2) biên dạng HE-AAC, và (3) biên dạng HE-AAC v2. Biên dạng AAC bao gồm loại đối tượng ACC có độ phức tạp thấp (hoặc “AAC-LC - AAC low complexity”). Đối tượng AAC-LC là dạng tương đương của biên dạng MPEG-2 AAC có độ phức tạp thấp, với một số điều chỉnh, và không bao gồm loại đối tượng tái tạo phổ tần số (“spectral band replication - SBR”) và cũng không bao gồm loại đối tượng mã hóa âm thanh nổi có tham số (“parametric stereo - PS”). Biên dạng HE-AAC là tập lớn của biên dạng AAC và ngoài ra bao gồm loại đối tượng SBR. Biên dạng HE-AAC v2 là tập lớn của biên dạng HE-AAC và ngoài ra còn bao gồm loại đối tượng PS.

Loại đối tượng SBR chứa công cụ tái tạo phổ tần số, công cụ này là công cụ mã hóa tái tạo tần số cao (“high frequency reconstruction - HFR”) quan trọng cải thiện đáng kể hiệu suất nén của các bộ mã hóa-giải mã âm thanh cảm nhận. SBR tái tạo các thành phần tần số cao của tín hiệu âm thanh ở phía bộ thu (ví dụ, ở bộ giải mã). Vì vậy, bộ mã hóa chỉ

cần mã hóa và truyền các thành phần tần số thấp, cho phép chất lượng âm thanh cao hơn nhiều ở tốc độ dữ liệu thấp. SBR dựa trên sự sao chép các chuỗi sóng hài, đã được cắt bớt trước đó để giảm tốc độ dữ liệu, từ tín hiệu giới hạn băng thông khả dụng và dữ liệu điều khiển thu được từ bộ mã hóa. Tỷ lệ giữa thành phần âm và thành phần giống nhiễu được duy trì bằng cách lọc ngược thích ứng cũng như bổ sung tùy ý nhiễu và dạng hình sin. Theo chuẩn MPEG-4 AAC, công cụ SBR thực hiện vá phổ (còn gọi là dịch tuyến tính hoặc dịch phổ), trong đó một số băng con của Bộ lọc cầu phương (Quadrature Mirror Filter - QMF) liên tiếp được sao chép (hoặc “vá” hoặc) từ phần băng thấp đã truyền của tín hiệu âm thanh đến phần băng cao của tín hiệu âm thanh, tín hiệu này được tạo ra ở bộ giải mã.

Vá phổ hoặc dịch tuyến tính có thể không lý tưởng đối với một số loại âm thanh nhất định, như nội dung âm nhạc có các tần số giao cắt tương đối thấp. Do đó, cần có các kỹ thuật để cải thiện tái tạo phổ tần số.

### **Bản chất kỹ thuật của sáng chế**

Nhóm phương án đầu tiên đề cập đến phương pháp giải mã dòng bit được âm thanh mã hóa được mô tả. Phương pháp bao gồm bước nhận dòng bit âm thanh được mã hóa và giải mã dữ liệu âm thanh để tạo ra tín hiệu âm thanh băng thấp đã giải mã. Phương pháp còn bao gồm bước trích siêu dữ liệu tái tạo tần số cao và lọc tín hiệu âm thanh băng thấp đã giải mã bằng giàn bộ lọc phân tích để tạo ra tín hiệu âm thanh băng thấp đã lọc. Phương pháp còn bao gồm bước trích cờ chỉ báo xem việc tịnh tiến phổ hoặc chuyển vị sóng hài có được thực hiện trên dữ liệu âm thanh hay không và tái tạo phần băng cao của tín hiệu âm thanh bằng cách sử dụng tín hiệu âm thanh băng thấp đã lọc và siêu dữ liệu tái tạo tần số cao theo cờ. Cuối cùng, phương pháp bao gồm bước kết hợp tín hiệu âm thanh băng thấp đã lọc và phần băng cao tái tạo để tạo ra tín hiệu âm thanh băng rộng.

Nhóm phương án thứ hai đề cập đến bộ giải mã âm thanh để giải mã dòng bit âm thanh được mã hóa. Bộ giải mã bao gồm giao diện đầu vào để nhận dòng bit âm thanh được mã hóa trong đó dòng bit âm thanh được mã hóa bao gồm dữ liệu âm thanh biểu diễn phần băng thấp của tín hiệu âm thanh và bộ giải mã lõi để giải mã dữ liệu âm thanh để tạo ra tín hiệu âm thanh băng thấp đã giải mã. Bộ giải mã còn bao gồm bộ giải ghép kênh để trích, từ dòng bit âm thanh được mã hóa, siêu dữ liệu tái tạo tần số cao trong đó siêu dữ liệu tái tạo tần số cao bao gồm các tham số hoạt động cho quy trình tái tạo tần số cao để dịch tuyến tính một số băng con liên tiếp từ phần băng thấp của tín hiệu âm thanh đến phần băng cao

của tín hiệu âm thanh và giàn bộ lọc phân tích để lọc tín hiệu âm thanh băng thấp đã giải mã để tạo ra tín hiệu âm thanh băng thấp đã lọc. Bộ giải mã còn bao gồm bộ giải ghép kênh để trích, từ dòng bit âm thanh được mã hóa, cờ chỉ báo liệu quy trình dịch tuyến tính hoặc chuyển vị sóng hài có được thực hiện trên dữ liệu âm thanh không và bộ tái tạo tần số cao để tái tạo phần băng cao của tín hiệu âm thanh nhờ sử dụng tín hiệu âm thanh băng thấp đã lọc và siêu dữ liệu tái tạo tần số cao theo cờ. Cuối cùng, bộ giải mã bao gồm giàn bộ lọc tổng hợp để kết hợp tín hiệu âm thanh băng thấp đã lọc và phần băng cao tái tạo để tạo ra tín hiệu âm thanh băng rộng.

Các nhóm phương án khác đề cập đến việc mã hóa và chuyển mã các dòng bit âm thanh chứa siêu dữ liệu xác định xem quy trình xử lý tái tạo phổ tần số nâng cao (enhanced spectral band replication - eSBR) có được thực hiện không.

### **Mô tả vắn tắt các hình vẽ**

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

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

Fig.3 là sơ đồ khối của hệ thống bao gồm bộ giải mã theo một phương án của đơn vị xử lý âm thanh theo sáng chế, và tùy ý cũng bao gồm bộ xử lý hậu kỳ được gắn vào đó.

Fig.4 là sơ đồ khối của bộ giải mã theo một phương án của đơn vị xử lý âm thanh theo sáng chế.

Fig.5 là sơ đồ khối của bộ giải mã theo một phương án khác của đơn vị xử lý âm thanh theo sáng chế.

Fig.6 là sơ đồ khối của một phương án khác của đơn vị xử lý âm thanh theo sáng chế.

Fig.7 là sơ đồ khối của dòng bit MPEG-4 AAC, bao gồm các đoạn đã được phân chia từ dòng bit này.

### **Mô tả chi tiết sáng chế**

### Kí hiệu và thuật ngữ

Trong suốt bản mô tả này, bao gồm cả phần yêu cầu bảo hộ, thuật ngữ thực hiện hoạt động “trên” tín hiệu hoặc dữ liệu (ví dụ, lọc, định tỷ lệ, biến đổi, hoặc áp dụng độ khuếch đại cho, tín hiệu hoặc dữ liệu) được sử dụng theo nghĩa rộng để chỉ việc thực hiện trực tiếp hoạt động 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 đã trải qua quy trình lọc sơ bộ hoặc tiền xử lý trước khi thực hiện hoạt động trên đó).

Trong suốt bản mô tả, bao gồm phần yêu cầu bảo hộ, cụm từ “đơn vị xử lý âm thanh” hoặc “bộ xử lý âm thanh” được dùng theo nghĩa rộng, để chỉ hệ thống, máy hoặc thiết bị, được tạo cấu hình để xử lý dữ liệu âm thanh. Ví dụ về các đơn vị xử lý âm thanh bao gồm, nhưng không bị giới hạn ở bộ mã hóa, bộ chuyển mã, bộ giải mã, bộ mã hóa-giải mã, hệ thống tiền xử lý, hệ thống xử lý hậu kỳ, và hệ thống xử lý dòng bit (đôi khi được gọi là công cụ xử lý dòng bit). Hầu như tất cả các thiết bị điện tử người dùng, như điện thoại di động, tivi, máy tính xách tay và máy tính bảng đều có đơn vị xử lý âm thanh hoặc bộ xử lý âm thanh.

Trong suốt bản mô tả này, bao gồm cả phần yêu cầu bảo hộ, thuật ngữ “ghép nối” hoặc “được ghép nối” được dùng theo nghĩa rộng nghĩa là kết nối trực tiếp hoặc gián tiếp. Do đó, nếu thiết bị thứ nhất ghép nối với thiết bị thứ hai, thì việc kết nối này có thể là thông qua kết nối trực tiếp, hoặc kết nối gián tiếp qua các thiết bị và kết nối khác. Hơn nữa, các thành phần được tích hợp vào hoặc với các thành phần khác cũng được ghép nối với nhau.

Chuẩn MPEG-4 AAC dự tính rằng dòng bit MPEG-4 AAC được mã hóa bao gồm siêu dữ liệu chỉ báo mỗi loại của quy trình xử lý tái tạo tần số cao (“high frequency reconstruction - HFR”) cần được áp dụng (nếu có sẽ được áp dụng) bởi bộ giải mã để giải mã nội dung âm thanh của dòng bit, và/hoặc kiểm soát quy trình xử lý HFR như vậy, và/hoặc chỉ báo ít nhất một đặc điểm hoặc tham số của ít nhất một công cụ HFR cần được dùng để giải mã nội dung âm thanh của dòng bit. Ở đây, các tác giả sáng chế sử dụng cụm từ “siêu dữ liệu SBR” để chỉ siêu dữ liệu thuộc loại được mô tả hoặc đề cập trong chuẩn MPEG-4 AAC để sử dụng với công cụ SBR. Người có hiểu biết trung bình trong lĩnh vực kỹ thuật này sẽ hiểu rằng SBR là một dạng của HFR.

SBR tốt hơn là được dùng dưới dạng hệ thống tốc độ kép, với bộ mã hóa-giải mã cơ bản hoạt động với một nửa tốc độ lấy mẫu ban đầu, còn SBR hoạt động với tốc độ lấy mẫu

ban đầu. Bộ mã hóa SBR hoạt động song song với bộ mã hóa-giải mã lõi cơ bản, mặc dù với tốc độ lấy mẫu cao hơn. Mặc dù SBR chủ yếu là xử lý hậu kỳ ở bộ giải mã, nhưng các tham số quan trọng được trích ở bộ mã hóa để bảo đảm sự tái tạo tần số cao chính xác nhất ở bộ giải mã. Bộ mã hóa ước lượng đường bao phổ của khoảng SBR trong một khoảng/độ phân giải thời gian và dải tần số thích hợp cho các đặc điểm đoạn tín hiệu đầu vào hiện thời. Đường bao phổ được ước lượng bằng cách phân tích QMF phức hợp và tính toán năng lượng sau đó. Độ phân giải thời gian và tần số của các đường bao phổ có thể được chọn với mức tự do cao, để bảo đảm độ phân giải thời gian tần số thích hợp nhất cho đoạn đầu vào cho trước. Việc ước lượng đường bao cần xem xét đến hiện tượng chuyển vị lúc ban đầu, chủ yếu ở vùng tần số cao (ví dụ, mức cao), sẽ xuất hiện với mức độ nhỏ trong băng cao được tạo ra bởi SBR trước khi điều chỉnh đường bao, vì băng cao ở bộ giải mã được dựa trên băng thấp trong đó hiện tượng chuyển tiếp ít rõ ràng hơn so với băng cao. Khía cạnh này đặt ra các yêu cầu khác nhau đối với độ phân giải thời gian tần số của dữ liệu đường bao phổ, so với việc ước lượng đường bao phổ thông thường như được sử dụng trong các thuật toán mã hóa âm thanh khác.

Ngoài đường bao phổ, một vài tham số bổ sung được trích biểu diễn các đặc tính phổ của tín hiệu đầu vào trong các vùng thời gian và tần số khác nhau. Do bản chất bộ mã hóa có quyền truy cập vào tín hiệu gốc cũng như thông tin về cách đơn vị SBR ở bộ giải mã sẽ tạo băng cao, dựa trên tập tham số điều khiển cụ thể, nên hệ thống có thể xử lý các tình huống mà ở đó băng thấp tạo thành chuỗi sóng hài mạnh và băng cao, cần được tái tạo, chủ yếu tạo thành các thành phần tín hiệu ngẫu nhiên, cũng như các tình huống mà ở đó các thành phần tone mạnh có mặt trong băng cao gốc mà không có các bản sao trong băng thấp, mà vùng băng cao được dựa vào đó. Hơn nữa, bộ mã hóa SBR làm việc có liên quan chặt chẽ với bộ mã hóa-giải mã lõi cơ sở để đánh giá dải tần số nào nên được bao trùm bởi SBR tại một thời điểm nhất định. Dữ liệu SBR được mã hóa một cách hiệu quả trước khi truyền bằng cách khai thác quy trình mã hóa entropy cũng như các mức phụ thuộc kênh của dữ liệu điều khiển, trong trường hợp có tín hiệu âm thanh nổi.

Các thuật toán trích tham số điều khiển thường cần được điều chỉnh một cách thận trọng cho bộ mã hóa giải mã cơ sở với tốc độ bit cho trước và tốc độ lấy mẫu cho trước. Điều này là do thực tế là tốc độ bit thấp hơn thường mang hàm ý khoảng SBR lớn hơn so

với tốc độ bit cao, và các tốc độ lấy mẫu khác nhau tương ứng với các độ phân giải thời gian khác nhau của các khung SBR.

Bộ giải mã SBR thường bao gồm một vài phần khác nhau. Bộ giải mã này bao gồm modul giải mã dòng bit, modul tái tạo tần số cao (HFR), modul thành phần tần số cao bổ sung, và modul điều chỉnh đường bao. Hệ thống dựa trên giàn lọc QMF có giá trị phức (đối với SBR chất lượng cao) hoặc giàn bộ lọc QMF có giá trị thực (đối với SBR công suất thấp). Các phương án của sáng chế có thể áp dụng được cho cả SBR chất lượng cao và SBR công suất thấp. Trong modul trích dòng bit, dữ liệu điều khiển được đọc từ dòng bit và được giải mã. Lưới tần số thời gian được thu nhận cho khung hiện thời, trước khi đọc dữ liệu đường bao từ dòng bit. Bộ giải mã lõi cơ sở giải mã tín hiệu âm thanh của khung hiện thời (mặc dù ở tốc độ lấy mẫu thấp hơn) để tạo ra các mẫu âm thanh miền thời gian. Khung dữ liệu âm thanh thu được được dùng để tái tạo tần số cao bởi modul HFR. Tín hiệu băng thấp được giải mã sau đó được phân tích bằng cách sử dụng giàn lọc QMF. Quy trình tái tạo tần số cao và điều chỉnh đường bao được thực hiện sau đó trên các mẫu băng con của giàn lọc QMF. Các tần số cao được tái tạo từ băng thấp theo cách linh hoạt, dựa trên tham số điều khiển cho trước. Hơn nữa, băng cao tái tạo này được lọc thích ứng trên cơ sở kênh băng con theo dữ liệu điều khiển để bảo đảm các đặc tính phổ thích hợp của vùng thời gian/tần số cho trước.

Mức đỉnh của dòng bit MPEG-4 AAC là chuỗi khối dữ liệu (các phân tử “raw\_data\_block”), mỗi trong số chúng là một đoạn dữ liệu (trong bản mô tả này được gọi là “khối”) chứa dữ liệu âm thanh (thường trong khoảng thời gian gồm 1024 hoặc 960 mẫu) và thông tin liên quan và/hoặc dữ liệu khác. Ở đây, các tác giả sáng chế sử dụng thuật ngữ “khối” để chỉ một đoạn của dòng bit MPEG-4 AAC gồm dữ liệu âm thanh (và siêu dữ liệu tương ứng và tùy ý còn có dữ liệu liên quan khác) xác định hoặc chỉ báo về một (nhưng không nhiều hơn một) phân tử “raw\_data\_block”.

Mỗi khối của dòng bit MPEG-4 AAC có thể bao gồm một số phân tử cú pháp (mỗi trong số đó còn được cụ thể hóa trong dòng bit dưới dạng đoạn dữ liệu). Bảy loại phân tử cú pháp như vậy được định nghĩa theo chuẩn MPEG-4 AAC. Mỗi phân tử cú pháp được nhận dạng bởi các giá trị khác nhau của phân tử dữ liệu “id\_syn\_ele.” Ví dụ về phân tử cú pháp bao gồm “single\_channel\_element()”, “channel\_pair\_element()”, và “fill\_element()”. Phân tử kênh đơn là phân tử chứa bao gồm dữ liệu âm thanh của kênh âm thanh đơn (tín hiệu

âm thanh đơn âm). Phần tử cặp kênh bao gồm dữ liệu âm thanh của hai kênh âm thanh (tức là, tín hiệu âm thanh nổi).

Phần tử làm đầy là phần chứa thông tin bao gồm mã định danh (ví dụ, giá trị của phần tử nêu trên “id\_syn\_ele”) theo sau là dữ liệu, được gọi là “dữ liệu làm đầy.” Các phần tử làm đầy đã từng được sử dụng trước đây để điều chỉnh tốc độ bit tức thời của các dòng bit mà sẽ được truyền qua kênh tốc độ không đổi. Bằng cách bổ sung lượng dữ liệu làm đầy thích hợp vào mỗi khối, có thể đạt được tốc độ dữ liệu không đổi.

Theo các phương án của sáng chế, dữ liệu làm đầy có thể bao gồm một hoặc nhiều tải tin mở rộng để mở rộng loại dữ liệu (ví dụ, siêu dữ liệu) có khả năng được truyền trong dòng bit. Bộ giải mã nhận các dòng bit có dữ liệu làm đầy chứa loại dữ liệu mới có thể tùy ý được sử dụng bởi thiết bị nhận dòng bit (ví dụ, bộ giải mã) để mở rộng chức năng của thiết bị. Vì vậy, người có hiểu biết trung bình trong lĩnh vực kỹ thuật này sẽ hiểu rằng, các phần tử làm đầy là loại cấu trúc dữ liệu đặc biệt và khác các cấu trúc dữ liệu thường được dùng để truyền dữ liệu âm thanh (ví dụ, tải tin âm thanh chứa dữ liệu kênh).

Theo một số phương án của sáng chế, mã định danh được dùng để nhận dạng phần tử làm đầy có thể bao gồm số nguyên không dấu 3 bit với bit có ý nghĩa nhất được truyền đầu tiên (“unsigned integer transmitted most significant bit first - uimbsf”) có giá trị 0x6. Trong một khối, một vài ví dụ của cùng loại phần tử cú pháp (ví dụ, một vài phần tử làm đầy) có thể xuất hiện.

Một chuẩn khác để mã hóa các dòng bit âm thanh là chuẩn MPEG mã hóa giọng nói và âm thanh thống nhất (Unified Speech and Audio Coding - USAC) (ISO/IEC 23003-3:2012). Chuẩn MPEG USAC mô tả kỹ thuật mã hóa và giải mã nội dung âm thanh bằng cách sử dụng quy trình xử lý tái tạo phổ tần số (bao gồm xử lý SBR như ược mô tả trong chuẩn MPEG-4 AAC, và còn bao gồm các dạng nâng cao khác của quy trình xử lý tái tạo phổ tần số). Quy trình xử lý này áp dụng các công cụ tái tạo phổ tần số (đôi khi được gọi là “công cụ SBR nâng cao” hoặc “công cụ eSBR”) của phiên bản được mở rộng và nâng cao của tập hợp các công cụ SBR được mô tả trong chuẩn MPEG-4 AAC. Vì vậy, eSBR (như được định nghĩa trong chuẩn USAC) là dạng nâng cao của SBR (như được định nghĩa theo chuẩn MPEG-4 AAC).

