



(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)



1-0036640

(51)⁷ H04N 7/015; H04N 7/24

(13) B

(21) 1-2012-03766

(22) 13/05/2011

(86) PCT/KR2011/003566 13/05/2011

(87) WO 2011/145841 24/11/2011

(30) 61/344,065 17/05/2010 US

(45) 25/08/2023 425

(43) 27/05/2013 302A

(73) SAMSUNG ELECTRONICS CO., LTD. (KR)

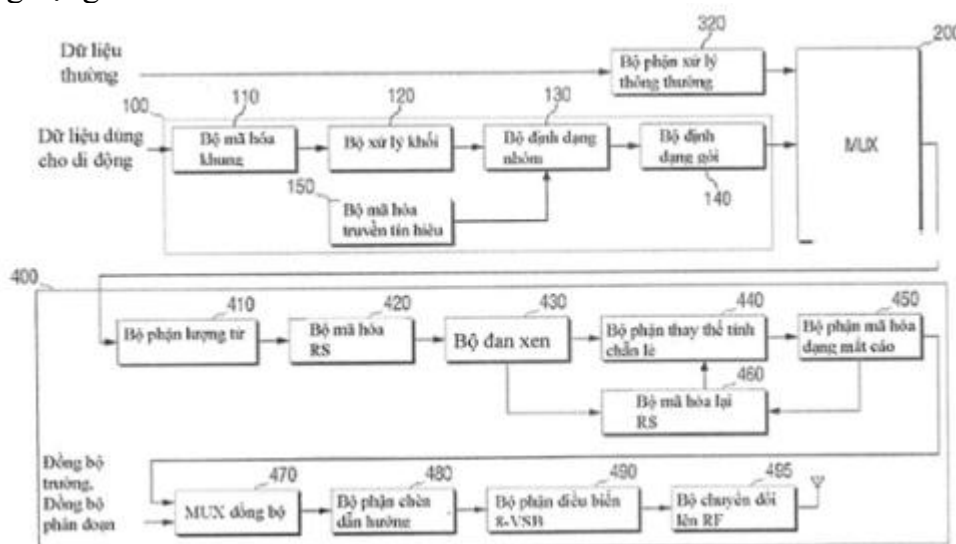
129, Samsung-ro, Yeongtong-gu, Suwon-si, Gyeonggi-do, 443-742, Republic of Korea

(72) Jin-hee JEONG (KR); Jung-Jin KIM (KR); Yong-sik KWON (KR).

(74) Công ty TNHH Tầm nhìn và Liên danh (VISION & ASSOCIATES CO.LTD.)

(54) BỘ TRUYỀN PHÁT RỘNG DẠNG SỐ, BỘ THU PHÁT RỘNG DẠNG SỐ,
PHƯƠNG PHÁP XỬ LÝ LUỒNG CỦA CÁC BỘ NÀY

(57) Sáng chế đề xuất phương pháp xử lý luồng cho bộ truyền phát rộng dạng số. Phương pháp này bao gồm các bước: tạo ra luồng, trong đó các rãnh gồm nhiều khối được đặt liên tục; và mã hóa và đan xen luồng, trong đó, nếu các rãnh của chế độ mở rộng khối thứ nhất được đặt liên tục, thì bước tạo ra luồng bao gồm việc nối chuỗi hướng dẫn được đặt ở các vị trí định trước của các rãnh liên kế để tạo ra chuỗi hướng dẫn dài. Sáng chế còn đề xuất bộ truyền phát rộng dạng số, phương pháp xử lý luồng của bộ thu phát rộng dạng số và bộ thu phát rộng dạng số.



Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến bộ truyền phát rộng dạng số, bộ thu phát rộng dạng số, và phương pháp tạo cấu hình và xử lý luồng của bộ thu phát rộng dạng số, và cụ thể hơn, sáng chế đề cập đến bộ truyền phát rộng dạng số, mà tạo cấu hình luồng vận chuyển gồm dữ liệu di động cùng với dữ liệu thường và truyền luồng vận chuyển, bộ thu phát rộng dạng số, mà nhận và xử lý luồng vận chuyển, và các phương pháp xử lý luồng.

Tình trạng kỹ thuật của sáng chế

Khi truyền phát rộng dạng số được sử dụng rộng rãi, nhiều loại thiết bị điện tử khác nhau hỗ trợ các dịch vụ phát rộng dạng số. Cụ thể, ngoài các thiết bị được bố trí trong các hộ gia đình nói chung như TV truyền phát rộng dạng số hoặc hộp chuyển đổi tín hiệu, các thiết bị di động mà dễ dàng mang theo, như điện thoại di động, hệ thống điều hướng, thiết bị trợ giúp cá nhân dạng số (personal digital assistance-PDA), và máy chơi đa phương tiện MP3 (ví dụ máy chơi MP3), v.v., được trang bị chức năng hỗ trợ các dịch vụ phát rộng dạng số.

Theo đó, các chuẩn truyền phát rộng dạng số để cung cấp các dịch vụ phát rộng dạng số cho thiết bị di động đó đã được thảo luận.

Một trong số chúng là chuẩn ATSC di động/cầm tay (MH). Chuẩn ATSC-MH bộc lộ công nghệ để đặt dữ liệu di động vào trong luồng vận chuyển, nó dùng để truyền dữ liệu cho các dịch vụ phát rộng dạng số thông thường, tức là dữ liệu thường, và truyền dữ liệu di động.

Do dữ liệu di động được nhận và xử lý bởi thiết bị di động, nên dữ liệu di động được xử lý theo định dạng mạnh với lỗi so với dữ liệu thường do tính di động của thiết bị di động và có trong luồng vận chuyển.

Fig.1 minh họa ví dụ về cấu hình luồng vận chuyển bao gồm dữ liệu di động và dữ liệu thường.

Phần (A) của Fig.1 minh họa luồng trong đó dữ liệu di động và dữ liệu thường được đặt trong các gói tương ứng được cấp phát cho nó và được dồn kênh.

Luồng trên phần (A) của Fig.1 được chuyển đổi thành luồng trên phần (B) của Fig.1 bằng cách đan xen. Như được thể hiện trên phần (B) của Fig.1, MH, tức là dữ liệu di động có thể được chia thành các vùng A và B bằng cách đan xen. Vùng A là vùng nằm trong phạm vi định trước dựa vào phần, tại đó dữ liệu di động lớn hơn kích thước định trước được tập hợp trong nhiều đơn vị truyền, và vùng B là vùng khác với vùng A. Dữ liệu di động được chia thành các vùng A và B theo cách làm ví dụ và có thể được chia theo cách khác. Tức là, trên phần (B) của Fig.1, phần mà không bao gồm dữ liệu thường có thể được thiết lập cho vùng A và phần tương ứng với đơn vị truyền trong đó dữ liệu thường ít nhất được đặt có thể được thiết lập cho vùng B.

Có vấn đề là vùng B tương đối dễ bị lỗi so với vùng A. Tức là, dữ liệu truyền phát rộng dạng số có thể bao gồm dữ liệu đã biết, ví dụ, chuỗi hướng dẫn, để được giải điều biến và cân bằng một cách thích hợp bởi bộ thu để hiệu chỉnh lỗi. Theo chuẩn ATSC-MH theo giải pháp kỹ thuật đã biết, dữ liệu đã biết không được đặt trong vùng B và theo đó có vấn đề là vùng B dễ bị lỗi.

Ngoài ra, luồng được xác định như được thể hiện trên Fig.1 có thể giới hạn việc truyền dữ liệu di động. Tức là, số lượng các trạm phát rộng và thiết bị hỗ trợ các dịch vụ phát rộng di động đã tăng lên, nhưng, một phần được cấp phát cho dữ liệu thường trên luồng được thể hiện trên Fig.1 không sử dụng được và theo đó hiệu quả của luồng giảm sút.

Do đó, cần có phương pháp sử dụng luồng vận chuyển một cách hiệu quả.

Bản chất kỹ thuật của sáng chế

Theo các khía cạnh của một hoặc nhiều phương án ví dụ, sáng chế đề xuất bộ truyền phát rộng dạng số, bộ thu phát rộng dạng số, và phương pháp tạo cấu hình và xử lý luồng của bộ thu phát rộng dạng số, chúng có thể sử dụng các gói của luồng vận chuyển được cấp phát cho dữ liệu thường theo các cách khác nhau, nhờ đó đa dạng hóa hiệu suất truyền dữ liệu di động và còn cải thiện hiệu suất thu nhận luồng vận chuyển.

Theo một khía cạnh của phương án ví dụ, sáng chế đề xuất phương pháp xử lý luồng của bộ truyền phát rộng dạng số, phương pháp này bao gồm các bước: tạo cấu hình luồng, trong đó các rãnh gồm nhiều khối được đặt; và mã hóa và đan xen luồng và truyền luồng làm luồng vận chuyển, trong đó bước tạo cấu hình luồng bao gồm, nếu các rãnh của

chế độ mở rộng khối 00 được đặt liên tục, thì nối dữ liệu đã biết đặt ở các vị trí định trước của các rãnh liên kế nhau để tạo ra chuỗi hướng dẫn dài.

Dữ liệu đã biết thứ nhất được đặt ở phần cuối của rãnh trước của các rãnh liên kế và dữ liệu đã biết thứ hai được đặt ở phần đầu của rãnh tiếp theo của các rãnh liên kế có thể được nối luân phiên với nhau trên ranh giới, và giá trị của dữ liệu đã biết thứ nhất và giá trị của dữ liệu đã biết thứ hai có thể là các giá trị định trước để tạo ra chuỗi hướng dẫn dài mà đã được biết đối với bộ truyền phát rộng dạng số và bộ thu phát rộng dạng số.

Dữ liệu đã biết có thể có cùng chuỗi như chuỗi hướng dẫn dài mà được sử dụng trong rãnh của chế độ mở rộng khối 01 trong đó một số khối của rãnh tương ứng được cung cấp cho một rãnh khác.

Bước truyền có thể bao gồm việc khởi tạo bộ mã hóa dạng mất cáo trước khi dữ liệu đã biết tương ứng với phần ban đầu của chuỗi hướng dẫn dài được mã hóa dạng mất cáo.

Bước truyền có thể bao gồm, nếu các rãnh của các chế độ mở rộng khối khác nhau được đặt liên tục, thì khởi tạo bộ mã hóa dạng mất cáo trước khi mã hóa dạng mất cáo dữ liệu đã biết được đặt ở phần rãnh của ranh giới giữa các rãnh được đặt liên tục.

Theo một khía cạnh của phương án ví dụ khác, sáng chế đề xuất bộ truyền phát rộng dạng số bao gồm: bộ phận tạo cấu hình luồng tạo cấu hình luồng, trong đó các rãnh gồm nhiều khối được đặt, và bộ phận kích thích mã hóa và đan xen luồng và truyền luồng làm luồng vận chuyển.

Nếu các rãnh của chế độ mở rộng khối 00 trong đó tất cả các khối của rãnh tương ứng được sử dụng được đặt liên tục, thì bộ phận tạo cấu hình luồng có thể đặt dữ liệu đã biết trong các phân đoạn định trước của các rãnh liên kế để tạo ra chuỗi hướng dẫn dài trên ranh giới giữa các rãnh liên kế được ghép khớp với kết cấu dạng răng cưa.

Dữ liệu đã biết thứ nhất được đặt ở phần cuối của rãnh trước của các rãnh liên kế và dữ liệu đã biết thứ hai được đặt ở phần đầu của rãnh tiếp theo của các rãnh liên kế có thể được nối luân phiên với nhau trên ranh giới, và giá trị của dữ liệu đã biết thứ nhất và giá trị của dữ liệu đã biết thứ hai có thể là các giá trị định trước để tạo ra chuỗi hướng dẫn dài mà đã được biết đối với bộ truyền phát rộng dạng số và bộ thu phát rộng dạng số.

Dữ liệu đã biết có thể có cùng chuỗi như chuỗi hướng dẫn dài mà được sử dụng trong rãnh của chế độ mở rộng khối 01, trong đó một số khối của rãnh tương ứng được cấp cho một rãnh khác.

Bộ phận kích thích có thể bao gồm: bộ phận mã hóa mà mã hóa luồng, bộ phận đan xen mà đan xen luồng đã được mã hóa, và bộ phận mã hóa dạng mắt cáo mà mã hóa dạng mắt cáo luồng đã được đan xen.

Bộ phận mã hóa dạng mắt cáo có thể được khởi tạo trước khi dữ liệu đã biết tương ứng với phần ban đầu của chuỗi hướng dẫn dài được mã hóa dạng mắt cáo.

Nếu các rãnh của các chế độ mở rộng khối khác nhau được đặt liên tục, thì bộ phận mã hóa dạng mắt cáo có thể được khởi tạo trước khi mã hóa dạng mắt cáo dữ liệu đã biết được đặt ở phần rãnh của ranh giới giữa các rãnh được đặt liên tục.

Theo một khía cạnh của phương án ví dụ khác, sáng chế đề xuất phương pháp xử lý luồng của bộ thu phát rộng dạng số, phương pháp này bao gồm các bước: nhận luồng vận chuyển được mã hóa và đan xen nếu các rãnh gồm nhiều khối được đặt liên tục, giải điều biến luồng vận chuyển nhận được, cân bằng luồng vận chuyển đã được giải điều biến, và giải mã dữ liệu di động mới từ luồng được cân bằng.

Mỗi rãnh của luồng vận chuyển có thể bao gồm ít nhất một trong số dữ liệu thường, dữ liệu di động hiện có, và dữ liệu di động mới, và nếu các rãnh của chế độ mở rộng khối 00 được đặt liên tục, trong luồng vận chuyển, dữ liệu đã biết mà được đặt ở các vị trí định trước của các rãnh liên kế có thể được nối với nhau để tạo ra chuỗi hướng dẫn dài.

Dữ liệu đã biết thứ nhất được đặt ở phần cuối của rãnh trước của các rãnh liên kế và dữ liệu đã biết thứ hai được đặt ở phần đầu của rãnh tiếp theo của các rãnh liên kế có thể được nối luân phiên với nhau trên ranh giới, và giá trị của dữ liệu đã biết thứ nhất và giá trị của dữ liệu đã biết thứ hai có thể là các giá trị định trước để tạo ra chuỗi hướng dẫn dài mà đã được biết đối với bộ truyền phát rộng dạng số và bộ thu phát rộng dạng số.

Dữ liệu đã biết có thể có cùng chuỗi như chuỗi hướng dẫn dài mà được sử dụng trong rãnh của chế độ mở rộng khối 01, trong đó một số khối của rãnh tương ứng được cung cấp cho một rãnh khác.

Phương pháp này còn có thể bao gồm bước giải mã dữ liệu truyền tín hiệu của mỗi rãnh và nhận diện chế độ mở rộng khối của mỗi một trong số các rãnh.

Phương pháp này còn có thể bao gồm, nếu hoàn thành bước giải mã dữ liệu truyền tín hiệu của rãnh tiếp theo của các rãnh liền kề và nó được nhận diện là chế độ mở rộng khối của rãnh tiếp theo là 00, thì phát hiện dữ liệu đã biết được đặt ở phần răng cưa của ranh giới giữa các rãnh liền kề như chuỗi hướng dẫn dài và xử lý dữ liệu đã biết.

Phương pháp này còn có thể bao gồm bước giải mã dữ liệu truyền tín hiệu của rãnh trước của các rãnh liền kề và nhận diện các chế độ mở rộng khối của cả rãnh trước và rãnh tiếp theo.

Theo một khía cạnh của phương án ví dụ của sáng chế, sáng chế đề xuất bộ thu phát rộng dạng số bao gồm: bộ phận thu nhận luồng vận chuyển được mã hóa và đan xen nếu các rãnh gồm nhiều khối được đặt, bộ giải điều biến giải điều biến luồng vận chuyển nhận được, bộ cân bằng cân bằng luồng vận chuyển đã được giải điều biến, và bộ phận giải mã mà giải mã dữ liệu di động mới từ luồng được cân bằng.

Mỗi rãnh của luồng vận chuyển có thể bao gồm ít nhất một trong số dữ liệu thường, dữ liệu di động hiện có, và dữ liệu di động mới, và nếu các rãnh của chế độ mở rộng khối 00 được đặt liên tục, thì trong luồng vận chuyển, dữ liệu đã biết mà được đặt ở các vị trí định trước của các rãnh liền kề có thể được nối với nhau để tạo ra chuỗi hướng dẫn dài.

Dữ liệu đã biết thứ nhất được đặt ở phần cuối của rãnh trước của các rãnh liền kề và dữ liệu đã biết thứ hai được đặt ở phần đầu của rãnh tiếp theo của các rãnh liền kề có thể được nối luân phiên với nhau trên ranh giới, và giá trị của dữ liệu đã biết thứ nhất và giá trị của dữ liệu đã biết thứ hai có thể là các giá trị định trước để tạo ra chuỗi hướng dẫn dài mà đã được biết đối với bộ truyền phát rộng dạng số và bộ thu phát rộng dạng số.

Chuỗi hướng dẫn dài có thể có cùng chuỗi như chuỗi hướng dẫn dài mà được sử dụng trong rãnh của chế độ mở rộng khối 01, trong đó một số khối của rãnh tương ứng được cung cấp cho một rãnh khác.

Bộ thu phát rộng dạng số còn có thể bao gồm bộ giải mã truyền tín hiệu mà giải mã dữ liệu truyền tín hiệu của mỗi rãnh và nhận diện chế độ mở rộng khối của mỗi rãnh trong các rãnh.

Bộ thu phát rộng dạng số còn có thể bao gồm bộ phận phát hiện mã, nếu hoàn thành bước giải mã dữ liệu truyền tín hiệu của rãnh tiếp theo của các rãnh liên kế và nó được nhận diện là chế độ mở rộng khối của rãnh tiếp theo là 00, thì phát hiện dữ liệu đã biết được đặt ở phần răng cưa của ranh giới giữa các rãnh liên kế như chuỗi hướng dẫn dài và xử lý dữ liệu đã biết.

Phần truyền phát rộng dạng số còn có thể bao gồm bộ giải mã truyền tín hiệu mà giải mã dữ liệu truyền tín hiệu của rãnh trước của các rãnh liên kế và nhận diện các chế độ mở rộng khối của cả rãnh trước và rãnh tiếp theo.

Theo các phương án ví dụ khác nhau, luồng vận chuyển được truyền theo các định dạng khác nhau, sao cho bộ thu có thể nhận các loại dữ liệu di động khác nhau.

Các khía cạnh và ưu điểm bổ sung của các phương án ví dụ của sáng chế sẽ được bộc lộ ở phần mô tả chi tiết, sẽ trở nên rõ ràng từ phần mô tả chi tiết, hoặc có thể được biết nhờ thực hiện các phương án ví dụ.

Mô tả vắn tắt các hình vẽ

Các khía cạnh trên và/hoặc các khía cạnh khác của sáng chế sẽ được hiểu rõ hơn từ phần mô tả sau đây của các phương án ví dụ, dựa vào các hình vẽ kèm theo, trong đó:

Fig.1 minh họa một ví dụ về cấu hình luồng vận chuyển theo chuẩn ATSC-MH trong giải pháp kỹ thuật đã biết;

Các hình vẽ từ Fig.2 đến Fig.4 minh họa bộ truyền phát rộng dạng số theo các phương án ví dụ khác nhau của sáng chế;

Fig.5 là sơ đồ khối minh họa bộ mã hóa khung, theo phương án ví dụ của sáng chế;

Fig.6 là sơ đồ khối minh họa một ví dụ về bộ mã hóa khung Reed Solomon (RS) của bộ mã hóa khung trên Fig.5;

Fig.7 là sơ đồ khối minh họa bộ xử lý khối, theo phương án ví dụ của sáng chế;

Fig.8 là hình vẽ giải thích ví dụ về phương pháp chia luồng thành các khối, theo phương án ví dụ của sáng chế;

Fig.9 là sơ đồ khối minh họa bộ mã hóa truyền tín hiệu, theo phương án ví dụ của sáng chế;

Các hình vẽ từ Fig.10 đến Fig.13 là các hình vẽ minh họa bộ mã hóa dạng mắt cáo, theo phương án ví dụ của sáng chế;

Fig.14 là hình vẽ giải thích khung dữ liệu di động theo phương án ví dụ của sáng chế;

Các hình vẽ từ Fig.15 đến Fig.21 là các hình vẽ minh họa ví dụ về kết cấu luồng theo các phương án ví dụ khác nhau của sáng chế;

Các hình vẽ từ Fig.22 đến Fig.28 là các hình vẽ minh họa mẫu chèn của dữ liệu đã biết theo các phương án ví dụ khác nhau của sáng chế;

Fig.29 là hình vẽ minh họa mẫu, trong đó dữ liệu di động được đặt trong vùng dữ liệu thường theo chế độ thứ nhất, theo phương án ví dụ của sáng chế;

Fig.30 là hình vẽ minh họa luồng trên Fig.29 ở trạng thái được đan xen;

Fig.31 là hình vẽ minh họa mẫu, trong đó dữ liệu di động được đặt trong vùng dữ liệu thường theo chế độ thứ hai, theo phương án ví dụ của sáng chế;

Fig.32 là hình vẽ minh họa luồng trên Fig.31 ở trạng thái được đan xen;

Fig.33 là hình vẽ minh họa mẫu, trong đó dữ liệu di động được đặt trong vùng dữ liệu thường theo chế độ thứ ba, theo phương án ví dụ của sáng chế;

Fig.34 là hình vẽ minh họa luồng trên Fig.33 ở trạng thái được đan xen;

Fig.35 là hình vẽ minh họa mẫu, trong đó dữ liệu di động được đặt trong vùng dữ liệu thường theo chế độ thứ tư, theo phương án ví dụ của sáng chế;

Fig.36 là hình vẽ minh họa luồng trên Fig.35 ở trạng thái được đan xen;

Các hình vẽ từ Fig.37 đến Fig.40 là các hình vẽ minh họa các mẫu, trong đó dữ liệu di động được đặt theo các chế độ khác nhau, theo phương án ví dụ của sáng chế;

Các hình vẽ từ Fig.41 đến Fig.43 là các hình vẽ minh họa các loại rãnh khác nhau được đặt lặp lại trong chuỗi, theo phương án ví dụ của sáng chế;

Các hình vẽ từ Fig.44 đến Fig.47 là các hình vẽ giải thích phương pháp cấp phát các khối theo các phương án ví dụ khác nhau của sáng chế;

Fig.48 là hình vẽ giải thích phương pháp xác định điểm bắt đầu của khung RS theo các phương án ví dụ khác nhau của sáng chế;

Fig.49 là hình vẽ giải thích vị trí chèn của dữ liệu truyền tín hiệu, theo phương án ví dụ của sáng chế;

Fig.50 là hình vẽ minh họa một ví dụ về kết cấu đồng bộ trường dữ liệu để truyền dữ liệu truyền tín hiệu, theo phương án ví dụ của sáng chế;

Các hình vẽ từ Fig.51 đến Fig.53 là các hình vẽ minh họa bộ thu phát rộng dạng số theo các phương án ví dụ khác nhau của sáng chế;

Fig.54 là hình vẽ minh họa một ví dụ về định dạng luồng sau khi đan xen theo phương án ví dụ của sáng chế;

Fig.55 là hình vẽ giải thích một ví dụ về phương pháp truyền trước tín hiệu thông tin của khung tiếp theo, theo phương án ví dụ của sáng chế;

Fig.56 là hình vẽ minh họa kết cấu luồng sau khi đan xen ở chế độ có thể mở rộng 11a, theo phương án ví dụ của sáng chế;

Fig.57 là hình vẽ minh họa kết cấu luồng trước khi đan xen ở chế độ có thể mở rộng 11a, theo phương án ví dụ của sáng chế;

Fig.58 là hình vẽ minh họa kết cấu luồng biểu thị vùng mờ còi loại thứ nhất sau khi đan xen, theo phương án ví dụ của sáng chế;

Fig.59 là hình vẽ minh họa kết cấu luồng biểu thị vùng mờ còi loại thứ nhất trước khi đan xen, theo phương án ví dụ của sáng chế;

Fig.60 là hình vẽ minh họa kết cấu luồng biểu thị vùng mờ còi loại thứ hai sau khi đan xen, theo phương án ví dụ của sáng chế;

Fig.61 là hình vẽ minh họa kết cấu luồng biểu thị vùng mờ còi loại thứ hai trước khi đan xen, theo phương án ví dụ của sáng chế;

Fig.62 là hình vẽ minh họa kết cấu luồng biểu thị vùng mô côi loại thứ ba sau khi đan xen, theo phương án ví dụ của sáng chế;

Fig.63 là hình vẽ minh họa kết cấu luồng biểu thị vùng mô côi loại thứ ba trước khi đan xen, theo phương án ví dụ của sáng chế;

Fig.64 là hình vẽ minh họa kết cấu luồng trước khi đan xen ở chế độ mở rộng khối 00, theo phương án ví dụ của sáng chế; và

Fig.65 là hình vẽ minh họa kết cấu luồng sau khi đan xen ở chế độ mở rộng khối 00, theo phương án ví dụ của sáng chế.

Mô tả chi tiết sáng chế

Sau đây, các phương án ví dụ của sáng chế sẽ được mô tả một cách chi tiết, các ví dụ của chúng được minh họa trên các hình vẽ kèm theo, trong đó các số chỉ dẫn giống nhau biểu thị các thành phần giống nhau trong suốt bản mô tả. Các phương án ví dụ được mô tả sau đây nhằm giải thích sáng chế dựa vào các hình vẽ.

Bộ truyền phát rộng dạng số

Trên Fig.2, bộ truyền phát rộng dạng số theo một phương án ví dụ bao gồm bộ tiền xử lý dữ liệu 100 và bộ dồn kênh 200.

Bộ tiền xử lý dữ liệu 100 nhận đầu vào của dữ liệu di động, xử lý dữ liệu di động một cách thích hợp, và chuyển đổi dữ liệu di động thành dữ liệu có định dạng phù hợp để truyền.

Bộ dồn kênh 200 tạo cấu hình luồng vận chuyển bao gồm dữ liệu di động được đưa ra từ bộ tiền xử lý dữ liệu 100. Nếu dữ liệu thường phải được truyền đầy đủ, thì bộ dồn kênh 200 tạo cấu hình luồng vận chuyển bằng cách dồn kênh dữ liệu di động và dữ liệu thường.

Bộ tiền xử lý dữ liệu 100 có thể xử lý dữ liệu di động sao cho dữ liệu di động được đặt trong tất cả hoặc một số gói của toàn bộ luồng được cấp phát cho dữ liệu thường.

Tức là, như được giải thích trên Fig.1, một số gói được cấp phát cho dữ liệu thường theo chuẩn ATSC-MH. Cụ thể, như được thể hiện trên Fig.1, luồng có thể được chia thành nhiều rãnh theo đơn vị thời gian và một rãnh có thể bao gồm 156 gói. Trong số

các gói này, 38 gói có thể được cấp phát cho dữ liệu thường và 118 gói còn lại có thể được cấp phát cho dữ liệu di động. Để giải thích thuận tiện, 118 gói được gọi là vùng được cấp phát cho dữ liệu di động hoặc vùng thứ nhất, và 38 gói được gọi là vùng được cấp phát cho dữ liệu thường hoặc vùng thứ hai. Dữ liệu thường chỉ các loại dữ liệu hiện có khác nhau mà có thể nhận được và xử lý bởi ti vi (television-TV) thông thường, và dữ liệu di động chỉ dữ liệu mà có thể được nhận và xử lý bởi thiết bị di động. Dữ liệu di động có thể được biểu diễn bởi các thuật ngữ khác nhau, như dữ liệu mạnh, dữ liệu turbo, và dữ liệu bổ sung, v.v..

Bộ tiền xử lý dữ liệu 100 có thể đặt dữ liệu di động vào trong vùng gói được cấp phát cho dữ liệu di động, và theo cách riêng rẽ, có thể đặt dữ liệu di động trong tất cả hoặc một số gói được cấp phát cho dữ liệu thường. Để giải thích thuận tiện, dữ liệu di động được đặt trong các gói được cấp phát cho dữ liệu di động được gọi là dữ liệu di động hiện có hoặc dữ liệu di động thứ nhất, và vùng được cấp phát cho dữ liệu di động hiện có được gọi là vùng thứ nhất như đã mô tả ở trên. Mặt khác, dữ liệu di động được đặt trong vùng thứ hai, tức là các gói được cấp phát cho dữ liệu thường, được gọi là dữ liệu di động mới, dữ liệu di động hoặc dữ liệu di động thứ hai để giải thích thuận tiện. Dữ liệu di động hiện có và dữ liệu di động có thể là cùng dữ liệu hoặc có thể là loại dữ liệu khác.

Bộ tiền xử lý dữ liệu 100 có thể đặt dữ liệu di động theo các mẫu khác nhau theo điều kiện thiết lập như chế độ khung hoặc chế độ thiết lập. Các mẫu, trong đó dữ liệu di động được đặt sẽ được giải thích sau đây dựa vào các hình vẽ.

Bộ dồn kênh 200 dồn kênh luồng và dữ liệu thường được đưa ra từ bộ tiền xử lý dữ liệu 100, theo đó tạo cấu hình luồng vận chuyển.

Fig.3 minh họa một phương án ví dụ, trong đó bộ điều khiển 310 được thêm vào bộ truyền phát rộng dạng số trên Fig.2. Trên Fig.3, bộ điều khiển 310 được bố trí trong bộ truyền phát rộng dạng số xác định điều kiện thiết lập của chế độ khung và điều khiển hoạt động của bộ tiền xử lý dữ liệu 100.

Cụ thể, nếu xác định được là chế độ khung thứ nhất được thiết lập, bộ điều khiển 310 điều khiển bộ tiền xử lý dữ liệu 100 để chỉ đặt dữ liệu di động vào trong vùng thứ nhất mà không đặt dữ liệu di động trong tất cả các gói được cấp phát cho dữ liệu thường. Tức là, bộ tiền xử lý dữ liệu 100 đưa ra luồng chỉ bao gồm dữ liệu di động hiện có. Do đó, dữ liệu thường được đặt trong các gói được cấp phát cho dữ liệu thường nhờ bộ dồn kênh 200 sao cho luồng vận chuyển được tạo cấu hình.

Mặt khác, nếu xác định được là chế độ khung thứ hai được thiết lập, bộ điều khiển 310 điều khiển bộ tiền xử lý dữ liệu 100 đặt dữ liệu di động hiện có trong các gói được cấp phát cho dữ liệu di động, tức là vùng thứ nhất, và đặt dữ liệu di động trong ít nhất một số gói được cấp phát cho dữ liệu thường, tức là ít nhất một phần của vùng thứ hai.

Trong trường hợp này, bộ điều khiển 310 có thể xác định điều kiện thiết lập của chế độ riêng khác với chế độ khung, tức là chế độ biểu thị số lượng các gói tại đó dữ liệu di động cần được đặt trong số các gói được cấp phát cho dữ liệu thường. Do đó, bộ điều khiển 310 có thể điều khiển bộ tiền xử lý dữ liệu 100 để đặt dữ liệu di động vào trong các gói nhiều như số lượng tương ứng với điều kiện thiết lập của chế độ, trong số toàn bộ các gói được cấp phát cho dữ liệu thường.

Chế độ được đề cập ở đây có thể được cung cấp theo các dạng khác nhau. Ví dụ, chế độ có thể bao gồm ít nhất một chế độ tương thích và chế độ không tương thích. Chế độ tương thích chỉ chế độ, trong đó tính tương thích với bộ thu dữ liệu thường hiện có, mà nhận và xử lý dữ liệu thường, được duy trì, và chế độ không tương thích chỉ chế độ trong đó tính tương thích không được duy trì.

Cụ thể, chế độ tương thích có thể bao gồm nhiều chế độ tương thích, trong đó dữ liệu di động mới được đặt ít nhất một phần của vùng thứ hai. Ví dụ, chế độ tương thích có thể là một trong số chế độ tương thích thứ nhất, trong đó dữ liệu di động chỉ được đặt trong một số gói được cấp phát cho dữ liệu thường hoặc chế độ tương thích thứ hai, trong đó dữ liệu di động được đặt trong tất cả các gói được cấp phát cho dữ liệu thường.

Chế độ tương thích thứ nhất có thể là chế độ, trong đó dữ liệu di động chỉ được đặt trong một phần của mỗi vùng dữ liệu của một số gói trong vùng thứ hai. Tức là, dữ liệu di động có thể được đặt trong một phần của toàn bộ vùng dữ liệu của một số gói, và dữ liệu thường có thể được đặt trong vùng dữ liệu khác.

Ngoài ra, chế độ tương thích thứ nhất có thể là chế độ, trong đó dữ liệu di động được đặt trong toàn bộ vùng dữ liệu của một số gói trong vùng thứ hai.

Chế độ có thể được cung cấp theo các dạng khác nhau tính đến số lượng các gói được cấp phát cho dữ liệu thường, kích thước của dữ liệu di động, loại dữ liệu di động, thời gian truyền, và môi trường truyền.

Ví dụ, như được thể hiện trên Fig.1, nếu 38 gói được cấp phát cho dữ liệu thường, chế độ tương thích thứ nhất có thể bao gồm:

- 1) chế độ thứ nhất, trong đó dữ liệu di động mới được đặt trong 38 gói theo tỷ lệ $1/4$;
- 2) chế độ thứ hai, trong đó dữ liệu di động mới được đặt trong 38 gói theo tỷ lệ $2/4$;
- 3) chế độ thứ ba, trong đó dữ liệu di động mới được đặt trong 38 gói theo tỷ lệ $3/4$; và
- 4) chế độ thứ tư, trong đó dữ liệu di động mới được đặt trong tất cả 38 gói.

Ở chế độ thứ nhất, dữ liệu di động mới có thể được đặt trong tổng của 2 trong số 38 gói và 9 gói tương ứng với thương của 36 gói còn lại chia cho 4, tức là tổng cộng 11 gói. Ở chế độ thứ hai, dữ liệu di động mới có thể được đặt trong tổng của 2 trong số 38 gói và 18 gói tương ứng với thương của 36 gói còn lại chia cho 2, tức là tổng cộng 20 gói. Ở chế độ thứ ba, dữ liệu di động mới có thể được đặt trong tổng của 2 trong số 38 gói và 27 gói là 36 gói còn lại nhân với $3/4$, tức là tổng cộng 29 gói. Ở chế độ thứ tư, dữ liệu di động mới có thể được đặt trong tất cả 38 gói.

Mặt khác, chế độ không tương thích chỉ chế độ, trong đó khả năng truyền tải của dữ liệu di động mới có thể tăng lên không quan tâm đến tính tương thích với bộ thu nhận dữ liệu thường. Cụ thể, chế độ không tương thích có thể là chế độ, trong đó dữ liệu di động mới được đặt bằng cách sử dụng phần đầu MPEG và vùng chắn lẻ Reed Solmon (RS) của vùng thứ nhất ngoài toàn bộ vùng thứ hai.

Kết quả là, bộ tiền xử lý dữ liệu 100 trên Fig.2 hoặc Fig.3 có thể tạo cấu hình luồng vận chuyển bằng cách đặt dữ liệu di động mới theo các chế độ khác nhau như sau:

- 1) chế độ thứ nhất, trong đó dữ liệu di động mới được đặt trong tổng cộng 11 gói trong số 38 gói được cấp phát cho dữ liệu thường;
- 2) chế độ thứ hai, trong đó dữ liệu di động mới được đặt trong tổng cộng 20 gói trong số 38 gói được cấp phát cho dữ liệu thường;
- 3) chế độ thứ ba, trong đó dữ liệu di động mới được đặt trong tổng cộng 29 gói trong số 38 gói được cấp phát cho dữ liệu thường;
- 4) chế độ thứ tư, trong đó dữ liệu di động mới được đặt trong tất cả 38 gói được cấp phát cho dữ liệu thường; và

5) chế độ thứ năm, trong đó dữ liệu di động mới được đặt trong tất cả 38 gói được cấp phát cho dữ liệu thường và trong vùng tương ứng với phần đầu MPEG và phần chắn lẻ của vùng được cấp phát cho dữ liệu di động hiện có.

Mặc dù chế độ thứ năm được gọi là chế độ không tương thích và các chế độ thứ nhất đến thứ tư được gọi là các chế độ tương thích theo phương án ví dụ này để giải thích thuận tiện, mỗi chế độ có thể được đặt tên khác nhau. Ngoài ra, mặc dù có tổng cộng năm chế độ bao gồm bốn chế độ tương thích và một chế độ không tương thích theo phương án ví dụ nêu trên, nhưng số lượng các chế độ tương thích có thể thay đổi khác đi. Ví dụ, các chế độ thứ nhất đến thứ ba có thể được sử dụng làm các chế độ tương thích như đã mô tả ở trên, và chế độ thứ tư có thể được thiết lập là chế độ thứ năm, tức là chế độ không tương thích.

Bộ tiền xử lý dữ liệu 100 có thể chèn dữ liệu đã biết ngoài dữ liệu di động. Dữ liệu đã biết được đề cập ở đây chỉ chuỗi được biết chung đối với bộ truyền phát rộng dạng số và bộ thu phát rộng dạng số. Bộ thu phát rộng dạng số nhận dữ liệu đã biết được truyền từ bộ truyền phát rộng dạng số và nhận diện sự khác biệt với chuỗi đã biết trước, và sau đó nắm được độ hiệu chỉnh lỗi. Dữ liệu đã biết có thể được biểu diễn bởi các thuật ngữ khác nhau như dữ liệu hướng dẫn, chuỗi hướng dẫn, dữ liệu tham chiếu, dữ liệu tham chiếu bổ sung, v.v. Tuy nhiên, thuật ngữ “dữ liệu đã biết” sẽ được sử dụng trong toàn bộ bản mô tả để mô tả thuận tiện.

Bộ tiền xử lý dữ liệu 100 chèn ít nhất một trong số dữ liệu di động và dữ liệu đã biết vào trong các phần khác nhau của toàn bộ luồng vận chuyển, nhờ đó cải thiện hiệu quả thu nhận.

Tức là, theo kết cấu luồng được thể hiện trên phần (B) của Fig.1, MH, tức là dữ liệu di động được tập hợp trong vùng A và được tạo ra trong vùng B theo kết cấu dạng nón. Do đó, vùng A có thể được gọi là vùng thân và vùng B có thể được gọi là vùng đầu/đuôi. Do dữ liệu đã biết không được đặt trong vùng đầu/đuôi, nên có vấn đề trong lĩnh vực liên quan là dữ liệu không thể hiện hiệu suất tốt so với dữ liệu của vùng thân.

Do đó, bộ tiền xử lý dữ liệu 100 chèn dữ liệu đã biết vào vị trí thích hợp để được đặt trong vùng đầu/đuôi. Dữ liệu đã biết có thể được đặt trong định dạng chuỗi hướng dẫn dài, trong đó dữ liệu lớn hơn kích thước định trước được đặt liên tục, hoặc có thể được đặt theo định dạng phân bố không liên tục.

Dữ liệu di động và dữ liệu đã biết có thể được chèn theo các định dạng khác nhau theo một phương án ví dụ. Điều này sẽ được giải thích chi tiết dựa vào các hình vẽ kèm theo. Tuy nhiên, kết cấu chi tiết của bộ truyền phát rộng dạng số trước tiên sẽ được giải thích theo cách chi tiết hơn.

Kết cấu chi tiết của bộ truyền phát rộng dạng số

Fig.4 là sơ đồ khối minh họa chi tiết bộ truyền phát rộng dạng số theo một phương án ví dụ. Trên Fig.4, bộ truyền phát rộng dạng số có thể bao gồm bộ xử lý thường 320 và bộ phận kích thích 400 (ví dụ, bộ kích thích) ngoài bộ tiền xử lý dữ liệu 100 và bộ dồn kênh 200. Để giải thích thuận tiện, bộ phận bao gồm bộ tiền xử lý dữ liệu 100, bộ xử lý thông thường 320, và bộ dồn kênh 200 có thể được gọi là bộ phận tạo cấu hình luồng.

Trên Fig.4, bộ điều khiển 310 được thể hiện trên Fig.3 được bỏ qua. Tuy nhiên, cần hiểu rằng, bộ điều khiển 310 có trong bộ truyền phát rộng dạng số. Ngoài ra, một hoặc nhiều thành phần của bộ truyền phát rộng dạng số được thể hiện trên Fig.4 có thể được loại bỏ hoặc thành phần mới có thể được thêm khi cần, và thứ tự bố trí các thành phần và số lượng các thành phần có thể thay đổi khác nhau.

Trên Fig.4, bộ xử lý thông thường 320 nhận dữ liệu thường và chuyển đổi nó thành dữ liệu có định dạng thích hợp mà phù hợp với cấu hình luồng vận chuyển. Tức là, do bộ truyền phát rộng dạng số tạo cấu hình luồng vận chuyển bao gồm dữ liệu thường và dữ liệu di động và truyền luồng vận chuyển, bộ thu mà nhận dữ liệu thường sẽ nhận và xử lý dữ liệu thường một cách phù hợp. Do đó, bộ xử lý thông thường 320 điều chỉnh việc định thời gói và tham chiếu đồng hồ chương trình (program clock reference-PCR) của dữ liệu thường (hoặc dữ liệu dịch vụ chính) để có định dạng phù hợp với chuẩn MPEG/ATSC, mà được sử dụng để giải mã dữ liệu thường. Giải thích chi tiết của chúng được bộc lộ trong Annex B của ATSC-MH (toàn bộ nội dung cả tài liệu này được đưa vào bản mô tả này bằng cách viện dẫn) và theo đó được bỏ qua ở đây.

Bộ tiền xử lý dữ liệu 100 bao gồm bộ mã hóa khung 110, bộ xử lý khối 120, bộ định dạng nhóm 130, bộ định dạng gói 140, và bộ mã hóa truyền tín hiệu 150.

Bộ mã hóa khung 110 thực hiện việc mã hóa khung RS. Cụ thể, bộ mã hóa khung 110 nhận dịch vụ đơn và xây dựng số lượng định trước của các khung RS. Ví dụ, nếu dịch vụ đơn là đơn vị quần thể M/H bao gồm nhiều tập hợp M/H, thì bộ mã hóa khung 110 xây dựng số lượng định trước các khung RS cho mỗi tập hợp M/H. Cụ thể, bộ mã hóa khung

110 ngẫu nhiên hóa dữ liệu di động đầu vào, thực hiện việc mã hóa RS-CRC, chia dữ liệu di động thành các khung RS theo chế độ khung định trước, và đưa ra số lượng định trước của các khung RS.

Fig.5 là sơ đồ khối minh họa một ví dụ về bộ mã hóa khung 110. Trên Fig.5, bộ mã hóa khung 110 bao gồm bộ giải dồn kênh đầu vào 111, nhiều bộ mã hóa khung RS 112-1~112-M, và bộ dồn kênh đầu ra 113.

Nếu dữ liệu di động của đơn vị dịch vụ định trước (ví dụ, quần thể M/H) được đưa vào, thì bộ giải dồn kênh đầu vào 111 giải dồn kênh dữ liệu di động thành nhiều quần thể, ví dụ, quần thể sơ cấp và quần thể thứ cấp, theo thông tin kết cấu đặt trước, tức là chế độ khung, và đưa các quần thể ra tới các bộ mã hóa khung RS 112-1~112-M. Mỗi một trong số các bộ mã hóa khung RS 112-1~112-M thực hiện ngẫu nhiên hóa, mã hóa RS-CRC, và chia thành các quần thể đầu vào, và đưa các quần thể ra tới bộ dồn kênh đầu ra 113. Bộ dồn kênh đầu ra 113 dồn kênh các phần khung được đưa ra từ các bộ mã hóa khung RS 112-1~112-M, và đưa ra phần khung RS sơ cấp và phần khung RS thứ cấp. Trong trường hợp này, chỉ có phần khung RS sơ cấp có thể được đưa ra theo điều kiện thiết lập của chế độ khung.

Fig.6 là sơ đồ khối minh họa một ví dụ về bộ mã hóa khung RS là một trong số các bộ mã hóa khung RS 112-1~112-M. Trên Fig.6, bộ mã hóa khung 112 bao gồm nhiều bộ ngẫu nhiên hóa M/H 112-1a và 112-1b, nhiều bộ mã hóa RS-CRC 112-2a và 112-2b, và nhiều bộ chia khung RS 112-3a và 112-3b.

Nếu quần thể M/H sơ cấp và quần thể M/H thứ cấp được đưa vào từ bộ giải dồn kênh đầu vào 111, mỗi bộ ngẫu nhiên hóa M/H 112-1a và 112-1b thực hiện ngẫu nhiên hóa và mỗi bộ mã hóa RS-CRC 112-2a và 112-2b mã hóa RS-CRC dữ liệu đã được ngẫu nhiên hóa. Mỗi bộ chia khung RS 112-3a và 112-3b chia dữ liệu để được mã hóa khối một cách phù hợp và đưa dữ liệu ra tới bộ dồn kênh đầu ra 113 sao cho bộ xử lý khối 120 được đặt ở đầu sau của bộ mã hóa khung 110 mã hóa khối dữ liệu một cách phù hợp. Bộ dồn kênh đầu ra 113 dồn kênh các phần khung bằng cách kết hợp chúng một cách phù hợp và đưa các phần khung ra tới bộ xử lý khối 120 sao cho bộ xử lý khối 120 mã hóa khối cho dữ liệu.

Bộ xử lý khối 120 mã hóa luồng được đưa ra từ bộ mã hóa khung 110 trong đơn vị khối, tức là mã hóa khối cho luồng.

Fig.7 là sơ đồ khối minh họa một ví dụ về bộ xử lý khối 120.

Trên Fig.7, bộ xử lý khối 120 bao gồm bộ chuyển đổi thứ nhất 121, bộ chuyển đổi byte thành bit 122, bộ mã hóa chập 123, bộ đan xen ký hiệu 124, bộ chuyển đổi ký hiệu thành byte 125, và bộ chuyển đổi thứ hai 126.

Bộ chuyển đổi thứ nhất 121 chuyển đổi khung RS được đưa vào từ bộ mã hóa khung 110 trong đơn vị khối. Tức là, bộ chuyển đổi thứ nhất 121 kết hợp dữ liệu di động trong khung RS theo chế độ khối định trước và đưa ra khối mã chập ghép nối tiếp (serially concatenated convolutional code-SCCC).

Ví dụ, nếu chế độ khối là “00”, thì khối M/H đơn trở thành khối SCCC đơn như vậy.

Fig.8 là hình vẽ minh họa dữ liệu di động được chia thành các khối M/H trong đơn vị khối. Trên Fig.8, đơn vị dữ liệu di động đơn, ví dụ, nhóm M/H có thể được chia thành 10 khối (B1 đến B10). Nếu chế độ khối là “00”, thì mỗi khối từ B1 đến B10 được đưa ra là khối SCCC. Mặt khác, nếu chế độ khối là “01”, thì hai khối M/H được kết hợp và đưa ra là khối SCCC đơn. Mẫu kết hợp có thể được thiết lập khác nhau. Ví dụ, các khối từ B1 và B6 có thể được kết hợp để tạo ra khối SCB1, và các khối B2 và B7, các khối B3 và B8, các khối B4 và B9, và các khối B5 và B10 được kết hợp để tạo ra các khối SCB2, SCB3, SCB4, và SCB5, theo cách tương ứng. Các khối có thể được kết hợp theo các cách khác nhau và số lượng của các khối kết hợp có thể khác nhau tùy theo các chế độ khối khác.

Bộ chuyển đổi byte thành bit 122 chuyển đổi khối SCCC từ đơn vị byte thành đơn vị bit. Sở dĩ như vậy là do bộ mã hóa chập 123 được vận hành theo đơn vị bit. Do đó, bộ mã hóa chập 123 mã hóa chập dữ liệu đã được chuyển đổi.

Sau đó, bộ đan xen ký hiệu 124 thực hiện đan xen ký hiệu. Việc đan xen ký hiệu có thể được thực hiện theo cùng một cách như theo kiểu đan xen khối. Dữ liệu đã được đan xen ký hiệu được chuyển đổi thành đơn vị byte nhờ bộ chuyển đổi ký hiệu thành byte 125, được chuyển đổi lại thành đơn vị khối M/H nhờ bộ chuyển đổi thứ hai 126, và được đưa ra.

Bộ định dạng nhóm 130 nhận luồng được xử lý bởi bộ xử lý khối 120 và định dạng luồng theo đơn vị nhóm. Cụ thể, bộ định dạng nhóm 130 ánh xạ dữ liệu được đưa ra từ bộ xử lý khối 120 vào vị trí thích hợp trong luồng, và thêm dữ liệu đã biết, dữ liệu

truyền tín hiệu, và dữ liệu khởi tạo vào luồng. Bộ định dạng nhóm 130 có thể thêm byte giữ chỗ cho dữ liệu thường, phần đầu MPEG-2, và phần chẵn lẻ RS không hệ thống, và byte giả để phù hợp với định dạng nhóm.

Dữ liệu truyền tín hiệu biểu thị nhiều loại thông tin cần thiết để xử lý luồng vận chuyển. Dữ liệu truyền tín hiệu có thể được xử lý phù hợp bởi bộ mã hóa truyền tín hiệu 150 và có thể được cấp cho bộ định dạng nhóm 130.

Để truyền dữ liệu di động, kênh thông số truyền (transmission parameter channel-TPC) và kênh thông tin nhanh (fast information channel-FIC) có thể được sử dụng. TPC là để cung cấp các thông số khác nhau như thông tin chế độ hiệu chỉnh lỗi tiến (forward error correction-FEC) và thông tin khung M/H, và FIC là để thu được dịch vụ nhanh của bộ thu và bao gồm thông tin lớp chéo giữa lớp vật lý và lớp trên. Nếu thông tin TPC và thông tin FIC được cấp cho bộ mã hóa truyền tín hiệu 150, bộ mã hóa truyền tín hiệu 150 xử lý thông tin một cách phù hợp và cung cấp thông tin làm dữ liệu truyền tín hiệu.

Fig.9 là sơ đồ khối minh họa một ví dụ về bộ mã hóa truyền tín hiệu 150.

Trên Fig.9, bộ mã hóa truyền tín hiệu 150 bao gồm bộ mã hóa RS 151 cho TPC, bộ dồn kênh 152, bộ mã hóa RS 153 cho FIC, bộ đan xen khối 154, bộ ngẫu nhiên hóa truyền dữ liệu 155, và bộ mã hóa mã chập ghép song song (Parallel Concatenated Convolutional Code-PCCC) 156. Bộ mã hóa RS 151 cho TPC mã hóa RS dữ liệu TPC đầu vào và tạo ra từ mã TPC. Bộ mã hóa RS 153 cho FIC và bộ đan xen khối 154 mã hóa RS và đan xen khối dữ liệu FIC đầu vào và tạo ra từ mã FIC. Bộ dồn kênh 152 đặt từ mã FIC sau từ mã TPC, theo đó tạo ra dãy gồm các chuỗi. Các chuỗi được ngẫu nhiên hóa bởi bộ ngẫu nhiên hóa truyền dữ liệu 155, được mã hóa PCCC nhờ bộ mã hóa PCCC 156, và được đưa ra bộ định dạng nhóm 130 làm dữ liệu truyền tín hiệu.

Dữ liệu đã biết chỉ chuỗi được biết chung đối với bộ truyền phát rộng dạng số và bộ thu phát rộng dạng số như đã mô tả ở trên. Bộ định dạng nhóm 130 chèn dữ liệu đã biết vào ở vị trí thích hợp theo tín hiệu điều khiển được cung cấp từ thành phần được bố trí riêng (ví dụ, bộ điều khiển 310), sao cho dữ liệu đã biết được đặt ở vị trí thích hợp của luồng sau khi được đan xen nhờ bộ phận kích thích 400. Ví dụ, dữ liệu đã biết có thể được chèn vào ở vị trí thích hợp sao cho dữ liệu đã biết có thể được đặt trong vùng B có kết cấu luồng trên Fig.1. Bộ định dạng nhóm 130 có thể tự xác định vị trí chèn dữ liệu đã biết, có tính đến luật đan xen.

Dữ liệu khởi tạo chỉ dữ liệu cần thiết cho bộ phận mã hóa dạng mắt cáo 450 (bộ mã hóa dạng mắt cáo) của bộ phận kích thích 400 để khởi tạo các bộ nhớ trong của nó tại thời điểm thích hợp. Điều này sẽ được giải thích chi tiết khi giải thích bộ phận kích thích 400.

Bộ định dạng nhóm 130 có thể bao gồm bộ phận tạo cấu hình định dạng nhóm để tạo cấu hình luồng theo định dạng nhóm bằng cách chen các vùng và tín hiệu khác nhau vào trong luồng như đã mô tả ở trên, và bộ giải đan xen dữ liệu để giải đan xen luồng được tạo cấu hình theo định dạng nhóm.

Bộ giải đan xen dữ liệu sắp xếp lại dữ liệu theo thứ tự ngược lại với thứ tự của bộ đan xen 430 được đặt ở đầu sau của luồng. Luồng được giải đan xen bởi bộ giải đan xen dữ liệu có thể được cấp cho bộ định dạng gói 140.

Bộ định dạng gói 140 loại bỏ các phần giữ chỗ khác nhau được bố trí trong luồng nhờ bộ định dạng nhóm 130 và thêm phần đầu MPEG có PID, nó là ID gói của dữ liệu di động. Do đó, bộ định dạng gói 140 đưa ra luồng theo đơn vị gồm số lượng gói định trước cho mọi nhóm. Ví dụ, 118 gói TS có thể được đưa ra.

Như đã mô tả ở trên, bộ tiền xử lý dữ liệu 100 được thực hiện theo các kết cấu khác nhau và tạo cấu hình dữ liệu di động theo định dạng phù hợp. Cụ thể, nếu cung cấp nhiều dịch vụ di động, thì số lượng các thành phần nằm trong bộ tiền xử lý dữ liệu 100 có thể là số nhiều.

Bộ dồn kênh 200 dồn kênh luồng thường được xử lý bởi bộ xử lý thông thường 320 và luồng di động được xử lý bởi bộ tiền xử lý dữ liệu 100, theo đó tạo cấu hình luồng vận chuyển. Luồng vận chuyển được đưa ra từ bộ dồn kênh 200 có thể bao gồm dữ liệu thường và dữ liệu di động và còn có thể bao gồm dữ liệu đã biết để cải thiện hiệu quả thu nhận.

Bộ phận kích thích 400 thực hiện việc mã hóa, đan xen, mã hóa dạng mắt cáo, và điều biến đối với luồng vận chuyển được tạo cấu hình nhờ bộ dồn kênh 200, và đưa ra luồng vận chuyển. Theo một số trường hợp, bộ phận kích thích 400 có thể được gọi là bộ xử lý sau dữ liệu.

Trên Fig.4, bộ phận kích thích 400 (ví dụ bộ kích thích) bao gồm bộ ngẫu nhiên hóa 410, bộ mã hóa RS 420, bộ đan xen 430, bộ phận thay thế chẵn lẻ 440, bộ phận mã hóa dạng mắt cáo 450 (ví dụ bộ mã hóa dạng mắt cáo), bộ mã hóa lại RS 460, bộ dồn

kênh đồng bộ 470, bộ phận chèn dẫn hướng 480 (ví dụ bộ chèn dẫn hướng), bộ điều biến 8-VSB 490, và bộ chuyển đổi lên RF 495.

Bộ ngẫu nhiên hóa 410 ngẫu nhiên hóa luồng vận chuyển được đưa ra từ bộ dồn kênh 200. Bộ ngẫu nhiên hóa 410 thực hiện cùng chức năng như chức năng của bộ ngẫu nhiên hóa theo chuẩn ATSC.

Bộ ngẫu nhiên hóa 410 có thể tính toán XOR phần đầu MPEG của dữ liệu di động và toàn bộ dữ liệu thường với chuỗi nhị phân giả ngẫu nhiên (pseudo random binary sequence-PRBS) có chiều dài tối đa 16 bit, nhưng có thể không tính toán XOR byte tải của dữ liệu di động. Tuy nhiên, trong trường hợp này, bộ tạo PRBS có thể tiếp tục xê dịch thanh ghi dịch chuyển. Tức là, bộ ngẫu nhiên hóa 410 bỏ qua byte tải của dữ liệu di động.

Bộ mã hóa RS 420 thực hiện việc mã hóa RS đối với luồng đã được ngẫu nhiên hóa.

Cụ thể, nếu một phần tương ứng với dữ liệu thường được đưa vào, thì bộ mã hóa RS 420 thực hiện việc mã hóa RS có hệ thống theo cùng cách như trong hệ thống ATSC trong lĩnh vực. Tức là, bộ mã hóa RS 420 thêm phần chẵn lẻ 20 byte vào phần cuối của mỗi trong số các gói có 187 byte. Mặt khác, nếu một phần tương ứng với dữ liệu di động được đưa vào, thì bộ mã hóa RS 420 thực hiện việc mã hóa RS không hệ thống. Trong trường hợp này, dữ liệu RS FEC 20 byte, thu được bằng cách mã hóa RS không hệ thống, được đặt ở vị trí byte chẵn lẻ định trước trong mỗi gói dữ liệu di động. Do đó, bộ truyền có thể có tính tương thích với bộ thu chuẩn ATSC trong lĩnh vực.

Bộ đan xen 430 đan xen luồng được mã hóa bởi bộ mã hóa RS 420. Việc đan xen có thể được thực hiện theo cùng một cách như việc đan xen của hệ thống ATSC trong lĩnh vực. Tức là, bộ đan xen 430 ghi và đọc dữ liệu, mặc dù lựa chọn nhiều đường dẫn, gồm số lượng các thanh ghi ghi dịch khác nhau, một cách tuần tự bằng cách sử dụng bộ chuyển mạch, sao cho việc đan xen được thực hiện nhiều như lượng các thanh ghi dịch chuyển trên đường dẫn.

Bộ phận thay thế chẵn lẻ 440 hiệu chỉnh phần chẵn lẻ được thay đổi nhờ thực hiện khởi tạo bộ nhớ bởi bộ phận mã hóa dạng mắt cáo 450 ở đầu sau.

Tức là, bộ phận mã hóa dạng mắt cáo 450 nhận luồng được đan xen và thực hiện việc mã hóa dạng mắt cáo đối với luồng. Bộ phận mã hóa dạng mắt cáo 450 thường sử dụng 12 bộ mã hóa dạng mắt cáo. Do đó, bộ giải điều biến chia luồng thành 12 luồng độc

lập và đưa các luồng vào trong mỗi bộ mã hóa dạng mắt cáo, và bộ dồn kênh kết hợp các luồng được mã hóa dạng mắt cáo bởi các bộ mã hóa dạng mắt cáo để tạo ra luồng đơn được sử dụng.

Mỗi một trong số các bộ mã hóa dạng mắt cáo thực hiện việc mã hóa dạng mắt cáo theo cách sao cho nó tính toán logic giá trị đầu vào mới và giá trị được lưu trữ trước trong bộ nhớ trong nhờ sử dụng nhiều bộ nhớ trong và đưa ra giá trị.

Như đã mô tả ở trên, luồng vận chuyển có thể bao gồm dữ liệu đã biết. Dữ liệu đã biết là chuỗi đã biết mà được biết chung đối với bộ truyền phát rộng dạng số và bộ thu phát rộng dạng số. Bộ thu phát rộng dạng số có thể xác định độ hiệu chỉnh lỗi bằng cách nhận diện trạng thái của dữ liệu đã biết nhận được. Tức là, dữ liệu đã biết phải được truyền theo cách mà bộ thu biết. Tuy nhiên, do giá trị được lưu trữ trong bộ nhớ trong được cung cấp trong bộ mã hóa dạng mắt cáo không được biết, nên bộ nhớ trong phải được khởi tạo để có giá trị nhất định trước khi đưa vào dữ liệu đã biết. Do đó, bộ phận mã hóa dạng mắt cáo 450 khởi tạo bộ nhớ trước khi mã hóa dạng mắt cáo dữ liệu đã biết. Việc khởi tạo bộ nhớ có thể được gọi là thiết lập lại dạng mắt cáo.