Ở đây, các tác giả sáng chế sử dụng cụm từ “xử lý SBR nâng cao” (hoặc “xử lý eSBR”) để chỉ quy trình xử lý tái tạo phổ tần số bằng cách sử dụng ít nhất một công cụ

eSBR (ví dụ, ít nhất một công cụ eSBR được mô tả hoặc đề cập trong chuẩn MPEG USAC) mà không được mô tả hoặc đề cập trong chuẩn MPEG-4 AAC. Ví dụ về các công cụ eSBR như vậy là chuyển vị sóng hài và tiền xử lý bổ sung và QMF hoặc “làm phẳng trước.”

Bộ chuyển vị sóng hài bậc  $T$  nguyên ánh xạ sóng hình sin với tần số  $\omega$  thành sóng hình sin có tần số  $T\omega$ , trong khi bảo toàn khoảng thời gian tín hiệu. Ba bậc,  $T = 2, 3, 4$ , thường được dùng theo chuỗi để tạo ra mỗi phần của khoảng tần số đầu ra mong muốn bằng cách sử dụng bậc chuyển vị nhỏ nhất có thể có. Nếu cần đầu ra cao hơn khoảng chuyển vị bậc bốn, thì đầu ra có thể được tạo ra bởi các phép dịch tần số. Khi có thể, các miền thời gian băng tần cơ sở được lấy mẫu gần tới hạn được tạo ra cho quy trình xử lý để tối thiểu hóa độ phức tạp tính toán.

Bộ chuyển vị sóng hài có thể là dựa trên QMF hoặc DFT. Khi sử dụng bộ chuyển vị sóng hài dựa trên QMF, việc mở rộng băng thông của tín hiệu miền thời gian của bộ mã hóa lõi được thực hiện toàn bộ trong miền QMF, bằng cách sử dụng cấu trúc bộ mã hóa tiếng nói pha sửa đổi, thực hiện quy trình lấy ra một phần mười tiếp theo là kéo thời gian cho mỗi băng con QMF. Quy trình chuyển vị sử dụng một vài hệ số chuyển vị (ví dụ,  $T = 2, 3, 4$ ) được thực hiện trong tầng biến đổi phân tích/tổng hợp QMF chung. Do bộ chuyển vị sóng hài dựa trên QMF không có đặc điểm tăng mẫu miền tần số thích ứng tín hiệu, nên cờ tương ứng trong dòng bit (`sbrOversamplingFlag[ch]`) có thể được bỏ qua.

Khi sử dụng bộ chuyển vị sóng hài dựa trên DFT, các bộ chuyển vị hệ số 3 và 4 (các bộ chuyển vị bậc thứ 3 và thứ 4) tốt hơn là được tích hợp vào bộ chuyển vị hệ số 2 (bộ chuyển vị bậc thứ hai) bằng phương pháp nội suy để làm giảm độ phức tạp. Đối với mỗi khung (tương ứng với các mẫu bộ mã hóa lõi `coreCoderFrameLength`), kích thước biến đổi "toàn kích thước" danh định của bộ chuyển vị được xác định trước tiên bởi cờ tăng mẫu trong miền tần số thích ứng tín hiệu (`sbrOversamplingFlag[ch]`) trong dòng bit.

Khi `sbrPatchingMode==1`, chỉ báo rằng chuyển vị tuyến tính sẽ được sử dụng để tạo ra băng cao, bước bổ sung có thể được đưa vào để tránh sự gián đoạn về hình dạng đường bao phổ của tín hiệu tần số cao đưa vào bộ điều chỉnh đường bao tiếp theo. Việc này cải thiện hoạt động của tầng điều chỉnh đường bao tiếp theo, kết quả là tín hiệu băng cao được cảm nhận là ổn định hơn. Hoạt động tiền xử lý bổ sung là có lợi cho các loại tín hiệu trong đó đường bao phổ thô của tín hiệu băng thấp được dùng để tái tạo tần số cao biểu thị các mức thay đổi lớn. Tuy nhiên, giá trị của phần tử dòng bit có thể được xác định

trong bộ mã hóa bằng cách áp dụng kiểu phân loại phụ thuộc vào tín hiệu bất kỳ. Quy trình tiền xử lý bổ sung tốt hơn là được kích hoạt qua phần tử dòng bit một bit, *bs\_sbr\_preprocessing*. Khi *bs\_sbr\_preprocessing* được đặt bằng một, quy trình xử lý bổ sung được kích hoạt. Khi *bs\_sbr\_preprocessing* được đặt bằng 0, quy trình tiền xử lý bổ sung bị vô hiệu hóa. Tốt hơn là quy trình xử lý bổ sung sử dụng đường cong *preGain* mà được bộ tạo tần số cao sử dụng để định tỷ lệ bằng thấp,  $X_{Low}$  cho mỗi phần v. Ví dụ, đường cong *preGain* có thể được tính toán theo:

$$preGain(k) = 10^{(meanNrg - lowEnv(k))/20}, 0 \leq k < k_0$$

trong đó  $k_0$  là băng con QMF thứ nhất trong bảng băng tần chủ và *lowEnvSlope* được tính toán bằng cách sử dụng hàm tính toán các hệ số của đa thức phù hợp nhất (theo nghĩa bình phương nhỏ nhất), như *polyfit()*. Ví dụ,

$$polyfit(3, k_0, x_{lowband}, lowEnv, lowEnvSlope);$$

có thể được sử dụng (bằng cách sử dụng đa thức bậc ba) và trong đó

$$lowEnv(k) = 10 \log_{10} \frac{\varphi_k(0,0)}{numTimeSlots \cdot RATE + 6}, 0 \leq k < k_0$$

$x_{lowband}(k)=[0 \dots k_0-1]$ , *numTimeSlot* là số lượng khe thời gian đường bao SBR hiện có trong khung, *RATE* là hằng số chỉ báo số lượng mẫu băng con QMF trên mỗi khe thời gian (ví dụ, 2),  $\varphi_k$  là hệ số bộ lọc dự đoán tuyến tính (có khả năng thu được từ phương pháp hiệp phương sai) và trong đó

$$meanNrg = \frac{\sum_{k=0}^{k_0-1} lowEnv(k)}{k_0}.$$

Dòng bit tạo ra theo chuẩn MPEG USAC (đôi khi được gọi trong bản mô tả này là “dòng bit USAC”) bao gồm nội dung âm thanh mã hóa và thường bao gồm siêu dữ liệu chỉ báo mỗi loại quy trình xử lý tái tạo phổ tần số cần được áp dụng bởi bộ giải mã để giải mã nội dung âm thanh của dòng bit USAC, và/hoặc siêu dữ liệu điều khiển quy trình xử lý tái tạo phổ tần số như vậy và/hoặc chỉ báo ít nhất một đặc điểm hoặc tham số của ít nhất một công cụ SBR và/hoặc công cụ eSBR cần được dùng để giải mã nội dung âm thanh của dòng bit USAC.

Ở đây, các tác giả sáng chế sử dụng cụm từ “siêu dữ liệu SBR nâng cao” (hoặc “siêu dữ liệu eSBR”) để chỉ siêu dữ liệu chỉ báo mỗi loại quy trình xử lý tái tạo phổ tần số cần

được áp dụng bởi bộ giải mã để giải mã nội dung âm thanh của dòng bit âm thanh được mã hóa (ví dụ, dòng bit USAC) và/hoặc điều khiển quy trình xử lý tái tạo phổ tần số như vậy, và/hoặc chỉ báo ít nhất một đặc điểm hoặc tham số của ít nhất một công cụ SBR và/hoặc công cụ eSBR cần được dùng để giải mã nội dung âm thanh như vậy, nhưng không được mô tả hoặc đề cập trong chuẩn MPEG-4 AAC. Ví dụ về siêu dữ liệu eSBR là siêu dữ liệu (chỉ báo, hoặc để điều khiển, quy trình xử lý tái tạo phổ tần số) được mô tả hoặc đề cập trong chuẩn MPEG USAC nhưng không có trong chuẩn MPEG-4 AAC. Vì vậy, siêu dữ liệu eSBR ở đây chỉ siêu dữ liệu mà không phải là siêu dữ liệu SBR, và siêu dữ liệu SBR ở đây chỉ siêu dữ liệu mà không phải là siêu dữ liệu eSBR.

Dòng bit USAC có thể bao gồm cả siêu dữ liệu SBR và siêu dữ liệu eSBR. Cụ thể hơn, dòng bit USAC có thể bao gồm siêu dữ liệu eSBR điều khiển việc thực hiện quy trình xử lý eSBR bằng bộ giải mã, và siêu dữ liệu SBR điều khiển việc thực hiện quy trình xử lý SBR bằng bộ giải mã. Theo các phương án điển hình của sáng chế, siêu dữ liệu eSBR (ví dụ, dữ liệu cấu hình eSBR riêng) được bao gồm (theo sáng chế) trong dòng bit MPEG-4 AAC (ví dụ, trong phần chứa `sbr_extension()` ở cuối tải tin SBR).

Việc thực hiện quy trình xử lý eSBR, trong khi giải mã dòng bit mã hóa sử dụng tập công cụ eSBR (gồm ít nhất một công cụ eSBR), bởi bộ giải mã tái tạo ra băng tần số cao của tín hiệu âm thanh, dựa trên việc tái tạo các chuỗi sóng hài đã cắt trong quá trình mã hóa. Quy trình xử lý eSBR như vậy thường điều chỉnh đường bao phổ của băng tần số cao được tạo ra và áp dụng quy trình lọc ngược, và thêm các thành phần nhiễu và dạng sin vào để tái tạo các đặc tính phổ của tín hiệu âm thanh gốc.

Theo các phương án điển hình của sáng chế, siêu dữ liệu eSBR được bao gồm (ví dụ, số lượng nhỏ các bit điều khiển là siêu dữ liệu eSBR được bao gồm) trong một hoặc nhiều trong số các đoạn siêu dữ liệu của dòng bit âm thanh được mã hóa (ví dụ, dòng bit MPEG-4 AAC) mà còn bao gồm dữ liệu âm thanh được mã hóa trong các đoạn khác (các đoạn dữ liệu âm thanh). Thông thường, ít nhất một đoạn siêu dữ liệu như vậy của mỗi khối trong dòng bit là (hoặc bao gồm) phần tử làm đầy (bao gồm mã định danh chỉ báo phần đầu của phần tử làm đầy), và siêu dữ liệu eSBR được bao gồm trong phần tử làm đầy sau mã định danh. Fig.1 là sơ đồ khối của chuỗi xử lý âm thanh làm ví dụ (hệ thống xử lý dữ liệu âm thanh), trong đó một hoặc nhiều trong số các phần tử của hệ thống có thể được tạo cấu hình theo một phương án của sáng chế. Hệ thống bao gồm các phần tử sau, được nối

với nhau như được thể hiện trên hình vẽ: bộ mã hóa 1, hệ thống phụ phân phối 2, bộ giải mã 3, và đơn vị xử lý hậu kỳ 4. Trong các thay đổi trên hệ thống được thể hiện, một hoặc nhiều trong số các phần tử này 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, bộ mã hóa 1 (mà tùy ý bao gồm bộ tiền xử lý) được tạo cấu hình để chấp nhận các mẫu PCM (miền thời gian) gồm nội dung âm thanh làm đầu vào, và xuất ra dòng bit âm thanh được mã hóa (có định dạng phù hợp với chuẩn MPEG-4 AAC) chỉ báo nội dung âm thanh. Dữ liệu của dòng bit chỉ báo nội dung âm thanh đôi khi được gọi trong bản mô tả là "dữ liệu âm thanh" hoặc "dữ liệu âm thanh mã hóa". Nếu bộ mã hóa được tạo cấu hình theo phương án điển hình của sáng chế, dòng bit âm thanh xuất ra từ bộ mã hóa bao gồm siêu dữ liệu eSBR (và thường là siêu dữ liệu khác nữa) cũng như dữ liệu âm thanh.

Một hoặc nhiều dòng bit âm thanh được mã hóa xuất ra từ bộ mã hóa 1 có thể được xác nhận với hệ thống phụ phân phối âm thanh mã hóa 2. Hệ thống phụ 2 được tạo cấu hình để lưu trữ và/hoặc phân phối mỗi dòng bit mã hóa xuất ra từ bộ mã hóa 1. Dòng bit âm thanh được mã hóa xuất ra từ bộ mã hóa 1 có thể được lưu trữ bởi hệ thống phụ 2 (ví dụ, ở dạng đĩa DVD hoặc đĩa Blu ray), hoặc được truyền bởi hệ thống phụ 2 (có thể thực hiện liên kết truyền hoặc mạng), hoặc có thể vừa được lưu trữ vừa được truyền bởi hệ thống phụ 2.

Bộ giải mã 3 được tạo cấu hình để giải mã dòng bit âm thanh MPEG-4 AAC mã hóa (tạo ra bởi bộ mã hóa 1) mà bộ giải mã nhận qua hệ thống phụ 2. Theo một số phương án, bộ giải mã 3 được tạo cấu hình để trích siêu dữ liệu eSBR từ mỗi khối của dòng bit, và để giải mã dòng bit (bao gồm bằng cách thực hiện xử lý eSBR nhờ sử dụng siêu dữ liệu eSBR đã trích) để tạo ra dữ liệu âm thanh giải mã (ví dụ, các dòng của các mẫu âm thanh PCM giải mã). Theo một số phương án, bộ giải mã 3 được tạo cấu hình để trích siêu dữ liệu SBR từ dòng bit (nhưng bỏ qua siêu dữ liệu eSBR bao gồm trong dòng bit), và để giải mã dòng bit (bao gồm bước thực hiện xử lý SBR sử dụng siêu dữ liệu SBR đã trích) để tạo ra dữ liệu âm thanh giải mã (ví dụ, các dòng của các mẫu âm thanh PCM giải mã). Thông thường, bộ giải mã 3 bao gồm bộ đệm lưu trữ (ví dụ, theo cách bất biến) các đoạn của dòng bit âm thanh được mã hóa nhận được từ hệ thống phụ 2.

Bộ xử lý hậu kỳ 4 trên Fig.1 được tạo cấu hình để chấp nhận dòng dữ liệu âm thanh giải mã từ bộ giải mã 3 (ví dụ, các mẫu âm thanh PCM giải mã), và để thực hiện quy trình xử lý hậu kỳ trên đó. Đơn vị xử lý hậu kỳ có thể còn được tạo cấu hình để kết xuất nội dung âm thanh xử lý hậu kỳ (hoặc âm thanh giải mã nhận được từ bộ giải mã 3) để phát lại bởi một hoặc nhiều loa.

Fig.2 là sơ đồ khối của bộ mã hóa (100) là một phương án của đơn vị xử lý âm thanh theo sáng chế. Bất kỳ trong số các thành phần hoặc phần tử của bộ mã hóa 100 có thể được thực thi dưới dạng một hoặc nhiều quy trình và/hoặc một hoặc nhiều mạch (ví dụ, ASIC, FPGA hoặc các mạch tích hợp khác), trong phần cứng, phần mềm, hoặc tổ hợp của phần cứng và phần mềm. Bộ mã hóa 100 bao gồm bộ mã hóa 105, tầng nhồi/định dạng 107, tầng tạo siêu dữ liệu 106, và bộ nhớ đệm 109, được kết nối như thể hiện. Thông thường, bộ mã hóa 100 còn bao gồm các phần tử xử lý khác (không được thể hiện trên hình vẽ). Bộ mã hóa 100 được tạo cấu hình để chuyển đổi dòng bit âm thanh đầu vào thành dòng bit MPEG-4 AAC đầu ra được mã hóa.

Bộ tạo siêu dữ liệu 106 được nối và được tạo cấu hình để tạo ra (và/hoặc đi qua tầng 107) siêu dữ liệu (bao gồm siêu dữ liệu eSBR và siêu dữ liệu SBR) để được đưa vào bởi tầng 107 trong dòng bit mã hóa cần được xuất ra từ bộ mã hóa 100.

Bộ mã hóa 105 được ghép nối và tạo cấu hình để mã hóa (ví dụ, bằng cách thực hiện nén trên đó) dữ liệu âm thanh đầu vào, và xác nhận âm thanh mã hóa thu được với tầng 107 để đưa vào dòng bit mã hóa cần được xuất ra từ tầng 107.

Tầng 107 được tạo cấu hình để ghép kênh âm thanh mã hóa từ bộ mã hóa 105 và siêu dữ liệu (bao gồm siêu dữ liệu eSBR và siêu dữ liệu SBR) từ bộ tạo 106 để tạo ra dòng bit mã hóa cần xuất ra từ tầng 107, tốt hơn là sao cho dòng bit mã hóa có định dạng như được xác định bởi một trong các phương án của sáng chế.

Bộ nhớ đệm 109 được tạo cấu hình để lưu trữ (ví dụ, theo cách bất biến) ít nhất một khối của dòng bit âm thanh được mã hóa xuất ra từ tầng 107, và chuỗi của các khối của dòng bit âm thanh được mã hóa sau đó được xác nhận từ bộ nhớ đệm 109 làm đầu ra từ bộ mã hóa 100 đến hệ thống phân phối.

Fig.3 là sơ đồ khối của hệ thống bao gồm bộ giải mã (200) là một phương án của đơn vị xử lý âm thanh theo sáng chế, và tùy ý còn có bộ xử lý hậu kỳ (300) được nối vào

đó. Bất kỳ trong số các thành phần hoặc phần tử của bộ giải mã 200 và bộ xử lý hậu kỳ 300 có thể được thực thi dưới dạng một hoặc nhiều quy trình và/hoặc một hoặc nhiều mạch (ví dụ, ASIC, FPGA, hoặc các mạch tích hợp khác), trong phần cứng, phần mềm, hoặc tổ hợp của phần cứng và phần mềm. Bộ giải mã 200 bao gồm bộ nhớ đệm 201, bộ khử định dạng tải tin dòng bit (bộ phân tích) 205, hệ thống phụ giải mã âm thanh 202 (đôi khi được gọi là tầng giải mã “lỗi” hoặc hệ thống phụ giải mã “lỗi”), tầng xử lý eSBR 203, và tầng tạo bit điều khiển 204, được kết nối như được thể hiện trên hình vẽ. Thông thường, bộ giải mã 200 còn bao gồm các phần tử xử lý khác (không được thể hiện trên hình vẽ).

Bộ nhớ đệm (bộ đệm) 201 lưu trữ (ví dụ, theo cách bất biến) ít nhất một khối của dòng bit âm thanh MPEG-4 AAC mã hóa nhận được bởi bộ giải mã 200. Trong quá trình hoạt động của bộ giải mã 200, chuỗi của các khối của dòng bit được xác nhận từ bộ đệm 201 với bộ khử định dạng 205.

Trong các thay đổi theo phương án trên Fig.3 (hoặc phương án trên Fig.4 sẽ được mô tả), APU mà không phải là bộ giải mã (ví dụ, APU 500 trên Fig.6) bao gồm bộ nhớ đệm (ví dụ, bộ nhớ đệm giống với bộ đệm 201) lưu trữ (ví dụ, theo cách bất biến) ít nhất một khối của dòng bit âm thanh được mã hóa (ví dụ, dòng bit âm thanh MPEG-4 AAC) thuộc cùng loại nhận được bởi bộ đệm 201 trên Fig.3 hoặc Fig.4 (tức là, dòng bit âm thanh được mã hóa bao gồm siêu dữ liệu eSBR).

Một lần nữa dựa vào Fig.3, bộ khử định dạng 205 được ghép nối và được tạo cấu hình để giải ghép kênh mỗi khối của dòng bit để trích siêu dữ liệu SBR (bao gồm dữ liệu đường bao lượng tử hóa) và siêu dữ liệu eSBR (và cũng thường là siêu dữ liệu khác) từ đó, để xác nhận ít nhất một siêu dữ liệu eSBR và siêu dữ liệu SBR với tầng xử lý eSBR 203, và cũng thường để xác nhận siêu dữ liệu trích khác với hệ thống phụ giải mã 202 (và tùy ý còn để điều khiển bộ tạo bit 204). Bộ khử định dạng 205 còn được ghép nối và tạo cấu hình để trích dữ liệu âm thanh từ mỗi khối của dòng bit, và để xác nhận dữ liệu âm thanh đã trích với hệ thống phụ giải mã (tầng giải mã) 202.