Fig.10 là hình vẽ minh họa một ví dụ về một trong nhiều bộ mã hóa dạng mắt cáo được bố trí trong bộ phận mã hóa dạng mắt cáo 450.

Trên Fig.10, bộ mã hóa dạng mắt cáo bao gồm các bộ dồn kênh thứ nhất 451 và thứ hai 452, các bộ cộng thứ nhất 453 và thứ hai 454, các bộ nhớ thứ nhất 455, thứ hai 456 và thứ ba 457, và bộ ánh xạ 458.

Bộ dồn kênh thứ nhất 451 nhận dữ liệu N trong luồng và giá trị I được lưu trữ trong bộ nhớ thứ nhất 455, và đưa ra một giá trị, tức là N hoặc I, theo tín hiệu điều khiển N/I. Cụ thể, tín hiệu điều khiển mà điều khiển để lựa chọn I khi giá trị tương ứng với đoạn dữ liệu khởi tạo được đưa vào được sử dụng sao cho bộ dồn kênh thứ nhất 451 đưa ra I. Trong các đoạn khác, N được đưa ra. Tương tự, bộ dồn kênh thứ hai 452 chỉ đưa ra I khi giá trị tương ứng với đoạn dữ liệu khởi tạo được đưa vào.

Do đó, bộ dồn kênh thứ nhất 451 đưa giá trị được đan xen ra đầu sau như vậy khi đoạn khác với đoạn dữ liệu khởi tạo được đưa vào, và giá trị đầu ra được đưa vào bộ cộng thứ nhất 453 cùng với giá trị được lưu trữ trước trong bộ nhớ thứ nhất 455. Bộ cộng thứ nhất 453 thực hiện tính toán logic, tức là OR loại trừ đối với các giá trị đầu vào, và đưa ra Z2. Trong giai đoạn này, nếu đoạn dữ liệu khởi tạo được đưa vào, thì giá trị được lưu trữ

trong bộ nhớ thứ nhất 455 được lựa chọn bởi bộ dồn kênh thứ nhất 451 như vậy và được đưa ra. Do đó, do hai giá trị giống nhau được đưa vào bộ cộng thứ nhất 453, giá trị tính toán logic luôn là giá trị không đổi. Tức là, nếu thực hiện OR loại trừ, thì 0 được đưa ra. Do giá trị đầu ra của bộ cộng thứ nhất 453 được đưa vào bộ nhớ thứ nhất 455 như vậy, bộ nhớ thứ nhất 455 được khởi tạo để có giá trị 0.

Nếu đoạn dữ liệu khởi tạo được đưa vào, bộ dồn kênh thứ hai 452 lựa chọn giá trị được lưu trữ trong bộ nhớ thứ ba 457 như vậy và đưa ra giá trị. Giá trị đầu ra được đưa vào bộ cộng thứ hai 454 cùng với giá trị được lưu trữ trong bộ nhớ thứ ba 457. Bộ cộng thứ hai 454 thực hiện tính toán logic đối với hai giá trị giống nhau và đưa giá trị thu được ra tới bộ nhớ thứ hai 456. Như đã mô tả ở trên, do các giá trị đầu vào của bộ cộng thứ hai 454 giống nhau, giá trị tính toán logic cho cùng các giá trị, ví dụ, 0 trong trường hợp của OR loại trừ được đưa vào bộ nhớ thứ hai 456. Do đó, bộ nhớ thứ hai 456 được khởi tạo. Giá trị được lưu trữ trong bộ nhớ thứ hai 456 được dịch và được lưu trữ trong bộ nhớ thứ ba 457. Do đó, khi dữ liệu khởi tạo tiếp theo được đưa vào, giá trị hiện tại của bộ nhớ thứ hai 456, tức là 0 được đưa vào bộ nhớ thứ ba 457 như vậy và theo đó bộ nhớ thứ ba 457 cũng được khởi tạo.

Bộ ánh xạ 458 nhận giá trị đầu ra của bộ cộng thứ nhất 453, giá trị đầu ra của bộ dồn kênh thứ hai 452, và giá trị đầu ra của bộ nhớ thứ hai 456, và ánh xạ các giá trị thành giá trị ký hiệu tương ứng R và đưa ra giá trị ký hiệu. Ví dụ, nếu Z0, Z1, và Z2 được lần lượt đưa vào là 0, 1, và 0, thì bộ ánh xạ 458 đưa ra ký hiệu -3.

Do bộ mã hóa RS 420 được đặt trước bộ mã hóa dạng mắt cáo 450, nên giá trị được đưa vào bộ mã hóa dạng mắt cáo 450 chứa phần chẵn lẻ đã được thêm vào đó. Do đó, khi bộ mã hóa dạng mắt cáo 450 thực hiện việc khởi tạo và theo đó một số giá trị của dữ liệu được thay đổi, phần chẵn lẻ cũng sẽ được thay đổi.

Bộ mã hóa lại RS 460 thay đổi giá trị của đoạn dữ liệu khởi tạo bằng cách sử dụng X1' và X2' được đưa ra từ bộ phận mã hóa dạng mắt cáo 450, nhờ đó tạo ra phần chẵn lẻ mới. Bộ mã hóa lại RS 460 có thể được gọi là bộ mã hóa RS không hệ thống.

Mặc dù bộ nhớ được khởi tạo để có giá trị là 0 trên Fig.10, nhưng theo các phương án ví dụ khác, bộ nhớ có thể được khởi tạo để có giá trị khác 0.

Fig.11 là hình vẽ minh họa một ví dụ khác của bộ mã hóa dạng mắt cáo.

Trên Fig.11, bộ mã hóa dạng mắt cáo có thể bao gồm các bộ dồn kênh thứ nhất 451 và thứ hai 452, các bộ cộng thứ nhất 453, thứ hai 454, thứ ba 459-1 và thứ tư 459-2, và các bộ nhớ thứ nhất 455, thứ hai 456 và thứ ba 457. Bộ ánh xạ 458 được bỏ qua trên Fig.11.

Cụ thể, bộ dồn kênh thứ nhất 451 có thể đưa ra một trong số giá trị đầu vào luồng X2 và giá trị của bộ cộng thứ ba 459-1. Giá trị I_X2 và giá trị lưu trữ của bộ nhớ thứ nhất 455 được đưa vào bộ cộng thứ ba 459-1. Giá trị I_X2 chỉ giá trị thiết lập lại bộ nhớ được đưa vào từ nguồn bên ngoài. Ví dụ, nếu bộ nhớ thứ nhất 455 cần được khởi tạo để có giá trị là 1, thì giá trị I_X2 được đưa vào là 1. Nếu giá trị lưu trữ của bộ nhớ thứ nhất 455 là 0, thì giá trị đầu ra của bộ cộng thứ ba 459-1 là 1 và bộ dồn kênh thứ nhất 451 đưa ra 1. Do đó, bộ cộng thứ nhất 453 thực hiện OR loại trừ đối với giá trị đầu ra của bộ dồn kênh thứ nhất 451, 1, và giá trị lưu trữ của bộ nhớ thứ nhất 455, 0, và lưu giá trị thu được, 1, trong bộ nhớ thứ nhất 455. Kết quả là, bộ nhớ thứ nhất 455 được khởi tạo để có giá trị là 1.

Nếu đoạn dữ liệu khởi tạo được đưa vào, bộ dồn kênh thứ hai 452 lựa chọn giá trị đầu ra của bộ cộng thứ tư 459-2 và đưa ra giá trị. Bộ cộng thứ tư 459-2 đưa ra giá trị thiết lập lại bộ nhớ I_X1 được nhập vào từ bên ngoài và giá trị OR loại trừ của bộ nhớ thứ ba 457. Ví dụ, nếu 1 và 0 lần lượt được lưu trữ trong các bộ nhớ thứ hai 456 và thứ ba 457, và hai bộ nhớ cần được khởi tạo để lần lượt có các giá trị là 1 và 1, thì bộ dồn kênh thứ hai 452 đưa ra giá trị OR loại trừ 1 của giá trị 0 được lưu trữ trong bộ nhớ thứ ba 457 và giá trị I_X1 là 1. Bộ cộng thứ hai 454 thực hiện OR loại trừ đối với giá trị đầu ra là 1 và giá trị là 0 được lưu trữ trong bộ nhớ thứ ba 457, đưa giá trị thu được là 1 vào bộ nhớ thứ hai 456. Giá trị là 1 được lưu trữ ban đầu trong bộ nhớ thứ hai 456 được dịch vào bộ nhớ thứ ba 457 sao cho giá trị của bộ nhớ thứ ba 457 là 1. Trong giai đoạn này, nếu I_X1 thứ hai được đưa vào là 1, thì thực hiện tính toán OR loại trừ đối với I_X1 và giá trị là 1 của bộ nhớ thứ ba 457, và giá trị thu được 0 được đưa ra từ bộ dồn kênh thứ hai 452. Nếu bộ cộng thứ hai 454 thực hiện OR loại trừ đối với giá trị là 0 được đưa ra từ bộ dồn kênh thứ hai 452 và giá trị là 1 được lưu trữ trong bộ nhớ thứ ba 457, thì giá trị thu được là 1 được đưa vào bộ nhớ thứ hai 456 và giá trị là 1 được lưu trữ trong bộ nhớ thứ hai 456 được dịch vào bộ nhớ thứ ba 457 và được lưu trữ trong bộ nhớ thứ ba 457. Kết quả là, các bộ nhớ thứ hai 456 và thứ ba 457 được khởi tạo để có giá trị là 1.

Fig.12 và Fig.13 là các hình vẽ minh họa bộ mã hóa dạng mắt cáo theo các phương án ví dụ khác nhau.

Trên Fig.12, bộ mã hóa dạng mắt cáo còn bao gồm các bộ dồn kênh thứ ba 459-3 và thứ tư 459-4 ngoài các thành phần trên Fig.11. Các bộ dồn kênh thứ ba 459-3 và thứ tư 459-4 có thể đưa ra các giá trị của các bộ cộng thứ nhất 453 và thứ hai 454 hoặc các giá trị I_X2 và I_X1 theo tín hiệu điều khiển N/I. Do đó, các bộ nhớ thứ nhất 455, thứ hai 456 và đến thứ ba 457 có thể được khởi tạo để có các giá trị mong muốn.

Fig.13 minh họa bộ mã hóa dạng mắt cáo trong kết cấu đơn giản hóa. Trên Fig.13, bộ mã hóa dạng mắt cáo có thể bao gồm các bộ cộng thứ nhất 453 và thứ hai 454, các bộ nhớ thứ nhất 455, thứ hai 456 và thứ ba 457, và các bộ dồn kênh thứ ba 459-3 và thứ tư 459-4. Do đó, các bộ nhớ thứ nhất 455, thứ hai 456 và thứ ba 457 có thể được khởi tạo theo các giá trị I_X1 và I_X2 lần lượt được đưa vào các bộ dồn kênh thứ ba 459-3 và thứ tư 459-4. Tức là, trên Fig.13, các giá trị I_X2 và I_X1 được đưa vào bộ nhớ thứ nhất 455 và bộ nhớ thứ hai 456 như vậy và trở thành các giá trị của bộ nhớ thứ nhất 455 và bộ nhớ thứ hai 456.

Quay trở lại Fig.4, đồng bộ trường và đồng bộ phân đoạn được thêm vào luồng được mã hóa dạng mắt cáo bởi bộ phận mã hóa dạng mắt cáo 450 nhờ bộ dồn kênh đồng bộ 470.

Như đã mô tả ở trên, nếu bộ tiền xử lý dữ liệu 100 đặt dữ liệu di động vào trong các gói được cấp phát cho dữ liệu thường hiện có và sử dụng dữ liệu di động, thì bộ truyền phát rộng dạng số có thể thông báo cho bộ thu về sự có mặt của dữ liệu di động mới. Sự có mặt của dữ liệu di động mới có thể được thông báo theo các cách khác nhau. Một trong các cách khác nhau là sử dụng đồng bộ trường. Cách này sẽ được mô tả chi tiết sau đây.

Bộ phận chèn dẫn hướng 480 chèn phần dẫn hướng vào luồng vận chuyển được xử lý bởi bộ dồn kênh đồng bộ 470, và bộ điều biến 8-VSB 490 thực hiện điều biến theo phương pháp điều biến 8-VSB. Bộ chuyển đổi lên RF 495 chuyển đổi luồng đã được điều biến thành tín hiệu dải RF trên để truyền, và truyền tín hiệu đã được chuyển đổi qua anten.

Như đã mô tả ở trên, luồng vận chuyển có thể được truyền tới bộ thu với dữ liệu thường, dữ liệu di động, và dữ liệu đã biết có trong đó.

Fig.14 là hình vẽ giải thích khung dữ liệu di động của luồng vận chuyển, tức là kết cấu đơn vị của khung M/H. Theo phần (a) trên Fig.14, một khung M/H dài 968ms (mili giây) theo đơn vị thời gian, và được chia thành 5 khung phụ như được thể hiện trên phần (b) trên Fig.14. Một khung phụ có thể có đơn vị thời gian là 193,6ms. Ngoài ra, như được

thể hiện trên phần (c) trên Fig.14, mỗi khung phụ có thể được chia thành 16 rãnh. Mỗi rãnh có thể có đơn vị thời gian là 12,1ms và có thể bao gồm tổng cộng 156 gói luồng vận chuyển. Như đã mô tả ở trên, tổng cộng 38 gói luồng vận chuyển được cấp phát cho dữ liệu thường và 118 gói được cấp phát cho dữ liệu di động. Tức là, một nhóm M/H bao gồm 118 gói.

Ở trạng thái này, bộ tiền xử lý dữ liệu 100 có thể đặt dữ liệu di động và dữ liệu đã biết trong các gói được cấp phát cho dữ liệu thường, nhờ đó cải thiện hiệu suất truyền dữ liệu di động và cải thiện hiệu quả thu nhận.

Các phương án ví dụ khác nhau của luồng vận chuyển được thay đổi

Các hình vẽ từ Fig.15 đến Fig.21 là các hình vẽ minh họa cấu hình luồng vận chuyển theo các phương án ví dụ khác nhau.

Fig.15 minh họa kết cấu cải biến đơn giản nhất, tức là kết cấu luồng được đan xen bởi dữ liệu di động đang được đặt trong các gói được cấp phát cho dữ liệu thường, tức là vùng thứ hai. Trong luồng trên Fig.15, dữ liệu đã biết có thể được đặt trong vùng thứ hai cùng với dữ liệu di động.

Do đó, một phần mà không được sử dụng cho dữ liệu di động trong ATSC-MH trong lĩnh vực, tức là 38 gói có thể được sử dụng cho dữ liệu di động. Do vùng thứ hai được sử dụng độc lập với vùng dữ liệu di động hiện có (tức là, vùng thứ nhất), nên có thể cung cấp thêm một hoặc nhiều dịch vụ. Nếu dữ liệu di động mới được sử dụng cho cùng dịch vụ như của dữ liệu di động hiện có, thì hiệu suất truyền dữ liệu có thể được cải thiện thêm.

Mặt khác, nếu dữ liệu di động mới và dữ liệu đã biết là để được truyền như được thể hiện trên Fig.15, thì sự có mặt của dữ liệu di động mới và dữ liệu đã biết và các vị trí của dữ liệu di động mới và dữ liệu đã biết có thể được thông báo cho bộ thu bằng cách sử dụng dữ liệu truyền tín hiệu hoặc đồng bộ trường.

Dữ liệu di động và dữ liệu đã biết có thể được đặt nhờ bộ tiền xử lý dữ liệu 100. Cụ thể, bộ định dạng nhóm 130 của bộ tiền xử lý dữ liệu 100 có thể đặt dữ liệu di động và dữ liệu đã biết trong 38 gói.

Có thể thấy từ Fig.15 là dữ liệu đã biết có định dạng 6 chuỗi hướng dẫn dài được đặt trong vùng thân, tại đó tập hợp dữ liệu di động hiện có. Còn có thể thấy được là dữ liệu

truyền tín hiệu được đặt giữa các chuỗi hướng dẫn dài thứ nhất và thứ hai để đạt được độ mạnh với lỗi của dữ liệu truyền tín hiệu. Mặt khác, dữ liệu đã biết có thể được đặt trong các gói được cấp phát cho dữ liệu thường theo định dạng phân bố cũng như định dạng chuỗi hướng dẫn dài.

Trên Fig.15, vùng gạch bóng được biểu diễn bởi số chỉ dẫn 1510 biểu thị phần đầu MPEG, vùng gạch bóng được biểu diễn bởi số chỉ dẫn 1520 biểu thị vùng chặn lẻ RS, vùng gạch bóng được biểu diễn bởi số chỉ dẫn 1530 biểu thị vùng giả, vùng gạch bóng được biểu diễn bởi số chỉ dẫn 1540 biểu thị dữ liệu truyền tín hiệu, và vùng gạch bóng được biểu diễn bởi số chỉ dẫn 1550 biểu thị dữ liệu khởi tạo. Trên Fig.15, dữ liệu khởi tạo được đặt trước khi dữ liệu đã biết xuất hiện. Số chỉ dẫn 1400 biểu thị dữ liệu M/H rãnh thứ N-1, số chỉ dẫn 1500 biểu thị dữ liệu M/H rãnh thứ N, và số chỉ dẫn 1600 biểu thị dữ liệu M/H rãnh thứ N+1.

Fig.16 minh họa cấu hình luồng vận chuyển để truyền dữ liệu di động và dữ liệu đã biết bằng cách sử dụng các gói được cấp phát cho dữ liệu thường, tức là vùng thứ hai, và một phần của vùng được cấp phát thứ nhất cho dữ liệu di động hiện có.

Trên Fig.16, trong vùng A, tức là vùng thân tại đó tập hợp dữ liệu di động hiện có, dữ liệu đã biết có định dạng 6 chuỗi hướng dẫn dài được đặt. Ngoài ra, trong vùng B, dữ liệu đã biết được đặt trong định dạng chuỗi hướng dẫn dài. Dữ liệu đã biết có trong một số gói trong 118 gói được cấp phát cho dữ liệu di động hiện có cũng như 38 gói, để được đặt trong vùng B theo định dạng chuỗi hướng dẫn dài. Dữ liệu di động mới được đặt trong vùng còn lại của 38 gói mà không bao gồm dữ liệu đã biết. Do đó, khả năng hiệu chỉnh lỗi của vùng B có thể được cải thiện.

Khi dữ liệu đã biết mới được thêm vào một phần của vùng cho dữ liệu di động hiện có, có thể thêm thông tin về vị trí của dữ liệu đã biết mới vào dữ liệu truyền tín hiệu hiện có nhằm tương thích với bộ thu dữ liệu di động hiện có, hoặc để xử lý phần đầu của gói di động hiện có mà dữ liệu đã biết mới được chèn vào có định dạng không thể nhận ra được bởi bộ thu dữ liệu di động hiện có, ví dụ, định dạng gói rỗng. Do đó, do bộ thu dữ liệu di động hiện có không nhận ra dữ liệu đã biết mới được thêm, nên không xảy ra trục trặc.

Fig.17 minh họa kết cấu luồng, trong đó ít nhất một trong số dữ liệu di động và dữ liệu đã biết được đặt ở phần đầu MPEG, phần chặn lẻ RS, ít nhất một phần của phần giả,

và dữ liệu MH hiện có. Trong trường hợp này, nhiều dữ liệu di động mới có thể được đặt theo vị trí.

Tức là, so với Fig.15, Fig.17 biểu thị là dữ liệu di động mới và dữ liệu đã biết mới được tạo ra ở phần đầu MPEG, phần chẵn lẻ RS, và một phần của phần giả. Dữ liệu di động được chèn vào trong các phần này và dữ liệu di động được chèn vào trong các gói dữ liệu thường có thể là dữ liệu khác hoặc là cùng dữ liệu.

Dữ liệu di động mới có thể được đặt trong vùng dữ liệu di động hiện có ngoài các phần này.

Nếu luồng được tạo cấu hình như được thể hiện trên Fig.17, thì hiệu suất truyền dữ liệu di động và dữ liệu đã biết có thể được cải thiện hơn nữa so với Fig.15 và Fig.16. Cụ thể, có thể cung cấp nhiều dịch vụ dữ liệu di động.

Nếu luồng được tạo cấu hình như được thể hiện trên Fig.17, dữ liệu truyền tín hiệu mới có trong vùng dữ liệu di động mới bằng cách sử dụng dữ liệu truyền tín hiệu hiện có và đồng bộ trường, sao cho nó được thông báo là liệu dữ liệu di động mới có được bao gồm hay không.

Fig.18 minh họa kết cấu luồng, trong đó dữ liệu di động mới và dữ liệu đã biết được đặt trong vùng B, tức là vùng thứ nhất tương ứng với vùng dịch vụ thứ cấp, cũng như vùng thứ hai.

Như được thể hiện trên Fig.18, toàn bộ luồng được chia thành vùng dịch vụ sơ cấp và vùng dịch vụ thứ cấp. Vùng dịch vụ sơ cấp có thể được gọi là vùng thân và vùng dịch vụ thứ cấp có thể được gọi là vùng đầu/đuôi. Như đã mô tả ở trên, do vùng đầu/đuôi không bao gồm dữ liệu đã biết và dữ liệu của các rãnh khác nhau cùng tồn tại, nên hiệu suất của vùng đầu/đuôi giảm đi so với hiệu suất của vùng thân. Theo đó, dữ liệu đã biết có thể được đặt và được sử dụng ở phần này cùng với dữ liệu di động mới. Dữ liệu đã biết có thể được đặt trong định dạng chuỗi hướng dẫn dài như trong vùng thân. Tuy nhiên, điều này không được coi là giới hạn. Dữ liệu đã biết có thể được đặt dưới định dạng phân bố hoặc có thể được đặt trong cả định dạng chuỗi hướng dẫn dài và định dạng phân bố.

Khi phần dữ liệu di động hiện có được sử dụng làm vùng dữ liệu di động mới, phần đầu của gói của một phần của vùng dữ liệu di động hiện có bao gồm dữ liệu di động mới hoặc dữ liệu đã biết được tạo cấu hình theo định dạng mà không thể nhận ra được bởi

bộ thu hiện có, sao cho có thể duy trì tính tương thích với bộ thu theo chuẩn ATSC-MH hiện có.

Ngoài ra, sự kiện trên có thể được thông báo thông qua dữ liệu truyền tín hiệu hiện có hoặc dữ liệu truyền tín hiệu mới.

Fig.19 minh họa một ví dụ về luồng vận chuyển để truyền dữ liệu di động mới và dữ liệu đã biết bằng cách sử dụng tất cả vùng dữ liệu thường hiện có, phần đầu MPEG, vùng chặn lẻ RS, ít nhất một phần của phần giả của dữ liệu di động hiện có, và vùng dữ liệu di động hiện có. Fig.17 minh họa trường hợp trong đó dữ liệu di động mới khác với dữ liệu di động mới được đặt trong vùng dữ liệu thường được truyền trong các vùng này, nhưng Fig.19 minh họa trường hợp, trong đó dữ liệu di động mới được truyền bằng cách sử dụng tất cả vùng dữ liệu thường và các vùng này.

Fig.20 minh họa luồng vận chuyển để truyền dữ liệu di động mới và dữ liệu đã biết bằng cách sử dụng tất cả toàn bộ vùng B, vùng dữ liệu thường, phần đầu MPEG, vùng chặn lẻ RS, và ít nhất một phần của phần giả của dữ liệu di động hiện có.

Trong trường hợp này, một phần bao gồm dữ liệu di động mới và dữ liệu đã biết có thể được xử lý sao cho phần này không thể được nhận ra nhằm tương thích với bộ thu hiện có.

Fig.21 minh họa cấu hình luồng vận chuyển, trong đó phần giả của vùng được sử dụng bởi dữ liệu di động hiện có được thay thế bởi phần chặn lẻ hoặc vùng dữ liệu di động mới, và dữ liệu di động và dữ liệu đã biết được đặt bằng cách sử dụng phần giả được thay thế và vùng dữ liệu thường. Trên Fig.21, phần giả của rãnh thứ N-1 và phần giả của rãnh thứ N được minh họa.

Như đã mô tả ở trên, các hình vẽ từ Fig.15 đến Fig.21 minh họa kết cấu luồng sau khi đan xen. Bộ tiền xử lý dữ liệu 100 đặt dữ liệu di động và dữ liệu đã biết ở các vị trí thích hợp để có kết cấu luồng được thể hiện trên các hình vẽ từ Fig.15 đến Fig.21 sau khi đan xen.

Cụ thể, bộ tiền xử lý dữ liệu 100 đặt gói dữ liệu di động trong vùng dữ liệu thường, tức là trong 38 gói trên kết cấu luồng được thể hiện trên phần (A) của Fig.1 theo mẫu định trước. Trong trường hợp này, dữ liệu di động có thể được đặt trong toàn bộ phần tải của gói hoặc có thể nằm trong một số vùng trong gói. Ngoài ra, dữ liệu di động có thể

được đặt trong vùng được đặt ở phần đầu hoặc phần đuôi của vùng di động hiện có sau khi đan xen, cũng như vùng dữ liệu thường.

Dữ liệu đã biết có thể được đặt trong mỗi gói dữ liệu di động hoặc gói dữ liệu thường. Trong trường hợp này, dữ liệu đã biết có thể được đặt liên tục theo phương thẳng đứng hoặc ở khoảng định trước như được thể hiện trên phần (A) của Fig.1, sao cho dữ liệu đã biết có định dạng của chuỗi hướng dẫn dài hoặc chuỗi hướng dẫn dài tương tự theo phương nằm ngang sau khi đan xen.

Dữ liệu đã biết có thể được đặt dưới định dạng phân bố ngoài định dạng chuỗi hướng dẫn dài như đã mô tả ở trên. Sau đây, các ví dụ khác nhau về việc đặt mẫu dữ liệu đã biết sẽ được giải thích.

Đặt dữ liệu đã biết

Như đã mô tả ở trên, dữ liệu đã biết được đặt ở vị trí thích hợp nhờ bộ định dạng nhóm 130 của bộ tiền xử lý dữ liệu 100 và sau đó được đan xen nhờ bộ đan xen 430 của bộ phận kích thích 400 cùng với luồng. Các hình vẽ từ Fig.22 đến Fig.28 là các hình vẽ giải thích phương pháp đặt dữ liệu đã biết theo các phương án ví dụ khác nhau.

Fig.22 minh họa phần thân trong đó dữ liệu đã biết được phân bố được đặt cùng với chuỗi hướng dẫn dài hiện có, và vùng đầu/đuôi, trong đó dữ liệu đã biết được đặt thêm ở phần dạng nón. Như đã mô tả ở trên, dữ liệu đã biết mới được thêm trong khi dữ liệu đã biết hiện có được duy trì như vậy sao cho có thể cải thiện sự đồng bộ và hiệu suất đánh giá kênh và hiệu suất cân bằng của bộ thu.

Việc đặt dữ liệu đã biết được thể hiện trên Fig.22 được thực hiện nhờ bộ định dạng nhóm 130 như đã mô tả ở trên. Bộ định dạng nhóm 130 có thể xác định vị trí chèn của dữ liệu đã biết có tính đến luật đan xen của bộ đan xen 430. Luật đan xen có thể khác nhau tùy theo các phương án ví dụ khác nhau. Tuy nhiên, nếu luật đan xen đã được biết, thì bộ định dạng nhóm 130 có thể xác định vị trí của dữ liệu đã biết một cách phù hợp. Ví dụ, nếu dữ liệu đã biết có kích thước định trước được chèn vào trong một phần của phần tải hoặc trường được bố trí riêng trong mỗi 4 gói, thì có thể thu được dữ liệu đã biết được phân bố theo mẫu đều thông qua đan xen.

Fig.23 minh họa kết cấu luồng thể hiện một ví dụ về một phương pháp chèn dữ liệu đã biết khác.

Trên Fig.23, dữ liệu đã biết được phân bố không được đặt trong vùng hình nón và chỉ được đặt trong vùng thân cùng với chuỗi hướng dẫn dài.

Fig.24 minh họa kết cấu luồng, trong đó chiều dài của chuỗi hướng dẫn dài được giảm đi so với chiều dài trên Fig.23 và dữ liệu đã biết được phân bố được đặt nhiều như số lượng giảm đi của các chuỗi. Do đó, hiệu suất theo dõi Doppler có thể được cải thiện, mặc dù hiệu suất dữ liệu được duy trì như cũ.

Fig.25 minh họa kết cấu luồng thể hiện một ví dụ về một phương pháp khác nữa để chèn dữ liệu đã biết.

Trên Fig.25, chỉ chuỗi thứ nhất trong số 6 chuỗi hướng dẫn dài trong vùng thân được giữ như vậy, và các chuỗi còn lại được thay thế bởi dữ liệu đã biết được phân bố. Do đó, việc đồng bộ ban đầu và hiệu suất đánh giá kênh có thể được duy trì nhờ chuỗi hướng dẫn dài thứ nhất bắt đầu vùng thân, và ngoài ra, hiệu suất theo dõi Doppler có thể được cải thiện.

Fig.26 minh họa kết cấu luồng thể hiện một ví dụ về một phương pháp khác nữa để chèn dữ liệu đã biết. Trên Fig.26, chuỗi thứ hai trong số 6 chuỗi hướng dẫn dài được thay thế bởi dữ liệu đã biết được phân bố.

Fig.27 minh họa kết cấu luồng trên Fig.26, trong đó dữ liệu đã biết được đặt lại theo định dạng phân bố được đặt luân phiên cùng với dữ liệu truyền tín hiệu.

Fig.28 minh họa kết cấu luồng, trong đó dữ liệu đã biết được phân bố được thêm vào vùng đuôi cũng như vùng đầu.

Như đã mô tả ở trên, dữ liệu đã biết có thể được đặt trong các định dạng khác nhau.

Nếu dữ liệu di động mới được cấp phát cho các gói được cấp phát cho dữ liệu thường, mẫu cấp phát có thể được thay đổi khác nhau. Sau đây, kết cấu của luồng vận chuyển bao gồm dữ liệu di động được đặt theo các cách khác nhau theo chế độ sẽ được giải thích.