Hệ thống trên Fig.3 tùy ý còn bao gồm bộ xử lý hậu kỳ 300. Bộ xử lý hậu kỳ 300 bao gồm bộ nhớ đệm (bộ đệm) 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 301 lưu trữ (ví dụ, theo cách bất biến) ít nhất một khối (hoặc khung) của dữ liệu âm thanh giải mã nhận được bởi bộ xử lý hậu kỳ 300 từ bộ giải mã 200. Các phần tử xử lý của bộ xử lý hậu kỳ

300 được nối và được tạo cấu hình để nhận và xử lý thích ứng chuỗi của các khối (hoặc khung) của âm thanh giải mã xuất ra từ bộ đệm 301, bằng cách sử dụng siêu dữ liệu xuất ra từ hệ thống phụ giải mã 202 (và/hoặc bộ khử định dạng 205) và/hoặc các bit điều khiển xuất ra từ tầng 204 của bộ giải mã 200.

Hệ thống phụ giải mã âm thanh 202 của bộ giải mã 200 được tạo cấu hình để giải mã dữ liệu âm thanh trích bởi bộ phân tích 205 (quy trình giải mã như vậy có thể được gọi là hoạt động giải mã “lỗi”) để tạo ra dữ liệu âm thanh giải mã, và để xác nhận dữ liệu âm thanh giải mã với tầng xử lý eSBR 203. Quy trình giải mã được thực hiện ở miền tần số và thường bao gồm quy trình lượng tử hóa ngược tiếp theo là xử lý phổ. Thông thường, tầng xử lý cuối trong hệ thống phụ 202 áp dụng kỹ thuật biến đổi từ miền tần số thành miền thời gian cho dữ liệu âm thanh miền tần số giải mã, sao cho đầu ra của hệ thống phụ là dữ liệu âm thanh miền thời gian giải mã. Tầng 203 được tạo cấu hình để áp dụng các công cụ SBR và công cụ eSBR được chỉ báo bởi siêu dữ liệu eSBR và eSBR (trích bởi bộ phân tích 205) cho dữ liệu âm thanh giải mã (tức là, để thực hiện xử lý SBR và eSBR trên đầu ra của hệ thống phụ giải mã 202 bằng cách sử dụng siêu dữ liệu SBR và eSBR) để tạo ra dữ liệu âm thanh giải mã hoàn toàn được xuất ra (ví dụ, vào bộ xử lý hậu kỳ 300) từ bộ giải mã 200. Thông thường, bộ giải mã 200 bao gồm bộ nhớ (truy cập được bởi hệ thống phụ 202 và tầng 203) lưu trữ dữ liệu âm thanh và siêu dữ liệu đã khử định dạng xuất ra từ bộ khử định dạng 205, và tầng 203 được tạo cấu hình để truy cập dữ liệu âm thanh và siêu dữ liệu (bao gồm siêu dữ liệu SBR và siêu dữ liệu eSBR) khi cần trong quá trình xử lý SBR và eSBR. Quy trình xử lý SBR và xử lý eSBR ở tầng 203 có thể được xem là quy trình xử lý hậu kỳ trên đầu ra của hệ thống phụ giải mã lỗi 202. Tùy ý, bộ giải mã 200 còn bao gồm hệ thống phụ tăng mức cuối cùng (mà có thể áp dụng các công cụ mã hóa âm thanh nổi có tham số (“parametric stereo - PS”) định nghĩa trong chuẩn MPEG-4 AAC, nhờ sử dụng siêu dữ liệu PS trích bởi bộ khử định dạng 205 và/hoặc các bit điều khiển tạo ra trong hệ thống phụ 204) mà được nối và được tạo cấu hình để thực hiện tăng mức trên đầu ra của tầng 203 cho âm thanh tăng mức được giải mã đầy đủ được tạo ra mà được xuất ra từ bộ giải mã 200. Theo cách khác, bộ xử lý hậu kỳ 300 được tạo cấu hình để thực hiện tăng mức trên đầu ra của bộ giải mã 200 (ví dụ, sử dụng siêu dữ liệu PS trích bởi bộ khử định dạng 205 và/hoặc các bit điều khiển tạo ra trong hệ thống phụ 204).

Để đáp lại siêu dữ liệu trích bởi bộ khử định dạng 205, bộ tạo bit điều khiển 204 có thể tạo ra dữ liệu điều khiển, và dữ liệu điều khiển này có thể được sử dụng trong bộ giải mã 200 (ví dụ, trong hệ thống phụ tăng mức cuối cùng) và/hoặc được xác nhận là đầu ra của bộ giải mã 200 (ví dụ, cho bộ xử lý hậu kỳ 300 dùng để xử lý hậu kỳ). Để đáp lại siêu dữ liệu trích từ dòng bit đầu vào (và tùy ý cũng để đáp lại dữ liệu điều khiển), tầng 204 có thể tạo ra (và xác nhận với bộ xử lý hậu kỳ 300) các bit điều khiển chỉ báo rằng dữ liệu âm thanh giải mã xuất ra từ tầng xử lý eSBR 203 cần trải qua kiểu quy trình xử lý hậu kỳ riêng. Theo một số phương án thực hiện, bộ giải mã 200 được tạo cấu hình để xác nhận siêu dữ liệu trích bởi bộ khử định dạng 205 từ dòng bit đầu vào với bộ xử lý hậu kỳ 300, và bộ xử lý hậu kỳ 300 được tạo cấu hình để thực hiện xử lý hậu kỳ trên dữ liệu âm thanh giải mã xuất ra từ bộ giải mã 200 bằng cách sử dụng siêu dữ liệu.

Fig.4 là sơ đồ khối của đơn vị xử lý âm thanh ("audio processing unit - APU") (210) là một phương án khác của đơn vị xử lý âm thanh theo sáng chế. APU 210 là bộ giải mã kế thừa không được tạo cấu hình để thực hiện quy trình xử lý eSBR. Bất kỳ trong số các thành phần hoặc phần tử APU 210 có thể được thực thi dưới dạng một hoặc nhiều quy trình và/hoặc một hoặc nhiều mạch (ví dụ, ASIC, FPGA hoặc các mạch tích hợp khác), trong phần cứng, phần mềm, hoặc tổ hợp của phần cứng và phần mềm. APU 210 bao gồm bộ nhớ đệm 201, bộ khử định dạng tải tin dòng bit (bộ phân tích) 215, hệ thống phụ giải mã âm thanh 202 (đôi khi được gọi là tầng giải mã "lỗi" hoặc hệ thống phụ giải mã "lỗi"), và tầng xử lý SBR 213, được kết nối như thể hiện trên hình vẽ. Thông thường, APU 210 còn bao gồm các phần tử xử lý khác (không được thể hiện trên hình vẽ). APU 210 có thể là, chẳng hạn, bộ mã hóa, bộ giải mã hoặc bộ chuyển mã âm thanh.

Các phần tử 201 và 202 của APU 210 giống với các phần tử được đánh số đồng nhất của bộ giải mã 200 (trên Fig.3) và phần mô tả trên đây về chúng sẽ không được lặp lại. Trong quá trình hoạt động của APU 210, chuỗi các khối của dòng bit âm thanh được mã hóa (dòng bit MPEG-4 AAC) nhận được bởi APU 210 được xác nhận từ bộ đệm 201 với bộ khử định dạng 215.

Bộ khử định dạng 215 được nối và được tạo cấu hình để giải ghép kênh mỗi khối của dòng bit để trích siêu dữ liệu SBR (bao gồm dữ liệu đường bao lượng tử hóa) và thường còn là siêu dữ liệu khác từ đó, nhưng bỏ qua siêu dữ liệu eSBR mà có thể được bao gồm trong dòng bit theo phương án bất kỳ của sáng chế. Bộ khử định dạng 215 được tạo cấu

hình để xác nhận ít nhất một siêu dữ liệu SBR với tầng xử lý SBR 213. Bộ khử định dạng 215 còn được ghép nối và được tạo cấu hình để trích dữ liệu âm thanh từ mỗi khối của dòng bit, và để xác nhận dữ liệu âm thanh trích với hệ thống phụ giải mã (tầng giải mã) 202.

Hệ thống phụ giải mã âm thanh 202 của bộ giải mã 200 được tạo cấu hình để giải mã dữ liệu âm thanh trích bởi bộ khử định dạng 215 (bước giải mã này có thể được gọi là hoạt động giải mã "lỗi") để tạo ra dữ liệu âm thanh giải mã, và để xác nhận dữ liệu âm thanh giải mã với tầng xử lý SBR 213. Quy trình giải mã được thực hiện trong miền tần số. Thông thường, tầng xử lý cuối cùng trong hệ thống phụ 202 áp dụng biến đổi từ miền tần số thành miền thời gian cho dữ liệu âm thanh miền tần số được giải mã, do đó đầu ra của hệ thống phụ là dữ liệu âm thanh miền thời gian đã giải mã. Tầng 213 được tạo cấu hình để áp dụng các công cụ SBR (mà không phải là các công cụ eSBR) được chỉ báo bởi siêu dữ liệu SBR (trích bởi bộ khử định dạng 215) cho dữ liệu âm thanh giải mã (tức là, để thực hiện xử lý SBR trên đầu ra của hệ thống phụ giải mã 202 sử dụng siêu dữ liệu SBR) để tạo ra dữ liệu âm thanh được giải mã đầy đủ mà được xuất ra (ví dụ, đến bộ xử lý hậu kỳ 300) từ APU 210. Thông thường, APU 210 bao gồm bộ nhớ (truy cập được bởi hệ thống phụ 202 và tầng 213) lưu trữ dữ liệu âm thanh và siêu dữ liệu đã khử định dạng xuất ra từ bộ khử định dạng 215, và tầng 213 được tạo cấu hình để truy cập dữ liệu âm thanh và siêu dữ liệu (bao gồm siêu dữ liệu SBR) khi cần trong quá trình xử lý SBR. Quy trình xử lý SBR ở tầng 213 có thể được xem là bước xử lý hậu kỳ trên đầu ra của hệ thống phụ giải mã lỗi 202. Tùy ý, APU 210 còn bao gồm hệ thống phụ tăng mức cuối cùng (có thể áp dụng các công cụ mã hóa âm thanh nổi có tham số (PS) được định nghĩa trong chuẩn MPEG-4 AAC, sử dụng siêu dữ liệu PS trích bởi bộ khử định dạng 215) mà được nối và được tạo cấu hình để thực hiện tăng mức trên đầu ra của tầng 213 cho âm thanh tăng mức được tạo ra đã giải mã đầy đủ mà được xuất ra từ APU 210. Theo cách khác, bộ xử lý hậu kỳ được tạo cấu hình để thực hiện tăng mức trên đầu ra APU 210 (ví dụ, sử dụng siêu dữ liệu PS trích bởi bộ khử định dạng 215 và/hoặc các bit điều khiển tạo ra trong APU 210).

Một số phương án thực hiện khác nhau của bộ mã hóa 100, bộ giải mã 200 và APU 210 được tạo cấu hình để thực hiện các phương án khác nhau của phương pháp theo sáng chế.

Theo một số phương án, siêu dữ liệu eSBR được bao gồm (ví dụ, số lượng nhỏ các bit điều khiển là siêu dữ liệu eSBR được bao gồm) trong dòng bit âm thanh được mã hóa (ví dụ, dòng bit MPEG-4 AAC), sao cho bộ giải mã kế thừa (mà không được tạo cấu hình để phân tích siêu dữ liệu eSBR, hoặc để sử dụng công cụ eSBR bất kỳ mà siêu dữ liệu eSBR gắn với nó) có thể bỏ qua siêu dữ liệu eSBR nhưng sẽ giải mã dòng bit ở mức độ có thể mà không sử dụng siêu dữ liệu eSBR hoặc công cụ eSBR bất kỳ mà siêu dữ liệu eSBR gắn với nó, thường không có điểm trừ đáng kể nào về chất lượng âm thanh được giải mã. Tuy nhiên, các bộ giải mã eSBR được tạo cấu hình để phân tích dòng bit để nhận dạng siêu dữ liệu eSBR và sử dụng ít nhất một công cụ eSBR để đáp lại siêu dữ liệu eSBR, sẽ hưởng các lợi ích của việc sử dụng ít nhất một công cụ eSBR như vậy. Do đó, các phương án của sáng chế đề xuất phương tiện để truyền một cách hiệu quả siêu dữ liệu hoặc dữ liệu điều khiển eSBR theo cách tương thích ngược.

Thông thường, siêu dữ liệu eSBR trong dòng bit chỉ báo (ví dụ, chỉ báo ít nhất một đặc điểm hoặc tham số của) một hoặc nhiều trong số các công cụ eSBR sau đây (được mô tả trong chuẩn MPEG USAC, và có thể có hoặc có thể không được áp dụng bởi bộ mã hóa trong quá trình tạo ra dòng bit):

- Chuyển vị sóng hài; và
- Tiền xử lý bổ sung và QMF (làm phẳng trước).

Chẳng hạn, siêu dữ liệu eSBR bao gồm trong dòng bit có thể chỉ báo các giá trị của các tham số (được mô tả trong chuẩn MPEG USAC và trong bản mô tả này): `sbrPatchingMode[ch]`, `sbrOversamplingFlag[ch]`, `sbrPitchInBins[ch]`, `sbrPitchInBins[ch]`, và `bs_sbr_preprocessing`.

Ở đây, ký hiệu `X[ch]`, trong đó `X` là tham số nào đó, chỉ ra rằng tham số thuộc về kênh (“ch”) của nội dung âm thanh của dòng bit mã hóa cần giải mã. Để đơn giản, đôi khi các tác giả sáng chế lược bỏ cụm từ `[ch]`, và giả định rằng tham số liên quan thuộc về kênh nội dung âm thanh.

Ở đây, ký hiệu `X[ch][env]`, trong đó `X` là tham số nào đó, chỉ ra rằng tham số này thuộc về đường bao SBR (“env”) của kênh (“ch”) của nội dung âm thanh của dòng bit mã hóa cần giải mã. Để đơn giản, đôi khi các tác giả sáng chế lược bỏ cụm từ `[env]` và `[ch]`, và giả định rằng tham số liên quan thuộc về đường bao SBR của kênh nội dung âm thanh.

Khi giải mã dòng bit mã hóa, việc thực hiện chuyển vị sóng hài trong tầng xử lý eSBR của quy trình giải mã (cho mỗi kênh, “ch”, của nội dung âm thanh chỉ báo bởi dòng bit) được điều khiển bởi các tham số siêu dữ liệu eSBR sau: `sbrPatchingMode[ch]`; `sbrOversamplingFlag[ch]`; `sbrPitchInBinsFlag[ch]`; và `sbrPitchInBins[ch]`.

Giá trị “`sbrPatchingMode[ch]`” chỉ báo loại bộ chuyển vị được sử dụng trong eSBR: `sbrPatchingMode[ch] = 1` biểu thị quy trình và chuyển vị tuyến tính như được mô tả trong Đoạn 4.6.18 của chuẩn MPEG-4 AAC (như được sử dụng với SBR chất lượng cao hoặc SBR công suất thấp); `sbrPatchingMode[ch] = 0` chỉ báo và SBR sóng hài như được mô tả trong Đoạn 7.5.3 hoặc 7.5.4 của chuẩn MPEG USAC.

Giá trị “`sbrOversamplingFlag[ch]`” biểu thị việc sử dụng quy trình tăng mẫu miền tần số thích ứng tín hiệu trong eSBR kết hợp với quy trình và SBR sóng hài dựa trên DFT như được mô tả trong Đoạn 7.5.3 của chuẩn MPEG USAC. Cờ này điều khiển kích thước của các DFT được sử dụng trong bộ chuyển vị. Giá trị 1 biểu thị quy trình tăng mẫu miền tần số thích ứng tín hiệu được kích hoạt như được mô tả trong Đoạn 7.5.3.1 của chuẩn MPEG USAC; giá trị 0 biểu thị quy trình tăng mẫu miền tần số thích ứng tín hiệu bị vô hiệu hóa như được mô tả trong Đoạn 7.5.3.1 của chuẩn MPEG USAC.

Giá trị “`sbrPitchInBinsFlag[ch]`” điều khiển việc dịch của tham số `sbrPitchInBins[ch]`: 1 biểu thị rằng giá trị `sbrPitchInBins[ch]` là hợp lệ và lớn hơn không; 0 biểu thị rằng giá trị `sbrPitchInBins[ch]` được thiết lập bằng không.

Giá trị “`sbrPitchInBins[ch]`” điều khiển việc bổ sung các số hạng tích có hướng trong bộ chuyển vị sóng hài SBR. Giá trị `sbrPitchInBins[ch]` là giá trị nguyên nằm trong khoảng  $[0, 127]$  và biểu diễn khoảng cách đo được ở các bin tần số đối với DFT 1536 dòng hoạt động trên tần số lấy mẫu của bộ mã hóa lỗi.

Trong trường hợp mà dòng bit MPEG-4 AAC chỉ báo cặp kênh SBR có các kênh của chúng không được nối (chứ không phải là kênh SBR đơn), dòng bit chỉ báo hai ví dụ của cú pháp ở trên (cho quy trình chuyển vị sóng hài hoặc không chuyển vị sóng hài), mỗi ví dụ cho mỗi kênh của `sbr_channel_pair_element()`.

Sự chuyển vị sóng hài của công cụ eSBR thường cải thiện chất lượng tín hiệu nhạc âm giải mã ở các tần số giao cắt vị sóng hài tương đối thấp. Việc không chuyển vị sóng hài (tức là, và phổ kế thừa) thường cải thiện tín hiệu tiếng nói. Vì thế, điểm khởi đầu trong

quyết định về loại chuyển tiếp nào thích hợp hơn để mã hóa nội dung âm thanh cụ thể là chọn phương pháp chuyển tiếp tùy thuộc vào việc phát hiện tiếng nói/âm nhạc với quy trình chuyển vị sóng hài được dùng trong quy trình vá nội dung và phổ âm nhạc trong nội dung tiếng nói.

Việc thực hiện quy trình làm phẳng trước trong quy trình xử lý eSBR được điều khiển bởi giá trị của tham số siêu dữ liệu eSBR một bit được gọi là “bs\_sbr\_preprocessing”, có nghĩa là quy trình làm phẳng trước được thực hiện hoặc không được thực hiện tùy thuộc vào giá trị của bit đơn này. Khi sử dụng thuật toán vá QMF SBR, như được mô tả trong Đoạn 4.6.18.6.3 của chuẩn MPEG-4 AAC, thì bước làm phẳng trước có thể được thực hiện (khi được chỉ báo bởi tham số “bs\_sbr\_preprocessing”) với nỗ lực tránh sự gián đoạn về hình dạng của đường bao phổ của tín hiệu tần số cao được đưa vào bộ điều chỉnh đường bao tiếp theo (bộ điều chỉnh đường bao thực hiện tầng xử lý eSBR khác). Quy trình làm phẳng trước thường cải thiện hoạt động của tầng điều chỉnh đường bao tiếp theo, kết quả là tín hiệu băng cao được coi là ổn định hơn.

Yêu cầu tốc độ bit tổng để bao gồm trong dòng bit MPEG-4 AAC siêu dữ liệu eSBR chỉ báo các công cụ eSBR đã nêu ở trên (chuyển vị sóng hài và làm phẳng trước) được kỳ vọng ở mức vài trăm bit trên giây vì chỉ cần có dữ liệu điều khiển vi sai để thực hiện xử lý eSBR được truyền theo một số phương án của sáng chế. Bộ giải mã kế thừa có thể bỏ qua thông tin này vì nó được bao gồm theo cách tương thích ngược (sẽ được giải thích sau). Do đó, ảnh hưởng bất lợi lên tốc độ bit kết liên quan đến việc bao gồm siêu dữ liệu eSBR là không đáng kể, vì một số lý do, bao gồm:

- Điểm trừ tốc độ bit (do bao gồm siêu dữ liệu eSBR) là phần rất nhỏ của tốc độ bit tổng vì chỉ cần có dữ liệu điều khiển vi sai để thực hiện xử lý eSBR được truyền (và không phải là việc truyền đồng thời dữ liệu điều khiển SBR); và
- Việc điều chỉnh thông tin điều khiển liên quan SBR thường không phụ thuộc vào các phần chi tiết của quy trình chuyển tiếp. Các ví dụ về thời điểm dữ liệu điều khiển phụ thuộc vào hoạt động của bộ chuyển vị được mô tả sau trong bản mô tả này.