Đặt dữ liệu di động

Bộ tiền xử lý dữ liệu 100 nhận diện điều kiện thiết lập của chế độ khung. Chế độ khung có thể được thiết lập khác nhau. Ví dụ, chế độ khung có thể bao gồm chế độ khung

thứ nhất, trong đó các gói được cấp phát cho dữ liệu thường được sử dụng cho dữ liệu thường như vậy và chỉ có các gói được cấp phát cho dữ liệu di động hiện có được sử dụng cho dữ liệu di động, và chế độ khung thứ hai, trong đó ít nhất một số gói được cấp phát cho dữ liệu thường cũng được sử dụng cho dữ liệu di động. Chế độ khung có thể được thiết lập tùy ý có tính đến mục đích của nhà cung cấp truyền phát rộng dạng số và môi trường truyền và nhận.

Nếu chế độ khung thứ nhất, trong đó dữ liệu thường được đặt trong tất cả các gói được cấp phát cho dữ liệu thường được thiết lập, thì bộ tiền xử lý dữ liệu 100 chỉ đặt dữ liệu di động trong các gói được cấp phát cho dữ liệu di động theo cùng một cách như của chuẩn ATSC-MH trong lĩnh vực.

Mặt khác, nếu khung thứ hai được thiết lập, thì bộ tiền xử lý dữ liệu 100 xác định điều kiện thiết lập của chế độ một lần nữa. Chế độ biểu thị dữ liệu di động được đặt theo mẫu nào trong các gói được cấp phát cho dữ liệu thường, tức là vùng thứ hai hoặc dữ liệu di động được đặt trong bao nhiêu gói, và các chế độ khác nhau được đưa ra theo một phương án ví dụ.

Cụ thể, chế độ có thể được thiết lập theo một chế độ, trong đó dữ liệu di động chỉ được đặt trong một số gói được cấp phát cho dữ liệu thường, chế độ, trong đó dữ liệu di động được đặt trong tất cả các gói được cấp phát cho dữ liệu thường, và chế độ không tương thích, trong đó dữ liệu di động được đặt trong tất cả các gói được cấp phát cho dữ liệu thường và dữ liệu di động cũng được đặt trong vùng chặn lẻ RS và vùng phân đầu, chúng được đưa ra nhằm tương thích với bộ thu để nhận dữ liệu thường. Trong trường hợp này, chế độ, trong đó dữ liệu di động chỉ được đặt trong một số gói có thể được chia thành chế độ, trong đó vùng dữ liệu của một số gói, tức là toàn bộ vùng tải được sử dụng cho dữ liệu di động và chế độ, trong đó chỉ có một phần của vùng tải được sử dụng cho dữ liệu di động.

Cụ thể, nếu 38 gói tương ứng với vùng thứ hai được cấp phát cho dữ liệu thường, thì chế độ bao gồm:

- 1) chế độ thứ nhất, trong đó dữ liệu di động mới được đặt trong 11 gói trong số 38 gói được cấp phát cho dữ liệu thường;
- 2) chế độ thứ hai, trong đó dữ liệu di động mới được đặt trong 20 gói trong số 38 gói được cấp phát cho dữ liệu thường;

3) chế độ thứ ba, trong đó dữ liệu di động mới được đặt trong 29 gói trong số 38 gói được cấp phát cho dữ liệu thường;

4) chế độ thứ tư, trong đó dữ liệu di động mới được đặt trong tất cả 38 gói được cấp phát cho dữ liệu thường; và

5) chế độ thứ năm, trong đó dữ liệu di động mới được đặt trong tất cả 38 gói được cấp phát cho dữ liệu thường và vùng tương ứng với phần đầu MPEG và phần chẵn lẻ trong vùng được cấp phát cho dữ liệu di động hiện có.

Như đã mô tả ở trên, chế độ thứ năm có thể được gọi chế độ không tương thích và các chế độ thứ nhất đến thứ tư có thể được gọi chế độ tương thích. Loại chế độ tương thích và số lượng các gói trong mỗi chế độ có thể được thay đổi khác nhau.

Fig.29 minh họa kết cấu luồng, trong đó dữ liệu di động và dữ liệu đã biết được đặt nhờ bộ định dạng nhóm 130 theo chế độ thứ nhất theo một phương án thực hiện, trong đó dữ liệu di động mới được truyền bằng cách sử dụng vùng thứ hai và vùng đầu/đuôi.

Trên Fig.29, dữ liệu di động mới 2950 và dữ liệu đã biết 2960 được đặt trong vùng thứ hai theo mẫu định trước, và ngoài ra, dữ liệu di động mới và dữ liệu đã biết được đặt ở phần 2950 tương ứng với vùng đầu/đuôi ngoài vùng thứ hai.

Có thể thấy rằng phần đầu MPEG 2910, dữ liệu đã biết 2920, dữ liệu truyền tín hiệu 2930, dữ liệu di động hiện có 2940, và phần giả 2970 được bố trí trên luồng theo phương thẳng đứng. Ở trạng thái này, khoảng trống trong vùng thứ hai được điền dữ liệu thường và sau đó kết cấu luồng được thể hiện trên Fig.30 được tạo ra nhờ mã hóa và đan xen.

Fig.30 minh họa kết cấu luồng ở trạng thái được đan xen ở chế độ thứ nhất.

Trên Fig.30, dữ liệu di động mới 3010 và dữ liệu đã biết 3030 được đặt trong một phần của vùng gói được cấp phát cho dữ liệu thường. Cụ thể, dữ liệu đã biết được bố trí không liên tục trong vùng thứ hai, theo đó tạo ra định dạng chuỗi hướng dẫn dài tương tự với chuỗi hướng dẫn dài của vùng thân.

Dữ liệu di động 2950 được đặt ở phần tương ứng với vùng đầu/đuôi trên Fig.29 tương ứng với dữ liệu di động 3020 được đặt trong vùng đầu/đuôi trên Fig.30, và dữ liệu đã biết 2955 được đặt cùng với dữ liệu di động 2950 tạo ra dữ liệu đã biết 3030 có định

dạng chuỗi hướng dẫn dài tương tự cùng với dữ liệu đã biết trong vùng thứ hai trên Fig.30.

Fig.31 minh họa kết cấu luồng, trong đó dữ liệu di động và dữ liệu đã biết được đặt nhờ bộ định dạng nhóm 130 theo chế độ thứ hai theo một phương án ví dụ, trong đó dữ liệu di động mới được truyền bằng cách sử dụng vùng thứ hai và vùng đầu/đuôi.

Fig.31 minh họa tỷ lệ tăng lên của dữ liệu di động nằm trong vùng thứ hai so với Fig.29. Có thể thấy rằng, một phần được chiếm bởi dữ liệu di động và dữ liệu đã biết tăng lên trên Fig.31 so với Fig.29.

Fig.32 minh họa luồng trên Fig.31 ở trạng thái được đan xen. Trên Fig.32, dữ liệu đã biết trong vùng thứ hai tạo ra chuỗi hướng dẫn dài tương tự với mật độ cao hơn trong dữ liệu đã biết trong vùng thứ hai trên Fig.30.

Fig.33 minh họa kết cấu luồng, trong đó dữ liệu di động và dữ liệu đã biết được đặt nhờ bộ định dạng nhóm 130 theo chế độ thứ ba theo một phương án ví dụ, trong đó dữ liệu di động mới được truyền bằng cách sử dụng vùng thứ hai và vùng đầu/đuôi. Fig.34 minh họa luồng trên Fig.33 ở trạng thái được đan xen.

Không có dấu hiệu bổ sung trên Fig.33 và Fig.34 trừ dữ liệu di động và dữ liệu đã biết được đặt dày đặc hơn so với các chế độ 1 và 2, và bởi vậy mô tả chi tiết được bỏ qua.

Fig.35 minh họa kết cấu luồng ở chế độ thứ tư mà sử dụng toàn bộ vùng dữ liệu thường theo một phương án ví dụ, trong đó tất cả các gói được cấp phát cho dữ liệu thường và vùng gói được cấp phát cho dữ liệu di động hiện có và tương ứng với vùng đầu/đuôi được sử dụng.

Trên Fig.35, dữ liệu đã biết được bố trí trong vùng thứ hai và vùng xung quanh của vùng thứ hai theo phương thẳng đứng, và vùng khác được điền dữ liệu di động mới.

Fig.36 minh họa luồng trên Fig.35 ở trạng thái được đan xen. Trên Fig.36, vùng đầu/đuôi và toàn bộ vùng dữ liệu thường được điền dữ liệu di động mới và dữ liệu đã biết, và cụ thể, dữ liệu đã biết được bố trí theo định dạng chuỗi hướng dẫn dài.

Dữ liệu đã biết được chèn lặp lại vào trong các vùng này từng chút một theo nhiều chu kỳ mẫu, sao cho dữ liệu đã biết trở thành dữ liệu đã biết được phân bố sau khi đan xen.

Fig.37 là hình vẽ giải thích phương pháp chèn dữ liệu di động mới vào trong vùng thứ hai, tức là các gói được cấp phát cho dữ liệu thường (ví dụ, 38 gói) ở các chế độ khác nhau. Để giải thích thuận tiện, sau đây, dữ liệu di động mới được gọi là dữ liệu 1.1 di động ATSC (hoặc dữ liệu phiên bản 1.1) và dữ liệu di động hiện có được gọi là dữ liệu 1.0 di động ATSC (hoặc dữ liệu phiên bản 1.0).

Trước tiên, a) ở chế độ thứ nhất, dữ liệu phiên bản 1.1 tương ứng được đặt trong gói khởi tạo và gói cuối cùng, và một gói 1.1 và 3 gói dữ liệu thường được chèn lặp lại vào trong các gói giữa gói khởi tạo và gói cuối cùng. Do đó, có thể sử dụng tổng cộng 11 gói để truyền dữ liệu phiên bản 1.1, tức là dữ liệu di động mới.

Tiếp theo, b) ở chế độ thứ hai, dữ liệu phiên bản 1.1 tương tự được đặt trong gói khởi tạo và gói cuối cùng, và một gói 1.1 và một gói dữ liệu thường được chèn luân phiên vào trong các gói giữa gói khởi tạo và gói cuối cùng. Do đó, có thể sử dụng tổng cộng 20 gói để truyền dữ liệu phiên bản 1.1, tức là dữ liệu di động mới.

Tiếp theo, c) ở chế độ thứ ba, dữ liệu phiên bản 1.1 tương tự được đặt trong gói khởi tạo và gói cuối cùng và ba gói 1.1 và một gói dữ liệu thường được chèn lặp lại vào trong các gói giữa gói khởi tạo và gói cuối cùng.

Tiếp theo, d) ở chế độ thứ tư, tất cả các gói tương ứng với vùng thứ hai được sử dụng để truyền dữ liệu phiên bản 1.1.

Chế độ thứ tư có thể được thực hiện bởi chế độ tương thích, trong đó chỉ có các gói tương ứng với vùng thứ hai được sử dụng để truyền dữ liệu phiên bản 1.1 hoặc chế độ không tương thích, trong đó không chỉ các gói tương ứng với vùng thứ hai mà còn phần đầu MPEG và vùng chẵn lẻ được đưa ra để tương thích với bộ thu dữ liệu thường được điền dữ liệu phiên bản 1.1. Chế độ không tương thích có thể được đưa ra làm chế độ thứ năm riêng.

Các chế độ thứ nhất đến thứ tư có thể tương ứng với việc sử dụng 1/4, 2/4, 3/4, và 4/4 trong toàn bộ các gói của vùng thứ hai để truyền dữ liệu di động, theo cách tương ứng. Tuy nhiên, tổng số lượng các gói là 38, không phải là bội của 4, và do đó một số gói được cố định làm gói để truyền dữ liệu di động mới hoặc dữ liệu thường và các gói còn lại được phân loại theo các tỷ lệ trên, sao cho các chế độ được phân loại. Tức là, như được giải thích trong mục a), b), và c) ở trên, 36 gói, là 38 gói trừ số lượng gói định trước, tức là 2 gói, có thể bao gồm dữ liệu 1.1 theo tỷ lệ là 1/4, 2/4, và 3/4.

Fig.38 là hình vẽ giải thích mẫu đặt dữ liệu di động ở chế độ khác.

Trên Fig.38, hai dữ liệu phiên bản 1.1 được đặt trong các gói trung gian, chúng được đặt ở giữa luồng của tất cả các gói trong vùng thứ hai, tức là 38 gói, và dữ liệu phiên bản 1.1 và dữ liệu thường được đặt trong các gói khác theo tỷ lệ được xác định theo mỗi chế độ.

Tức là, a) ở chế độ thứ nhất, đối với các gói trừ hai gói ở phần giữa, 3 gói dữ liệu thường và một gói dữ liệu phiên bản 1.1 được đặt lặp lại ở phần trên và một gói dữ liệu phiên bản 1.1 và 3 gói dữ liệu thường được đặt lặp lại ở phần dưới.

b) Ở chế độ thứ hai, đối với các gói trừ hai gói ở phần giữa, hai gói dữ liệu thường và hai gói dữ liệu phiên bản 1.1 được đặt lặp lại ở phần trên và hai gói dữ liệu phiên bản 1.1 và hai gói dữ liệu thường được đặt lặp lại ở phần dưới.

c) Ở chế độ thứ ba, đối với các gói trừ hai gói ở phần giữa, một gói dữ liệu thường và ba gói dữ liệu phiên bản 1.1 được đặt lặp lại ở phần trên và ba gói dữ liệu phiên bản 1.1 và một gói dữ liệu thường được đặt lặp lại ở phần dưới.

d) Ở chế độ thứ tư, dữ liệu phiên bản 1.1 được đặt trong tất cả các gói. Nó giống như ở chế độ thứ tư trên Fig.37.

Tiếp theo, Fig.39 minh họa một phương án ví dụ trong đó dữ liệu phiên bản 1.1 được đặt từ gói giữa đến các gói trên và dưới một cách tuần tự có dựa vào vị trí trên luồng.

Tức là, a) ở chế độ thứ nhất trên Fig.39, 11 gói trong số toàn bộ các gói của vùng thứ hai được đặt từ phần giữa tuần tự hướng lên trên và xuống dưới.

b) Ở chế độ thứ hai trên Fig.39, tổng cộng 20 gói được đặt từ phần giữa tuần tự hướng lên trên và xuống dưới. c) Ở chế độ thứ ba trên Fig.39, 30 gói tổng cộng được đặt từ phần giữa tuần tự hướng lên trên và xuống dưới. d) Ở chế độ thứ tư trên Fig.39, tất cả các gói được điền dữ liệu phiên bản 1.1.

Fig.40 minh họa kết cấu luồng theo một phương án ví dụ, trong đó dữ liệu di động được đặt từ các gói trên và dưới đến phần giữa một cách tuần tự theo thứ tự ngược với thứ tự trên Fig.39. Ngoài ra, trên Fig.40, số lượng các gói dữ liệu di động mới ở các chế độ thứ nhất đến thứ tư được thiết lập khác với các chế độ của các phương án thực hiện nêu trên.

Tức là, a) ở chế độ thứ nhất trên Fig.40, bốn gói dữ liệu phiên bản 1.1 được đặt từ gói trên hướng xuống dưới và bốn gói dữ liệu phiên bản 1.1 được đặt từ gói dưới hướng lên trên. Tức là, tổng cộng tám gói dữ liệu phiên bản 1.1 được đặt.

b) Ở chế độ thứ hai, tám gói dữ liệu phiên bản 1.1 được đặt từ gói trên hướng xuống dưới và tám gói dữ liệu phiên bản 1.1 được đặt từ gói dưới hướng lên trên. Tức là, tổng cộng mười sáu gói dữ liệu phiên bản 1.1 được đặt.

c) Ở chế độ thứ ba, mười hai gói dữ liệu phiên bản 1.1 được đặt từ gói trên hướng xuống dưới và mười hai gói dữ liệu phiên bản 1.1 được đặt từ gói dưới hướng lên trên. Tức là, tổng cộng 24 gói dữ liệu phiên bản 1.1 được đặt.

Các gói còn lại được điền dữ liệu thường. Ở chế độ thứ tư, mẫu gói giống như trên Fig.37, Fig.38, và Fig.39 và được bỏ qua trên Fig.40.

Mặc dù mẫu chèn dữ liệu đã biết không được minh họa trên các hình vẽ từ Fig.37 đến Fig.40, nhưng dữ liệu đã biết có thể được chèn vào trong một số vùng của cùng gói như của dữ liệu di động hoặc có thể được chèn vào trong một số vùng của gói riêng hoặc toàn bộ vùng tải. Phương pháp chèn dữ liệu đã biết đã được mô tả ở trên và theo đó được bỏ qua trên các hình vẽ từ Fig.37 đến Fig.40.

Ở chế độ thứ năm, tức là ở chế độ không tương thích, dữ liệu di động mới được điền thêm vào trong vùng chặn lẻ RS và vùng phần đầu trong vùng dữ liệu di động hiện có khác vùng dữ liệu thường, và theo đó chế độ thứ năm được bỏ qua trên các hình vẽ từ Fig.37 đến Fig.40.

Mặc dù chế độ thứ năm được mô tả ở trên có thể là chế độ mới tách riêng với chế độ thứ tư, nhưng chế độ thứ tư hoặc chế độ thứ năm có thể được kết hợp với các chế độ thứ nhất đến thứ ba, sao cho có thể thực hiện tổng số bốn chế độ.

Trên các hình vẽ từ Fig.37 đến Fig.40, phương pháp chèn dữ liệu di động mới vào trong vùng thứ hai, tức là các gói được cấp phát cho dữ liệu thường (ví dụ, 38 gói) ở các chế độ khác nhau được mô tả. Phương pháp đặt dữ liệu di động mới trong các gói được cấp phát cho dữ liệu thường theo chế độ định trước là khác nhau theo các chế độ thứ nhất đến thứ tư như đã mô tả ở trên trên các hình vẽ từ Fig.37 đến Fig.40. Chế độ thứ tư có thể được thực hiện theo chế độ, trong đó chỉ 38 gói được điền dữ liệu di động mới hoặc chế độ, trong đó dữ liệu di động mới được đặt trong vùng chặn lẻ RS và vùng phần đầu ngoài

38 gói. Ngoài ra, như đã mô tả ở trên, chế độ có thể bao gồm tất cả các chế độ từ thứ nhất đến thứ năm.

Nếu chế độ để xác định có bao nhiêu gói trong số 38 gói được cấp phát cho dữ liệu di động mới và cách thức các khối được tạo cấu hình trong nhóm M/H là chế độ có thể mở rộng, a) chế độ có thể mở rộng 00, b) chế độ có thể mở rộng 01, c) chế độ có thể mở rộng 10, và d) chế độ có thể mở rộng 11 được xác định bằng cách sử dụng trường tín hiệu gồm hai bit trên Fig.37. Kể cả nếu tất cả 38 gói được cấp phát cho dữ liệu di động mới như ở phần d) trên Fig.37, thì 118 gói, là vùng dữ liệu di động hiện có, và 38 gói mà dữ liệu di động mới được cấp phát cho có thể tạo ra nhóm M/H đơn.

Trong trường hợp này, hai chế độ có thể mở rộng được xác định theo cách thức các khối được tạo cấu hình trong nhóm M/H. Tùy theo liệu toàn bộ tốc độ truyền dữ liệu 19,4Mbps có được cấp phát cho dữ liệu di động hay không, nhóm M/H có kết cấu khối khác có thể được tạo ra kể cả nếu tất cả 38 gói trong một rãnh được cấp phát cho dữ liệu di động như được thể hiện trên Fig.37.

Nếu toàn bộ tốc độ truyền dữ liệu 19,4Mbps được cấp phát cho dữ liệu di động, thì tốc độ dữ liệu thường là 0Mbps. Trong trường hợp này, nhà cung cấp truyền phát rộng không tính đến bộ thu dữ liệu thường và chỉ tính đến bộ thu dữ liệu di động. Trong trường hợp này, vùng trong đó có mặt phần giữ chỗ cho phần đầu MPEG và phần chẵn lẻ RS, mà giữ nguyên nhằm tương thích với bộ thu dữ liệu thường hiện có, được xác định là vùng cho dữ liệu di động và khả năng truyền tải của dữ liệu di động có thể tăng lên thành 21,5Mbps.

Để cấp phát toàn bộ tốc độ truyền dữ liệu 19,4Mbps cho dữ liệu di động, 156 gói của các rãnh M/H tạo cấu hình khung M/H phải được cấp phát cho dữ liệu di động. Có nghĩa là tất cả 16 rãnh trong mỗi khung phụ M/H được thiết lập là chế độ có thể mở rộng 11. Trong trường hợp này, 38 gói, tương ứng với vùng dữ liệu thường, được điền dữ liệu di động, và ngoài ra, khối SB5 tương ứng với vùng, trong đó có mặt phần giữ chỗ cho phần đầu MPEG và phần chẵn lẻ RS của vùng thân có thể được tạo ra. Nếu tất cả 16 rãnh của khung phụ M/H được thiết lập là chế độ có thể mở rộng 11 và chế độ khung RS được thiết lập là 00 (chế độ khung đơn), thì khối SB5 không có mặt riêng rẽ và phần giữ chỗ tương ứng với SB5 được gộp vào trong các khối M/H B4, B5, B6, và B7. Nếu tất cả 16 rãnh của khung phụ M/H được thiết lập là chế độ có thể mở rộng 11 và chế độ khung RS là 01 (chế độ khung kép), thì phần giữ chỗ được đặt trong SB5 tạo cấu hình khối SB5. Vùng giữ chỗ cho phần chẵn lẻ RS có mặt trong vùng đầu/đuôi khác vùng thân cũng được

điền dữ liệu di động và được gộp vào trong khối mà phân đoạn tại đó có mặt phần giữ chỗ cho phần chẵn lẻ RS thuộc về. Phần giữ chỗ được đặt trong phân đoạn tương ứng của các khối M/H B8 và B9 được gộp vào trong SB1. Phần giữ chỗ được đặt trong 14 phân đoạn đầu của khối M/H B10 được gộp vào trong SB2. Phần giữ chỗ được đặt trong 14 phân đoạn cuối của khối M/H B1 của rãnh tiếp theo được gộp vào trong SB3. Phần giữ chỗ được đặt trong các phân đoạn tương ứng của các khối M/H B2 và B3 của rãnh tiếp theo được gộp vào trong SB4. Như được thể hiện trên Fig.20 được mô tả ở trên, vùng cho phần đầu MPEG và phần chẵn lẻ RS không có mặt theo định dạng nhóm sau khi đan xen.

Nếu tốc độ truyền dữ liệu hiện có 19,4Mbps không được cấp phát cho dữ liệu di động, thì tốc độ dữ liệu thường không bằng 0Mbps. Trong trường hợp này, nhà cung cấp truyền phát rộng cung cấp các dịch vụ có tính đến cả bộ thu dữ liệu thường và bộ thu dữ liệu di động. Trong trường hợp này, để duy trì tính tương thích với bộ thu dữ liệu thường hiện có, phần đầu MPEG và phần chẵn lẻ RS không được xác định lại là dữ liệu di động và sẽ được truyền như vậy. Tức là, khi ở chế độ tương thích được mô tả ở trên, kể cả nếu chỉ một số trong 38 gói được điền dữ liệu di động mới hoặc tất cả 38 gói được điền dữ liệu di động mới, phần đầu MPEG và phần chẵn lẻ RS không được điền dữ liệu di động mới. Do đó, thậm chí nếu tất cả 38 gói, là vùng dữ liệu thường trong rãnh nhất định, được điền dữ liệu di động, thì khối SB5 tương ứng với vùng, tại đó có phần đầu MPEG và phần chẵn lẻ RS của vùng thân không được tạo ra.

Fig.57 minh họa định dạng nhóm có đơn vị gói trước khi đan xen có tính đến tính tương thích nếu tất cả 38 gói, là vùng dữ liệu thường, được điền dữ liệu di động. Như ở phần d) của các hình vẽ từ Fig.37 đến Fig.40, tất cả 38 gói được cấp phát cho dữ liệu di động, nhưng, như được thể hiện trên Fig.56, vùng trong đó có mặt phần đầu MPEG và phần chẵn lẻ RS được duy trì theo định dạng nhóm có đơn vị phân đoạn sau khi đan xen và vùng khối SB5 không được tạo ra. Định dạng nhóm này có thể được xác định là định dạng nhóm tương ứng với chế độ thứ tư hoặc chế độ có thể mở rộng 11. Ngoài ra, chế độ thứ tư, trong đó chỉ có 38 gói được điền dữ liệu di động mới có tính đến sự tương thích có thể được gọi là chế độ có thể mở rộng 11a.

Nếu sử dụng chế độ có thể mở rộng 11, là chế độ không tương thích, thì rãnh được điền dữ liệu di động mới ở các chế độ khác không thể được sử dụng. Tức là, tất cả các rãnh, tức là các rãnh từ thứ 0 đến thứ 15 phải được điền dữ liệu di động mới theo chế độ có thể mở rộng 11. Mặt khác, các chế độ thứ nhất đến thứ tư có thể được sử dụng kết hợp.

Như đã mô tả ở trên, vùng dữ liệu thường của mỗi rãnh có thể được điền dữ liệu di động theo các định dạng khác nhau. Do đó, định dạng của rãnh có thể thay đổi theo điều kiện thiết lập của chế độ khung và chế độ.

Nếu bốn chế độ được đưa ra như đã mô tả ở trên, thì các rãnh trong đó dữ liệu di động được đặt ở các chế độ thứ nhất đến thứ tư có thể được gọi là các rãnh loại thứ nhất đến các rãnh loại thứ tư.

Bộ truyền phát rộng dạng số có thể tạo cấu hình rãnh cùng loại trong mọi rãnh. Tuy nhiên, ngược lại, bộ truyền phát rộng dạng số có thể tạo cấu hình luồng để có các loại rãnh khác nhau được lặp lại theo đơn vị gồm số lượng rãnh định trước.

Tức là, như được thể hiện trên Fig.41, bộ tiền xử lý dữ liệu 100 có thể đặt dữ liệu di động sao cho một rãnh loại thứ nhất và ba rãnh loại 0 được đặt lặp lại. Rãnh loại 0 chỉ rãnh, trong đó dữ liệu thường được cấp phát cho các gói được cấp phát cho dữ liệu thường như vậy.

Các loại rãnh này có thể được xác định bằng cách sử dụng dữ liệu truyền tín hiệu hiện có, ví dụ, một phần cụ thể của TPC hoặc FIC.

Nếu chế độ khung được thiết lập là 1 như đã mô tả ở trên, chế độ có thể được thiết lập là một trong số nhiều chế độ, các chế độ thứ nhất đến thứ tư. Chế độ thứ tư có thể là chế độ có thể mở rộng 11 hoặc chế độ có thể mở rộng 11a được mô tả ở trên. Ngoài ra, chế độ thứ tư có thể bao gồm các chế độ có thể mở rộng 11 và 11a và có thể là một trong số tổng cộng năm chế độ. Chế độ có thể được chia thành ít nhất một chế độ tương thích và chế độ không tương thích, tức là chế độ có thể mở rộng 11.

Nếu chế độ bao gồm các chế độ thứ nhất đến thứ tư theo một phương án ví dụ, các rãnh tương ứng với các chế độ có thể được gọi là các rãnh loại 1-1, 1-2, 1-3, và 1-4.

Tức là, rãnh loại 1-1 chỉ rãnh trong đó 38 gói được đặt ở chế độ thứ nhất, rãnh loại 1-2 chỉ rãnh trong đó 38 gói được đặt ở chế độ thứ hai, rãnh loại 1-3 chỉ rãnh trong đó 38 gói được đặt ở chế độ thứ ba, và rãnh loại 1-4 chỉ rãnh trong đó 38 gói được đặt ở chế độ thứ tư.

Fig.42 minh họa các ví dụ về luồng, trong đó các loại rãnh khác nhau được đặt lặp lại.

Ví dụ 1 trên Fig.42 minh họa luồng, trong đó rãnh loại 0 và các rãnh loại 1-1, 1-2, 1-3, và 1-4 được lặp lại tuần tự.

Ví dụ 2 trên Fig.42 minh họa luồng trong đó rãnh loại 1-4 và rãnh loại 0 được lặp lại luân phiên. Như đã mô tả ở trên, chế độ thứ tư là chế độ, trong đó toàn bộ vùng dữ liệu thường được điền dữ liệu di động. Theo đó, trong toàn bộ vùng dữ liệu thường của ví dụ 2, rãnh được sử dụng cho dữ liệu di động và rãnh được sử dụng cho dữ liệu thường được đặt luân phiên.

Như trong các ví dụ 3, 4, và 5, các loại rãnh khác nhau có thể được đặt lặp lại theo các cách khác nhau. Cụ thể, như trong ví dụ 6, toàn bộ các rãnh có thể được kết hợp thành loại đơn, theo đó tạo cấu hình cho luồng.

Fig.43 là hình vẽ minh họa kết cấu luồng theo ví dụ 2 trên Fig.42. Trên Fig.43, trong rãnh loại 0, vùng dữ liệu thường được sử dụng cho dữ liệu thường, và trong rãnh loại thứ nhất, toàn bộ vùng dữ liệu thường được sử dụng cho dữ liệu di động và đồng thời đã biết được đặt theo định dạng chuỗi hướng dẫn dài. Như đã mô tả ở trên, các loại rãnh có thể được thực hiện theo cách khác nhau.

Các hình vẽ từ Fig.44 đến Fig.47 minh họa các kết cấu luồng để giải thích phương pháp cấp phát các khối ở các chế độ từ 1 đến 4. Như đã mô tả ở trên, mỗi vùng trong số vùng thứ nhất và vùng thứ hai có thể được chia thành nhiều khối.

Bộ tiền xử lý dữ liệu 100 có thể thực hiện việc mã hóa khối theo đơn vị khối đơn hoặc kết hợp của nhiều khối theo chế độ khối định trước.

Fig.44 minh họa việc chia khối ở chế độ thứ nhất. Trên Fig.44, vùng thân được chia thành các khối từ B3 đến B8 và vùng đầu/đuôi được chia thành các khối từ BN1 đến BN4.

Fig.45 và Fig.46 minh họa việc chia khối ở chế độ thứ hai và chế độ thứ ba. Tương tự trên Fig.44, mỗi vùng trong số vùng thân và vùng đầu/đuôi được chia thành nhiều khối.

Fig.47 minh họa việc chia khối ở chế độ thứ tư, trong đó vùng đầu/đuôi được điền hoàn toàn bởi dữ liệu di động. Do vùng dữ liệu thường được điền hoàn toàn bởi dữ liệu di động, nên phần đầu MPEG của vùng thân và phần chẵn lẻ của dữ liệu thường là thừa. Các phần này được xác định như khối BN5 trên Fig.47. Khối BN5 được điền dữ liệu di động

mới ở chế độ không tương thích và được sử dụng cho phần đầu và phần chẵn lẻ ở chế độ tương thích. So với các hình vẽ từ Fig.44 đến Fig.46, Fig.47 minh họa vùng đầu/đuôi được chia thành các khối từ BN1 đến BN5.

Như đã mô tả ở trên, bộ xử lý khối 120 của bộ tiền xử lý dữ liệu 100 chuyển đổi khung RS theo đơn vị khối và xử lý các khối. Tức là, như được thể hiện trên Fig.7, bộ xử lý khối 120 bao gồm bộ chuyển đổi thứ nhất 121, mà kết hợp dữ liệu di động trong khung RS theo chế độ khối định trước, theo đó đưa ra khối mã chập ghép nối tiếp (serially concatenated convolutional code-SCCC).

Chế độ khối có thể được thiết lập khác nhau.

Ví dụ, nếu chế độ khối được thiết lập là 0, thì các khối BN1, BN2, BN3, BN4, và BN5 được đưa vào như khối SCCC đơn và hoạt động như đơn vị mã hóa SCCC.

Mặt khác, nếu chế độ khối được thiết lập là 1, thì các khối được thêm vào, theo đó tạo cấu hình cho khối SCCC. Cụ thể, $BN1+BN3=SCBN1$ và $BN2+BN4=SCBN2$. Khối BN5 có thể trở thành khối SCBN3.

Ngoài ra dữ liệu di động được đặt trong vùng thứ hai, dữ liệu di động hiện có được đặt trong vùng thứ nhất có thể được mã hóa khối nhờ được kết hợp thành khối đơn hoặc nhiều khối theo chế độ khối. Nó giống như ATSC-MH trong lĩnh vực và bởi vậy mô tả chi tiết của chúng được bỏ qua.

Thông tin về chế độ khối có thể được ghi trên dữ liệu truyền tín hiệu hiện có hoặc có trong vùng được cung cấp trong dữ liệu truyền tín hiệu mới, và có thể được thông báo cho bộ thu. Bộ thu nhận diện thông tin về chế độ khối và giải mã phù hợp dữ liệu, nhờ đó khôi phục luồng ban đầu.

Như đã mô tả ở trên, dữ liệu để được mã hóa khối có thể được kết hợp để tạo cấu hình khung RS. Tức là, bộ mã hóa khung 110 trong bộ tiền xử lý dữ liệu 100 tạo ra khung RS bằng cách kết hợp các phần khung sao cho bộ xử lý khối 120 có thể thực hiện sự mã hóa khối một cách phù hợp.

Cụ thể, khung RS 0 được tạo ra bằng cách kết hợp các khối SCBN1 và SCBN2 và khung RS 1 được tạo ra bằng cách kết hợp các khối SCBN3 và SCBN4.

Ngoài ra, khung RS 0 có thể được tạo ra bằng cách kết hợp các khối SCBN1, SCBN2, SCBN3, và SCBN4 và khung RS 1 có thể được tạo ra bởi khối SCBN5.

Ngoài ra, khung RS đơn có thể được tạo ra bằng cách thêm các khối SCBN1, SCBN2, SCBN3, SCBN4, và SCBN5.

Ngoài ra, khung RS có thể được tạo ra bằng cách thêm các khối tương ứng với dữ liệu di động hiện có và các khối mới được thêm vào (SCBN1~SCBN5).

Fig.48 là hình vẽ giải thích các phương pháp khác nhau để xác định điểm bắt đầu của khung RS. Trên Fig.48, luồng vận chuyển được chia thành nhiều khối. Trong ATSC-MH trong lĩnh vực, khung RS được nhận diện giữa các khối BN2 và BN3. Tuy nhiên, do dữ liệu di động và dữ liệu đã biết được chèn vào trong vùng dữ liệu thường như theo phương án ví dụ của sáng chế, nên điểm bắt đầu của khung RS có thể được xác định theo cách khác.

Ví dụ, khung RS có thể bắt đầu từ ranh giới giữa các khối BN1 và B8, có thể bắt đầu từ ranh giới giữa các khối BN2 và BN3 tương tự như điểm tham chiếu hiện tại, hoặc có thể bắt đầu từ ranh giới giữa các khối B8 và BN1. Điểm bắt đầu của khung RS có thể được xác định theo cách khác theo điều kiện kết hợp của mã hóa khối.

Thông tin cấu hình của khung RS được mô tả ở trên có thể có trong vùng được đưa ra trong dữ liệu truyền tín hiệu hiện có hoặc dữ liệu truyền tín hiệu mới và có thể được thông báo cho bộ thu.

Do dữ liệu di động mới và dữ liệu đã biết được chèn vào trong vùng được cấp phát ban đầu cho dữ liệu thường và vùng được cấp phát cho dữ liệu di động hiện có như đã mô tả ở trên, nên cần có nhiều loại thông tin để thông báo sự kiện này cho bộ thu. Thông tin này có thể được truyền bằng cách sử dụng bit dự trữ trong vùng TPC theo chuẩn ATSC-MH trong lĩnh vực, và ngoài ra, vùng dữ liệu truyền tín hiệu mới thu được và dữ liệu truyền tín hiệu mới có thể được truyền bằng cách sử dụng vùng đó. Do vùng truyền tín hiệu mới được cung cấp phải được đặt trong cùng vị trí trong mọi chế độ, vùng truyền tín hiệu mới được đặt ở phần đầu/phần đuôi.

Fig.49 minh họa kết cấu luồng biểu thị vị trí đặt của dữ liệu truyền tín hiệu hiện có và vị trí đặt của dữ liệu truyền tín hiệu mới.

Trên Fig.49, dữ liệu truyền tín hiệu hiện có được đặt giữa các chuỗi hướng dẫn dài của vùng thân và dữ liệu truyền tín hiệu mới được đặt trong vùng đầu/đuôi. Dữ liệu truyền tín hiệu mới được mã hóa bởi bộ mã hóa truyền tín hiệu 150 được chèn vào vị trí định trước được thể hiện trên Fig.49 nhờ bộ định dạng nhóm 130.

Bộ mã hóa truyền tín hiệu 150 có thể cải thiện hiệu suất bằng cách sử dụng mã khác với mã của bộ mã hóa truyền tín hiệu trong lĩnh vực hoặc bằng cách thực hiện việc mã hóa ở tỷ lệ mã hóa khác.

Tức là, cùng dữ liệu được truyền hai lần sử dụng mã PCCC 1/8 ngoài mã RS hiện có hoặc sử dụng mã RS+1/4 PCCC sao cho có thể thu được cùng tác dụng như khi sử dụng mã PCCC tỷ lệ 1/8.

Do dữ liệu đã biết có trong luồng vận chuyển như đã mô tả ở trên, nên bộ nhớ trong bộ mã hóa dạng mắt cáo phải được khởi tạo trước khi mã hóa dạng mắt cáo được thực hiện cho dữ liệu đã biết.

Nếu chuỗi hướng dẫn dài được cung cấp như ở chế độ 4, thì có thể xử lý chuỗi tương ứng nhờ khởi tạo đơn và theo đó không gặp vấn đề lớn. Tuy nhiên, nếu dữ liệu đã biết được đặt không liên tục như ở các chế độ khác, thì có vấn đề là việc khởi tạo phải được thực hiện vài lần. Ngoài ra, nếu bộ nhớ được khởi tạo để có giá trị là 0, thì khó tạo ra cùng ký hiệu như ở chế độ 4.

Tính đến điều này, việc thiết lập lại dạng mắt cáo không được thực hiện và giá trị bộ nhớ của bộ mã hóa dạng mắt cáo (tức là, giá trị thanh ghi) ở chế độ 4 ở cùng vị trí có thể được nạp vào trong bộ mã hóa dạng mắt cáo, sao cho cùng ký hiệu như ở chế độ 4 có thể được tạo ra ở các chế độ từ 1 đến 3. Để đạt được điều này, các giá trị lưu trữ bộ nhớ của bộ mã hóa dạng mắt cáo ở chế độ 4 được ghi lại và được lưu trữ dưới dạng bảng, và có thể được mã hóa dạng mắt cáo để có các giá trị của các vị trí tương ứng của bảng được lưu trữ. Ngoài ra, bộ mã hóa dạng mắt cáo riêng có thể hoạt động ở chế độ 4 được bố trí và giá trị thu được từ bộ mã hóa dạng mắt cáo có thể được sử dụng.

Như đã mô tả ở trên, dữ liệu di động có thể được cung cấp theo các cách khác nhau bằng cách chủ động sử dụng vùng dữ liệu thường và vùng dữ liệu di động hiện có trong luồng vận chuyển. Do đó, so với chuẩn ATSC trong lĩnh vực, sáng chế có thể tạo ra luồng thích hợp hơn để truyền dữ liệu di động.

Truyền tín hiệu

Do dữ liệu di động mới và dữ liệu đã biết được thêm vào luồng vận chuyển như đã mô tả ở trên, kỹ thuật để thông báo sự kiện này cho bộ thu để xử lý dữ liệu này được sử dụng. Có thể thực hiện việc thông báo theo các cách khác nhau.

Theo phương pháp thứ nhất, sự có mặt/không có mặt của dữ liệu di động mới có thể được thông báo bằng cách sử dụng đồng bộ trường dữ liệu được sử dụng để truyền của dữ liệu di động hiện có.

Fig.50 là hình vẽ minh họa một ví dụ về kết cấu đồng bộ trường dữ liệu. Trên Fig.50, đồng bộ trường dữ liệu bao gồm tổng cộng 832 ký hiệu. 104 trong 832 ký hiệu tương ứng với vùng dự trữ. Các ký hiệu thứ 83 đến 92 trong vùng dự trữ, tức là 10 ký hiệu tương ứng với vùng tăng cường.

Nếu chỉ bao gồm dữ liệu phiên bản 1.0, thì ký hiệu thứ 85 trong trường dữ liệu được đánh số lẻ được thiết lập là +5 và các ký hiệu còn lại, tức là các ký hiệu thứ 83, 84, và từ 87 đến 92, được thiết lập là -5. Trường dữ liệu được đánh số chẵn có dấu ký hiệu ngược lại với dấu của trường dữ liệu được đánh số lẻ. Tức là, nó được thông báo là liệu dữ liệu phiên bản 1.1 có được bao gồm hay không bằng cách sử dụng ký hiệu thứ 86.

Một ký hiệu khác trong vùng tăng cường có thể được sử dụng để thông báo liệu dữ liệu phiên bản 1.1 có được bao gồm hay không. Tức là, nhờ thiết lập một hoặc nhiều ký hiệu trừ ký hiệu thứ 85 là +5 hoặc các giá trị khác, nó được thông báo là liệu dữ liệu phiên bản 1.1 có được bao gồm hay không. Ví dụ, ký hiệu thứ 87 có thể được sử dụng.

Đồng bộ trường dữ liệu được tạo ra bởi bộ điều khiển hoặc bộ mã hóa truyền tín hiệu trên Fig.3 hoặc bộ tạo đồng bộ trường được bố trí riêng, và được cung cấp tới bộ dồn kênh đồng bộ 470 trên Fig.4 và theo đó được dồn kênh vào trong luồng nhờ bộ dồn kênh đồng bộ 470.

Theo phương pháp thứ hai, sự có mặt/không có mặt của dữ liệu phiên bản 1.1 có thể được thông báo bằng cách sử dụng TPC. TPC có thể bao gồm các cú pháp như trong bảng 1 sau đây.

Bảng 1

Số cú pháp của định dạng bit		
TPC_data { sub-frame_number slot_number parade_id	34743322222	uimsbfuimsbf
starting_group_number number_of_groups_minus_1	222545215	uimsbfuimsbf
parade_repetition_cycle_minus_1 rs_frame_mode		uimsbfuimsbf
rs_code_mode_primary rs_code_mode_secondary		bslbfbslbfbslbf
sccc_block_mode sccc_outer_code_mode_a		bslbfbslbfbslbf
sccc_outer_code_mode_b sccc_outer_code_mode_c		bslbfbslbfuims
sccc_outer_code_mode_d fic_version		bfiimsbfuims
parade_continuity_counter total_number_of_groups		bfbfbslbf
reserved tpc_protocol_version }		

Có vùng dự trữ trong thông tin TPC như được thể hiện trong bảng 1. Do đó, có thể truyền tín hiệu là liệu dữ liệu di động có trong các gói được cấp phát cho dữ liệu thường, tức là trong vùng thứ hai, vị trí của dữ liệu di động, liệu dữ liệu đã biết mới có được thêm vào hay không, và vị trí của dữ liệu đã biết được thêm vào, bằng cách sử dụng một hoặc nhiều bit trong vùng dự trữ.

Thông tin chèn vào có thể được biểu diễn như trong bảng 2 sau đây:

Bảng 2

Trường cần thiết	Bit (có thể thay đổi)
Chế độ khung 1.1	3
Chế độ di động 1.1	2
Chế độ khối SCCC 1.1	2
1.1 SCCC BM1	2
1.1 SCCC BM2	2
1.1 SCCC BM3	2
1.1 SCCC BM4	2
1.1 SCCC BM5	2

Trong bảng 2, chế độ khung 1.1 chỉ thông tin biểu thị liệu các gói có được cấp phát cho dữ liệu thường được sử dụng cho dữ liệu thường như vậy hoặc được sử dụng cho dữ liệu di động mới, tức là dữ liệu phiên bản 1.1.

Chế độ di động 1.1 chỉ thông tin biểu thị dữ liệu di động được đặt trong các gói được cấp phát cho dữ liệu thường theo mẫu nào. Một trong bốn chế độ, các chế độ từ 1 đến 4, được mô tả ở trên có thể được biểu diễn bằng cách tạo ra một trong các giá trị “00”, “01”, “10”, và “11” sử dụng hai bit. Do đó, luồng có thể được bố trí trong các định dạng khác nhau như trên Fig.29, Fig.31, Fig.33, Fig.35, Fig.37, Fig.38, Fig.39, và Fig.40. Bộ thu nhận diện thông tin chế độ di động và theo đó nhận diện vị trí đặt của dữ liệu di động.

Chế độ khối SCCC 1.1 chỉ thông tin biểu thị chế độ khối của dữ liệu phiên bản 1.1. SCCC BM1~SCCC BM5 1.1 chỉ thông tin biểu thị đơn vị mã hóa của dữ liệu phiên bản 1.1.

Ngoài thông tin được thể hiện trong bảng 2, có thể cung cấp thêm nhiều loại thông tin cho phép bộ thu phát hiện dữ liệu di động mới một cách phù hợp và giải mã dữ liệu di động mới. Số lượng bit được cấp phát cho mỗi mẫu thông tin có thể thay đổi khi cần và vị trí của mỗi trường có thể được bố trí theo thứ tự khác với thứ tự của bảng 2.

Thông tin FIC có thể được sử dụng cho bộ thu phát rộng dạng số, mà nhận luồng bao gồm dữ liệu di động mới, để nhận ra liệu dữ liệu di động mới có được bao gồm hay không.

Tức là, bộ thu phiên bản 1.1, mà nhận và xử lý dữ liệu di động mới, phải có thể xử lý thông tin dịch vụ 1.0 và thông tin dịch vụ 1.1 một cách đồng thời. Mặt khác, bộ thu phiên bản 1.0 phải có thể không xét đến thông tin dịch vụ 1.1.

Do đó, vùng để thông báo sự có mặt/không có mặt của dữ liệu phiên bản 1.1 có thể thu được nhờ thay đổi cú pháp phân đoạn FIC hiện có.

Trước tiên, cú pháp của phân đoạn FIC hiện có có thể được tạo cấu hình như trong bảng 3 sau đây:

Bảng 3

Số cú pháp của định dạng bit		
FIC_segment_header() { FIC_segment_type	2221144	uimsbf11'uimsb
reserved FIC_chunk_major_protocol_version		fbslbfbslbfuims
current_next_indicator error_indicator		bfuimsbf
FIC_segment_num FIC_last_segment_num }		

Phân đoạn FIC thể hiện trong bảng 3 có thể thay đổi để thông báo sự có mặt/không có mặt của dữ liệu phiên bản 1.1 như trong bảng 4 sau đây:

Bảng 4

Số cú pháp của định dạng bit			
FIC_segment_header() { FIC_segment_type	211255	uimsbfb	slbfb
current_next_indicator error_indicator		fuimsb	fuimsbfu
FIC_chunk_major_protocol_version		imsbf	
FIC_segment_num FIC_last_segment_num }			

Trên Fig.4, thay cho vùng dự trữ, FIC_segment_num và FIC_last_segment_num có thể được mở rộng thành 5 bit.

Trong bảng 4, 01 được thêm vào giá trị của FIC_segement_type sao cho sự có mặt/không có mặt của dữ liệu phiên bản 1.1 có thể được thông báo. Tức là, nếu FIC_segement_type được thiết lập là 01, thì bộ thu phiên bản 1.1 giải mã thông tin FIC và xử lý dữ liệu phiên bản 1.1. Trong trường hợp này, bộ thu phiên bản 1.0 không thể phát hiện thông tin FIC. Mặt khác, nếu FIC_segment_type được xác định là 00 hoặc phân đoạn rỗng, thì bộ thu phiên bản 1.0 giải mã thông tin FIC và xử lý dữ liệu di động hiện có.

Sự có mặt/không có mặt của dữ liệu phiên bản 1.1 có thể được thông báo bằng cách sử dụng một số vùng của cú pháp của khúc FIC, ví dụ, vùng dự trữ, trong khi duy trì cú pháp của khúc FIC mà không thay đổi cú pháp FIC hiện có.

FIC có thể bao gồm 16 bit là tối đa khi khúc FIC được tạo cấu hình. Trạng thái của dữ liệu phiên bản 1.1 có thể được biểu thị bằng cách thay đổi một số cú pháp của khúc FIC.

Cụ thể, “MH 1.1 service_status” có thể được thêm vào vùng dự trữ của vòng lặp quản thể dịch vụ như trong bảng 5 sau đây:

Bảng 5

Số cú pháp của định dạng bit		
FIC_chunk_payload(){ for(i=0; i<num_ensembles; i++){ ensemble_id reserved ensemble_protocol_version SLT_ensemble_indicator GAT_ensemble_indicator reserved MH_service_signaling_channel_version num_MH_services for (j=0; j<num_MH_services; j++){ MH_service_id MH1.1_service_status reserved multi_ensemble_service MH_service_status SP_indicator } } FIC_chunk_stuffing() }	8351115816212 21var	uimsbfbf111'ui sbfbfslbfslbf'1' uimsbfuimsbfui msbfuimsbf'1'ui msbfuimsbfslbf

Trên Fig.5, MH1.1_service_status có thể được biểu thị bằng cách sử dụng hai trong ba bit của vùng dự trữ. MH1.1_service_status có thể là dữ liệu biểu thị dữ liệu phiên bản 1.1 có mặt trong luồng hay không.

Ngoài MH1.1_service_status, MH1.1_ensemble_indicator có thể được thêm vào. Tức là, cú pháp của khúc FIC có thể được tạo cấu hình như trong bảng 6 sau đây:

Bảng 6

Số cú pháp của định dạng bit		
FIC_chunk_payload(){ for(i=0; i<num_ensembles; i++){ ensemble_id MH1.1_ensemble_indicator reserved ensemble_protocol_version SLT_ensemble_indicator GAT_ensemble_indicator reserved MH_service_signaling_channel_version num_MH_services for (j=0; j<num_MH_services; j++){ MH_service_id MH1.1_service_status_extension reserved multi_ensemble_service MH_service_status SP_indicator } } FIC_chunk_stuffing() }	8125111581622 21var	uimsbfbfslbf'1'1' uimsbfbfslbfslbf f'1'uimsbfuimsb fuimsbfuimsbf 1'uimsbfuimsbf bslbf

Trên Fig.6, 1 bit trong 3 bit của vùng dự trữ thứ nhất được cấp phát cho MH1.1_ensemble_indicator. MH1.1_ensemble_indicator chỉ thông tin liên quan đến quần thể, là đơn vị dịch vụ của dữ liệu phiên bản 1.1. Trong bảng 6, MH1.1_service_status_extension có thể được biểu thị bằng cách sử dụng 2 bit trong 3 bit của vùng dự trữ thứ hai.

Nếu phiên bản giao thức quần thể được thay đổi và theo đó dịch vụ phiên bản 1.1 được cung cấp như trong bảng 7, thì phiên bản 1.1 được biểu thị bằng cách sử dụng giá trị được cấp phát cho vùng dự trữ 1.0:

Bảng 7

Số cú pháp của định dạng bit		
FIC_chunk_payload(){ for(i=0; i<num_ensembles; i++){	8351115816322	uimsbf111'uim
ensemble_id reserved ensemble_protocol_version	1var	sbfbfslbfbslbf1'
SLT_ensemble_indicator GAT_ensemble_indicator		uimsbfuimsbfui
reserved MH_service_signaling_channel_version		msbf111'uimsb
num_MH_services for (j=0; j<num_MH_services; j++){		fuimsbfbslbf
MH_service_id reserved multi_ensemble_service		
MH_service_status SP_indicator } }		
FIC_chunk_stuffing() }		

Ngoài ra, dữ liệu truyền tín hiệu có thể được truyền nhờ thay đổi chiều dài mở rộng phần đầu vòng lặp quần thể trong các trường cú pháp của phần đầu khúc FIC, thêm phần mở rộng quần thể trong các trường cú pháp của phần tải khúc FIC, và thêm MH1.1_service_status vào 3 bit phục vụ vòng lặp dịch vụ trong số các cú pháp của phần tải khúc FIC như trong bảng 8 sau đây:

Bảng 8

Số cú pháp của định dạng bit		
FIC_chunk_payload(){ for(i=0; i<num_ensembles; i++){	8351115358162	uimsbf111'uim
ensemble_id reserved ensemble_protocol_version	13221var	sbfbfslbfbfslbf1'
SLT_ensemble_indicator GAT_ensemble_indicator		uimsbfuimsbfui
reserved MH_service_signaling_channel_version		msbfuimsbf111
reserved ensemble_extension num_MH_services for (j=0;		'uimsbfuimsbf
j<num_MH_services; j++){ MH_service_id		slbf
MH_service_status_extention reserved reserved		
multi_ensemble_service MH_service_status SP_indicator		
} } FIC_chunk_stuffing() }		

Ngoài ra, MH_service_loop_extension_length từ trong các trường cú pháp của phần đầu khúc FIC có thể thay đổi và trường thông tin về trạng thái MH1.1_service có thể được thêm vào trường tải của khúc FIC:

Bảng 9

Số cú pháp của định dạng bit		
FIC_chunk_payload(){ for(i=0; i<num_ensembles; i++){	8351115816322	uimsbf111'uim
ensemble_id reserved ensemble_protocol_version	153var	sbfbfslbfbfslbf1'
SLT_ensemble_indicator GAT_ensemble_indicator		uimsbfuimsbfui
reserved MH_service_signaling_channel_version		msbf111'uimsb
num_MH_services for (j=0; j<num_MH_services; j++){		fuimsbfslbfui
MH_service_id reserved multi_ensemble_service		msbfuimsbf
MH_service_status SP_indicator reserved		
MH1.1_Detailed_service_Info } }		
FIC_chunk_stuffing() }		

Như đã mô tả ở trên, dữ liệu truyền tín hiệu có thể được cấp cho bộ thu bằng cách sử dụng các vùng khác nhau như đồng bộ trường, thông tin TPC, và thông tin FIC.

Dữ liệu truyền tín hiệu có thể được chèn vào trong vùng khác với các vùng được mô tả ở trên. Tức là, dữ liệu truyền tín hiệu có thể được chèn vào ở phần tải gói của dữ liệu hiện có. Trong trường hợp này, sự có mặt của dữ liệu phiên bản 1.1 hoặc dữ liệu truyền tín hiệu nhận diện vị trí được ghi lại sử dụng thông tin FIC như được thể hiện trong bảng 5, và dữ liệu truyền tín hiệu phiên bản 1.1 được cung cấp riêng, sao cho bộ thu phiên bản 1.1 có thể phát hiện và sử dụng dữ liệu truyền tín hiệu tương đương.

Dữ liệu truyền tín hiệu có thể được tạo cấu hình như luồng riêng và có thể được truyền tới bộ thu bằng cách sử dụng kênh riêng khác với kênh truyền luồng.

Ngoài ra, dữ liệu truyền tín hiệu có thể bao gồm thông tin khác để truyền tín hiệu ít nhất một mẫu thông tin như thông tin biểu thị liệu hiện có hoặc dữ liệu di động mới có được bao gồm hay không, vị trí của dữ liệu di động, thông tin biểu thị liệu dữ liệu đã biết được thêm vào hay không, vị trí của dữ liệu đã biết được thêm vào, mẫu đặt dữ liệu di động và dữ liệu đã biết, chế độ khối, và đơn vị mã hóa, ngoài các thông tin khác nhau được mô tả ở trên.

Bộ truyền phát rộng dạng số, mà sử dụng dữ liệu truyền tín hiệu, có thể bao gồm bộ tiền xử lý dữ liệu để đặt ít nhất một trong số dữ liệu di động và dữ liệu đã biết trong ít nhất một phần của vùng dữ liệu thường trong số tất cả các gói của luồng, và bộ dồn kênh để tạo ra luồng vận chuyển bao gồm dữ liệu di động và dữ liệu truyền tín hiệu. Kết cấu chi tiết của bộ tiền xử lý dữ liệu có thể được thực hiện theo một trong các phương án ví dụ nêu trên, và một số thành phần có thể được bỏ qua, thêm vào, hoặc sửa đổi. Cụ thể, dữ liệu truyền tín hiệu được tạo ra bởi bộ mã hóa truyền tín hiệu, bộ điều khiển, hoặc bộ tạo đồng bộ trường được bố trí riêng (không được thể hiện trên hình vẽ), và có thể được chèn vào trong luồng vận chuyển nhờ bộ dồn kênh hoặc bộ dồn kênh đồng bộ. Trong trường hợp này, dữ liệu truyền tín hiệu có thể được thực hiện như dữ liệu để thông báo ít nhất một mẫu thông tin về thông tin liên quan đến liệu dữ liệu di động có được đặt hay không và mẫu đặt, như đồng bộ trường dữ liệu hoặc TPC hoặc thông tin FIC như đã mô tả ở trên.

Ngoài ra, như đã mô tả ở trên, nếu có chế độ có thể mở rộng 11a ngoài chế độ có thể mở rộng 11, tức là có các chế độ từ thứ nhất đến thứ năm, thì phương pháp biểu diễn chế độ trong dữ liệu truyền tín hiệu có thể khác đi.

Theo một phương án ví dụ, trường truyền tín hiệu trong trường TPC có thể có tên của chế độ có thể mở rộng, và bốn chế độ như ở phần a) đến d) trên các hình vẽ từ Fig.37 đến Fig.40 có thể được xác định là 00, 01, 10, và 11 bằng cách sử dụng hai bit. Trong trường hợp này, chế độ thứ tư có cùng giá trị bit là 11 không quan tâm đến liệu chế độ thứ tư là chế độ tương thích hay chế độ không tương thích. Tuy nhiên, phần đầu MPEG và vùng chẵn lẻ được sử dụng hay không tùy theo hai chế độ và theo đó định dạng nhóm có thể khác nhau theo hai chế độ.

Bộ thu nhận diện TPC của rãnh bao gồm nhóm M/H của tập hợp M/H cần được nhận và TPC của các rãnh khác, và nếu chế độ có thể mở rộng của tất cả các rãnh là 11 và không có rãnh CMM, tức là nếu tốc độ dữ liệu thường là 0Mbps, thì bộ thu xác định giá trị bit là 11 như chế độ có thể mở rộng 11 và giải mã dữ liệu.

Mặt khác, nếu chế độ có thể mở rộng của tất cả các rãnh không phải là 11 hoặc nếu có rãnh CMM, tức là nếu tốc độ dữ liệu thường không bằng 0Mbps, thì bộ thu xác định giá trị bit là 11 như chế độ có thể mở rộng 11a có tính đến sự tương thích, và giải mã dữ liệu.

Theo một phương án ví dụ khác, trường truyền tín hiệu trong trường TPC có tên của chế độ có thể mở rộng và ba bit có thể được cấp phát cho trường đó. Do đó, tổng cộng năm định dạng nhóm bao gồm ba định dạng nhóm tương ứng với a) đến c) trên các hình vẽ từ Fig.37 đến Fig.40, tức là các chế độ thứ nhất đến thứ ba, và hai định dạng nhóm tương ứng với d) trên các hình vẽ từ Fig.37 đến Fig.40, tức là chế độ thứ tư và chế độ thứ năm.

Tức là, như đã mô tả ở trên, toàn bộ chế độ có thể bao gồm:

- 1) chế độ thứ nhất, trong đó dữ liệu di động mới được đặt trong 11 gói trong số 38 gói được cấp phát cho dữ liệu thường;
- 2) chế độ thứ hai, trong đó dữ liệu di động mới được đặt trong 20 gói trong số 38 gói được cấp phát cho dữ liệu thường;
- 3) chế độ thứ ba, trong đó dữ liệu di động mới được đặt trong 29 gói trong số 38 gói được cấp phát cho dữ liệu thường;
- 4) chế độ thứ tư, trong đó dữ liệu di động mới được đặt trong tất cả 38 gói được cấp phát cho dữ liệu thường; và
- 5) chế độ thứ năm, trong đó dữ liệu di động mới được đặt trong tất cả 38 gói được cấp phát cho dữ liệu thường và trong vùng tương ứng với phần đầu MPEG và phần chẵn lẻ trong số các vùng được cấp phát cho dữ liệu di động hiện có.