Vì vậy, các phương án của sáng chế đề xuất phương tiện để truyền một cách hiệu quả dữ liệu hoặc siêu dữ liệu điều khiển eSBR theo cách tương thích ngược. Quy trình truyền hiệu quả dữ liệu điều khiển eSBR này giảm yêu cầu bộ nhớ của bộ giải mã, bộ mã

hóa, và bộ chuyển mã nhờ sử dụng các khía cạnh của sáng chế, đồng thời không có tác dụng ngược rõ ràng lên tốc độ bit. Ngoài ra, độ phức tạp và các yêu cầu xử lý liên quan đến việc thực hiện eSBR theo các phương án của sáng chế cũng giảm vì dữ liệu SBR chỉ cần được xử lý một lần và không truyền đồng thời, đó là trường hợp nếu eSBR được xử lý dưới dạng loại đối tượng hoàn toàn riêng theo MPEG-4 AAC thay vì được tích hợp vào bộ mã hóa giải mã MPEG-4 AAC theo cách tương thích ngược.

Tiếp theo, như được thể hiện trên Fig.7, sáng chế mô tả các phần tử của khối (“raw\_data\_block”) của dòng bit MPEG-4 AAC mà siêu dữ liệu eSBR được bao gồm trong đó theo một số phương án của sáng chế. Fig.7 là sơ đồ của khối (“raw\_data\_block”) của dòng bit MPEG-4 AAC, thể hiện một số đoạn của dòng bit này.

Khối dòng bit MPEG-4 AAC có thể bao gồm ít nhất một “single\_channel\_element()” (ví dụ, phần tử kênh đơn thể hiện trên Fig.7), và/hoặc ít nhất một “channel\_pair\_element()” (không được thể hiện cụ thể trên Fig.7 mặc dù nó có thể có mặt), bao gồm dữ liệu âm thanh cho chương trình âm thanh. Khối có thể còn bao gồm một số “fill\_elements” (ví dụ, phần tử làm đầy 1 và/hoặc phần tử làm đầy 2 trên Fig.7) bao gồm dữ liệu (ví dụ, siêu dữ liệu) liên quan đến chương trình. Mỗi “single\_channel\_element()” bao gồm mã định danh (ví dụ, “ID1” trên Fig.7) chỉ báo phần bắt đầu của phần tử kênh đơn, và có thể bao gồm dữ liệu âm thanh chỉ báo một kênh khác của chương trình âm thanh đa kênh. Mỗi “channel\_pair\_element” bao gồm một mã định danh (không được thể hiện trên Fig.7) chỉ báo phần bắt đầu của phần tử cặp kênh, và có thể bao gồm dữ liệu âm thanh chỉ báo hai kênh của chương trình.

fill\_element (được gọi là phần tử làm đầy) của dòng bit MPEG-4 AAC bao gồm mã định danh (“ID2” trên Fig.7) chỉ báo phần bắt đầu của phần tử làm đầy, và dữ liệu làm đầy sau mã định danh. Mã định danh ID2 có thể gồm có số nguyên không dấu 3 bit với bit có ý nghĩa nhất được truyền đầu tiên (“uimsbf”) có giá trị 0x6. Dữ liệu làm đầy có thể bao gồm phần tử extension\_payload() (đôi khi được gọi là tải tin mở rộng) có cú pháp được thể hiện trên Bảng 4.57 của chuẩn MPEG-4 AAC. Một vài loại tải tin mở rộng có mặt và được nhận dạng thông qua tham số “extension\_type”, là số nguyên không dấu 4 bit với bit có ý nghĩa nhất được truyền đầu tiên (uimsbf).

Dữ liệu làm đầy (ví dụ, tải tin mở rộng của nó) có thể bao gồm phần đầu hoặc mã định danh (ví dụ, “header1” trên Fig.7) biểu thị một đoạn của dữ liệu làm đầy chỉ báo đối

trạng (tức là, phần đầu khởi tạo loại “đối tượng SBR”, được gọi là `sbr_extension_data()` trong chuẩn MPEG-4 AAC). Chẳng hạn, tải tin mở rộng SBR được nhận dạng với giá trị ‘1101’ hoặc ‘1110’ đối với trường `extension_type` trong phần đầu, với mã định danh ‘1101’ nhận dạng tải tin mở rộng với dữ liệu SBR và ‘1110’ nhận dạng và tải tin mở rộng với dữ liệu SBR bằng kỹ thuật kiểm tra dư vòng (Cyclic Redundancy Check - CRC) để xác nhận độ chính xác của dữ liệu SBR.

Khi phần đầu (ví dụ, trường `extension_type`) khởi tạo loại đối tượng SBR, siêu dữ liệu SBR (đôi khi được gọi là “dữ liệu tái tạo phổ tần số,” và được gọi là `sbr_data()` trong chuẩn MPEG-4 AAC) tiếp theo phần đầu, và ít nhất một phần tử mở rộng tái tạo phổ tần số (ví dụ, “phần tử mở rộng SBR” của phần tử làm đầy 1 trên Fig.7) có thể theo sau siêu dữ liệu SBR. Phần tử mở rộng tái tạo phổ tần số như vậy (một đoạn của dòng bit) được gọi là phần chứa “`sbr_extension()`” trong chuẩn MPEG-4 AAC. Phần tử mở rộng tái tạo phổ tần số tùy ý bao gồm phần đầu (ví dụ, “phần đầu mở rộng SBR” của phần tử làm đầy 1 trên Fig.7).

Chuẩn MPEG-4 AAC dự tính rằng phần tử mở rộng tái tạo phổ tần số có thể bao gồm dữ liệu PS (mã hóa âm thanh nổi có tham số) đối với dữ liệu âm thanh của chương trình. Chuẩn MPEG-4 AAC dự tính rằng khi phần đầu của phần tử làm đầy (ví dụ, của tải tin mở rộng của nó) khởi tạo loại đối tượng SBR (header1” trên Fig.7) và phần tử mở rộng tái tạo phổ tần số của phần tử làm đầy bao gồm dữ liệu PS, phần tử làm đầy (ví dụ, tải tin mở rộng của nó) bao gồm dữ liệu tái tạo phổ tần số, và tham số “`bs_extension_id`” có giá trị (tức là, `bs_extension_id = 2`) chỉ báo rằng dữ liệu PS được bao gồm trong phần tử mở rộng tái tạo phổ tần số của phần tử làm đầy.

Theo một số phương án của sáng chế, siêu dữ liệu eSBR (ví dụ, cờ chỉ báo liệu quy trình xử lý eSBR có được thực hiện trên nội dung âm thanh của khối hay không) được bao gồm trong phần tử mở rộng tái tạo phổ tần số của phần tử làm đầy. Chẳng hạn, cờ như vậy được chỉ báo trong phần tử làm đầy 1 trên Fig.7, trong đó cờ xuất hiện sau phần đầu (“phần đầu mở rộng SBR” của phần tử làm đầy 1) của “phần tử mở rộng SBR” của phần tử làm đầy 1. Tùy ý, cờ và siêu dữ liệu eSBR bổ sung như vậy được bao gồm trong phần tử mở rộng tái tạo phổ tần số sau phần đầu của phần tử mở rộng tái tạo phổ tần số (ví dụ, trong phần tử mở rộng SBR của phần tử làm đầy 1 trên Fig.7, sau phần đầu mở rộng SBR). Theo một số phương án của sáng chế, phần tử làm đầy bao gồm siêu dữ liệu eSBR còn bao gồm

tham số “bs\_extension\_id” có giá trị (ví dụ, bs\_extension\_id = 3) chỉ báo rằng siêu dữ liệu eSBR được bao gồm trong phần tử làm đầy và chỉ báo rằng quy trình xử lý eSBR sẽ được thực hiện trên nội dung âm thanh của khối liên quan.

Theo một số phương án của sáng chế, siêu dữ liệu eSBR được bao gồm trong phần tử làm đầy (ví dụ, phần tử làm đầy 2 trên Fig.7) của dòng bit MPEG-4 AAC ngoài phần tử mở rộng tái tạo phổ tần số (phần tử mở rộng SBR) của phần tử làm đầy. Điều này là do các phần tử làm đầy chứa extension\_payload() với dữ liệu SBR hoặc dữ liệu SBR có CRC không chứa tải tin mở rộng khác bất kỳ của loại mở rộng khác bất kỳ. Do đó, theo các phương án trong đó siêu dữ liệu eSBR được lưu trữ tải tin mở rộng của chính nó, phần tử làm đầy riêng được dùng để lưu trữ siêu dữ liệu eSBR. Phần tử làm đầy như vậy bao gồm mã định danh (ví dụ, “ID2” trên Fig.7) chỉ báo phần bắt đầu của phần tử làm đầy, và dữ liệu làm đầy sau mã định danh. Dữ liệu làm đầy có thể bao gồm phần tử extension\_payload() (đôi khi được gọi là tải tin mở rộng) có cú pháp được thể hiện trong Bảng 4.57 của chuẩn MPEG-4 AAC. Dữ liệu làm đầy (ví dụ, tải tin mở rộng của nó) bao gồm phần đầu (ví dụ, “header2” của phần tử làm đầy 2 trên Fig.7) chỉ báo đối tượng eSBR (tức là, phần đầu khởi tạo loại đối tượng eSBR), và dữ liệu làm đầy (ví dụ, tải tin mở rộng của nó) bao gồm siêu dữ liệu eSBR sau phần đầu. Chẳng hạn, phần tử làm đầy 2 trên Fig.7 bao gồm phần đầu như vậy (“header2”) và còn bao gồm, sau phần đầu, siêu dữ liệu eSBR (tức là, “cờ” trong phần tử làm đầy 2, chỉ báo liệu quy trình xử lý eSBR có phải được thực hiện trên nội dung âm thanh của khối hay không). Tùy ý, siêu dữ liệu eSBR bổ sung còn được bao gồm trong dữ liệu làm đầy của phần tử làm đầy 2 trên Fig.7, sau header2. Theo các phương án được mô tả trong đoạn này, phần đầu (ví dụ, header2 trên Fig.7) có giá trị nhận dạng mà không phải là một trong các giá trị thông thường được chỉ rõ trong Bảng 4.57 của chuẩn MPEG-4 AAC, và thay vào đó chỉ báo tải tin mở rộng eSBR (do đó trường extension\_type của phần đầu chỉ báo rằng dữ liệu làm đầy bao gồm siêu dữ liệu eSBR).

Theo nhóm phương án thứ nhất, sáng chế đề xuất bộ xử lý âm thanh (ví dụ, bộ giải mã), gồm:

bộ nhớ (ví dụ, bộ đệm 201 trên Fig.3 hoặc Fig.4) được tạo cấu hình để lưu trữ ít nhất một khối của dòng bit âm thanh được mã hóa (ví dụ, ít nhất một khối của dòng bit MPEG-4 AAC);

bộ khử định dạng tải tin dòng bit (ví dụ, phần tử 205 trên Fig.3 hoặc phần tử 215 trên Fig.4) được ghép nối với bộ nhớ và được tạo cấu hình để giải ghép kênh ít nhất một phần của khối của dòng bit này; và

hệ thống phụ giải mã (ví dụ, phần tử 202 và 203 trên Fig.3, hoặc phần tử 202 và 213 trên Fig.4), được ghép nối và được tạo cấu hình để giải mã ít nhất một phần nội dung âm thanh của khối của dòng bit này, trong đó khối bao gồm:

phần tử làm đầy, bao gồm mã định danh chỉ báo phần bắt đầu của phần tử làm đầy (ví dụ, mã định danh “id\_syn\_ele” có giá trị 0x6, trong bảng 4.85 của chuẩn MPEG-4 AAC), và dữ liệu làm đầy sau mã định danh, trong đó dữ liệu làm đầy bao gồm:

ít nhất một cờ xác định xem quy trình xử lý eSBR có được thực hiện trên nội dung âm thanh của khối hay không (ví dụ, sử dụng dữ liệu tái tạo phổ tần số và siêu dữ liệu eSBR bao gồm trong khối).

Cờ là siêu dữ liệu eSBR, và ví dụ về cờ là cờ sbrPatchingMode. Một ví dụ khác về cờ là cờ harmonicSBR. Cả hai cờ này chỉ báo liệu dạng tái tạo phổ tần số cơ bản hay dạng tái tạo phổ tần số nâng cao sẽ được thực hiện trên dữ liệu âm thanh của khối. Dạng tái tạo phổ tần số cơ bản là dạng vá phổ, và dạng tái tạo phổ tần số nâng cao là dạng chuyển vị sóng hài.

Theo một số phương án, dữ liệu làm đầy còn bao gồm siêu dữ liệu eSBR bổ sung (tức là, siêu dữ liệu eSBR ngoài cờ).

Bộ nhớ có thể là bộ nhớ đệm (ví dụ, phương án thực hiện của bộ đệm 201 trên Fig.4) lưu trữ (ví dụ, theo cách bất biến) ít nhất một khối của dòng bit âm thanh được mã hóa.

Ước lượng rằng độ phức tạp của việc thực hiện quy trình xử lý eSBR (sử dụng chuyển vị sóng hài eSBR và làm phẳng trước) bởi bộ giải mã eSBR trong quá trình giải mã dòng bit MPEG-4 AAC bao gồm siêu dữ liệu eSBR (chỉ báo các công cụ eSBR này) có thể sẽ như sau (đối với việc giải mã thông thường với các tham số được chỉ báo):

- Chuyển vị sóng hài (16 kbps, 14400/28800 Hz)
  - dựa trên DFT: 3,68 WMOPS (hàng triệu phép toán trên giây có trọng số);
  - Dựa trên QMF: 0,98 WMOPS;

- Quy trình xử lý trước và QMF (làm phẳng trước). 0,1WMOPS.

Đã biết rằng phép chuyển vị dựa trên DFT thường thực hiện tốt hơn phép chuyển vị dựa trên QMF đối với các phân chuyển tiếp.

Theo một số phương án của sáng chế, phần tử làm đầy (của dòng bit âm thanh được mã hóa) bao gồm siêu dữ liệu eSBR còn bao gồm tham số (ví dụ, tham số “bs\_extension\_id”) có giá trị (ví dụ, bs\_extension\_id = 3) báo hiệu rằng siêu dữ liệu eSBR được bao gồm trong phần tử làm đầy và việc xử lý eSBR phải được thực hiện trên nội dung âm thanh của khối liên quan, và/hoặc tham số (ví dụ, cùng tham số “bs\_extension\_id”) có giá trị (ví dụ, bs\_extension\_id = 2) báo hiệu rằng phần chứa sbr\_extension() của phần tử làm đầy bao gồm dữ liệu PS. Ví dụ, như được chỉ báo trong Bảng 1 dưới đây, tham số này có giá trị bs\_extension\_id = 2 có thể báo hiệu rằng phần chứa sbr\_extension() của phần tử làm đầy bao gồm dữ liệu PS, và các tham số này có giá trị bs\_extension\_id = 3 có thể báo hiệu rằng phần chứa sbr\_extension() của phần tử làm đầy bao gồm siêu dữ liệu eSBR:

Bảng 1

| bs_extension_id | Ý nghĩa           |
|-----------------|-------------------|
| 0               | Dự trữ            |
| 1               | Dự trữ            |
| 2               | EXTENSION_ID_PS   |
| 3               | EXTENSION_ID_ESBR |

Theo một số phương án của sáng chế, cú pháp của mỗi phần tử mở rộng tái tạo phổ tần số bao gồm siêu dữ liệu eSBR và/hoặc dữ liệu PS là như được chỉ ra trong Bảng 2 dưới đây (trong đó “sbr\_extension()” chỉ phần chứa là phần tử mở rộng tái tạo phổ tần số, “bs\_extension\_id” là như được mô tả trong Bảng 1 trên đây, “ps\_data” chỉ dữ liệu PS, và “esbr\_data” chỉ siêu dữ liệu eSBR):

Bảng 2

|                                               |  |
|-----------------------------------------------|--|
| sbr_extension(bs_extension_id, num_bits_left) |  |
| {                                             |  |
| switch (bs_extension_id) {                    |  |

|                                                     |           |
|-----------------------------------------------------|-----------|
| case EXTENSION_ID_PS:                               |           |
| num_bits_left -= ps_data();                         | Ghi chú 1 |
| break;                                              |           |
| case EXTENSION_ID_ESBR:                             |           |
| num_bits_left -= esbr_data();                       | Ghi chú 2 |
| break;                                              |           |
| default:                                            |           |
| bs_fill_bits;                                       |           |
| num_bits_left = 0;                                  |           |
| break;                                              |           |
| }                                                   |           |
| }                                                   |           |
| Ghi chú 1: ps_data() trả lại số lượng bit đã đọc.   |           |
| Ghi chú 2: esbr_data() trả lại số lượng bit đã đọc. |           |

Theo phương án làm ví dụ, esbr\_data() được nói đến trong Bảng 2 ở trên chỉ báo các giá trị của các tham số siêu dữ liệu sau:

1. tham số siêu dữ liệu một bit, “bs\_sbr\_preprocessing”; và
2. cho mỗi kênh (“ch”) của nội dung âm thanh của dòng bit mã hóa cần được giải mã, mỗi trong số các tham số nêu trên: “sbrPatchingMode[ch]”; “sbrOversamplingFlag[ch]”; “sbrPitchInBinsFlag[ch]”; và “sbrPitchInBins[ch]”.

Chẳng hạn, theo một số phương án, esbr\_data() có thể có cú pháp chỉ ra trong Bảng 3, để chỉ báo các tham số siêu dữ liệu này:

Bảng 3

| Cú pháp                        | Số lượng bit |
|--------------------------------|--------------|
| esbr_data(id_aac, bs_coupling) |              |
| {                              |              |
| bs_sbr_preprocessing;          | 1            |
| if (id_aac == ID_SCE) {        |              |

|                                  |   |
|----------------------------------|---|
| if (sbrPatchingMode[0] == 0) {   | 1 |
| sbrOversamplingFlag[0];          | 1 |
| if (sbrPitchInBinsFlag[0])       | 1 |
| sbrPitchInBins[0];               | 7 |
| else                             |   |
| sbrPitchInBins[0] = 0;           |   |
| } else {                         |   |
| sbrOversamplingFlag[0] = 0;      |   |
| sbrPitchInBins[0] = 0;           |   |
| }                                |   |
| } else if (id_aac == ID_CPE) {   |   |
| If (bs_coupling) {               |   |
| if (sbrPatchingMode[0,1] == 0) { | 1 |
| sbrOversamplingFlag[0,1];        | 1 |
| if (sbrPitchInBinsFlag[0,1])     | 1 |
| sbrPitchInBins[0,1];             | 7 |
| else                             |   |
| sbrPitchInBins[0,1] = 0;         |   |
| } else {                         |   |
| sbrOversamplingFlag[0,1] = 0;    |   |
| sbrPitchInBins[0,1] = 0;         |   |
| }                                |   |
| } else { /* bs_coupling == 0 */  |   |
| if (sbrPatchingMode[0] == 0) {   | 1 |
| sbrOversamplingFlag[0];          | 1 |
| if (sbrPitchInBinsFlag[0])       | 1 |
| sbrPitchInBins[0];               | 7 |
| else                             |   |
| sbrPitchInBins[0] = 0;           |   |
| } else {                         |   |
| sbrOversamplingFlag[0] = 0;      |   |

|                                                                                                                                                                                                                                                                                                                             |   |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
| <code>sbrPitchInBins[0] = 0;</code>                                                                                                                                                                                                                                                                                         |   |
| <code>}</code>                                                                                                                                                                                                                                                                                                              |   |
| <code>if (sbrPatchingMode[1] == 0) {</code>                                                                                                                                                                                                                                                                                 | 1 |
| <code>sbrOversamplingFlag[1];</code>                                                                                                                                                                                                                                                                                        | 1 |
| <code>if (sbrPitchInBinsFlag[1])</code>                                                                                                                                                                                                                                                                                     | 1 |
| <code>sbrPitchInBins[1];</code>                                                                                                                                                                                                                                                                                             | 7 |
| <code>else</code>                                                                                                                                                                                                                                                                                                           |   |
| <code>sbrPitchInBins[1] = 0;</code>                                                                                                                                                                                                                                                                                         |   |
| <code>} else {</code>                                                                                                                                                                                                                                                                                                       |   |
| <code>sbrOversamplingFlag[1] = 0;</code>                                                                                                                                                                                                                                                                                    |   |
| <code>sbrPitchInBins[1] = 0;</code>                                                                                                                                                                                                                                                                                         |   |
| <code>}</code>                                                                                                                                                                                                                                                                                                              |   |
| <code>}</code>                                                                                                                                                                                                                                                                                                              |   |
| <code>}</code>                                                                                                                                                                                                                                                                                                              |   |
| <code>}</code>                                                                                                                                                                                                                                                                                                              |   |
| Ghi chú: <i>bs_sbr_preprocessing</i> được định nghĩa như được mô tả trong<br>mục 6.2.12 theo ISO/IEC 23003-3:2012. <i>sbrPatchingMode[ch]</i> ,<br><i>sbrOversamplingFlag[ch]</i> , <i>sbrPitchInBinsFlag[ch]</i> và<br><i>sbrPitchInBins[ch]</i> được định nghĩa như được mô tả trong mục 7.5 của<br>ISO/IEC 23003-3:2012. |   |

Cú pháp trên cho phép việc thực hiện hiệu quả dạng nâng cao của phép tái tạo phổ tần số, như chuyển vị sóng hài, dưới dạng mở rộng cho bộ giải mã kế thừa. Cụ thể, dữ liệu eSBR của Bảng 3 chỉ bao gồm các tham số cần thiết để thực hiện dạng nâng cao của phép tái tạo phổ tần số mà không được hỗ trợ trong dòng bit hoặc trực tiếp bắt nguồn từ các tham số đã được hỗ trợ trong dòng bit. Tất cả các tham số và dữ liệu xử lý khác cần thiết để thực hiện dạng nâng cao của phép tái tạo phổ tần số được trích từ các tham số có ở các vị trí được xác định trước trong dòng bit.