Chế độ thứ nhất được xác định là chế độ có thể mở rộng 000, chế độ thứ hai được xác định là chế độ có thể mở rộng 001, chế độ thứ ba được xác định là chế độ có thể mở rộng 010, chế độ thứ tư, trong đó 38 gói được điền dữ liệu di động và sự tương thích phải được tính đến được xác định là chế độ có thể mở rộng 011, và chế độ thứ năm, trong đó 38 gói được điền dữ liệu di động và sự tương thích không cần được tính đến được xác định là chế độ có thể mở rộng 111.

Ngoài ra, để xác định định dạng nhóm bổ sung, giá trị bit của chế độ có thể mở rộng có thể được cấp phát hoặc bit truyền tín hiệu có thể được thêm vào.

Bộ truyền phát rộng dạng số theo các phương án ví dụ khác nhau được mô tả ở trên có thể đặt dữ liệu di động hiện có, dữ liệu di động mới, và dữ liệu thường trong luồng theo các cách khác nhau, và có thể truyền dữ liệu.

Ví dụ, như được thể hiện trên Fig.4, bộ phận tạo cấu hình luồng, tức là bộ định dạng nhóm 130 được bố trí trong bộ tiền xử lý dữ liệu 100 có thể thêm dữ liệu đã biết, dữ liệu truyền tín hiệu, và dữ liệu khởi tạo vào luồng được xử lý bởi bộ xử lý khối 120, nhờ đó định dạng dữ liệu theo đơn vị nhóm.

Do đó, nếu bộ định dạng gói thực hiện định dạng gói, thì bộ dồn kênh 200 thực hiện dồn kênh. Trong trường hợp này, ở các chế độ thứ nhất đến thứ ba, bộ dồn kênh 200 dồn kênh dữ liệu thường được xử lý bởi bộ xử lý thông thường 320. Mặt khác, ở các chế độ thứ tư và thứ năm, bộ xử lý thông thường 320 không đưa ra dữ liệu thường và bộ dồn kênh 200 đưa ra luồng được cung cấp bởi bộ định dạng gói 140 như vậy.

Bộ truyền phát rộng dạng số

Như đã mô tả ở trên, bộ truyền phát rộng dạng số có thể truyền dữ liệu di động mới bằng cách sử dụng một số hoặc tất cả các gói được cấp phát cho dữ liệu thường và một số hoặc tất cả các gói được cấp phát cho dữ liệu di động hiện có trong kết cấu luồng hiện có.

Bộ thu phát rộng dạng số nhận ít nhất một trong số dữ liệu di động hiện có, dữ liệu thường, và dữ liệu di động mới theo phiên bản của nó.

Tức là, nếu bộ thu phát rộng dạng số là để xử lý dữ liệu thường hiện có và nhận luồng được mô tả ở trên có các kết cấu khác nhau, thì bộ thu phát rộng dạng số nhận diện dữ liệu truyền tín hiệu, và phát hiện và giải mã dữ liệu thường. Nếu luồng được tạo cấu hình mà không có dữ liệu thường, thì bộ thu xử lý dữ liệu thường không thể cung cấp dịch vụ dữ liệu thường.

Nếu bộ thu là bộ thu phát rộng dạng số phiên bản 1.0 và nhận luồng được mô tả ở trên có các kết cấu khác nhau, thì bộ thu nhận diện dữ liệu truyền tín hiệu, và phát hiện và giải mã dữ liệu di động hiện có. Nếu dữ liệu di động phiên bản 1.1 được đặt trong toàn bộ vùng, thì bộ thu phát rộng dạng số phiên bản 1.0 không thể cung cấp dịch vụ di động.

Bộ thu phát rộng dạng số phiên bản 1.1 có thể phát hiện và xử lý dữ liệu phiên bản 1.0 cũng như dữ liệu phiên bản 1.1. Trong trường hợp này, nếu bao gồm khối giải mã để xử lý dữ liệu thường, thì dịch vụ dữ liệu thường có thể được hỗ trợ.

Fig.51 là sơ đồ khối minh họa một ví dụ về bộ thu phát rộng dạng số theo một phương án ví dụ. Bộ thu phát rộng dạng số bao gồm các thành phần tương ứng với các thành phần của bộ truyền phát rộng dạng số trên các hình vẽ từ Fig.2 đến Fig.4 và được bố trí theo thứ tự ngược lại. Tuy nhiên, để giải thích thuận tiện, Fig.51 chỉ minh họa các thành phần cơ bản để thu nhận.

Tức là, trên Fig.51, bộ thu phát rộng dạng số bao gồm bộ phận thu 5100 (ví dụ bộ thu), bộ giải điều biến 5200, bộ cân bằng 5300, và bộ phận giải mã 5400 (ví dụ bộ giải mã).

Bộ phận thu 5100 nhận luồng vận chuyển được truyền từ bộ truyền phát rộng dạng số qua anten hoặc cáp.

Bộ giải điều biến 5200 giải điều biến luồng vận chuyển nhận được qua bộ phận thu 5100. Tần số của tín hiệu và tín hiệu đồng hồ nhận được qua bộ phận thu 5100 được đồng bộ với bộ truyền phát rộng dạng số, khi đi qua bộ giải điều biến 5200.

Bộ cân bằng 5300 cân bằng luồng vận chuyển đã được giải điều biến.

Bộ giải điều biến 5200 và bộ cân bằng 5300 có thể thực hiện đồng bộ và cân bằng nhanh chóng hơn bằng cách sử dụng dữ liệu đã biết có trong luồng vận chuyển, cụ thể, dữ liệu đã biết mới được thêm vào cùng với dữ liệu di động.

Bộ phận giải mã 5400 phát hiện dữ liệu di động từ luồng vận chuyển đã được cân bằng và giải mã dữ liệu di động.

Các vị trí chèn và kích thước của dữ liệu di động và dữ liệu đã biết có thể được thông báo qua dữ liệu truyền tín hiệu có trong luồng vận chuyển hoặc dữ liệu truyền tín hiệu nhận được qua kênh riêng.

Bộ phận giải mã 5400 nhận diện vị trí của dữ liệu di động thích hợp với bộ thu phát rộng dạng số bằng cách sử dụng dữ liệu truyền tín hiệu và sau đó phát hiện dữ liệu di động từ vị trí đó và giải mã dữ liệu di động.

Bộ phận giải mã 5400 có thể được thực hiện theo các cách khác nhau theo một phương án ví dụ.

Tức là, bộ phận giải mã 5400 có thể bao gồm hai bộ giải mã bao gồm bộ giải mã dạng mắt cáo và bộ giải mã nhân chập. Hai bộ giải mã trao đổi thông tin tin cậy về giải mã với nhau, nhờ đó cải thiện hiệu suất. Thông tin được đưa ra từ bộ giải mã chập có thể giống như đầu vào của bộ mã hóa RS của bộ truyền.

Fig.52 là sơ đồ khối minh họa chi tiết một ví dụ về bộ thu phát rộng dạng số theo một phương án ví dụ.

Trên Fig.52, bộ thu phát rộng dạng số có thể bao gồm bộ phận thu 5100 (ví dụ bộ thu), bộ giải điều biến 5200, bộ cân bằng 5300, bộ phận giải mã 5400 (bộ giải mã), bộ phận phát hiện 5500 (ví dụ bộ phát hiện), và bộ giải mã truyền tín hiệu 5600.

Các chức năng của bộ phận thu 5100, bộ giải điều biến 5200, và bộ cân bằng 5300 giống như các bộ phận trên Fig.51 và bởi vậy mô tả chi tiết của chúng được bỏ qua.

Bộ phận giải mã 5400 có thể bao gồm bộ giải mã thứ nhất 5410 và bộ giải mã thứ hai 5420.

Bộ giải mã thứ nhất 5410 thực hiện việc giải mã đối với ít nhất một trong số dữ liệu di động hiện có và dữ liệu di động mới. Bộ giải mã thứ nhất 5410 có thể thực hiện việc giải mã SCCC, được thực hiện theo đơn vị khối.

Bộ giải mã thứ hai 5420 thực hiện việc giải mã RS đối với luồng được giải mã nhờ bộ giải mã thứ nhất 5410.

Các bộ giải mã thứ nhất 5410 và thứ hai 5420 có thể xử lý dữ liệu di động bằng cách sử dụng giá trị đầu ra của bộ giải mã truyền tín hiệu 5600.

Tức là, bộ giải mã truyền tín hiệu 5600 phát hiện dữ liệu truyền tín hiệu từ luồng và giải mã dữ liệu truyền tín hiệu. Cụ thể, bộ giải mã truyền tín hiệu 5600 giải dãn kênh vùng dự trữ, vùng thông tin TPC, hoặc vùng thông tin FIC của dữ liệu đồng bộ trường từ luồng vận chuyển. Do đó, bộ giải mã truyền tín hiệu 5600 thực hiện việc giải mã chập và giải mã RS đối với phần đã được giải dãn kênh và sau đó thực hiện ngẫu nhiên hóa đảo, nhờ đó khôi phục dữ liệu truyền tín hiệu. Dữ liệu truyền tín hiệu đã được khôi phục được đưa tới các thành phần của bộ thu phát rộng dạng số, tức là bộ giải điều biến 5200, bộ cân bằng 5300, bộ phận giải mã 5400, và bộ phận phát hiện 5500. Dữ liệu truyền tín hiệu có

thể bao gồm nhiều loại thông tin để được sử dụng bởi các thành phần đó, tức là thông tin chế độ khối, thông tin chế độ, thông tin mẫu chèn dữ liệu đã biết và chế độ khung. Các loại và chức năng của thông tin này đã được mô tả ở trên và bởi vậy mô tả chi tiết được bỏ qua.

Ngoài thông tin trên, nhiều loại thông tin như tỷ lệ mã hóa, tốc độ dữ liệu, và vị trí chèn của dữ liệu di động, loại mã hiệu chỉnh lỗi được sử dụng, thông tin của dịch vụ sơ cấp, thông tin cần thiết để hỗ trợ phân khoảng thời gian, mô tả liên quan đến dữ liệu di động, thông tin liên quan đến thay đổi về thông tin chế độ, và thông tin hỗ trợ dịch vụ IP có thể được cấp cho bộ thu theo định dạng của dữ liệu truyền tín hiệu hoặc dữ liệu bổ sung.

Trên Fig.52, dữ liệu truyền tín hiệu có trong luồng. Tuy nhiên, nếu tín hiệu dữ liệu truyền tín hiệu được truyền thông qua kênh riêng, bộ giải mã truyền tín hiệu 5600 giải mã tín hiệu dữ liệu truyền tín hiệu và đưa ra thông tin trên.

Bộ phận phát hiện 5500 phát hiện dữ liệu đã biết từ luồng bằng cách sử dụng thông tin mẫu chèn dữ liệu đã biết được cung cấp bởi bộ giải mã truyền tín hiệu 5600. Trong trường hợp này, dữ liệu đã biết được thêm vào cùng với dữ liệu di động hiện có cũng như dữ liệu đã biết được thêm vào cùng với dữ liệu di động mới có thể được xử lý.

Cụ thể, như được thể hiện trên các hình vẽ từ Fig.22 đến Fig.36, dữ liệu đã biết có thể được chèn vào trong ít nhất một trong số vùng thân và vùng đầu/đuôi của dữ liệu di động ở các vị trí khác nhau và theo các định dạng khác nhau. Thông tin liên quan đến mẫu chèn của dữ liệu đã biết, tức là vị trí, điểm bắt đầu, và chiều dài của dữ liệu đã biết có thể có trong dữ liệu truyền tín hiệu. Bộ phận phát hiện 5500 phát hiện dữ liệu đã biết từ vị trí thích hợp theo dữ liệu truyền tín hiệu và có thể đưa dữ liệu đã biết tới bộ giải điều biến 5200, bộ cân bằng 5300, và bộ phận giải mã 5400.

Fig.53 là hình vẽ minh họa chi tiết một ví dụ về bộ thu phát rộng dạng số theo một phương án ví dụ khác.

Trên Fig.53, bộ thu phát rộng dạng số bao gồm bộ phận thu 5100 (ví dụ bộ thu), bộ giải điều biến 5200, bộ cân bằng 5300, bộ xử lý FEC 5411, bộ phận giải mã TCM 5412 (ví dụ bộ giải mã TCM), bộ phận giải đan xen CV 5412 (ví dụ bộ giải đan xen CV), bộ phận giải đan xen ngoài 5414 (ví dụ bộ giải đan xen ngoài), bộ phận giải mã ngoài 5415 (ví dụ bộ giải mã ngoài), bộ phận giải mã RS 5416 (ví dụ bộ giải mã RS), bộ ngẫu

nhiên hóa đảo 5417, bộ phận đan xen ngoài 5418, bộ phận đan xen CV 5419 (ví dụ bộ đan xen CV), và bộ giải mã truyền tín hiệu 5600.

Do bộ phận thu 5100, bộ giải điều biến 5200, bộ cân bằng 5300, và bộ giải mã truyền tín hiệu 5600 đã được mô tả ở trên dựa vào Fig.52, nên giải thích trùng lặp được bỏ qua. Không như trên Fig.52, bộ phận phát hiện 5500 được bỏ qua. Tức là, các thành phần có thể phát hiện trực tiếp dữ liệu đã biết bằng cách sử dụng dữ liệu truyền tín hiệu được giải mã nhờ bộ giải mã truyền tín hiệu 5600.

Bộ xử lý FEC 5411 thực hiện việc hiệu chỉnh lỗi tiến đối với luồng vận chuyển được cân bằng bởi bộ cân bằng 5300. Bộ xử lý FEC 5411 phát hiện dữ liệu đã biết từ luồng vận chuyển bằng cách sử dụng thông tin về vị trí hoặc mẫu chèn của dữ liệu đã biết trong số thông tin được cung cấp bởi bộ giải mã truyền tín hiệu và sử dụng dữ liệu đã biết để thực hiện hiệu chỉnh lỗi tiến. Theo một phương án ví dụ, dữ liệu tham chiếu bổ sung có thể không được sử dụng trong hiệu chỉnh lỗi tiến.

Trên Fig.53, các thành phần được bố trí theo cách sao cho việc giải mã được thực hiện cho dữ liệu di động sau khi xử lý FEC được thực hiện. Tức là, việc xử lý FEC được thực hiện đối với toàn bộ luồng vận chuyển. Tuy nhiên, chỉ có dữ liệu di động có thể được phát hiện từ luồng vận chuyển và sau đó có thể thực hiện xử lý FEC đối với chỉ dữ liệu di động.

Bộ phận giải mã TCM 5412 phát hiện dữ liệu di động từ luồng vận chuyển được đưa ra từ bộ xử lý FEC 5411, và thực hiện việc giải mã dạng mất cáo đối với dữ liệu di động. Trong trường hợp này, nếu bộ xử lý FEC đã phát hiện dữ liệu di động và đã thực hiện hiệu chỉnh lỗi tiến đối với chỉ phần được phát hiện, bộ phận giải mã TCM 5412 có thể trực tiếp thực hiện việc giải mã dạng mất cáo đối với dữ liệu đầu vào.

Bộ phận giải đan xen CV 5413 thực hiện giải đan xen chập đối với dữ liệu đã được giải mã dạng mất cáo. Như đã mô tả ở trên, do các thành phần của bộ thu phát rộng dạng số tương ứng với các thành phần của bộ truyền phát rộng dạng số, mà tạo cấu hình và xử lý luồng vận chuyển, nên bộ phận giải đan xen CV 5413 có thể không cần tuân theo kết cấu của bộ truyền.

Bộ phận giải đan xen ngoài 5414 thực hiện giải đan xen ngoài đối với dữ liệu đã được giải đan xen chập. Sau đó, bộ phận giải mã ngoài 5415 thực hiện việc giải mã và loại bỏ phần chẵn lẻ khỏi dữ liệu di động.

Trong một số tình huống, các xử lý từ bộ phận giải mã TCM 5412 đến bộ phận giải mã ngoài 5415 có thể được lặp lại một hoặc nhiều lần sao cho có thể cải thiện hiệu quả thu nhận của dữ liệu di động. Để lặp lại các xử lý, dữ liệu giải mã của bộ phận giải mã ngoài 5415 có thể được cấp cho bộ phận giải mã TCM 5412 thông qua bộ phận đan xen ngoài 5418 và bộ phận đan xen CV 5419 làm đầu vào của bộ phận giải mã TCM 5412. Ở thời điểm này, bộ phận đan xen CV 5419 có thể không cần tuân theo kết cấu của bộ truyền.

Như đã mô tả ở trên, dữ liệu đã được giải mã dạng mắt cáo được cấp cho bộ phận giải mã RS 5416. Bộ phận giải mã RS 5416 giải mã RS dữ liệu và bộ ngẫu nhiên hóa đảo 5417 thực hiện ngẫu nhiên hóa đảo. Thông qua quy trình được mô tả ở trên, dữ liệu di động, cụ thể, luồng của dữ liệu phiên bản 1.1 mới được xác định có thể được xử lý.

Như đã mô tả ở trên, nếu bộ thu phát rộng dạng số là bộ thu phiên bản 1.1, thì bộ thu có thể xử lý dữ liệu phiên bản 1.0 ngoài dữ liệu phiên bản 1.1.

Tức là, ít nhất một trong số bộ xử lý FEC 5411 và bộ phận giải mã TCM 5412 có thể phát hiện toàn bộ dữ liệu di động trừ dữ liệu thường và có thể xử lý dữ liệu được phát hiện.

Nếu bộ thu phát rộng dạng số là bộ thu chung, thì bộ thu có thể bao gồm khối để xử lý dữ liệu thường, khối để xử lý dữ liệu phiên bản 1.0, và khối để xử lý dữ liệu phiên bản 1.1. Trong trường hợp này, nhiều đường dẫn xử lý được bố trí ở đầu sau của bộ cân bằng 5300 và các khối được mô tả ở trên được đặt trong các đường dẫn xử lý, theo cách tương ứng. Ít nhất một đường dẫn xử lý được lựa chọn theo sự điều khiển của bộ điều khiển được bố trí riêng, sao cho dữ liệu thích hợp có trong luồng vận chuyển.

Ngoài ra, như đã mô tả ở trên, dữ liệu di động có thể được đặt trong luồng vận chuyển theo mẫu khác nhau theo rãnh. Tức là, các rãnh khác nhau, như rãnh thuộc loại thứ nhất bao gồm dữ liệu thường như vậy, rãnh thuộc loại thứ hai bao gồm dữ liệu di động mới trong toàn bộ vùng dữ liệu thường, rãnh thuộc loại thứ ba bao gồm dữ liệu di động mới trong một phần của vùng dữ liệu thường, và rãnh thuộc loại thứ tư bao gồm dữ liệu di động mới trong vùng dữ liệu thường và toàn bộ vùng di động hiện có, có thể được lặp lại theo mẫu định trước.

Bộ giải mã truyền tín hiệu 5600 giải mã dữ liệu truyền tín hiệu và thông báo thông tin chế độ khung hoặc thông tin chế độ với mỗi thành phần. Do đó, mỗi thành phần, cụ

thể, bộ xử lý FEC 5411 hoặc bộ phận giải mã TCM 5412 phát hiện dữ liệu di động từ vị trí xác định của mỗi rãnh và xử lý dữ liệu di động.

Các hình vẽ từ Fig.51 đến Fig.53 không minh họa bộ điều khiển, nhưng có thể bao gồm thêm bộ điều khiển để tác dụng tín hiệu điều khiển thích hợp tới mỗi khối bằng cách sử dụng dữ liệu truyền tín hiệu được giải mã nhờ bộ giải mã truyền tín hiệu 5600. Bộ điều khiển này có thể điều khiển hoạt động điều chỉnh của bộ phận thu 5100 theo lựa chọn của người dùng.

Nếu bộ thu phát rộng dạng số là bộ thu phiên bản 1.1, thì bộ thu có thể cung cấp dữ liệu phiên bản 1.0 hoặc dữ liệu phiên bản 1.1 theo cách có chọn lọc theo lựa chọn của người sử dụng. Ngoài ra, nếu nhiều dữ liệu phiên bản 1.1 được cung cấp, thì bộ thu có thể cung cấp một trong các dịch vụ theo lựa chọn của người sử dụng.

Cụ thể, như ở các chế độ thứ nhất đến thứ tư (trong đó các chế độ thứ nhất đến thứ tư có thể là chế độ tương thích hoặc chỉ có chế độ thứ tư có thể là chế độ không tương thích) hoặc các chế độ từ thứ nhất đến thứ năm, ít nhất một trong số dữ liệu thường, dữ liệu di động hiện có, và dữ liệu di động mới có thể được đặt trong luồng và có thể được truyền.

Trong trường hợp này, bộ thu phát rộng dạng số phát hiện dữ liệu từ vị trí thích hợp theo chế độ, và áp dụng phương pháp giải mã phù hợp và thực hiện việc giải mã.

Cụ thể, theo một phương án ví dụ, trong đó chế độ được biểu diễn bởi hai bit và trường truyền tín hiệu TPC được ghi là 00, 01, 10, và 11 được khôi phục, nếu giá trị là 11 được nhận diện từ dữ liệu truyền tín hiệu, thì bộ thu phát rộng dạng số nhận diện các TPC của tất cả các rãnh bao gồm rãnh nhóm M/H của tập hợp M/H cần được nhận. Do đó, nếu thông tin chế độ của tất cả các rãnh là 11 và không có rãnh CMM, thì xác định được chế độ thứ tư là chế độ không tương thích. Do đó, bộ thu phát rộng dạng số có thể giải mã phần đầu MPEG và vùng chặn lẻ trong đó dữ liệu di động mới được đặt, ví dụ, vùng SB5 được mô tả ở trên, theo cùng cách như trong luồng của vùng thân còn lại. Mặt khác, nếu chế độ có thể mở rộng của tất cả các rãnh không phải là 11 hoặc nếu có rãnh CMM, thì bộ thu xác định là chế độ thiết lập là chế độ tương thích, tức là chế độ có thể mở rộng 11a và có thể giải mã phần đầu MPEG và vùng chặn lẻ, tức là vùng SB5 theo cách khác với cách của luồng vùng thân còn lại, tức là theo cách giải mã tương ứng với cách mã hóa của dữ liệu di động mới. TPC của mỗi rãnh và chế độ có thể được nhận diện nhờ bộ giải mã truyền tín hiệu hoặc bộ điều khiển được bố trí riêng.

Theo một phương án ví dụ, trong đó chế độ được biểu diễn bởi ba bit và các bit truyền tín hiệu như 000, 001, 010, 011, và 111 được truyền, bộ thu phát rộng dạng số nhận diện chế độ theo giá trị bit và thực hiện việc giải mã phù hợp.

Bộ truyền phát rộng dạng số kết hợp dữ liệu thường, dữ liệu di động hiện có, và dữ liệu di động mới, theo đó tạo cấu hình luồng vận chuyển, và truyền luồng vận chuyển.

Do đó, bộ thu phát rộng dạng số mà nhận và xử lý luồng vận chuyển có thể được thực hiện theo các dạng khác nhau. Tức là, bộ thu phát rộng dạng số được thực hiện như bộ thu dữ liệu thường để xử lý chỉ dữ liệu thường, bộ thu dữ liệu di động hiện có để xử lý chỉ dữ liệu di động hiện có (ví dụ lĩnh vực liên quan), bộ thu dữ liệu di động mới xử lý chỉ dữ liệu di động mới, và bộ thu chung để xử lý ít nhất hai trong số các dữ liệu này.

Nếu bộ thu phát rộng dạng số là bộ thu dữ liệu thường, thì không có dữ liệu được xử lý ở chế độ thứ tư hoặc thứ năm mà không có tính tương thích không như ở chế độ thứ nhất đến thứ tư có tính tương thích. Do đó, bộ thu phát rộng dạng số có thể không xét đến luồng vận chuyển mà không thể nhận ra và xử lý.

Mặt khác, nếu bộ thu phát rộng dạng số là bộ thu dữ liệu di động hiện có hoặc bộ thu chung có khả năng xử lý dữ liệu di động hiện có và dữ liệu thường, thì bộ thu giải mã dữ liệu thường mà có trong rãnh chỉ bao gồm các gói thường hoặc có trong tất cả hoặc một số gói trong số 38 gói để xử lý dữ liệu thường, và phát hiện dữ liệu di động hiện có có trong các gói khác 38 gói và giải mã dữ liệu di động hiện có để xử lý dữ liệu di động hiện có. Cụ thể, nếu rãnh bao gồm dữ liệu di động mới và chế độ khối là chế độ riêng như đã mô tả ở trên, thì phần quản thể sơ cấp được điền dữ liệu di động hiện có và phần quản thể thứ cấp được điền dữ liệu di động mới, sao cho có thể truyền cả dữ liệu di động hiện có và dữ liệu di động mới trong rãnh đơn. Do đó, nếu chế độ là chế độ có thể mở rộng 11, thì bộ thu giải mã vùng thân trừ SB5 để xử lý dữ liệu di động hiện có. Mặt khác, nếu chế độ là chế độ có thể mở rộng 11a, SB5 không được điền dữ liệu di động mới và theo đó bộ thu giải mã toàn bộ vùng thân để xử lý dữ liệu di động hiện có. Nếu chế độ khối là chế độ đôi, toàn bộ khối chỉ được điền dữ liệu di động 1.1 và theo đó bộ thu không xét đến rãnh tương ứng để xử lý dữ liệu di động hiện có.

Nếu bộ thu phát rộng dạng số là bộ thu dữ liệu di động mới hoặc bộ thu chung có khả năng xử lý tất cả dữ liệu di động mới và dữ liệu khác, thì bộ thu thực hiện việc giải mã theo chế độ khối và chế độ. Tức là, nếu chế độ khối là chế độ riêng và chế độ là chế độ có thể mở rộng 11, thì khối độc lập của vùng SB5 và khối được cấp phát cho dữ liệu di động

mới được giải mã theo phương pháp giải mã tương ứng với phương pháp mã hóa của dữ liệu di động mới, và nếu chế độ là chế độ có thể mở rộng 11a, thì khối được cấp phát cho dữ liệu di động mới được giải mã theo phương pháp giải mã tương ứng với phương pháp mã hóa của dữ liệu di động mới. Mặt khác, nếu chế độ khối là chế độ đôi, thì tất cả các khối được giải mã.

Trên các hình vẽ từ Fig.51 đến Fig.53, bộ điều khiển riêng hoặc bộ giải mã truyền tín hiệu nhận diện chế độ khối và chế độ và điều khiển việc giải mã như đã mô tả ở trên. Cụ thể, nếu hai bit của dữ liệu truyền tín hiệu biểu thị chế độ và giá trị bit 11 được truyền, thì bộ điều khiển hoặc bộ giải mã truyền tín hiệu nhận diện các TPC của tất cả các rãnh bao gồm rãnh gồm nhóm M/H của tập hợp M/H cần được nhận. Do đó, nếu nhận diện được tốc độ dữ liệu thường là 0Mbps, thì xác định được rằng giá trị bit là 11 biểu thị chế độ có thể mở rộng 11 và thực hiện việc giải mã. Mặt khác, nếu chế độ có thể mở rộng của tất cả các rãnh không phải là 11 hoặc nếu có rãnh CMM, tức là nếu tốc độ dữ liệu thường không bằng 0Mbps, thì xác định được là giá trị bit là 11 biểu thị chế độ có thể mở rộng 11a và thực hiện việc giải mã.

Bộ thu phát rộng dạng số trên các hình vẽ từ Fig.51 đến Fig.53 có thể được thực hiện bởi hộp chuyển đổi tín hiệu hoặc ti vi. Tuy nhiên, bộ thu phát rộng dạng số có thể được thực hiện nhờ các loại thiết bị di động khác nhau như điện thoại di động, thiết bị trợ giúp cá nhân dạng số (personal digital assistant-PDA), máy chơi MP3, từ điển điện tử, và máy tính xách tay. Ngoài ra, mặc dù không được thể hiện trên các hình vẽ từ Fig.51 đến Fig.53, nhưng có thể bao gồm thành phần để mở rộng hoặc chuyển đổi dữ liệu đã được giải mã một cách thích hợp và đưa dữ liệu lên trên màn hình có thể có dưới dạng dữ liệu âm thanh và video.

Phương pháp tạo cấu hình luồng của bộ truyền phát rộng dạng số và phương pháp xử lý luồng của bộ thu phát rộng dạng số theo một phương án ví dụ có thể được mô tả có dựa vào các sơ đồ khối và các hình vẽ kết cấu luồng được mô tả ở trên.

Tức là, phương pháp tạo cấu hình luồng của bộ truyền phát rộng dạng số thường bao gồm bước đặt dữ liệu di động trong ít nhất một số gói được cấp phát cho dữ liệu thường trong số tất cả các gói của luồng, và chèn dữ liệu thường vào trong luồng, trong đó dữ liệu di động được đặt, theo đó tạo cấu hình luồng vận chuyển.

Hoạt động đặt dữ liệu di động có thể được thực hiện nhờ bộ tiền xử lý dữ liệu 100 được thể hiện trên các hình vẽ từ Fig.2 đến Fig.4.

Dữ liệu di động có thể được đặt một mình ở các vị trí khác nhau hoặc cùng với dữ liệu thường và dữ liệu di động hiện có như được mô tả theo các phương án ví dụ trên. Tức là, dữ liệu di động và dữ liệu đã biết có thể được đặt theo các cách khác nhau như được giải thích trên các hình vẽ từ Fig.15 đến Fig.40.

Ngoài ra, trong hoạt động tạo cấu hình luồng, luồng vận chuyển có thể được tạo cấu hình bằng cách dồn kênh dữ liệu thường, nó được xử lý riêng với dữ liệu di động, cùng với dữ liệu di động.

Luồng vận chuyển trải qua các xử lý khác nhau như mã hóa RS, đan xen, mã hóa dạng mắt cáo, dồn kênh đồng bộ, và điều biến, và sau đó được truyền tới bộ thu. Hoạt động xử lý luồng vận chuyển có thể được thực hiện nhờ các thành phần của bộ truyền phát rộng dạng số được thể hiện trên Fig.4.