Chẳng hạn, bộ giải mã theo chuẩn MPEG-4 HE-AAC hoặc HE-AAC v2 có thể được mở rộng để bao gồm dạng nâng cao của phép tái tạo phổ tần số, như chuyển vị sóng hài. Dạng nâng cao này của phép tái tạo phổ tần số, ngoài dạng cơ sở của phép tái tạo phổ tần

số, đã được hỗ trợ bởi bộ giải mã. Trong ngữ cảnh bộ giải mã theo chuẩn MPEG-4 HE-AAC hoặc HE-AAC v2, dạng cơ sở này của phép tái tạo phổ tần số là công cụ SBR và phổ QMF như định nghĩa trong Đoạn 4.6.18 của chuẩn MPEG-4 AAC.

Trong khi thực hiện dạng nâng cao của phép tái tạo phổ tần số, bộ giải mã HE-AAC mở rộng có thể tái sử dụng nhiều tham số dòng bit đã được bao gồm trong tải tin mở rộng SBR của dòng bit. Các tham số cụ thể có thể được tái sử dụng bao gồm, ví dụ, các tham số khác nhau xác định bảng băng tần số chủ. Các tham số này bao gồm `bs_start_freq` (tham số xác định sự bắt đầu của bảng tần số chủ), `bs_stop_freq` (tham số xác định kết thúc của bảng tần số chủ), `bs_freq_scale` (tham số xác định số lượng băng tần số trên octave), và `bs_alter_scale` (tham số biến đổi thang tỉ lệ của các băng tần số). Các tham số có thể được tái sử dụng còn bao gồm các tham số xác định bảng băng tần nhiễu (`bs_noise_bands`) và tham số bảng băng tần giới hạn (`bs_limiter_bands`). Do đó, theo các phương án khác nhau, ít nhất một số trong các tham số tương đương quy định trong chuẩn USAC được lược bỏ khỏi dòng bit, nhờ đó làm giảm chi phí điều khiển trong dòng bit. Thông thường, trường hợp tham số quy định trong chuẩn AAC có tham số tương đương quy định trong chuẩn USAC, tham số tương đương quy định trong chuẩn USAC này có cùng tên với tham số quy định trong chuẩn AAC, ví dụ hệ số định tỉ lệ đường bao `EOrigMapped`. Tuy nhiên, tham số tương đương được quy định trong chuẩn USAC thường có giá trị khác, được “điều chỉnh” cho quy trình xử lý SBR nâng cao được định nghĩa trong chuẩn USAC thay vì cho quy trình xử lý SBR được định nghĩa trong chuẩn AAC.

Để cải tiến chất lượng chủ quan cho nội dung âm thanh với cấu trúc tần số sóng hài và các đặc tính âm điệu mạnh, cụ thể tại các tốc độ bit thấp, sự kích hoạt SBR nâng cao được khuyến cáo. Các giá trị của phần tử dòng bit tương ứng (tức là `esbr_data()`), điều khiển các công cụ này, có thể được xác định trong bộ mã hóa bằng cách áp dụng cơ chế phân loại phụ thuộc vào tín hiệu. Nói chung, việc sử dụng phương pháp và sóng hài (`sbrPatchingMode == 1`) tốt hơn là để mã hóa các tín hiệu nhạc tại tốc độ bit rất thấp, trong đó bộ mã hóa-giải mã lỗi có thể xem xét giới hạn trong băng thông âm thanh. Điều này đặc biệt là đúng nếu các tín hiệu này bao gồm cấu trúc sóng hài rõ ràng. Ngược lại, việc sử dụng phương pháp và SBR thông thường được ưu tiên cho các tín hiệu tiếng nói và hỗn hợp, do nó cung cấp sự bảo toàn tốt hơn của cấu trúc thời gian trong tiếng nói.

Để cải thiện hiệu năng của bộ chuyển vị sóng hài, bước tiền xử lý có thể được kích hoạt (`bs_sbr_preprocessing == 1`) cố gắng để tránh việc đưa vào gián đoạn phổ trong tín hiệu đi vào bộ điều chỉnh đường bao tiếp theo. Hoạt động của công cụ này là có lợi cho các loại tín hiệu trong đó đường bao phổ thô của tín hiệu băng thấp được sử dụng để tái tạo tần số cao biểu thị sự thay đổi lớn theo mức độ.

Để cải thiện việc đáp ứng chuyển tiếp của kỹ thuật vá SBR sóng hài, có thể áp dụng kỹ thuật tăng mẫu miền tần số thích ứng tín hiệu (`sbrOversamplingFlag == 1`). Do việc tăng mẫu miền tần số thích ứng tín hiệu làm tăng mức độ phức tạp tính toán của bộ chuyển vị, nhưng chỉ mang lại các lợi ích cho các khung chứa chuyển tiếp, nên việc sử dụng của công cụ này được điều khiển bởi phần tử dòng bit, phần tử này được truyền một lần trên mỗi khung và mỗi kênh SBR độc lập.

Bộ giải mã hoạt động trong chế độ SBR nâng cao được đề xuất thường cần có khả năng chuyển đổi giữa sự kế thừa và việc vá SBR nâng cao. Do đó, độ trễ có thể được đưa vào có thể miễn là thời lượng của một khung âm thanh lõi, phụ thuộc vào sự cài đặt bộ giải mã. Thông thường, độ trễ cho cả sự kế thừa và việc vá SBR nâng cao sẽ là tương tự.

Ngoài các tham số này, các phần tử dữ liệu khác có thể cũng được tái sử dụng bởi bộ giải mã HE-AAC mở rộng khi thực hiện dạng nâng cao của phép tái tạo phổ tần số theo các phương án của sáng chế. Chẳng hạn, dữ liệu đường bao và dữ liệu nhiễu nền có thể được trích từ dữ liệu `bs_data_env` (hệ số tỉ lệ đường bao) và `bs_noise_env` (hệ số tỉ lệ nhiễu nền) và được sử dụng trong dạng nâng cao của phép tái tạo phổ tần số.

Về bản chất, các phương án này khai thác các tham số cấu hình và dữ liệu đường bao đã được hỗ trợ bởi bộ giải mã HE-AAC hoặc HE-AAC v2 kế thừa trong tải tin mở rộng SBR để kích hoạt dạng nâng cao của phép tái tạo phổ tần số đòi hỏi càng ít dữ liệu truyền bổ sung càng tốt. Siêu dữ liệu được điều chỉnh ban đầu cho dạng cơ sở của HFR (ví dụ, hoạt động dịch phổ của SBR), nhưng theo các phương án, được dùng cho dạng nâng cao của HFR (ví dụ, chuyển vị sóng hài của eSBR). Như đã được mô tả trên đây, siêu dữ liệu thường biểu diễn các tham số hoạt động (ví dụ, hệ số tỉ lệ đường bao, hệ số tỉ lệ nhiễu nền, tham số lưới thời gian/tần số, thông tin bổ sung hình sin, băng tần/tần số giao cắt biến đổi, chế độ lọc ngược, độ phân giải đường bao, chế độ làm mượt, chế độ nội suy tần số) được điều chỉnh và dự tính để được sử dụng với dạng cơ sở của HFR (ví dụ, dịch phổ tuyến tính). Tuy nhiên, siêu dữ liệu này, kết hợp với các tham số siêu dữ liệu bổ sung dành riêng

cho dạng nâng cao của HFR (ví dụ, chuyển vị sóng hài), có thể được sử dụng để xử lý một cách hiệu quả và có tác dụng dữ liệu âm thanh nhờ sử dụng dạng nâng cao của HFR.

Do đó, bộ giải mã mở rộng hỗ trợ dạng nâng cao của phép tái tạo phổ tần số có thể được tạo ra theo cách rất hiệu quả bằng cách nhờ vào phần tử dòng bit đã được định nghĩa (ví dụ, các phần tử trong tải tin mở rộng SBR) và chỉ bổ sung các tham số cần để hỗ trợ dạng nâng cao của phép tái tạo phổ tần số (trong tải tin mở rộng của phần tử làm đầy). Đặc điểm làm giảm dữ liệu này kết hợp với sự sắp xếp của các tham số thêm mới trong trường dữ liệu dự trữ, như phần chứa mở rộng, gần như làm giảm các trở ngại đối với việc tạo ra bộ giải mã hỗ trợ dạng nâng cao của phép tái tạo phổ tần số bằng cách bảo đảm rằng dòng bit tương thích ngược với bộ giải mã kế thừa không hỗ trợ dạng nâng cao của phép tái tạo phổ tần số. Sẽ nhận ra rằng trường dữ liệu dự trữ là trường dữ liệu tương thích ngược, nghĩa là nó là trường dữ liệu mà đã được hỗ trợ bởi các bộ giải mã trước đó, như các bộ giải mã HE-AAC hoặc HE-AAC v2 kế thừa. Tương tự, phần chứa mở rộng là tương thích ngược, nghĩa là nó là phần chứa mở rộng đã được hỗ trợ bởi các bộ giải mã trước đó, như các bộ giải mã HE-AAC hoặc HE-AAC v2 kế thừa.

Trong bảng 3, số trong cột phải chỉ số lượng của các bit của tham số tương ứng trong cột trái.

Theo một số phương án, loại đối tượng SBR được định nghĩa trong MPEG-4 AAC được cập nhật để chứa công cụ SBR và các khía cạnh của công cụ SBR nâng cao (eSBR) như được báo hiệu trong phần tử mở rộng SBR (`bs_extension_id`==`EXTENSION_ID_ESBR`). Nếu bộ giải mã phát hiện phần tử mở rộng SBR này, thì bộ giải mã sử dụng các khía cạnh được báo hiệu của công cụ SBR nâng cao.

Theo một số phương án, sáng chế đề xuất phương pháp bao gồm bước mã hóa dữ liệu âm thanh để tạo ra dòng bit mã hóa (ví dụ, dòng bit MPEG-4 AAC), bao gồm bằng cách bao gồm siêu dữ liệu eSBR trong ít nhất một đoạn của ít nhất một khối của dòng bit mã hóa và dữ liệu âm thanh trong ít nhất một đoạn khác của khối. Theo các phương án phổ biến, phương pháp bao gồm bước ghép kênh dữ liệu âm thanh có siêu dữ liệu eSBR trong mỗi khối của dòng bit mã hóa. Theo quá trình giải mã điển hình dòng bit được mã hóa ở bộ giải mã eSBR, bộ giải mã trích siêu dữ liệu eSBR từ dòng bit (bao gồm bằng cách phân tích và giải ghép kênh siêu dữ liệu eSBR và dữ liệu âm thanh) và sử dụng siêu dữ liệu eSBR để xử lý dữ liệu âm thanh để tạo ra dòng dữ liệu âm thanh giải mã.

Một khía cạnh khác của sáng chế là bộ giải mã eSBR được tạo cấu hình để thực hiện xử lý eSBR (ví dụ, nhờ sử dụng ít nhất một trong số các công cụ eSBR đã biết là chuyển vị sóng hài hoặc làm phẳng trước) trong quá trình giải mã của dòng bit âm thanh được mã hóa (ví dụ, dòng bit MPEG-4 AAC) không bao gồm siêu dữ liệu eSBR. Ví dụ về bộ giải mã này sẽ được mô tả dựa trên Fig.5.

Bộ giải mã eSBR (400) trên Fig.5 bao gồm bộ nhớ đệm 201 (giống với bộ nhớ 201 trên Fig.3 và Fig.4), bộ khử định dạng tải tin dòng bit 215 (giống với bộ khử định dạng 215 trên Fig.4), hệ thống phụ giải mã âm thanh 202 (đôi khi được gọi là tầng giải mã "lỗi" hoặc hệ thống phụ giải mã "lỗi", và giống với hệ thống phụ giải mã lỗi 202 trên Fig.3), hệ thống phụ tạo dữ liệu điều khiển eSBR 401, và tầng xử lý eSBR 203 (giống với tầng 203 trên Fig.3), được kết nối như được thể hiện trên hình vẽ. Thông thường, bộ giải mã 400 bao gồm các phần tử xử lý khác (không được thể hiện trên hình vẽ).

Trong quá trình hoạt động của bộ giải mã 400, chuỗi các khối của dòng bit âm thanh được mã hóa (dòng bit MPEG-4 AAC) nhận được bởi bộ giải mã 400 được xác nhận từ bộ đệm 201 đến bộ khử định dạng 215.

Bộ khử định dạng 215 được nối và được tạo cấu hình để giải ghép kênh mỗi khối của dòng bit để trích siêu dữ liệu SBR (bao gồm dữ liệu đường bao lượng tử hóa) và thường là còn có siêu dữ liệu khác từ đó. Bộ khử định dạng 215 được tạo cấu hình để xác nhận ít nhất siêu dữ liệu SBR đến tầng xử lý eSBR 203. Bộ khử định dạng 215 còn được nối và tạo cấu hình để trích dữ liệu âm thanh từ mỗi khối của dòng bit, và để xác nhận dữ liệu âm thanh trích được vào hệ thống phụ giải mã (tầng giải mã) 202.

Hệ thống phụ giải mã âm thanh 202 của bộ giải mã 400 được tạo cấu hình để mã hóa dữ liệu âm thanh trích bởi bộ khử định dạng 215 (bước giải mã này có thể được gọi là quá trình giải mã "lỗi") để tạo ra dữ liệu âm thanh giải mã, và để xác nhận dữ liệu âm thanh giải mã đến tầng xử lý eSBR 203. Việc giải mã được thực hiện ở miền tần số. Thông thường, tầng xử lý cuối trong hệ thống phụ 202 áp dụng biến đổi từ miền tần số thành miền thời gian cho dữ liệu âm thanh miền tần số được giải mã, do đó đầu ra của hệ thống phụ là dữ liệu âm thanh giải mã, miền thời gian. Tầng 203 được tạo cấu hình để áp dụng các công cụ SBR (và các công cụ eSBR) chỉ báo bởi siêu dữ liệu SBR (trích bởi bộ khử định dạng 215) và bởi siêu dữ liệu eSBR tạo ra trong hệ thống phụ 401, cho dữ liệu âm thanh giải mã (tức là, để thực hiện việc xử lý SBR và eSBR trên đầu ra của hệ thống phụ giải mã 202 nhờ

sử dụng siêu dữ liệu SBR và eSBR) để tạo ra dữ liệu âm thanh được giải mã đầy đủ được xuất ra từ bộ giải mã 400. Thông thường, bộ giải mã 400 bao gồm bộ nhớ (truy cập được bởi hệ thống phụ 202 và tầng 203) lưu trữ đầu ra dữ liệu âm thanh và siêu dữ liệu được khử định dạng từ bộ khử định dạng 215 (và tùy ý còn hệ thống phụ 401), và tầng 203 được tạo cấu hình để truy cập dữ liệu âm thanh và siêu dữ liệu khi cần trong quá trình xử lý SBR và eSBR. Việc xử lý SBR ở tầng 203 có thể được xem là bước xử lý hậu kỳ trên đầu ra của hệ thống phụ giải mã lõi 202. Tùy ý, bộ giải mã 400 còn bao gồm hệ thống phụ tăng mức cuối (có thể áp dụng các công cụ mã hóa âm thanh nổi có tham số ("PS") định nghĩa trong chuẩn MPEG-4 AAC, nhờ sử dụng siêu dữ liệu PS trích bởi bộ khử định dạng 215) được nối và tạo cấu hình để thực hiện phép tăng mức trên đầu ra của tầng 203 cho âm thanh tăng mức được giải mã đầy đủ được tạo ra mà được xuất ra từ APU 210.

Mã hóa âm thanh nổi có tham số là công cụ mã hóa biểu diễn tín hiệu âm thanh nổi bằng cách sử dụng giảm mức tuyến tính các kênh trái và phải trong tín hiệu âm thanh nổi và các tập hợp của các tham số không gian mô tả hình ảnh nổi. Mã hóa âm thanh nổi có tham số thường sử dụng ba loại tham số không gian: (1) chênh lệch cường độ liên kênh (inter-channel intensity difference - IID) mô tả các chênh lệch cường độ giữa các kênh; (2) chênh lệch pha liên kênh (inter-channel phase difference - IPD) mô tả chênh lệch pha giữa các kênh; và (3) sự nhất quán giữa các kênh (inter-channel coherence - ICC) mô tả sự nhất quán (hoặc sự tương tự) giữa các kênh. Sự nhất quán có thể được đo bằng giá trị cực đại của tương quan chéo là hàm số theo thời gian hoặc pha. Ba tham số này nói chung cho phép tái tạo chất lượng cao ảnh nổi. Tuy nhiên, các tham số IPD chỉ chỉ rõ các hiệu số pha tương đối giữa các kênh của tín hiệu đầu vào nổi và không chỉ báo sự phân bố của các hiệu số pha này trên các kênh trái và phải. Do đó, loại tham số thứ tư mô tả độ lệch pha toàn bộ hoặc hiệu số pha toàn bộ (OPD) có thể được sử dụng theo cách bổ sung. Trong quá trình tái tạo âm thanh nổi, các đoạn được lập cửa sổ liên tiếp của cả tín hiệu giảm mức thu được,  $s[n]$ , và phiên bản được khử tương quan của tín hiệu giảm mức thu được,  $d[n]$ , được xử lý cùng với các tham số không gian để tạo ra các tín hiệu được tái tạo trái ( $l_k(n)$ ) và phải ( $r_k(n)$ ) theo:

$$l_k(n) = H_{11}(k, n)s_k(n) + H_{21}(k, n)d_k(n)$$

$$r_k(n) = H_{12}(k, n)s_k(n) + H_{22}(k, n)d_k(n)$$

trong đó  $H_{11}$ ,  $H_{12}$ ,  $H_{21}$  và  $H_{22}$  được xác định bởi các tham số tái tạo âm thanh nổi. Cuối cùng, các tín hiệu  $l_k(n)$  và  $r_k(n)$  được biến đổi trở lại miền thời gian bằng phương pháp biến đổi tần số thành thời gian.

Hệ thống phụ tạo dữ liệu điều khiển 401 trên Fig.5 được nối và tạo cấu hình để phát hiện ít nhất một đặc tính của dòng bit âm thanh được mã hóa cần được giải mã, và để tạo ra dữ liệu điều khiển eSBR (mà có thể là hoặc bao gồm siêu dữ liệu eSBR thuộc loại bất kỳ trong số các loại được bao gồm trong dòng bit âm thanh được mã hóa theo các phương án khác của sáng chế) đáp lại ít nhất một kết quả của bước phát hiện. Dữ liệu điều khiển eSBR được xác nhận đến tầng 203 để kích hoạt việc áp dụng các công cụ eSBR riêng hoặc tổ hợp của các công cụ eSBR khi phát hiện thấy một đặc tính cụ thể (hoặc tổ hợp các đặc tính) của dòng bit, và/hoặc để kiểm soát việc áp dụng các công cụ eSBR này. Chẳng hạn, để kiểm soát hiệu suất của xử lý eSBR nhờ sử dụng phép chuyển vị sóng hài, một số phương án của hệ thống phụ tạo dữ liệu điều khiển 401 sẽ bao gồm: bộ phát hiện nhạc (ví dụ, phiên bản đơn giản hóa của bộ phát hiện nhạc thông thường) để thiết lập tham số `sbrPatchingMode[ch]` (và xác nhận tham số thiết lập này đến tầng 203) đáp lại việc phát hiện rằng dòng bit là hoặc không chỉ báo nhạc; bộ phát hiện chuyển tiếp để thiết lập tham số `sbrOversamplingFlag[ch]` (và xác nhận tham số thiết lập này đến tầng 203) đáp lại việc phát hiện sự có mặt hoặc vắng mặt của sự chuyển tiếp ở nội dung âm thanh được chỉ báo bởi dòng bit; và/hoặc bộ phát hiện độ cao âm thanh để thiết lập tham số `sbrPitchInBinsFlag[ch]` và `sbrPitchInBins[ch]` (và xác nhận các tham số thiết lập này đến tầng 203) đáp lại việc phát hiện độ cao âm thanh của nội dung âm thanh được chỉ báo bởi dòng bit. Các khía cạnh khác của sáng chế là các phương pháp giải mã dòng bit âm thanh thực hiện bởi phương án bất kỳ của bộ giải mã theo sáng chế được mô tả trong đoạn này và các đoạn trước.

Các khía cạnh của sáng chế bao gồm phương pháp mã hóa hoặc giải mã thuộc loại mà phương án bất kỳ của APU, hệ thống hoặc thiết bị theo sáng chế được tạo cấu hình (ví dụ, được lập trình) để thực hiện. Các khía cạnh khác của sáng chế bao gồm hệ thống hoặc thiết bị được tạo cấu hình (ví dụ, được lập trình) để thực hiện phương án bất kỳ của phương pháp theo sáng chế, và phương tiện đọc được bằng máy tính (ví dụ, đĩa) lưu trữ mã (ví dụ, theo cách bất biến) 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 đa năng

lập trình được, bộ xử lý tín hiệu số, hoặc bộ vi xử lý, được lập trình với phần mềm hoặc firmware và/hoặc theo cách khác được tạo cấu hình để thực hiện hoạt động bất kỳ trong số các hoạt động khác nhau trên dữ liệu, bao gồm phương pháp hoặc các bước của nó theo một phương án của sáng chế. Bộ xử lý đa năng như vậy có thể là hoặc bao gồm hệ thống máy tính gồm thiết bị đầu vào, bộ nhớ, và hệ mạch xử lý được lập trình (và/hoặc theo cách khác được tạo cấu hình) để thực hiện phương pháp (hoặc các bước của nó) theo một phương án của sáng chế để đáp lại dữ liệu được xác nhận vào đó.

Các phương án theo sáng chế có thể được thực hiện trong phần cứng, firmware, hoặc phần mềm, hoặc kết hợp cả hai (ví dụ, như mảng logic lập trình được). Trừ phi được quy định khác, các thuật toán hoặc quy trình được bao gồm như một phần của sáng chế bản thân chúng vốn không liên quan đến máy tính cụ thể hoặc thiết bị khác bất kỳ. Cụ thể, nhiều máy đa năng khác nhau có thể được sử dụng với các chương trình được viết theo nội dung mô tả 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 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 phần tử của nó), hoặc bộ giải mã 200 trên Fig.3 (hoặc phần tử của nó), hoặc bộ giải mã 210 trên Fig.4 (hoặc phần tử của nó) hoặc bộ giải mã 400 trên Fig.5 (hoặc 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 trong bản mô tả này và tạo thông tin đầu ra. Thông tin đầu ra được áp dụng cho một hoặc nhiều thiết bị đầu ra, theo cách đã biết.

Mỗi chương trình như vậy có thể được thực thi theo ngôn ngữ máy tính mong muốn bất kỳ (bao gồm ngôn ngữ lập trình cho máy móc, dây chuyền, hoặc thủ tục, logic hoặc ngôn ngữ lập trình hướng đối tượng bậc cao) để giao tiếp với hệ thống máy tính. Trong bất kỳ trường hợp nào, ngôn ngữ này có thể là ngôn ngữ biên dịch hoặc diễn dịch.

Ví dụ, khi được triển khai bởi các chuỗi 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 chuỗi 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 như vậy 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) đọc được bằng máy tính lập trình được đa năng hoặc chuyên dụng, để tạo cấu hình và vận hành máy tính khi phương tiện hoặc thiết bị lưu trữ được đọc bởi hệ thống máy tính để thực hiện các thủ tục được mô tả ở đây. Hệ thống theo sáng chế cũng có thể được thực thi dưới dạng phương tiện lưu trữ đọc được bởi máy tính, được cấu hình với (tức là, lưu trữ) chương trình máy tính, ở đó phương tiện lưu trữ được cấu hình như vậy khiến hệ thống máy tính vận hành theo cách thức cụ thể và xác định trước để thực hiện các chức năng được mô tả trong bản mô tả này.

Một số phương án theo sáng chế đã được mô tả. Tuy nhiên, cần hiểu rằng các cải biến khác nhau có thể được thực hiện mà không nằm ngoài ý tưởng và phạm vi của sáng chế. Nhiều cải biến và biến thể của sáng chế có thể thực hiện nằm trong phạm vi các nội dung mô tả ở trên. Chẳng hạn, để tạo thuận tiện cho các phương án thực hiện hiệu quả, các độ dịch pha có thể được sử dụng kết hợp với các giàn bộ lọc phân tích và tổng hợp QMF phức hợp. Giàn bộ lọc phân tích chịu trách nhiệm lọc tín hiệu băng thấp miền thời gian tạo ra bởi bộ giải mã lỗi thành nhiều băng con (ví dụ, băng con QMF). Giàn bộ lọc tổng hợp chịu trách nhiệm kết hợp băng cao được tái tạo tạo ra bởi kỹ thuật HFR được chọn (như được chỉ báo bởi tham số `sbrPatchingMode` nhận được) có băng thấp được giải mã để tạo ra tín hiệu âm thanh đầu ra băng rộng. Tuy nhiên, phương án thực hiện của giàn lọc cho trước vận hành theo chế độ tốc độ lấy mẫu nhất định, ví dụ, hoạt động tốc độ kép bình thường hoặc chế độ SBR giảm mẫu, tuy nhiên, không nên có độ dịch pha phụ thuộc vào dòng bit. Các giàn QMF được sử dụng trong SBR là dạng mở rộng hàm số mũ phức của nguyên lý của các giàn lọc điều chế cosin. Có thể thấy rằng các ràng buộc triệt tiêu hiện tượng chồng phổ trở nên lỗi thời khi mở rộng giàn lọc điều chế cosin với phép điều chế hàm mũ phức. Vì vậy, đối với các giàn QMF SBR, cả bộ lọc phân tích,  $h_k(n)$ , và bộ lọc tổng hợp,  $f_k(n)$ , có thể được xác định bởi:

$$h_k(n) = f_k(n) = p_0(n) \exp \left\{ i \frac{\pi}{M} \left( k + \frac{1}{2} \right) \left( n - \frac{N}{2} \right) \right\}, \quad 0 \leq n \leq N; 0 \leq k < M$$

(1)

trong đó  $p_0(n)$  là bộ lọc nguyên mẫu không đối xứng hoặc đối xứng giá trị thực (thường là, bộ lọc nguyên mẫu thông thấp),  $M$  chỉ số lượng kênh và  $N$  là thứ tự bộ lọc nguyên mẫu. Số lượng của kênh sử dụng trong giàn bộ lọc phân tích có thể khác với số lượng kênh được sử dụng trong giàn bộ lọc tổng hợp. Chẳng hạn, giàn bộ lọc phân tích có thể có 32 kênh và giàn bộ lọc tổng hợp có thể có 64 kênh. Khi vận hành giàn bộ lọc tổng hợp ở chế độ giảm mẫu, giàn bộ lọc tổng hợp có thể chỉ có 32 kênh. Do các mẫu băng con từ giàn lọc có giá trị phức, bước dịch pha phụ thuộc kênh bổ sung có thể có thể được thêm vào giàn bộ lọc phân tích. Các độ dịch pha bổ sung này cần được bù trước giàn bộ lọc tổng hợp. Mặc dù thuật ngữ dịch pha về nguyên tắc có thể có giá trị tùy ý mà không phá hỏng hoạt động của chuỗi QMF phân tích/tổng hợp, nhưng chúng có thể cũng bị ràng buộc ở các giá trị nhất định để kiểm chứng tương thích. Tín hiệu SBR sẽ bị ảnh hưởng bởi việc lựa chọn các yếu tố pha trong khi tín hiệu thông thấp đến từ bộ giải mã lỗi sẽ không bị ảnh hưởng. Chất lượng âm thanh của tín hiệu đầu ra không bị ảnh hưởng.

Các hệ số của bộ lọc nguyên mẫu,  $p_0(n)$ , có thể có thể được xác định với độ dài,  $L$ , là 640, như thể hiện trong bảng 4 dưới đây.

Bảng 4

| $n$ | $p_0(n)$      | $n$ | $p_0(n)$      | $n$ | $p_0(n)$     |
|-----|---------------|-----|---------------|-----|--------------|
| 0   | 0,0000000000  | 214 | 0,0019765601  | 428 | 0,0117623832 |
| 1   | -0,0005525286 | 215 | -0,0032086896 | 429 | 0,0163701258 |
| 2   | -0,0005617692 | 216 | -0,0085711749 | 430 | 0,0207997072 |
| 3   | -0,0004947518 | 217 | -0,0141288827 | 431 | 0,0250307561 |
| 4   | -0,0004875227 | 218 | -0,0198834129 | 432 | 0,0290824006 |
| 5   | -0,0004893791 | 219 | -0,0258227288 | 433 | 0,0329583930 |
| 6   | -0,0005040714 | 220 | -0,0319531274 | 434 | 0,0366418116 |
| 7   | -0,0005226564 | 221 | -0,0382776572 | 435 | 0,0401458278 |
| 8   | -0,0005466565 | 222 | -0,0447806821 | 436 | 0,0434768782 |
| 9   | -0,0005677802 | 223 | -0,0514804176 | 437 | 0,0466303305 |
| 10  | -0,0005870930 | 224 | -0,0583705326 | 438 | 0,0495978676 |
| 11  | -0,0006132747 | 225 | -0,0654409853 | 439 | 0,0524093821 |

|    |               |     |               |     |              |
|----|---------------|-----|---------------|-----|--------------|
| 12 | -0,0006312493 | 226 | -0,0726943300 | 440 | 0,0550460034 |
| 13 | -0,0006540333 | 227 | -0,0801372934 | 441 | 0,0575152691 |
| 14 | -0,0006777690 | 228 | -0,0877547536 | 442 | 0,0598166570 |
| 15 | -0,0006941614 | 229 | -0,0955533352 | 443 | 0,0619602779 |
| 16 | -0,0007157736 | 230 | -0,1035329531 | 444 | 0,0639444805 |
| 17 | -0,0007255043 | 231 | -0,1116826931 | 445 | 0,0657690668 |
| 18 | -0,0007440941 | 232 | -0,1200077984 | 446 | 0,0674525021 |
| 19 | -0,0007490598 | 233 | -0,1285002850 | 447 | 0,0689664013 |
| 20 | -0,0007681371 | 234 | -0,1371551761 | 448 | 0,0703533073 |
| 21 | -0,0007724848 | 235 | -0,1459766491 | 449 | 0,0715826364 |
| 22 | -0,0007834332 | 236 | -0,1549607071 | 450 | 0,0726774642 |
| 23 | -0,0007779869 | 237 | -0,1640958855 | 451 | 0,0736406005 |
| 24 | -0,0007803664 | 238 | -0,1733808172 | 452 | 0,0744664394 |
| 25 | -0,0007801449 | 239 | -0,1828172548 | 453 | 0,0751576255 |
| 26 | -0,0007757977 | 240 | -0,1923966745 | 454 | 0,0757305756 |
| 27 | -0,0007630793 | 241 | -0,2021250176 | 455 | 0,0761748321 |
| 28 | -0,0007530001 | 242 | -0,2119735853 | 456 | 0,0765050718 |
| 29 | -0,0007319357 | 243 | -0,2219652696 | 457 | 0,0767204924 |
| 30 | -0,0007215391 | 244 | -0,2320690870 | 458 | 0,0768230011 |
| 31 | -0,0006917937 | 245 | -0,2423016884 | 459 | 0,0768173975 |
| 32 | -0,0006650415 | 246 | -0,2526480309 | 460 | 0,0767093490 |
| 33 | -0,0006341594 | 247 | -0,2631053299 | 461 | 0,0764992170 |
| 34 | -0,0005946118 | 248 | -0,2736634040 | 462 | 0,0761992479 |
| 35 | -0,0005564576 | 249 | -0,2843214189 | 463 | 0,0758008358 |
| 36 | -0,0005145572 | 250 | -0,2950716717 | 464 | 0,0753137336 |
| 37 | -0,0004606325 | 251 | -0,3059098575 | 465 | 0,0747452558 |
| 38 | -0,0004095121 | 252 | -0,3168278913 | 466 | 0,0741003642 |
| 39 | -0,0003501175 | 253 | -0,3278113727 | 467 | 0,0733620255 |
| 40 | -0,0002896981 | 254 | -0,3388722693 | 468 | 0,0725682583 |
| 41 | -0,0002098337 | 255 | -0,3499914122 | 469 | 0,0717002673 |
| 42 | -0,0001446380 | 256 | 0,3611589903  | 470 | 0,0707628710 |

|    |               |     |              |     |              |
|----|---------------|-----|--------------|-----|--------------|
| 43 | -0,0000617334 | 257 | 0,3723795546 | 471 | 0,0697630244 |
| 44 | 0,0000134949  | 258 | 0,3836350013 | 472 | 0,0687043828 |
| 45 | 0,0001094383  | 259 | 0,3949211761 | 473 | 0,0676075985 |
| 46 | 0,0002043017  | 260 | 0,4062317676 | 474 | 0,0664367512 |
| 47 | 0,0002949531  | 261 | 0,4175696896 | 475 | 0,0652247106 |
| 48 | 0,0004026540  | 262 | 0,4289119920 | 476 | 0,0639715898 |
| 49 | 0,0005107388  | 263 | 0,4402553754 | 477 | 0,0626857808 |
| 50 | 0,0006239376  | 264 | 0,4515996535 | 478 | 0,0613455171 |
| 51 | 0,0007458025  | 265 | 0,4629308085 | 479 | 0,0599837480 |
| 52 | 0,0008608443  | 266 | 0,4742453214 | 480 | 0,0585915683 |
| 53 | 0,0009885988  | 267 | 0,4855253091 | 481 | 0,0571616450 |
| 54 | 0,0011250155  | 268 | 0,4967708254 | 482 | 0,0557173648 |
| 55 | 0,0012577884  | 269 | 0,5079817500 | 483 | 0,0542452768 |
| 56 | 0,0013902494  | 270 | 0,5191234970 | 484 | 0,0527630746 |
| 57 | 0,0015443219  | 271 | 0,5302240895 | 485 | 0,0512556155 |
| 58 | 0,0016868083  | 272 | 0,5412553448 | 486 | 0,0497385755 |
| 59 | 0,0018348265  | 273 | 0,5522051258 | 487 | 0,0482165720 |
| 60 | 0,0019841140  | 274 | 0,5630789140 | 488 | 0,0466843027 |
| 61 | 0,0021461583  | 275 | 0,5738524131 | 489 | 0,0451488405 |
| 62 | 0,0023017254  | 276 | 0,5845403235 | 490 | 0,0436097542 |
| 63 | 0,0024625616  | 277 | 0,5951123086 | 491 | 0,0420649094 |
| 64 | 0,0026201758  | 278 | 0,6055783538 | 492 | 0,0405349170 |
| 65 | 0,0027870464  | 279 | 0,6159109932 | 493 | 0,0390053679 |
| 66 | 0,0029469447  | 280 | 0,6261242695 | 494 | 0,0374812850 |
| 67 | 0,0031125420  | 281 | 0,6361980107 | 495 | 0,0359697560 |
| 68 | 0,0032739613  | 282 | 0,6461269695 | 496 | 0,0344620948 |
| 69 | 0,0034418874  | 283 | 0,6559016302 | 497 | 0,0329754081 |
| 70 | 0,0036008268  | 284 | 0,6655139880 | 498 | 0,0315017608 |
| 71 | 0,0037603922  | 285 | 0,6749663190 | 499 | 0,0300502657 |
| 72 | 0,0039207432  | 286 | 0,6842353293 | 500 | 0,0286072173 |
| 73 | 0,0040819753  | 287 | 0,6933282376 | 501 | 0,0271859429 |

|     |              |     |              |     |               |
|-----|--------------|-----|--------------|-----|---------------|
| 74  | 0,0042264269 | 288 | 0,7022388719 | 502 | 0,0257875847  |
| 75  | 0,0043730719 | 289 | 0,7109410426 | 503 | 0,0244160992  |
| 76  | 0,0045209852 | 290 | 0,7194462634 | 504 | 0,0230680169  |
| 77  | 0,0046606460 | 291 | 0,7277448900 | 505 | 0,0217467550  |
| 78  | 0,0047932560 | 292 | 0,7358211758 | 506 | 0,0204531793  |
| 79  | 0,0049137603 | 293 | 0,7436827863 | 507 | 0,0191872431  |
| 80  | 0,0050393022 | 294 | 0,7513137456 | 508 | 0,0179433381  |
| 81  | 0,0051407353 | 295 | 0,7587080760 | 509 | 0,0167324712  |
| 82  | 0,0052461166 | 296 | 0,7658674865 | 510 | 0,0155405553  |
| 83  | 0,0053471681 | 297 | 0,7727780881 | 511 | 0,0143904666  |
| 84  | 0,0054196775 | 298 | 0,7794287519 | 512 | -0,0132718220 |
| 85  | 0,0054876040 | 299 | 0,7858353120 | 513 | -0,0121849995 |
| 86  | 0,0055475714 | 300 | 0,7919735841 | 514 | -0,0111315548 |
| 87  | 0,0055938023 | 301 | 0,7978466413 | 515 | -0,0101150215 |
| 88  | 0,0056220643 | 302 | 0,8034485751 | 516 | -0,0091325329 |
| 89  | 0,0056455196 | 303 | 0,8087695004 | 517 | -0,0081798233 |
| 90  | 0,0056389199 | 304 | 0,8138191270 | 518 | -0,0072615816 |
| 91  | 0,0056266114 | 305 | 0,8185776004 | 519 | -0,0063792293 |
| 92  | 0,0055917128 | 306 | 0,8230419890 | 520 | -0,0055337211 |
| 93  | 0,0055404363 | 307 | 0,8272275347 | 521 | -0,0047222596 |
| 94  | 0,0054753783 | 308 | 0,8311038457 | 522 | -0,0039401124 |
| 95  | 0,0053838975 | 309 | 0,8346937361 | 523 | -0,0031933778 |
| 96  | 0,0052715758 | 310 | 0,8379717337 | 524 | -0,0024826723 |
| 97  | 0,0051382275 | 311 | 0,8409541392 | 525 | -0,0018039472 |
| 98  | 0,0049839687 | 312 | 0,8436238281 | 526 | -0,0011568135 |
| 99  | 0,0048109469 | 313 | 0,8459818469 | 527 | -0,0005464280 |
| 100 | 0,0046039530 | 314 | 0,8480315777 | 528 | 0,0000276045  |
| 101 | 0,0043801861 | 315 | 0,8497805198 | 529 | 0,0005832264  |
| 102 | 0,0041251642 | 316 | 0,8511971524 | 530 | 0,0010902329  |
| 103 | 0,0038456408 | 317 | 0,8523047035 | 531 | 0,0015784682  |
| 104 | 0,0035401246 | 318 | 0,8531020949 | 532 | 0,0020274176  |