Các phương án thực hiện khác nhau của phương pháp tạo cấu hình luồng có liên quan đến các hoạt động khác nhau của bộ truyền phát rộng dạng số được mô tả ở trên. Do đó, lưu đồ của phương pháp tạo cấu hình luồng được bỏ qua.

Phương pháp xử lý luồng của bộ thu phát rộng dạng số theo một phương án ví dụ có thể bao gồm các bước: nhận luồng vận chuyển được chia thành vùng thứ nhất được cấp phát cho dữ liệu di động hiện có và vùng thứ hai được cấp phát cho dữ liệu thường và trong đó dữ liệu di động được đặt trong ít nhất một phần của vùng thứ hai riêng với dữ liệu di động hiện có, giải điều biến luồng vận chuyển nhận được, cân bằng luồng vận chuyển đã được giải điều biến, và giải mã ít nhất một trong số dữ liệu di động hiện có và dữ liệu di động từ luồng vận chuyển đã được cân bằng.

Luồng vận chuyển nhận được theo phương pháp này có thể là luồng vận chuyển được tạo cấu hình nhờ bộ truyền phát rộng dạng số theo các phương án ví dụ khác nhau và được truyền từ bộ truyền phát rộng dạng số. Tức là, trong luồng vận chuyển, dữ liệu di động có thể được đặt theo các cách khác nhau như được thể hiện trên các hình vẽ từ Fig.15 đến Fig.21 và các hình vẽ từ Fig.29 đến Fig.40. Ngoài ra, dữ liệu đã biết có thể được đặt trong các định dạng khác nhau như được thể hiện trên các hình vẽ từ Fig.22 đến Fig.28.

Các phương án ví dụ khác nhau về phương pháp xử lý luồng có liên quan đến các phương án ví dụ nêu trên của bộ thu phát rộng dạng số.

Các ví dụ về kết cấu của luồng được minh họa trên Fig.15 đến Fig.40 không cố định và có thể được chuyển thành kết cấu khác tùy theo tình huống. Tức là, dữ liệu di động và dữ liệu đã biết có thể được đặt và được mã hóa khối nhờ áp dụng các chế độ khung, các chế độ, và các chế độ khối khác nhau theo tín hiệu điều khiển được sử dụng bởi bộ điều khiển riêng được bố trí trong bộ tiền xử lý dữ liệu 100 hoặc tín hiệu điều khiển được đưa vào từ nguồn bên ngoài. Do đó, nhà cung cấp truyền phát rộng dạng số có thể cung cấp dữ liệu mong muốn, cụ thể, dữ liệu di động có các kích thước khác nhau.

Ngoài ra, dữ liệu di động mới được mô tả ở trên, tức là dữ liệu phiên bản 1.1 có thể giống như dữ liệu di động hiện có, tức là dữ liệu phiên bản 1.0 hoặc có thể là dữ liệu khác được đưa vào từ nguồn khác. Ngoài ra, nhiều dữ liệu phiên bản 1.1 có thể có trong rãnh đơn và được truyền cùng nhau. Do đó, người sử dụng bộ thu phát rộng dạng số có thể xem dữ liệu với các định dạng khác nhau mà họ muốn xem.

Phương pháp xử lý khối

Các phương án ví dụ nêu trên có thể được cải biến khác nhau.

Ví dụ, bộ xử lý khối 120 trên Fig.4 được mô tả ở trên có thể thực hiện việc mã hóa khối nhờ kết hợp một cách phù hợp dữ liệu di động hiện có, dữ liệu thường, dữ liệu di động mới, và dữ liệu đã biết được đặt trong luồng. Dữ liệu di động mới và dữ liệu đã biết có thể được đặt trong không chỉ ít nhất một phần của dữ liệu thường vùng được cấp phát cho dữ liệu thường, mà còn ít nhất một phần của vùng dữ liệu di động hiện có được cấp phát cho dữ liệu di động hiện có. Tức là, dữ liệu thường, dữ liệu di động mới, và dữ liệu di động hiện có cùng tồn tại.

Fig.54 minh họa một ví dụ về định dạng luồng sau khi đan xen. Trên Fig.54, luồng bao gồm nhóm dữ liệu di động bao gồm 208 phân đoạn dữ liệu. Năm phân đoạn thứ nhất tương ứng với dữ liệu chẵn lẻ RS và theo đó được loại khỏi nhóm dữ liệu di động. Do đó, nhóm dữ liệu di động gồm 203 phân đoạn dữ liệu được chia thành 15 khối dữ liệu di động. Cụ thể, nhóm dữ liệu di động bao gồm các khối từ B1 đến B10 và các khối từ SB1 đến SB5. Trong số các khối, các khối từ B1 đến B10 có thể tương ứng với dữ liệu di động được đặt trong vùng dữ liệu di động hiện có như được thể hiện trên Fig.8. Mặt khác, các khối từ SB1 đến SB5 có thể tương ứng với dữ liệu di động mới được cấp phát cho vùng dữ liệu thường hiện có. Khối SB5 bao gồm phần đầu MPEG và phần chẵn lẻ RS để tương thích ngược.

Mỗi một trong số các khối từ B1 đến B10 bao gồm 16 phân đoạn, mỗi khối SB1 và SB4 bao gồm 31 phân đoạn, và mỗi khối SB2 và SB3 bao gồm 14 phân đoạn.

Các khối này, tức là các khối từ B1 đến B10 và các khối từ SB1 đến SB5 được mã hóa khối nhờ được kết hợp theo các định dạng khác nhau.

Tức là, như đã mô tả ở trên, chế độ khối có thể được thiết lập là các giá trị khác nhau như 00 và 01. Bảng sau thể hiện các khối SCB và chiều dài khối đầu ra SCCC (SCCC output block length-SOBL) và chiều dài khối đầu vào SCCC (SCCC input block length-SIBL) của mỗi khối SCB nếu chế độ khối được thiết lập là “00”:

Bảng 10

Khối SCCC	SOBL	SIBL tỷ lệ $\frac{1}{2}$	Tỷ lệ $\frac{1}{4}$
SCB1(B1)	528	264	132
SCB2(B2)	1536	768	384
SCB3(B3)	2376	1188	594
SCB4(B4)	2388	1194	597
SCB5(B5)	2772	1386	693
SCB6(B6)	2472	1236	618
SCB7(B7)	2772	1386	693
SCB8(B8)	2508	1254	627
SCB9(B9)	1416	708	354
SCB10(B10)	480	240	120

Theo bảng 10, các khối từ B1 đến B10 trở thành các khối từ SCB1 đến SCB10.

Nếu chế độ khối được thiết lập là “01”, thì mỗi khối SCB và SOBL và SIBL của mỗi khối SCB như sau:

Bảng 11

Khối SCCC	SOBL	SIBL tỷ lệ $\frac{1}{2}$	Tỷ lệ $\frac{1}{4}$
SCB1(B1+B6)	3000	1500	750
SCB2(B2+B7)	4308	2154	1077
SCB3(B3+B8)	4884	2442	1221
SCB4(B4+B9)	3804	1902	951
SCB5(B5+B10)	3252	1626	813

Theo bảng 11, các khối B1 và B6 được kết hợp để tạo cấu hình khối SCB1, và các khối B2 và B7, các khối B3 và B8, các khối B4 và B9, và các khối B5 và B10 được kết hợp để tạo cấu hình các khối SCB2, SCB3, SCB4, và SCB5, theo cách tương ứng.

Ngoài ra, chiều dài khối đầu vào là khác nhau tùy theo liệu tỷ lệ dữ liệu là tỷ lệ 1/2 hoặc tỷ lệ 1/4.

Hoạt động tạo cấu hình cho khối SCB bằng cách kết hợp các khối từ B1 đến B10 có thể được thực hiện nếu dữ liệu di động mới không được đặt, tức là ở chế độ CMM.

Ở chế độ SFCMM trong đó đặt dữ liệu di động mới, các khối được kết hợp khác nhau để tạo cấu hình khối SCB. Tức là, có thể thực hiện việc mã hóa khối SCCC nhờ kết hợp dữ liệu di động hiện có và dữ liệu di động mới. Các bảng 12 và 13 sau đây minh họa một ví dụ về các khối được kết hợp khác nhau theo chế độ khung RS và chế độ rãnh:

Bảng 12

Chế độ khung	00		01	
RS				
Chế độ khối	00	01	00	01
Chế độ khối SCCC	Chế độ khối SCCC đơn	Chế độ khối SCCC đôi	Chế độ khối SCCC đơn	Chế độ khối SCCC đôi
Mô tả SCB	Đầu vào SCB, các khối M/H	Đầu vào SCB, các khối M/H	Đầu vào SCB, các khối M/H	Đầu vào SCB, các khối M/H
SCB1	B1	B1+B6+SB3	B1	B1+SB3+B9+SB1
SCB2	B2	B2+B7+SB4	B2	B2+SB4+B10+SB2
SCB3	B3	B3+B8	B9+SB1	
SCB4	B4	B4+B9+SB1	B10+SB2	
SCB5	B5	B5+B10+SB2	SB3	
SCB6	B6		SB4	
SCB7	B7			
SCB8	B8			
SCB9	B9+SB1			
SCB10	B10+SB2			
SCB11	SB3			
SCB12	SB4			

Trong bảng 12, chế độ khung RS chỉ thông tin biểu thị liệu một rãnh bao gồm một quần thể (trong trường hợp của chế độ khung RS 00) hoặc liệu một rãnh bao gồm nhiều quần thể như quần thể sơ cấp và quần thể thứ cấp (trong trường hợp của khung RS 01). Ngoài ra, chế độ khối SCCC chỉ thông tin biểu thị liệu việc xử lý khối SCCC riêng được thực hiện hoặc liệu việc xử lý khối SCCC được thực hiện bằng cách kết hợp nhiều khối, như chế độ khối được mô tả ở trên.

Bảng 12 biểu thị trường hợp trong đó chế độ rãnh là 00. Chế độ rãnh là thông tin biểu thị tiêu chuẩn dựa trên đó phần đầu và cuối của rãnh được phân biệt với nhau. Tức là, nếu chế độ rãnh là 00, thì một phần bao gồm các khối từ B1 đến B10 và các khối từ SB1 đến SB5 cho cùng rãnh do chúng được xác định là một rãnh, và nếu chế độ rãnh là 01, thì

các khối B1 và B2 được gửi tới rãnh trước và các khối B1 và B2 của rãnh tiếp theo có trong rãnh hiện tại, sao cho một phần bao gồm tổng cộng 15 khối được xác định là một rãnh. Chế độ rãnh có thể có các tên khác nhau theo phiên bản của tài liệu chuẩn. Ví dụ, chế độ rãnh có thể được gọi chế độ mở rộng khối. Điều này sẽ được mô tả sau đây.

Theo bảng 12, nếu chế độ khung RS là 00 và chế độ khối SCCC là 00, thì các khối từ B1 đến B8 được sử dụng lần lượt làm các khối từ SCB1 đến SCB8, các khối B9 và SB1 được kết hợp để tạo cấu hình khối SCB9, các khối B10 và SB2 được kết hợp để tạo cấu hình khối SCB10, và các khối SB3 và SB4 được sử dụng lần lượt làm các khối SCB11 và SCB12. Mặt khác, nếu chế độ khối SCCC là 01, thì các khối B1, B6, và SB3 được kết hợp và được sử dụng làm khối SCB1, và B2+B7+SB4 được sử dụng làm khối SB2 và B3+B8, B4+B9+SB1, và B5+B10+SB2 được sử dụng lần lượt làm các khối SCB3, SCB4, và SCB5.

Mặt khác, nếu khung RS là 01 và chế độ khối SCCC là 00, thì các khối B1, B2, B9+SB1, B10+SB2, SB3, và SB4 được sử dụng lần lượt làm các khối từ SCB1 đến SCB6. Nếu chế độ khối SCCC là 01, thì B1+SB3+B9+SB1 được sử dụng làm khối SCB1 và B2+SB4+B10+SB2 được sử dụng làm khối SCB2.

Nếu chế độ rãnh là 01 và dữ liệu di động mới được đặt theo các chế độ thứ nhất đến thứ ba được mô tả ở trên, khối SCCC có thể được kết hợp như sau:

Bảng 13

Khung RS	00		01	
Chế độ	00	01	00	01
Chế độ khối SCCC	Chế độ khối SCCC đơn	Chế độ khối SCCC đôi	Chế độ khối SCCC đơn	Chế độ khối SCCC đôi
Mô tả SCB	Đầu vào SCB, các khối M/H	Đầu vào SCB, các khối M/H	Đầu vào SCB, các khối M/H	Đầu vào SCB, Các khối M/H
SCB1	B1+SB3	B1+B6+SB3	B9+SB1	
SCB2	B2+SB4	B2+B7+SB4	B10+SB2	
SCB3	B3	B3+B8		
SCB4	B4	B4+B9+SB1		
SCB5	B5	B5+B10+SB2		
SCB6	B6			
SCB7	B7			
SCB8	B8			
SCB9	B9+SB1			
SCB10	B10+SB2			

Theo bảng 13, các khối từ B1 đến B10 và các khối từ SB1 đến SB5 có thể được kết hợp theo các cách khác nhau theo điều kiện thiết lập như chế độ khung RS và chế độ khối SCCC.

Nếu chế độ rãnh là 01 và dữ liệu di động mới được đặt trong toàn bộ vùng dữ liệu thường theo chế độ thứ tư được mô tả ở trên, thì các khối SCB có thể được tạo cấu hình theo các kết hợp khác nhau như sau:

Bảng 14

Khung RS	00		01	
Chế độ	00	01	00	01
Chế độ khối SCCC	Chế độ khối SCCC đơn	Chế độ khối SCCC đôi	Chế độ khối SCCC đơn	Chế độ khối SCCC đôi
Mô tả SCB	Đầu vào SCB, các khối M/H	Đầu vào SCB, các khối M/H	Đầu vào SCB, các khối M/H	Đầu vào SCB, Các khối M/H
SCB1	B1+SB3	B1+B6+SB3+SB5	B1+SB3	B1+SB3+B9+SB1
SCB2	B2+SB4	B2+B7+SB4	B2+SB4	B2+SB4+B10+SB2
SCB3	B3	B3+B8	B9+SB1	
SCB4	B4	B4+B9+SB1	B10+SB2	
SCB5	B5	B5+B10+SB2		
SCB6	B6+SB5			
SCB7	B7			
SCB8	B8			
SCB9	B9+SB1			
SCB10	B10+SB2			

Như đã mô tả ở trên, mỗi trong số dữ liệu di động hiện có, dữ liệu thường, và dữ liệu di động mới được chia thành các khối và các khối được kết hợp khác nhau theo chế độ, theo đó tạo cấu hình khối SCCC. Do đó, các khối SCCC được kết hợp để tạo cấu hình khung RS.

Việc kết hợp và mã hóa các khối có thể được thực hiện trong bộ tiền xử lý dữ liệu 100 như được minh họa trong các phương án ví dụ nêu trên. Cụ thể, bộ xử lý khối 120 trong bộ tiền xử lý dữ liệu 100 kết hợp các khối và thực hiện việc mã hóa khối. Các xử lý khác trừ phương pháp kết hợp đã được mô tả theo các phương án ví dụ trên và giải thích trùng lặp được bỏ qua.

Tỷ lệ mã hóa để mã hóa các khối SCCC, tức là tỷ lệ mã hóa ngoài SCCC có thể được xác định theo cách khác theo chế độ mã hóa ngoài. Cụ thể, tỷ lệ mã hóa được xác định như sau:

Bảng 15

Chế độ mã hóa ngoài SCCC	Mô tả
00	Tỷ lệ mã hóa ngoài của khối SCCC là tỷ lệ 1/2
01	Tỷ lệ mã hóa ngoài của khối SCCC là tỷ lệ 1/4
10	Tỷ lệ mã hóa ngoài của khối SCCC là tỷ lệ 1/3
11	Dự trữ

Như được mô tả trong bảng 15, chế độ mã hóa ngoài SCCC có thể được thiết lập là các giá trị khác nhau như 00, 01, 10, và 11. Nếu chế độ mã hóa ngoài SCCC là 00, thì khối SCCC được mã hóa ở tỷ lệ mã hóa là 1/2, nếu chế độ mã hóa ngoài SCCC là 01, thì khối SCCC được mã hóa ở tỷ lệ mã hóa là 1/4, và nếu chế độ mã hóa ngoài SCCC là 10, thì khối SCCC được mã hóa ở tỷ lệ mã hóa là 1/3. Tỷ lệ mã hóa có thể thay đổi khác nhau theo phiên bản của chuẩn. Tỷ lệ mã mới được thêm có thể được gán cho chế độ mã hóa ngoài SCCC 11. Mỗi tương quan thích ứng giữa chế độ mã hóa ngoài SCCC và tỷ lệ mã hóa có thể thay đổi. Bộ tiền xử lý dữ liệu 100 có thể mã hóa khối SCCC ở tỷ lệ mã hóa phù hợp theo điều kiện thiết lập của chế độ mã hóa ngoài. Điều kiện thiết lập của chế độ mã hóa ngoài có thể được thông báo nhờ bộ điều khiển 310 hoặc thành phần khác hoặc có thể được nhận diện thông qua kênh truyền tín hiệu riêng. Tỷ lệ mã hóa 1/3 chỉ tỷ lệ tại đó 1 bit được đưa vào và 3 bit được đưa ra, và bộ mã hóa có thể được tạo cấu hình khác nhau. Ví dụ, bộ mã hóa có thể được tạo cấu hình theo kết hợp của tỷ lệ mã hóa 1/2 và tỷ lệ mã hóa 1/4. Bộ mã hóa có thể được tạo cấu hình bằng cách đánh thùng đầu ra của bộ mã hóa chập 4 trạng thái.

Chế độ mở rộng khối: BEM

Như đã mô tả ở trên, các khối hiện có trong rãnh được mã hóa theo cách khác theo chế độ rãnh hoặc chế độ mở rộng khối. Nếu chế độ mở rộng khối là 00, thì một phần bao gồm các khối từ B1 đến B10 và các khối từ SB1 đến SB5 cho cùng một khối được xác định là một rãnh như đã mô tả ở trên, và nếu chế độ mở rộng khối là 01, thì các khối B1 và B2 được gửi tới rãnh trước và các khối B1 và B2 của rãnh tiếp theo có trong rãnh hiện tại, sao cho phần bao gồm tổng cộng 15 khối được xác định là một rãnh.

Các vùng nhóm có thể được phân loại bởi các khối trong rãnh. Ví dụ, bốn khối B4 đến B7 được nhóm thành vùng nhóm A, hai khối B3 và B8 được nhóm thành vùng nhóm B, hai khối B2 và B9 được nhóm thành vùng nhóm C, và hai khối B1 và B10 được nhóm thành vùng nhóm D. Bốn khối từ SB1 đến SB4, được tạo ra khi 38 gói, là vùng dữ liệu thường, được đan xen, có thể được nhóm thành vùng nhóm E.

Nếu chế độ mở rộng khối của khối nhất định là 01, các vùng nhóm A và B bao gồm các khối B3 đến B8 có thể được xác định là quần thể sơ cấp. Các khối B1 và B2 được gửi tới rãnh trước, và các khối B9 và B10, các khối từ SB1 đến SB4, và các khối B1 và B2 của rãnh tiếp theo có trong các vùng nhóm C, D, và E, và các vùng nhóm C, D, và E có thể được xác định là quần thể thứ cấp. Tương tự với quần thể sơ cấp, quần thể thứ cấp có thể điền vùng đầu/đuôi bằng dữ liệu hướng dẫn dài có chiều dài tương ứng với một phân đoạn dữ liệu, và theo đó có thể cải thiện hiệu quả thu nhận của vùng đầu/đuôi lên tới mức tương đương với mức của vùng thân.

Nếu chế độ mở rộng khối của gói nhất định là 00, thì quần thể sơ cấp giống như trong trường hợp của BEM 01, nhưng quần thể thứ cấp thì khác. Quần thể thứ hai có thể được xác định là bao gồm các khối B1 và B2 của rãnh hiện tại, các khối B9 và B10, và các khối từ SB1 đến SB4. Quần thể thứ cấp này có vùng đầu/đuôi dạng răng cưa không như quần thể sơ cấp và theo đó không thể điền vùng đầu/đuôi bằng dữ liệu hướng dẫn dài. Theo đó, hiệu suất nhận của vùng đầu/đuôi kém hơn hiệu suất của vùng thân.

Nếu hai rãnh nhất định liền kề nhau ở chế độ BEM 00, thì một phần tại đó vùng đầu/đuôi dạng răng cưa của các rãnh giao nhau có thể được điền bởi dữ liệu hướng dẫn dài. Như được thể hiện trên Fig.64 và Fig.65, trong vùng tại đó các răng cưa của hai rãnh liền kề của chế độ BEM 00 được ghép khớp với nhau, phần hướng dẫn được phân đoạn của các rãnh được nối sao cho phần hướng dẫn dài có chiều dài tương đương với một phân đoạn dữ liệu có thể được tạo ra. Trên Fig.64 và Fig.65, vị trí bộ mã hóa dạng mắt cáo byte khởi tạo và vị trí byte đã biết được thể hiện.

Khi khung M/H được tạo cấu hình theo loại dịch vụ, rãnh được điền dữ liệu di động mới (rãnh SFCMM) có thể được đặt gần với rãnh được điền dữ liệu di động hiện có (rãnh SMM) hoặc rãnh trong đó 156 gói chỉ được điền dữ liệu thường (toàn bộ rãnh chính). Ở thời điểm này, nếu chế độ BEM của rãnh SFCMM là 00, thì kết hợp là có thể kể cả nếu rãnh CMM hoặc toàn bộ rãnh chính được đặt như rãnh liền kề. Nếu rãnh BEM 00 trong số 16 rãnh trong khung phụ M/H được đặt trong rãnh #0 và rãnh CMM được đặt trong rãnh #1, thì việc mã hóa khối được thực hiện bằng cách kết hợp các khối từ B1 đến B10 và các khối từ SB1 đến SB4 trong rãnh #0. Tương tự, trong rãnh #1, việc mã hóa khối được thực hiện bằng cách kết hợp các khối từ B1 đến B10 trong rãnh #1.

Nếu chế độ BEM của rãnh SFCMM là 01 và rãnh CMM hoặc toàn bộ rãnh chính được đặt như rãnh liền kề, thì vùng mô côi phải được tính đến. Vùng mô côi chỉ vùng mà khó sử dụng trong rãnh bất kỳ vì nhiều rãnh thuộc các loại khác nhau được đặt liên tục.

Ví dụ, nếu rãnh BEM 01 trong số 16 rãnh trong khung phụ M/H được đặt trong rãnh #0 và rãnh CMM được đặt trong rãnh #1, thì các khối B1 và B2 trong rãnh #0 được gửi tới rãnh trước và thực hiện việc mã hóa khối bằng cách bao gồm các khối từ B3 đến B10 và các khối từ SB1 đến SB4 và các khối B1 và B2 của rãnh tiếp theo. Tức là, hai rãnh được điền dữ liệu di động 1.0 và dữ liệu di động 1.1 không tương thích với nhau phải được xử lý để không chịu cản trở giữa đó bởi phương pháp mã hóa khối của BEM 01.

Rãnh của BEM 00 và rãnh của BEM 01 có thể được thiết lập sao cho chúng không thể được kết hợp và được sử dụng cùng nhau. Mặt khác, trong trường hợp của chế độ BEM 01, chế độ CMM, chế độ BEM 01, và toàn bộ chế độ chính các rãnh có thể được kết hợp để sử dụng. Trong trường hợp này, vùng mà khó sử dụng do sự khác biệt về chế độ có thể được coi là vùng mờ côi.

Vùng mờ côi

Vùng mờ côi để ngăn giao nhau giữa hai rãnh thay đổi theo rãnh nào liền kề với rãnh của BEM 01 hoặc theo thứ tự của các rãnh liền kề.

Trước tiên, nếu rãnh thứ (i) là rãnh CMM và rãnh thứ (i+1), là rãnh tiếp theo, là rãnh BEM 01, thì các khối B1 và B2 hiện có trong vùng đầu của rãnh BEM 01 được gửi tới rãnh trước. Tuy nhiên, do rãnh CMM không được mã hóa khối bằng cách sử dụng các khối B1 và B2 của rãnh tiếp theo, vùng của các khối B1 và B2 của rãnh thứ (i+1) không được cấp phát cho dịch vụ bất kỳ. Vùng này được xác định là loại mờ côi 1. Tương tự, nếu rãnh thứ (i) là toàn bộ rãnh chính và rãnh thứ (i+1), là rãnh tiếp theo, là rãnh BEM 01, thì vùng của các khối B1 và B2 của rãnh (i+1) không được cấp phát cho dịch vụ bất kỳ và theo đó loại mờ côi 1 cũng được tạo ra.

Thứ hai, nếu rãnh thứ (i) là rãnh BEM 01 và rãnh thứ (i+1), là rãnh tiếp theo, là rãnh CMM, rãnh BEM 01 thứ (i) thực hiện việc mã hóa khối bằng cách sử dụng các khối B1 và B2 của rãnh tiếp theo và theo đó rãnh tiếp theo không thể sử dụng các khối B1 và B2. Tức là, rãnh CMM, là rãnh tiếp theo, được thiết lập là chế độ khung kép và cấp phát dịch vụ chỉ cho quần thể sơ cấp và làm trống quần thể thứ cấp. Ở thời điểm này, khối B1 và B2 của quần thể thứ cấp, bao gồm các khối B1 và B2 và các khối B9 và B10, được gửi tới rãnh thứ (i), là rãnh trước, để sử dụng, nhưng vùng của các khối B9 và B10 không được cấp phát cho dịch vụ bất kỳ. Vùng này được xác định là loại mờ côi 2.

Cuối cùng, nếu rãnh thứ (i) liền kề với rãnh BEM 01 và rãnh thứ (i+1) liền kề với toàn bộ rãnh chính, loại mô côi 3 được tạo ra. Nếu rãnh BEM 01 mang vùng tương ứng với các khối B1 và B2 từ toàn bộ rãnh chính, là rãnh tiếp theo, và sử dụng vùng này, thì dữ liệu thường không được truyền tới 32 gói trên, trong đó vùng của các khối B1 và B2 có mặt trong số 156 rãnh tiếp theo. Tức là, một số gói trong 32 gói đầu tiên của rãnh tiếp theo tương ứng với vùng của các khối B1 và B2 và được sử dụng trong rãnh BEM 01, là rãnh thứ (i), nhưng các gói còn lại mà không tương ứng với vùng của các khối B1 và B2 không được cấp phát cho dịch vụ bất kỳ. Vùng còn lại mà không tương ứng với vùng của các khối B1 và B2 trong số 32 gói đầu tiên của rãnh tiếp theo được phân bố trên một phần của các vùng nhóm A và B theo định dạng nhóm sau khi đan xen. Do đó, loại mô côi 3 được tạo ra trong vùng thân của rãnh tiếp theo.

Phương pháp sử dụng phần mô côi

Vùng mô côi có thể bao gồm dữ liệu di động mới, dữ liệu hướng dẫn hoặc byte giả khi cần. Nếu vùng mô côi được điền dữ liệu di động mới, thì sự có mặt của dữ liệu tương ứng và loại dữ liệu và thông tin truyền tín hiệu cần thiết để bộ thu để nhận ra và giải mã có thể được thêm.

Nếu vùng mô côi được điền dữ liệu hướng dẫn, thì bộ mã hóa dạng mắt cáo được khởi tạo theo chuỗi hướng dẫn để được tạo ra và byte đã biết được xác định sao cho bộ thu có thể nhận ra chuỗi hướng dẫn.

Bảng 16 minh họa một ví dụ về vị trí của phần mô côi nếu BEM=01 và phương pháp sử dụng chúng:

Bảng 16

Rãnh (i)	Rãnh (i+1)	Tồn thất (byte)	Vị trí phần mô côi	Sử dụng phần mô côi
CMM	BEM=01	1850	Phần đầu rãnh (i+1)	Phần hướng dẫn (141/89)
BEM=01	CMM	1570	Phần đuôi rãnh (i+1)	Phần hướng dẫn (195/141)
Toàn bộ phần chính	BEM=01	1850	Phần đầu rãnh (i+1)	Phần hướng dẫn (141/89)
BEM=01	Toàn bộ phần chính	3808	Phần rãnh (i+1) của Vùng A và B	Phần giả

Nếu BEM=01, vùng mô côi có thể được tạo ra như trong bảng 17 sau đây:

Bảng 17

Loại mô côi	Rãnh (i)	Rãnh (i+1)	Tồn thất (byte)	Vị trí vùng mô côi	Sử dụng phần mô côi (Byte đã biết/Byte khởi tạo)
Loại 1	Rãnh CMM	Rãnh SFCMM có BEM=01	1618	Phần đầu rãnh (i+1)	Phần hướng dẫn (210/252)
Loại 2	Rãnh SFCMM có BEM=01	Rãnh CMM	1570	Phần đuôi rãnh (i+1)	Phần hướng dẫn (195/141)
Loại 1	Rãnh M/H chỉ có các gói chính	Rãnh SFCMM có BEM=01	1618	Phần đầu rãnh (i+1)	Phần hướng dẫn (210/252)
Loại 3	Rãnh SFCMM có BEM=01	Rãnh M/H chỉ có các gói chính	3808	Phần rãnh (i+1) của Vùng A và B	Phần giả

Như được thể hiện trong bảng 17 trên, vùng mô côi có thể được tạo ra ở các vị trí khác nhau và với các kích thước khác nhau theo các loại của hai rãnh liên tiếp. Ngoài ra, vùng mô côi này có thể được sử dụng cho các mục đích khác nhau như dữ liệu hướng dẫn hoặc phần giả. Mặc dù các bảng 16 và 17 không minh họa vùng mô côi trong đó dữ liệu di động được sử dụng, nhưng dữ liệu di động có thể được sử dụng trong vùng mô côi.