|     |                |     |              |     |              |
|-----|----------------|-----|--------------|-----|--------------|
| 105 | 0,0032091885   | 319 | 0,8535720573 | 533 | 0,0024508540 |
| 106 | 0,0028446757   | 320 | 0,8537385600 | 534 | 0,0028446757 |
| 107 | 0,0024508540   | 321 | 0,8535720573 | 535 | 0,0032091885 |
| 108 | 0,0020274176   | 322 | 0,8531020949 | 536 | 0,0035401246 |
| 109 | 0,0015784682   | 323 | 0,8523047035 | 537 | 0,0038456408 |
| 110 | 0,0010902329   | 324 | 0,8511971524 | 538 | 0,0041251642 |
| 111 | 0,0005832264   | 325 | 0,8497805198 | 539 | 0,0043801861 |
| 112 | 0,0000276045   | 326 | 0,8480315777 | 540 | 0,0046039530 |
| 113 | -0,0005464280  | 327 | 0,8459818469 | 541 | 0,0048109469 |
| 114 | -0,0011568135  | 328 | 0,8436238281 | 542 | 0,0049839687 |
| 115 | -0,0018039472  | 329 | 0,8409541392 | 543 | 0,0051382275 |
| 116 | -0,0024826723  | 330 | 0,8379717337 | 544 | 0,0052715758 |
| 117 | -0,0031933778  | 331 | 0,8346937361 | 545 | 0,0053838975 |
| 118 | -0,0039401124  | 332 | 0,8311038457 | 546 | 0,0054753783 |
| 119 | -0,0047222596  | 333 | 0,8272275347 | 547 | 0,0055404363 |
| 120 | -0,0055337211  | 334 | 0,8230419890 | 548 | 0,0055917128 |
| 121 | -0,0063792293  | 335 | 0,8185776004 | 549 | 0,0056266114 |
| 122 | -0,0072615816  | 336 | 0,8138191270 | 550 | 0,0056389199 |
| 123 | -0,0081798233  | 337 | 0,8087695004 | 551 | 0,0056455196 |
| 124 | -0,0091325329  | 338 | 0,8034485751 | 552 | 0,0056220643 |
| 125 | -0,0101150215  | 339 | 0,7978466413 | 553 | 0,0055938023 |
| 126 | -0,01111315548 | 340 | 0,7919735841 | 554 | 0,0055475714 |
| 127 | -0,0121849995  | 341 | 0,7858353120 | 555 | 0,0054876040 |
| 128 | 0,0132718220   | 342 | 0,7794287519 | 556 | 0,0054196775 |
| 129 | 0,0143904666   | 343 | 0,7727780881 | 557 | 0,0053471681 |
| 130 | 0,0155405553   | 344 | 0,7658674865 | 558 | 0,0052461166 |
| 131 | 0,0167324712   | 345 | 0,7587080760 | 559 | 0,0051407353 |
| 132 | 0,0179433381   | 346 | 0,7513137456 | 560 | 0,0050393022 |
| 133 | 0,0191872431   | 347 | 0,7436827863 | 561 | 0,0049137603 |
| 134 | 0,0204531793   | 348 | 0,7358211758 | 562 | 0,0047932560 |
| 135 | 0,0217467550   | 349 | 0,7277448900 | 563 | 0,0046606460 |

|     |              |     |              |     |              |
|-----|--------------|-----|--------------|-----|--------------|
| 136 | 0,0230680169 | 350 | 0,7194462634 | 564 | 0,0045209852 |
| 137 | 0,0244160992 | 351 | 0,7109410426 | 565 | 0,0043730719 |
| 138 | 0,0257875847 | 352 | 0,7022388719 | 566 | 0,0042264269 |
| 139 | 0,0271859429 | 353 | 0,6933282376 | 567 | 0,0040819753 |
| 140 | 0,0286072173 | 354 | 0,6842353293 | 568 | 0,0039207432 |
| 141 | 0,0300502657 | 355 | 0,6749663190 | 569 | 0,0037603922 |
| 142 | 0,0315017608 | 356 | 0,6655139880 | 570 | 0,0036008268 |
| 143 | 0,0329754081 | 357 | 0,6559016302 | 571 | 0,0034418874 |
| 144 | 0,0344620948 | 358 | 0,6461269695 | 572 | 0,0032739613 |
| 145 | 0,0359697560 | 359 | 0,6361980107 | 573 | 0,0031125420 |
| 146 | 0,0374812850 | 360 | 0,6261242695 | 574 | 0,0029469447 |
| 147 | 0,0390053679 | 361 | 0,6159109932 | 575 | 0,0027870464 |
| 148 | 0,0405349170 | 362 | 0,6055783538 | 576 | 0,0026201758 |
| 149 | 0,0420649094 | 363 | 0,5951123086 | 577 | 0,0024625616 |
| 150 | 0,0436097542 | 364 | 0,5845403235 | 578 | 0,0023017254 |
| 151 | 0,0451488405 | 365 | 0,5738524131 | 579 | 0,0021461583 |
| 152 | 0,0466843027 | 366 | 0,5630789140 | 580 | 0,0019841140 |
| 153 | 0,0482165720 | 367 | 0,5522051258 | 581 | 0,0018348265 |
| 154 | 0,0497385755 | 368 | 0,5412553448 | 582 | 0,0016868083 |
| 155 | 0,0512556155 | 369 | 0,5302240895 | 583 | 0,0015443219 |
| 156 | 0,0527630746 | 370 | 0,5191234970 | 584 | 0,0013902494 |
| 157 | 0,0542452768 | 371 | 0,5079817500 | 585 | 0,0012577884 |
| 158 | 0,0557173648 | 372 | 0,4967708254 | 586 | 0,0011250155 |
| 159 | 0,0571616450 | 373 | 0,4855253091 | 587 | 0,0009885988 |
| 160 | 0,0585915683 | 374 | 0,4742453214 | 588 | 0,0008608443 |
| 161 | 0,0599837480 | 375 | 0,4629308085 | 589 | 0,0007458025 |
| 162 | 0,0613455171 | 376 | 0,4515996535 | 590 | 0,0006239376 |
| 163 | 0,0626857808 | 377 | 0,4402553754 | 591 | 0,0005107388 |
| 164 | 0,0639715898 | 378 | 0,4289119920 | 592 | 0,0004026540 |
| 165 | 0,0652247106 | 379 | 0,4175696896 | 593 | 0,0002949531 |
| 166 | 0,0664367512 | 380 | 0,4062317676 | 594 | 0,0002043017 |

|     |              |     |               |     |               |
|-----|--------------|-----|---------------|-----|---------------|
| 167 | 0,0676075985 | 381 | 0,3949211761  | 595 | 0,0001094383  |
| 168 | 0,0687043828 | 382 | 0,3836350013  | 596 | 0,0000134949  |
| 169 | 0,0697630244 | 383 | 0,3723795546  | 597 | -0,0000617334 |
| 170 | 0,0707628710 | 384 | -0,3611589903 | 598 | -0,0001446380 |
| 171 | 0,0717002673 | 385 | -0,3499914122 | 599 | -0,0002098337 |
| 172 | 0,0725682583 | 386 | -0,3388722693 | 600 | -0,0002896981 |
| 173 | 0,0733620255 | 387 | -0,3278113727 | 601 | -0,0003501175 |
| 174 | 0,0741003642 | 388 | -0,3168278913 | 602 | -0,0004095121 |
| 175 | 0,0747452558 | 389 | -0,3059098575 | 603 | -0,0004606325 |
| 176 | 0,0753137336 | 390 | -0,2950716717 | 604 | -0,0005145572 |
| 177 | 0,0758008358 | 391 | -0,2843214189 | 605 | -0,0005564576 |
| 178 | 0,0761992479 | 392 | -0,2736634040 | 606 | -0,0005946118 |
| 179 | 0,0764992170 | 393 | -0,2631053299 | 607 | -0,0006341594 |
| 180 | 0,0767093490 | 394 | -0,2526480309 | 608 | -0,0006650415 |
| 181 | 0,0768173975 | 395 | -0,2423016884 | 609 | -0,0006917937 |
| 182 | 0,0768230011 | 396 | -0,2320690870 | 610 | -0,0007215391 |
| 183 | 0,0767204924 | 397 | -0,2219652696 | 611 | -0,0007319357 |
| 184 | 0,0765050718 | 398 | -0,2119735853 | 612 | -0,0007530001 |
| 185 | 0,0761748321 | 399 | -0,2021250176 | 613 | -0,0007630793 |
| 186 | 0,0757305756 | 400 | -0,1923966745 | 614 | -0,0007757977 |
| 187 | 0,0751576255 | 401 | -0,1828172548 | 615 | -0,0007801449 |
| 188 | 0,0744664394 | 402 | -0,1733808172 | 616 | -0,0007803664 |
| 189 | 0,0736406005 | 403 | -0,1640958855 | 617 | -0,0007779869 |
| 190 | 0,0726774642 | 404 | -0,1549607071 | 618 | -0,0007834332 |
| 191 | 0,0715826364 | 405 | -0,1459766491 | 619 | -0,0007724848 |
| 192 | 0,0703533073 | 406 | -0,1371551761 | 620 | -0,0007681371 |
| 193 | 0,0689664013 | 407 | -0,1285002850 | 621 | -0,0007490598 |
| 194 | 0,0674525021 | 408 | -0,1200077984 | 622 | -0,0007440941 |
| 195 | 0,0657690668 | 409 | -0,1116826931 | 623 | -0,0007255043 |
| 196 | 0,0639444805 | 410 | -0,1035329531 | 624 | -0,0007157736 |
| 197 | 0,0619602779 | 411 | -0,0955533352 | 625 | -0,0006941614 |

|     |              |     |               |     |               |
|-----|--------------|-----|---------------|-----|---------------|
| 198 | 0,0598166570 | 412 | -0,0877547536 | 626 | -0,0006777690 |
| 199 | 0,0575152691 | 413 | -0,0801372934 | 627 | -0,0006540333 |
| 200 | 0,0550460034 | 414 | -0,0726943300 | 628 | -0,0006312493 |
| 201 | 0,0524093821 | 415 | -0,0654409853 | 629 | -0,0006132747 |
| 202 | 0,0495978676 | 416 | -0,0583705326 | 630 | -0,0005870930 |
| 203 | 0,0466303305 | 417 | -0,0514804176 | 631 | -0,0005677802 |
| 204 | 0,0434768782 | 418 | -0,0447806821 | 632 | -0,0005466565 |
| 205 | 0,0401458278 | 419 | -0,0382776572 | 633 | -0,0005226564 |
| 206 | 0,0366418116 | 420 | -0,0319531274 | 634 | -0,0005040714 |
| 207 | 0,0329583930 | 421 | -0,0258227288 | 635 | -0,0004893791 |
| 208 | 0,0290824006 | 422 | -0,0198834129 | 636 | -0,0004875227 |
| 209 | 0,0250307561 | 423 | -0,0141288827 | 637 | -0,0004947518 |
| 210 | 0,0207997072 | 424 | -0,0085711749 | 638 | -0,0005617692 |
| 211 | 0,0163701258 | 425 | -0,0032086896 | 639 | -0,0005525280 |
| 212 | 0,0117623832 | 426 | 0,0019765601  |     |               |
| 213 | 0,0069636862 | 427 | 0,0069636862  |     |               |

Bộ lọc nguyên mẫu,  $p_0(n)$ , có thể cũng được suy ra từ Bảng 4 bởi một hoặc nhiều phép toán như làm tròn, lấy mẫu phụ, nội suy, và lấy thập phân.

Mặc dù việc điều chỉnh thông tin điều khiển liên quan tới SBR thường không phụ thuộc vào chi tiết của quá trình chuyển tiếp (như được mô tả trước đó), nhưng trong một số các phương án các phần tử nhất định của dữ liệu điều khiển có thể được truyền đồng thời trong phân chứa mở rộng eSBR (`bs_extension_id == EXTENSION_ID_ESBR`) để cải thiện chất lượng của tín hiệu được tái tạo. Một số phần tử được truyền đồng thời có thể bao gồm dữ liệu nhiều nền (ví dụ, hệ số tỉ lệ nhiều nền và tham số chỉ báo hướng, trong hướng tần số hoặc thời gian, của mã hóa delta cho mỗi nhiều nền), dữ liệu lọc ngược (ví dụ, tham số chỉ báo chế độ lọc ngược được chọn từ không có lọc ngược, mức lọc ngược thấp, mức lọc ngược trung bình, và mức lọc ngược mạnh), và dữ liệu sóng hài bị thiếu (ví dụ, tham số chỉ báo hình sin có nên được bổ sung vào băng tần số cụ thể trong băng cao được tái tạo hay không). Tất cả các phần tử này dựa vào sự mô phỏng tổng hợp trong bộ chuyển vị của bộ giải mã được thực hiện trong bộ mã hóa và do đó nếu được điều chỉnh

chính xác cho bộ chuyển vị được lựa chọn có thể làm tăng chất lượng của tín hiệu được tái tạo.

Cụ thể, trong một số phương án, dữ liệu điều khiển bộ lọc ngược và sóng hài bị thiếu được truyền trong phần chứa mở rộng eSBR (cùng với các tham số dòng bit khác của Bảng 3) và được điều chỉnh đối với bộ chuyển vị sóng hài của eSBR. Tốc độ bit tăng thêm được yêu cầu để truyền hai lớp siêu dữ liệu này cho bộ chuyển vị sóng hài của eSBR là tương đối thấp. Do đó, việc gửi sóng hài bị thiếu được điều chỉnh và/hoặc dữ liệu điều khiển bộ lọc ngược trong phần chứa mở rộng eSBR sẽ làm tăng chất lượng âm thanh được tạo ra bằng bộ chuyển vị đồng thời chỉ ảnh hưởng một cách tối thiểu đến tốc độ bit. Để đảm bảo tính tương thích ngược với các bộ giải mã kế thừa, các tham số được điều chỉnh cho hoạt động dịch phổ của SBR có thể còn được gửi trong dòng bit dưới dạng một phần của dữ liệu điều khiển SBR bằng cách sử dụng báo hiệu ẩn hoặc rõ ràng.

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. Số tham chiếu bất kỳ có trong yêu cầu bảo hộ kèm theo là chỉ dùng cho mục đích minh họa và không được sử dụng để giải thích hoặc giới hạn phạm vi của các yêu cầu bảo hộ theo bất kỳ cách nào. Theo các khía cạnh khác nhau, sáng chế sẽ được hiểu qua các phương án ví dụ được liệt kê dưới đây (enumerated example embodiment - EEE):

EEE1. Phương pháp thực hiện tái tạo tần số cao của tín hiệu âm thanh, phương pháp này bao gồm các bước:

nhận dòng bit âm thanh được mã hóa, dòng bit âm thanh được mã hóa này bao gồm dữ liệu âm thanh biểu diễn phần băng thấp của tín hiệu âm thanh và siêu dữ liệu tái tạo tần số cao;

giải mã dữ liệu âm thanh để tạo ra tín hiệu âm thanh băng thấp được giải mã;

trích từ dòng bit âm thanh được mã hóa của siêu dữ liệu tái tạo tần số cao, siêu dữ liệu tái tạo tần số cao bao gồm các tham số hoạt động cho quy trình tái tạo tần số cao, các tham số hoạt động bao gồm tham số chế độ và được đặt trong phần chứa mở rộng của dòng bit âm thanh được mã hóa, trong đó giá trị thứ nhất của tham số chế độ và chỉ báo sự dịch phổ và giá trị thứ hai của tham số chế độ và chỉ báo sự chuyển vị sóng hài bằng cách trải tần số của bộ mã hóa tiếng nói pha;

lọc tín hiệu âm thanh băng thấp đã giải mã để tạo ra tín hiệu âm thanh băng thấp đã lọc;

tái tạo phần băng cao của tín hiệu âm thanh bằng cách sử dụng tín hiệu âm thanh băng thấp được lọc và siêu dữ liệu tái tạo tần số cao, trong đó việc tái tạo bao gồm sự dịch phổ nếu tham số chế độ vá là giá trị thứ nhất và việc tái tạo bao gồm sự chuyển vị sóng hài bằng cách trải tần số của bộ mã hóa tiếng nói pha nếu tham số chế độ vá là giá trị thứ hai; và

kết hợp tín hiệu âm thanh băng thấp đã lọc với phần băng cao tái tạo để tạo ra tín hiệu âm thanh băng rộng,

EEE2. Phương pháp theo EEE 1 trong đó phần chứa mở rộng bao gồm dữ liệu điều khiển bộ lọc ngược cần sử dụng khi tham số chế độ vá bằng giá trị thứ hai.

EEE3. Phương pháp theo phương án bất kỳ trong các EEE từ 1 đến 2 trong đó phần chứa mở rộng còn bao gồm dữ liệu điều khiển sóng hài bị thiếu cần sử dụng khi tham số chế độ vá bằng giá trị thứ hai.

EEE4. Phương pháp theo bất kỳ EEE trước đó trong đó dòng bit âm thanh được mã hóa còn bao gồm phần tử làm đầy với mã định danh chỉ báo bắt đầu của phần tử làm đầy và dữ liệu làm đầy sau mã định danh, trong đó dữ liệu làm đầy bao gồm phần chứa mở rộng.

EEE5. Phương pháp theo EEE 4 trong đó mã định danh là số nguyên không dấu 3 bit với bit có ý nghĩa nhất được truyền đầu tiên và có giá trị 0x6.

EEE6. Phương pháp theo EEE 4 hoặc EEE 5, trong đó dữ liệu làm đầy bao gồm tải tin mở rộng, tải tin mở rộng này bao gồm dữ liệu mở rộng tái tạo phổ tần số, và tải tin mở rộng được nhận dạng bởi số nguyên không dấu 4 bit với bit có ý nghĩa nhất được truyền đầu tiên và có giá trị '1101' hoặc '1110', và, tùy ý,

trong đó dữ liệu mở rộng tái tạo phổ tần số bao gồm:

phần đầu tái tạo phổ tần số tùy ý,

dữ liệu tái tạo phổ tần số sau phần đầu này, và

phần tử mở rộng tái tạo phổ tần số sau dữ liệu tái tạo phổ tần số, và trong đó còn được bao gồm trong phần tử mở rộng tái tạo phổ tần số này.

EEE7. Phương pháp theo phương án bất kỳ trong các EEE từ 1 đến 6 trong đó siêu dữ liệu tái tạo tần số cao bao gồm hệ số tỉ lệ đường bao, hệ số tỉ lệ nhiễu nền, thông tin lưới thời gian/tần số, hoặc tham số chỉ báo tần số giao cắt.

EEE8. Phương pháp theo phương án bất kỳ trong các EEE từ 1 đến 7 trong đó việc lọc được thực hiện bởi giàn bộ lọc phân tích bao gồm các bộ lọc phân tích,  $h_k(n)$ , là các phiên bản được điều chế của bộ lọc nguyên mẫu,  $p_0(n)$ , theo:

$$h_k(n) = p_0(n) \exp \left\{ i \frac{\pi}{M} \left( k + \frac{1}{2} \right) \left( n - \frac{N}{2} \right) \right\}, \quad 0 \leq n \leq N; 0 \leq k < M$$

trong đó  $p_0(n)$  là bộ lọc nguyên mẫu không đối xứng hoặc đối xứng giá trị thực,  $M$  là số lượng kênh trong giàn lọc phân tích và  $N$  là thứ tự bộ lọc nguyên mẫu.

EEE9. Phương pháp theo EEE 8 trong đó bộ lọc nguyên mẫu,  $p_0(n)$ , được suy ra từ các hệ số của bảng 4 ở đây.

EEE10. Phương pháp theo EEE 8 trong đó bộ lọc nguyên mẫu,  $p_0(n)$ , được suy ra từ các hệ số của bảng 4 ở đây bằng một hoặc nhiều phép toán được chọn từ nhóm gồm có làm tròn, lấy mẫu phụ, nội suy, hoặc lấy thập phân.

EEE11. Phương pháp theo phương án bất kỳ trong các EEE từ 1 đến 10 trong đó độ dịch pha được thêm vào tín hiệu âm thanh băng thấp đã lọc sau khi lọc và được bù trước khi kết hợp để làm giảm độ phức tạp của phương pháp.

EEE12. Phương pháp theo bất kỳ các EEE trước trong đó phần chứa mở rộng còn bao gồm cờ chỉ báo liệu quy trình tiền xử lý bổ sung có được sử dụng để tránh sự gián đoạn về hình dạng của đường bao phổ trong phần băng cao khi tham số chế độ vắ bằng giá trị thứ nhất không, trong đó giá trị thứ nhất của cờ cho phép tiền xử lý bổ sung và giá trị thứ hai của cờ vô hiệu hóa tiền xử lý bổ sung.

EEE13. Phương pháp theo EEE 12 trong đó quy trình tiền xử lý bổ sung bao gồm bước tính toán đường cong khuếch đại trước bằng cách sử dụng hệ số bộ lọc dự đoán tuyến tính.

EEE14. Phương pháp theo phương án bất kỳ trong các EEE từ 1 đến 13 trong đó phần chứa mở rộng là phần chứa mở rộng tương thích ngược.

EEE15. Phương pháp theo phương án bất kỳ trong các EEE từ 1 đến 14 trong đó dòng âm thanh được mã hóa được mã hóa theo một định dạng, và trong đó phần chứa mở

rộng là phần chứa mở rộng được xác định trong ít nhất một phiên bản kế thừa của định dạng đó.

EEE16. Phương tiện bất biến đọc được bằng máy tính chứa các lệnh mà, khi được thực thi bởi bộ xử lý, thực hiện phương pháp theo phương án bất kỳ trong các EEE từ 1 đến 15.

EEE17. Đơn vị xử lý âm thanh thực hiện tái tạo tần số cao của tín hiệu âm thanh, đơn vị xử lý âm thanh được tạo cấu hình để thực hiện phương pháp theo phương án bất kỳ trong các EEE từ 1 đến 15.

## YÊU CẦU BẢO HỘ

1. Phương pháp thực hiện tái tạo tần số cao tín hiệu âm thanh, phương pháp này bao gồm các bước:

nhận dòng bit âm thanh được mã hóa, dòng bit âm thanh được mã hóa bao gồm dữ liệu âm thanh biểu diễn phần băng thấp của tín hiệu âm thanh và siêu dữ liệu tái tạo tần số cao;

giải mã dữ liệu âm thanh để tạo ra tín hiệu âm thanh băng thấp giải mã;

trích siêu dữ liệu tái tạo tần số cao từ dòng bit âm thanh được mã hóa, siêu dữ liệu tái tạo tần số cao bao gồm các tham số hoạt động cho quy trình tái tạo tần số cao, các tham số hoạt động bao gồm tham số chế độ và được đặt trong phần chứa mở rộng tương thích ngược của dòng bit âm thanh được mã hóa, trong đó giá trị thứ nhất của tham số chế độ và chỉ báo sự dịch phổ và giá trị thứ hai của tham số chế độ và chỉ báo sự chuyển vị sóng hài bằng cách trải tần số của bộ mã hóa tiếng nói pha;

lọc tín hiệu âm thanh băng thấp đã giải mã để tạo ra tín hiệu âm thanh băng thấp đã lọc;

tái tạo phần băng cao của tín hiệu âm thanh bằng cách sử dụng tín hiệu âm thanh băng thấp được lọc và siêu dữ liệu tái tạo tần số cao, trong đó bước tái tạo bao gồm bước dịch phổ nếu tham số chế độ và là giá trị thứ nhất và bước tái tạo bao gồm bước chuyển vị sóng hài bằng cách trải tần số của bộ mã hóa tiếng nói pha nếu tham số chế độ và là giá trị thứ hai; và

kết hợp tín hiệu âm thanh băng thấp đã lọc với phần băng cao tái tạo để tạo ra tín hiệu âm thanh băng rộng,

2. Phương pháp theo điểm 1 trong đó phần chứa mở rộng tương thích ngược bao gồm dữ liệu điều khiển bộ lọc ngược sẽ được sử dụng khi tham số chế độ và bằng giá trị thứ hai.

3. Phương pháp theo điểm 1 trong đó phần chứa mở rộng tương thích ngược còn bao gồm dữ liệu điều khiển sóng hài bị thiếu sẽ được sử dụng khi tham số chế độ và bằng giá trị thứ hai.

4. Phương pháp theo điểm 1 trong đó dòng bit âm thanh được mã hóa còn bao gồm phần tử làm đầy với mã định danh chỉ báo điểm bắt đầu của phần tử làm đầy và dữ liệu làm đầy sau mã định danh, trong đó dữ liệu làm đầy bao gồm phần chứa mở rộng tương thích ngược.

5. Phương pháp theo điểm 4 trong đó mã định danh là số nguyên không dấu 3 bit với bit có ý nghĩa nhất được truyền đầu tiên và có giá trị 0x6.

6. Phương pháp theo điểm 4, trong đó dữ liệu làm đầy bao gồm tải tin mở rộng, tải tin mở rộng này bao gồm dữ liệu mở rộng tái tạo phổ tần số, và tải tin mở rộng được nhận dạng bởi số nguyên không dấu 4 bit với bit có ý nghĩa nhất được truyền đầu tiên và có giá trị '1101' hoặc '1110', và, tùy ý,

trong đó dữ liệu mở rộng tái tạo phổ tần số bao gồm:

phần đầu tái tạo phổ tần số tùy ý,

dữ liệu tái tạo phổ tần số sau phần đầu, và

phần tử mở rộng tái tạo phổ tần số sau dữ liệu tái tạo phổ tần số, và trong đó cờ được bao gồm trong phần tử mở rộng tái tạo phổ tần số.

7. Phương pháp theo điểm 1 trong đó siêu dữ liệu tái tạo tần số cao bao gồm hệ số tỉ lệ đường bao, hệ số tỉ lệ nhiều nền, thông tin lưới thời gian/tần số, hoặc tham số chỉ báo tần số giao cắt.

8. Phương pháp theo điểm 1 trong đó bước lọc được thực hiện bởi giàn bộ lọc phân tích bao gồm các bộ lọc phân tích,  $h_k(n)$ , là các phiên bản được điều chế của bộ lọc nguyên mẫu,  $p_0(n)$ , theo:

$$h_k(n) = p_0(n) \exp \left\{ i \frac{\pi}{M} \left( k + \frac{1}{2} \right) \left( n - \frac{N}{2} \right) \right\}, \quad 0 \leq n \leq N; \quad 0 \leq k < M$$

trong đó  $p_0(n)$  là bộ lọc nguyên mẫu không đối xứng hoặc đối xứng giá trị thực,  $M$  là số lượng kênh trong giàn lọc phân tích và  $N$  là thứ tự bộ lọc nguyên mẫu.

9. Phương pháp theo điểm 8 trong đó bộ lọc nguyên mẫu,  $p_0(n)$ , được suy ra từ các hệ số của bảng 4 ở đây.

10. Phương pháp theo điểm 8 trong đó bộ lọc nguyên mẫu,  $p_0(n)$ , được suy ra từ các hệ số của bảng 4 ở đây bằng một hoặc nhiều phép toán được chọn từ nhóm gồm có làm tròn, lấy mẫu phụ, nội suy, hoặc lấy thập phân.

11. Phương pháp theo điểm 1 trong đó giá trị dịch pha được thêm vào tín hiệu âm thanh bằng thấp đã lọc sau khi lọc và được bù trước khi kết hợp để làm giảm độ phức tạp của phương pháp.

12. Phương pháp theo điểm 1 trong đó phần chứa mở rộng tương thích ngược còn bao gồm cờ biểu thị liệu quy trình tiền xử lý bổ sung có được sử dụng để tránh sự gián đoạn về hình dạng của đường bao phổ trong phần băng cao khi tham số chế độ vó bằng giá trị thứ nhất không, trong đó giá trị thứ nhất của cờ cho phép quy trình tiền xử lý bổ sung và giá trị thứ hai của cờ vô hiệu hóa quy trình tiền xử lý bổ sung.

13. Phương pháp theo điểm 12 trong đó quy trình tiền xử lý bổ sung bao gồm bước tính toán đường cong khuếch đại trước bằng cách sử dụng hệ số bộ lọc dự đoán tuyến tính.

14. Phương tiện bất biến đọc được bằng máy tính chứa các lệnh, mà khi được thực thi bởi bộ xử lý, thực hiện phương pháp theo điểm 1.

15. Đơn vị xử lý âm thanh thực hiện tái tạo tần số cao của tín hiệu âm thanh, đơn vị xử lý âm thanh này bao gồm các bước:

giao diện đầu vào để nhận dòng bit âm thanh được mã hóa, dòng bit âm thanh được mã hóa bao gồm dữ liệu âm thanh biểu diễn phần băng thấp của tín hiệu âm thanh và siêu dữ liệu tái tạo tần số cao;

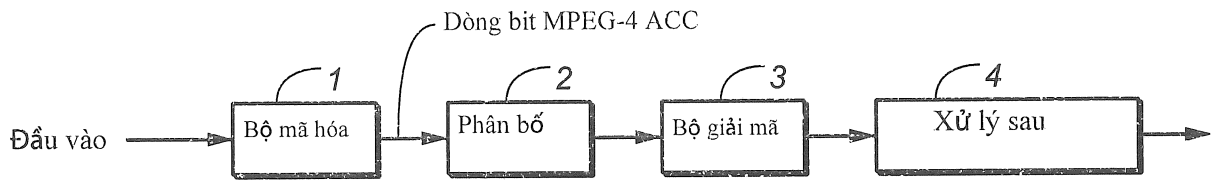
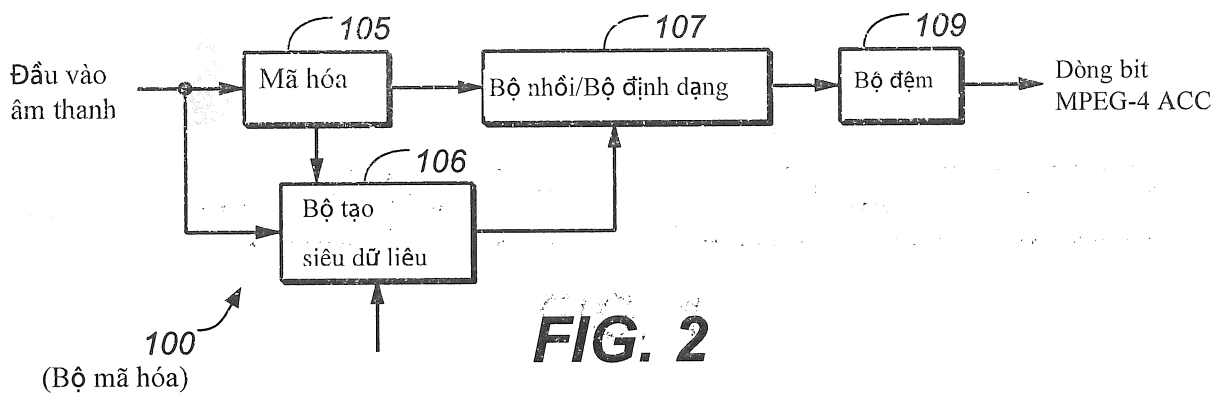
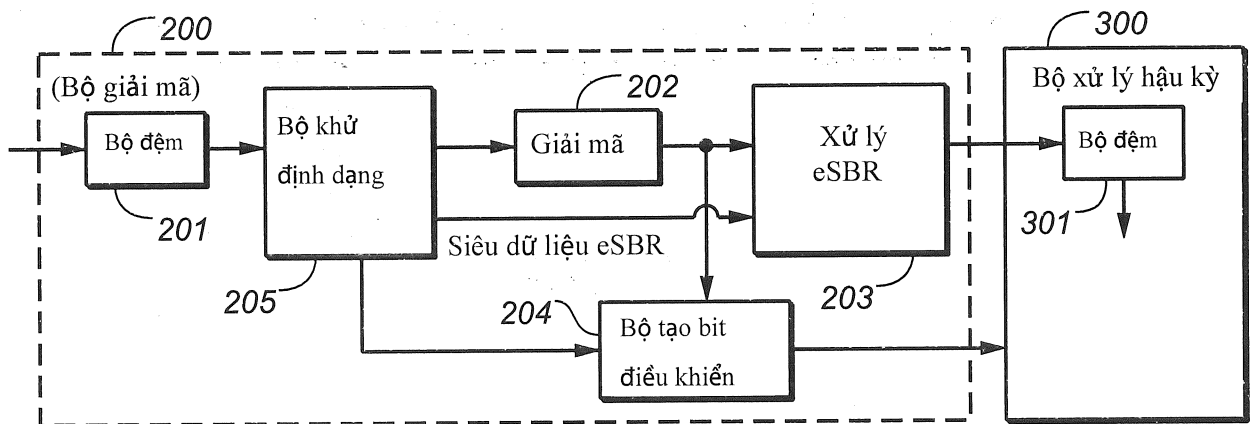
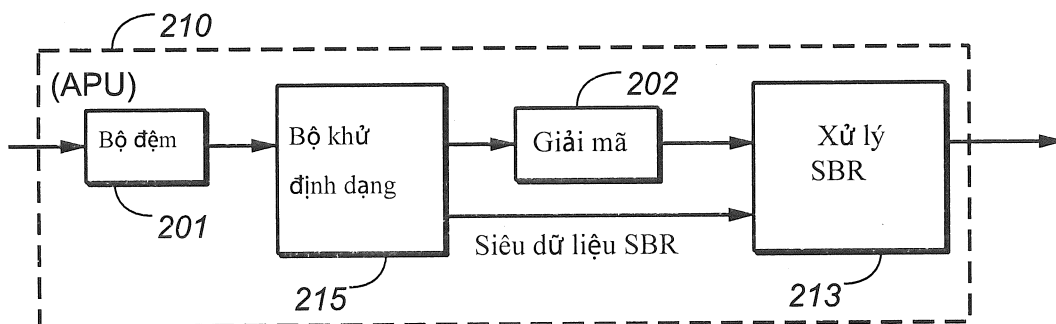
bộ giải mã âm thanh lõi để giải mã dữ liệu âm thanh để tạo ra tín hiệu âm thanh băng thấp đã giải mã;

bộ khử định dạng để trích siêu dữ liệu tái tạo tần số cao từ dòng bit âm thanh được mã hóa, siêu dữ liệu tái tạo tần số cao bao gồm các tham số hoạt động cho quy trình tái tạo tần số cao, các tham số hoạt động bao gồm tham số chế độ và được đặt trong phần chứa mở rộng tương thích ngược của dòng bit âm thanh được mã hóa, trong đó giá trị thứ nhất của tham số chế độ và chỉ báo sự dịch phổ và giá trị thứ hai của tham số chế độ và chỉ báo sự chuyển vị sóng hài bằng cách trải tần số của bộ mã hóa tiếng nói pha;

giàn bộ lọc phân tích để lọc tín hiệu âm thanh băng thấp đã giải mã để tạo ra tín hiệu âm thanh băng thấp đã lọc;

bộ tái tạo tần số cao để tái tạo phần băng cao của tín hiệu âm thanh bằng cách sử dụng tín hiệu âm thanh băng thấp được lọc và siêu dữ liệu tái tạo tần số cao, trong đó bước tái tạo bao gồm bước dịch phổ nếu tham số chế độ và là giá trị thứ nhất và bước tái tạo bao gồm bước chuyển vị sóng hài bằng cách trải tần số của bộ mã hóa tiếng nói pha nếu tham số chế độ và là giá trị thứ hai; và

giàn bộ lọc tổng hợp để kết hợp tín hiệu âm thanh băng thấp đã lọc với phần băng cao tái tạo để tạo ra tín hiệu âm thanh băng rộng.

**FIG. 1****FIG. 2****FIG. 3****FIG. 4**

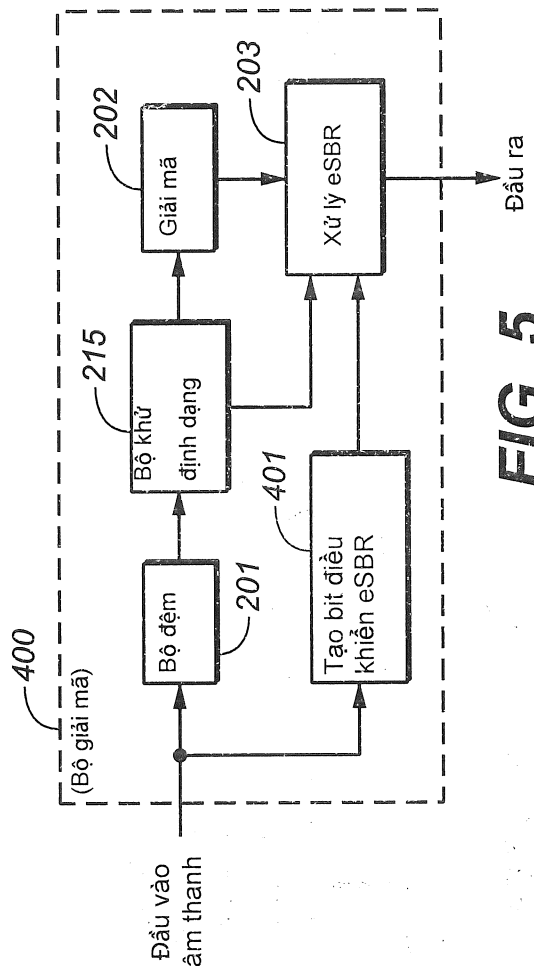


FIG. 6

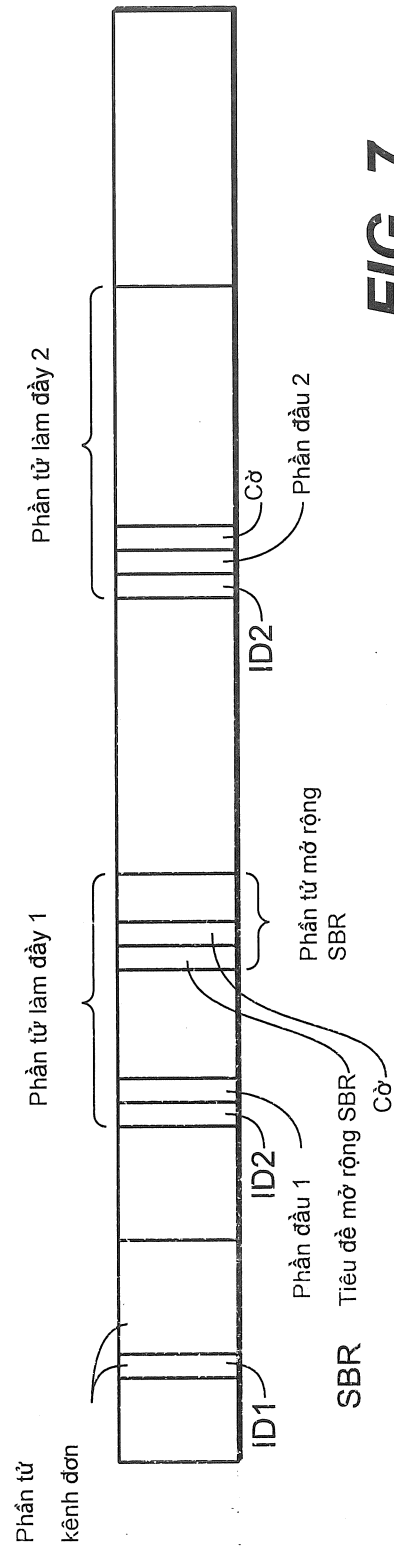


FIG. 7