Nếu vùng mô côi được sử dụng, thì phương pháp xử lý luồng của bộ truyền phát rộng dạng số có thể bao gồm bước tạo cấu hình luồng, trong đó nhiều rãnh thuộc các loại khác nhau trong đó ít nhất một trong số dữ liệu di động hiện có, dữ liệu thường, và dữ liệu di động mới được đặt trong định dạng khác được bố trí liên tiếp; và mã hóa và đan xen luồng để được đưa ra làm luồng vận chuyển. Hoạt động truyền có thể được thực hiện nhờ bộ phận kích thích 400 của bộ truyền phát rộng dạng số được mô tả ở trên.

Hoạt động tạo cấu hình luồng có thể đặt ít nhất một trong số dữ liệu di động mới, dữ liệu hướng dẫn, và dữ liệu giả trong vùng mô côi mà dữ liệu không được cấp phát cho nó do sự khác biệt về định dạng giữa các rãnh liên tiếp. Phương pháp sử dụng vùng mô côi được mô tả ở trên.

Vùng mô côi có thể thuộc các loại khác nhau như đã mô tả ở trên.

Tức là, nếu rãnh CMM và rãnh SFCMM của chế độ mở rộng khối 01 được đặt tuần tự hoặc nếu toàn bộ rãnh chính chỉ bao gồm dữ liệu thường và rãnh SFCMM của chế

độ mở rộng khối 01 được đặt tuần tự, thì vùng mờ còi thuộc loại 1 có thể được tạo ra ở phần đầu của rãnh SFCMM.

Nếu rãnh SFCMM của chế độ mở rộng khối 01 và rãnh CMM được đặt tuần tự, thì vùng mờ còi thuộc loại 2 có thể được tạo ra ở phần đuôi của rãnh CMM.

Nếu rãnh SFCMM của chế độ mở rộng khối 01 và toàn bộ rãnh chính chỉ bao gồm dữ liệu thường được đặt tuần tự, thì vùng mờ còi thuộc loại 3 có thể được tạo ra ở phần thân của toàn bộ rãnh chính.

Như đã mô tả ở trên, rãnh CMM là rãnh trong đó dữ liệu di động hiện có được đặt trong vùng thứ nhất được cấp phát cho dữ liệu di động hiện có và dữ liệu thường được đặt trong vùng thứ hai được cấp phát cho dữ liệu thường.

Rãnh SFCMM là rãnh trong đó dữ liệu di động mới được đặt trong một phần của toàn bộ vùng bao gồm vùng thứ nhất và vùng thứ hai theo chế độ xác định.

Fig.58 minh họa kết cấu luồng biểu thị vùng mờ còi thuộc loại 1 sau khi đan xen, và Fig.59 minh họa kết cấu luồng biểu thị vùng mờ còi thuộc loại 1 trước khi đan xen.

Fig.60 minh họa kết cấu luồng biểu thị vùng mờ còi thuộc loại 2 sau khi đan xen, và Fig.61 minh họa kết cấu luồng biểu thị vùng mờ còi thuộc loại 2 trước khi đan xen.

Fig.62 minh họa kết cấu luồng biểu thị vùng mờ còi thuộc loại 3 sau khi đan xen, và Fig.63 minh họa kết cấu luồng biểu thị vùng mờ còi thuộc loại 3 trước khi đan xen.

Trên các hình vẽ này, phần mờ còi có thể được tạo ra ở các vị trí khác nhau theo mẫu đặt của rãnh.

Luồng vận chuyển được truyền từ bộ truyền phát rộng dạng số được nhận và xử lý nhờ bộ thu phát rộng dạng số.

Tức là, bộ thu phát rộng dạng số bao gồm bộ phận thu để nhận luồng vận chuyển được mã hóa và đan xen với nhiều rãnh thuộc các loại khác nhau, trong đó ít nhất một trong số dữ liệu di động hiện có, dữ liệu thường, và dữ liệu di động mới được đặt trong định dạng khác, được bố trí liên tiếp, bộ giải điều biến giải điều biến luồng vận chuyển, bộ cân bằng cân bằng luồng vận chuyển đã được giải điều biến, và bộ phận giải mã giải mã dữ liệu di động mới từ luồng được cân bằng. Luồng vận chuyển có thể bao gồm vùng mờ còi mà dữ liệu không được cấp phát cho nó do sự khác biệt về định dạng giữa các rãnh

liên tiếp, và ít nhất một trong số dữ liệu di động mới, dữ liệu hướng dẫn, và dữ liệu giả có thể được đặt trong vùng mờ còi.

Bộ thu phát rộng dạng số chỉ có thể phát hiện dữ liệu mà nó có thể xử lý theo loại bộ thu phát rộng dạng số, tức là tùy theo liệu bộ thu phát rộng dạng số là bộ thu dữ liệu thường, bộ thu chuyên dụng CMM, bộ thu chuyên dụng SFCMM, hoặc bộ thu chung.

Ngoài ra, như đã mô tả ở trên, sự có mặt/không có mặt của dữ liệu trong vùng mờ còi và loại dữ liệu có thể được thông báo bằng cách sử dụng thông tin truyền tín hiệu. Tức là, bộ thu phát rộng dạng số còn có thể bao gồm bộ giải mã truyền tín hiệu giải mã thông tin truyền tín hiệu và nhận diện sự có mặt/không có mặt của dữ liệu trong vùng mờ còi và loại dữ liệu.

Dữ liệu truyền tín hiệu

Thông tin như số lượng của các gói dữ liệu di động hiện có hoặc mới được thêm vào hoặc tỷ lệ mã hóa có thể được truyền tới bộ thu như thông tin truyền tín hiệu.

Ví dụ, thông tin truyền tín hiệu này có thể được truyền bằng cách sử dụng vùng dự trữ của TPC. Trong trường hợp này, khung phụ nhất định truyền thông tin về khung hiện tại và một khung phụ khác truyền thông tin về khung tiếp theo, sao cho có thể thực hiện “Truyền tín hiệu trước”. Tức là, thông số TPC định trước và dữ liệu FIC có thể được tạo tín hiệu trước.

Cụ thể, như được thể hiện trên Fig.55, một khung M/H có thể được chia thành 5 khung phụ. Các thông số TPC như `sub_frame_number`, `slot_number`, `parade_id`, `parade_repetition_cycle_minus_1`, `parade_continuity_counter`, `fic_version`, và chế độ rãnh thêm vào được mô tả ở trên có thể truyền thông tin về khung hiện tại trong năm khung phụ. Ngoài ra, các thông số TPC như SGN, `number_of_groups_minus_1`, các chế độ FEC, TNoG, số lượng của các gói dữ liệu di động hiện có hoặc mới được thêm vào được mô tả ở trên, và tỷ lệ mã hóa có thể được ghi lại theo cách khác nhau theo số lượng khung phụ. Tức là, các khung phụ #0 và #1 có thể truyền thông tin về khung hiện tại và các khung phụ #2, #3, và #4 có thể truyền thông tin về khung tiếp theo tính đến chu kỳ lặp tập hợp (parade repetition cycle-PRC). Trong trường hợp của TNoG, các khung phụ #0 và #1 chỉ truyền thông tin về khung hiện tại và các khung phụ #2, #3, #4 truyền thông tin về khung hiện tại và khung tiếp theo.

Cụ thể, thông tin TPC có thể được tạo cấu hình như trong bảng 18 sau đây:

Bảng 18

Cú pháp	Số bit	Định dạng
TPC_data {		
sub-frame_number	3	uimsbf
slot_number	4	uimsbf
parade_id	7	uimsbf
if (sub-frame_number ≤ 1){		
current_starting_group_number	4	uimsbf
current_number_of_groups_minus_1 }	3	uimsbf
if (sub-frame_number ≥ 2){		
next_starting_group_number	4	uimsbf
next_number_of_groups_minus_1 }	3	uimsbf
parade_repetition_cycle_minus_1	3	uimsbf
if(sub-frame_number ≤ 1){		
current_rs_frame_mode	2	bslbf
current_rs_code_mode_primary	2	bslbf
current_rs_code_mode_secondary	2	bslbf
current_sccc_block_mode	2	bslbf
current_sccc_outer_code_mode_a	2	bslbf
current_sccc_outer_code_mode_b	2	bslbf
current_sccc_outer_code_mode_c	2	bslbf
current_sccc_outer_code_mode_d }	2	bslbf
if(sub-frame_number ≥ 2){		
next_rs_frame_mode	2	bslbf
next_rs_code_mode_primary	2	bslbf
next_rs_code_mode_secondary	2	bslbf
next_sccc_block_mode	2	bslbf
next_sccc_outer_code_mode_a	2	bslbf
next_sccc_outer_code_mode_b	2	bslbf
next_sccc_outer_code_mode_c	2	bslbf
next_sccc_outer_code_mode_d }	2	bslbf
fic_version	5	uimsbf
parade_continuity_counter	4	uimsbf
if(sub-frame_number ≤ 1){		
current_TNoG	5	uimsbf
reserved }	5	bslbf
if(sub-frame_number ≥ 2){		
next_TNoG	5	uimsbf
current_TNoG }	5	uimsbf
if(sub-frame_number ≤ 1){		
current_sccc_outer_code_mode_e	2	bslbf
current_scalable_mode }	2	uimsbf
if(sub-frame_number ≥ 2){		
next_sccc_outer_code_mode_e	2	bslbf
next_scalable_mode }	2	uimsbf
slot mode	2	uimsbf
reserved	10	bslbf
tpc_protocol_version	5	bslbf
}		

Như được thể hiện trong bảng 18, nếu số lượng của khung phụ nhỏ hơn hoặc bằng 1, tức là #0 hoặc #1, thì nhiều loại thông tin về khung M/H hiện tại được truyền, và nếu số lượng của khung phụ lớn hơn hoặc bằng 2, tức là #2, #3, và #4, thì nhiều loại thông tin về khung M/H tiếp theo có thể được truyền có tính đến PRC. Do đó, thông tin về khung tiếp theo có thể được biết trước sao cho tốc độ xử lý có thể được cải thiện hơn nữa.

Kết cấu của bộ thu có thể được thay đổi theo biến đổi được mô tả ở trên của một phương án ví dụ.

Tức là, bộ thu giải mã dữ liệu mà được mã hóa khối nhờ được kết hợp theo cách khác nhau theo chế độ khối, nhờ đó khôi phục dữ liệu di động hiện có, dữ liệu thường, và dữ liệu di động mới. Ngoài ra, thông tin truyền tín hiệu về khung tiếp theo được nhận diện trước, sao cho việc xử lý có thể được chuẩn bị theo thông tin đã được nhận diện.

Cụ thể, trong bộ thu phát rộng dạng số có kết cấu được thể hiện trên Fig.51, bộ phận thu 5100 nhận luồng mà được tạo cấu hình bằng cách thực hiện việc mã hóa SCCC bằng cách kết hợp dữ liệu được đặt trong vùng dữ liệu di động hiện có và dữ liệu di động mới được đặt trong vùng dữ liệu thường theo đơn vị khối.

Luồng được chia thành các khung và một khung được chia thành nhiều khung phụ. Một số khung phụ bao gồm thông tin truyền tín hiệu về khung hiện tại và các khung phụ khác bao gồm thông tin truyền tín hiệu về khung tiếp theo có tính đến PRC. Ví dụ, trong tổng số 5 khung phụ, khung phụ #0 và #1 bao gồm thông tin về khung hiện tại và các khung phụ #2, #3, và #4 bao gồm thông tin về khung tiếp theo có tính đến PRC.

Luồng được mô tả ở trên có thể là luồng được mã hóa SCCC nhờ bộ truyền phát rộng dạng số theo một trong số các tỷ lệ 1/2, 1/3, và 1/4.

Nếu luồng được mô tả ở trên được truyền, bộ giải điều biến 5200 giải điều biến luồng và bộ cân bằng 5300 cân bằng luồng đã được giải điều biến.

Bộ phận giải mã 5400 giải mã ít nhất một trong số dữ liệu di động hiện có và dữ liệu di động mới từ luồng được cân bằng. Trong trường hợp này, việc xử lý cho khung tiếp theo có thể được chuẩn bị trước bằng cách sử dụng thông tin khung có trong mỗi khung phụ.

Như đã mô tả ở trên, bộ thu phát rộng dạng số có thể xử lý phù hợp luồng được truyền từ bộ truyền phát rộng dạng số theo các phương án ví dụ khác nhau. Phương pháp xử lý luồng của bộ thu phát rộng dạng số sẽ không được giải thích và minh họa.

Kết cấu của bộ thu theo các phương án ví dụ khác nhau được mô tả ở trên tương tự với kết cấu của các phương án ví dụ khác được mô tả ở trên, và theo đó minh họa và giải thích của chúng được bỏ qua.

Fig.56 là hình vẽ minh họa định dạng nhóm M/H trước khi dữ liệu được đan xen ở chế độ tương thích được mô tả ở trên, tức là chế độ có thể mở rộng 11a.

Trên Fig.56, nhóm M/H bao gồm dữ liệu di động gồm 208 phân đoạn dữ liệu. Nếu nhóm M/H trong rãnh M/H được tạo cấu hình theo đơn vị 156 gói được phân bố trên 156 gói, thì 156 gói được phân bố trên 208 phân đoạn dữ liệu nhờ đan xen theo luật đan xen của bộ đan xen 430.

Nhóm dữ liệu di động của tổng cộng 208 phân đoạn dữ liệu được chia thành 15 khối dữ liệu di động. Cụ thể, nhóm dữ liệu di động bao gồm các khối từ B1 đến B10 và các khối từ SB1 đến SB5. Các khối từ B1 đến B10 có thể tương ứng với dữ liệu di động được đặt trong vùng dữ liệu di động hiện có như được thể hiện trên Fig.8. Mặt khác, các khối từ SB1 đến SB5 có thể tương ứng với dữ liệu di động mới được cấp phát cho vùng dữ liệu thường hiện có. Khối SB5 là vùng bao gồm phần đầu MPEG và phần chẵn lẻ RS để tương thích ngược.

Mỗi một trong số các khối từ B1 đến B10 có thể bao gồm 16 phân đoạn như vùng dữ liệu di động hiện có, và khối SB4 có thể bao gồm 31 phân đoạn và mỗi khối SB2 và SB3 có thể bao gồm 14 phân đoạn. Trong khối SB1, chiều dài của các phân đoạn được phân bố có thể thay đổi theo chế độ. Nếu dữ liệu thường không được truyền qua tất cả các khung, tức là nếu toàn bộ tốc độ dữ liệu 19,4Mbps được điền dữ liệu di động, thì khối SB1 có thể bao gồm 32 phân đoạn. Nếu dữ liệu thường được truyền một phần, khối SB1 có thể bao gồm 31 phân đoạn.

Khối SB5 là vùng trong đó phần đầu MPEG và phần chẵn lẻ RS hiện có trong 51 phân đoạn của vùng thân được phân bố. Nếu dữ liệu thường không được truyền qua tất cả các khung, tức là nếu toàn bộ tốc độ dữ liệu 19,4Mbps được điền dữ liệu di động, khối SB5 có thể được xác định là được điền dữ liệu di động. Nó tương ứng với chế độ không tương thích được mô tả ở trên. Nếu tất cả dữ liệu được cấp phát là dữ liệu di động và theo đó tính tương thích không cần được tính đến, thì vùng trong đó phần đầu MPEG và phần chẵn lẻ RS, mà có mặt nhằm tương thích với bộ thu dữ liệu thường hiện có, đã được phân bố có thể được xác định lại làm dữ liệu di động.

Như đã mô tả ở trên, các khối này, tức là các khối từ B1 đến B10 và các khối từ SB1 đến SB5 có thể được mã hóa nhờ được kết hợp theo các định dạng khác nhau.

Tức là, nếu chế độ khối SCCC là 00 (khối riêng), thì chế độ mã hóa ngoài SCCC có thể được áp dụng khác nhau theo các vùng nhóm A, B, C và D. Mặt khác, nếu chế độ khối SCCC là 01 (khối đôi), thì chế độ mã hóa ngoài SCCC của tất cả các vùng sẽ giống nhau. Ví dụ, các khối SB1 và SB4, là các khối dữ liệu di động mới được thêm vào, chấp nhận chế độ mã hóa ngoài SCCC được thiết lập cho vùng nhóm C, và các khối SB2 và SB3 chấp nhận chế độ mã hóa ngoài SCCC được thiết lập cho vùng nhóm C. Cuối cùng, khối SB5 chấp nhận chế độ mã hóa ngoài SCCC được thiết lập cho vùng nhóm A.

Cụ thể, khối SB 5 được tạo ra nếu dịch vụ được thực hiện với chỉ dữ liệu di động. Trong trường hợp này, có tính đến tính tương thích giữa bộ thu để nhận dữ liệu di động hiện có và bộ thu để nhận dữ liệu di động mới, ngoài ra, khối SB5 có thể được mã hóa theo cách khác nhau.

Nếu chế độ khối của rãnh mà từ đó khối SB5 được tạo ra là chế độ riêng, thì quần thể sơ cấp phải được điền dữ liệu di động 1.0 và quần thể thứ cấp phải được điền dữ liệu di động 1.1, và theo đó tính tương thích với bộ thu để nhận dữ liệu di động sẽ được duy trì. Do đó, khối SB5 có thể được mã hóa độc lập.

Nếu chế độ khối của rãnh mà từ đó khối SB5 được tạo ra là chế độ đôi, khung là khung đơn, mà chỉ được điền dữ liệu di động 1.1 và theo đó tính tương thích với bộ thu dữ liệu di động hiện có không cần được tính đến. Do đó, khối SB5 có thể được mã hóa nhờ được gộp vào trong một phần của vùng thân.

Cụ thể, nếu dữ liệu di động mới được đặt trong toàn bộ vùng thứ hai trong một rãnh như ở chế độ tương thích, tức là ở chế độ có thể mở rộng 11, thì khối SB5 có thể được mã hóa khác nhau theo chế độ khối. Nếu chế độ khối được thiết lập cho rãnh tương ứng là chế độ riêng trong đó dữ liệu di động hiện có và dữ liệu di động mới cùng tồn tại, khối bao gồm phần đầu MPEG và vùng chặn lẻ RS, tức là khối SB5 có thể được mã hóa độc lập với vùng thân trong rãnh tương ứng. Mặt khác, nếu chế độ khối là chế độ đôi trong đó chỉ có mặt dữ liệu di động mới, thì khối bao gồm phần đầu MPEG và vùng chặn lẻ RS, tức là khối SB5 có thể được mã hóa cùng với phần còn lại của vùng thân. Như đã mô tả ở trên, việc mã hóa khối có thể được thực hiện theo các cách khác nhau.

Do đó, bộ thu phát rộng dạng số, mà nhận luồng vận chuyển, nhận diện chế độ theo dữ liệu truyền tín hiệu và sau đó phát hiện dữ liệu di động mới theo chế độ và tạo lại dữ liệu di động mới. Tức là, nếu chế độ khối là chế độ đôi ở chế độ không tương thích được mô tả ở trên (tức là, chế độ thứ năm hoặc chế độ có thể mở rộng 11) và dữ liệu di động mới được truyền, khối SB5 có thể không được giải mã riêng và có thể được giải mã cùng với dữ liệu di động có trong vùng thân hiện có.

Ngoài ra, nếu dữ liệu đã biết, tức là chuỗi hướng dẫn có mặt như đã mô tả ở trên, các bộ nhớ trong bộ mã hóa dạng mắt cáo phải được khởi tạo trước khi chuỗi hướng dẫn được mã hóa dạng mắt cáo. Trong trường hợp này, vùng để khởi tạo bộ nhớ, tức là byte khởi tạo phải được đặt trước chuỗi hướng dẫn.

Fig.56 minh họa kết cấu luồng sau khi đan xen. Trên Fig.56, chuỗi hướng dẫn xuất hiện trong vùng thân theo định dạng của nhiều chuỗi hướng dẫn dài, và xuất hiện trong vùng đầu/đuôi ở dưới dạng nhiều chuỗi hướng dẫn dài. Cụ thể, trong vùng đầu/đuôi, tổng cộng 5 chuỗi hướng dẫn dài xuất hiện. Không như trong các chuỗi hướng dẫn thứ nhất và thứ năm, trong các chuỗi hướng dẫn thứ hai, thứ ba và thứ năm, byte khởi tạo dạng mắt cáo không bắt đầu từ byte thứ nhất của mỗi phân đoạn và được thiết lập để bắt đầu sau byte định trước.

Sự dịch chuyển vị trí của byte khởi tạo dạng mắt cáo này không bị giới hạn ở vùng đầu/đuôi. Tức là, trong một số chuỗi trong nhiều chuỗi hướng dẫn dài có trong vùng thân, byte khởi tạo dạng mắt cáo có thể được thiết lập để bắt đầu sau byte định trước.

Kích thước của PL, SOBL, và SIBL theo chế độ khối

Các kích thước của chiều dài phần (Portion Length-PL) khung RS, chiều dài khối đầu ra SCCC (SCCC output block length-SOBL), và chiều dài khối đầu vào SCCC (SCCC input block length-SIBL) có thể được thực hiện theo cách khác nhau theo chế độ khối. Bảng sau biểu thị PL của khung RS sơ cấp nếu chế độ khung RS là 00 (tức là, khung đơn), chế độ khối SCCC là 00 (tức là, khối riêng), và chế độ mở rộng khối SCCC là 01:

Bảng 19

Tổ hợp chế độ mã ngoài SCCC								
Cho khu vực A và khối M/H SB5	Cho khu vực B	Cho khu vực C, M/H	Cho khu vực D, M/H	PL				
		Các khối SB1 và SB4	Các khối SB2 và SB3	Chế độ mở rộng được 00	Chế độ mở rộng được 01	Chế độ mở rộng được 10	Chế độ mở rộng được 11	Chế độ mở rộng được 11a
00	00	00	00	10440	11094	11748	13884	12444
00	00	00	10	10138	10678	11216	13126	11766
00	00	00	01	9987	10470	10950	12747	11427
00	00	10	00	9810	10360	10912	12698	11522
00	00	10	10	9508	9944	10380	11940	101
00	00	10	01	9357	9736	10114	11561	10505
00	00	01	00	9495	9993	10494	12105	11061
00	00	01	10	9193	9577	9962	11347	10383
00	00	01	01	9042	9369	9696	10968	10044
00	10	00	00	9626	10280	10934	13070	11630
00	10	00	10	9324	9864	10402	12312	10952
00	10	00	01	9173	9656	10136	11933	10613
00	10	10	00	8996	9546	10098	11884	10708
00	10	10	10	8694	9130	9566	11126	10030
00	10	10	01	8543	8922	9300	10747	9691
00	10	01	00	8681	9179	9680	11291	10247
00	10	01	10	8379	8763	9148	10533	9569
00	10	01	01	8228	8555	8882	10154	9230
00	01	00	00	9219	9873	10527	12663	11223
00	01	00	10	8917	9457	9995	11905	10545
00	01	00	01	8766	9249	9729	11526	10206
00	01	10	00	8589	9139	9691	11477	10301
00	01	10	10	8287	8723	9159	10719	9623
00	01	10	01	8136	8515	8893	10340	9284
00	01	01	00	8274	8772	9273	10884	9840
00	01	01	10	7972	8356	8741	10126	9162
00	01	01	01	7821	8148	8475	9747	8823
10	00	00	00	8706	9360	10014	12422	10710
10	00	00	10	8404	8944	9482	11256	10032
10	00	00	01	8253	8736	9216	10877	9693
10	00	10	00	8076	8626	9178	10828	9788
10	00	10	10	7774	8210	8646	10070	9110
10	00	10	01	7623	8002	8380	9691	8771
10	00	01	00	7761	8259	8760	10235	9327
10	00	01	10	7459	7843	8228	9477	8649
10	00	01	01	7308	7635	7962	9098	8310
10	10	00	00	7892	8546	9200	11200	9896
10	10	00	10	7590	8130	8668	10442	9218
10	10	00	01	7439	7922	8402	10063	8879
10	10	10	00	7262	7812	8364	10014	8974
10	10	10	10	6960	7396	7832	9256	8296
10	10	10	01	6809	7188	7566	8877	7957
10	10	01	00	6947	7445	7946	9421	8513
10	10	01	10	6645	7029	7414	8663	7835
10	10	01	01	6494	6821	7148	8284	7496
10	01	00	00	7485	8139	8793	10793	9489
10	01	00	10	7183	7723	8261	10035	8811
10	01	00	01	7032	7515	7995	9656	8472
10	01	10	00	6855	7405	7957	9607	8567
10	01	10	10	6553	6989	7425	8849	7889
10	01	10	01	6402	6781	7159	8470	7550
10	01	01	00	6540	7038	7539	9014	8106
10	01	01	10	6238	6622	7007	8256	7428
10	01	01	01	6087	6414	6741	7877	7089
01	00	00	00	7839	8493	9147	11079	9843
01	00	00	10	7537	8077	8615	10321	9165
01	00	00	01	7386	7869	8349	9942	8826
01	00	10	00	7209	7759	8311	9893	8921
01	00	10	10	6907	7343	7779	9135	8243
01	00	10	01	6756	7135	7513	8756	7904
01	00	01	00	6894	7392	7893	9300	8460

Bảng 19 - tiếp tục

Tô hợp chế độ mã ngoài SCCC								
Cho khu vực A và khối M/H SB5	Cho khu vực B	Cho khu vực C, M/H	Cho khu vực D, M/H	PL				
		Các khối SB1 và SB4	Các khối SB2 và SB3	Chế độ mở rộng được 00	Chế độ mở rộng được 01	Chế độ mở rộng được 10	Chế độ mở rộng được 11	Chế độ mở rộng được 11a
01	00	01	10	6592	6976	7361	8542	7782
01	00	01	01	6441	6768	7095	8163	7443
01	10	00	00	7025	7679	8333	10265	9029
01	10	00	10	6723	7263	7801	9507	8351
01	10	00	01	6572	7055	7535	9128	8012
01	10	10	00	6395	6945	7497	9079	8107
01	10	10	10	6093	6529	6965	8321	7429
01	10	10	01	5942	6321	6699	7942	7090
01	10	01	00	6080	6578	7079	8486	7646
01	10	01	10	5778	6162	6547	7728	6968
01	10	01	01	5627	5954	6281	7349	6629
01	01	00	00	6618	7272	7926	9858	8622
01	01	00	10	6316	6856	7394	9100	7944
01	01	00	01	6165	6648	7128	8721	7605
01	01	10	00	5988	6538	7090	8672	7700
01	01	10	10	5686	6122	6558	7914	7022
01	01	10	01	5535	5914	6292	7535	6683
01	01	01	00	5673	6171	6672	8079	7239
01	01	01	10	5371	5755	6140	7321	6561
01	01	01	01	5220	5547	5874	6942	6222
		Khác		Không xác định	Không xác định	Không xác định	Không xác định	Không xác định

Bảng sau biểu thị PL của khung RS sơ cấp nếu chế độ khung RS là 00 (tức là, khung đơn), chế độ khối SCCC là 01 (tức là, khối đôi), và chế độ mở rộng khối SCCC là 01:

Bảng 20

Chế độ mã ngoài SCCC	PL				
	Chế độ mở rộng được 00	Chế độ mở rộng được 01	Chế độ mở rộng được 10	Chế độ mở rộng được 11	Chế độ mở rộng được 11a
00	10440	11094	11748	13884	12444
10	6960	7396	7832	9256	8296
01	5220	5547	5874	6942	6222
Khác	Không xác định				

Bảng sau biểu thị PL của khung RS thứ cấp nếu chế độ khung RS là 01 (tức là, khung kép), chế độ khối SCCC là 00 (tức là, khối đơn), và chế độ mở rộng khối SCCC là 01:

Bảng 21

Tổ hợp chế độ mã ngoài SCCC

Cho khu vực C, M/H	Cho khu vực D, M/H	Cho M/H Khối SB5	PL				
			Chế độ mở rộng được 00	Chế độ mở rộng được 01	Chế độ mở rộng được 10	Chế độ mở rộng được 11	Chế độ mở rộng được 11a
00	00	00	2796	3450	4104	6240	4800
00	10	00	2494	3034	3572	5482	4122
00	01	00	2343	2826	3306	5103	3783
10	00	00	2166	2716	3268	5054	3878
10	10	00	1864	2300	2736	4296	3200
10	01	00	1713	2092	2470	3917	2861
01	00	00	1851	2349	2850	4461	3417
	10	00	1549	1933	2318	3703	2739
	01	00	1398	1725	2052	3324	2400
00	00	01	2796	3450	4104	6036	4800
00	10	01	2494	3034	3572	5278	4122
00	01	01	2343	2826	3306	4899	3783
10	00	01	2166	2716	3268	4850	3878
10	10	01	1864	2300	2736	4092	3200
10	01	01	1713	2092	2470	3713	2861
01	00	01	1851	2349	2850	4257	3417
01	10	01	1549	1933	2318	3499	2739
01	01	01	1398	1725	2052	3120	2400
	Khác		Không xác định	Không xác định	Không xác định	Không xác định	Không xác định

Ngoài ra, bảng sau biểu thị SOBL và SIBL nếu chế độ khối SCCC là 00 (tức là, khối riêng), chế độ khung RS là 00 (tức là, khung đơn), và chế độ mở rộng khối SCCC là 01:

Bảng 22

Khối SCCC	SOBL					SIBL				
	Chế độ mở rộng được 00	Chế độ mở rộng được 01	Chế độ mở rộng được 10	Chế độ mở rộng được 11	Chế độ mở rộng được 11a	Chế độ mở rộng được 00	Chế độ mở rộng được 01	Chế độ mở rộng được 10	Chế độ mở rộng được 11	Chế độ mở rộng được 11a
	Tỷ lệ 1/2									
SCB1 (B1 + SB3)	888	1212	1536	2280	1932	444	606	768	1140	966
SCB2 (B2 + SB4)	1872	2160	2412	3432	2568	936	1080	1206	1716	1284
SCB3 (B3)	2376	2376	2376	2376	2376	1188	1188	1188	1188	1188
SCB4 (B4)	2388	2388	2388	2388	2388	1194	1194	1194	1194	1
SCB5 (B5)	2772	2772	2772	2772	2772	1386	1386	1386	1386	1386
SCB6 (B6)	2472	2472	2472	2472	2472	1236	1236	1236	1236	1236
SCB7 (B7)	2772	2772	2772	2772	2772	1386	1386	1386	1386	1386
SCB8 (B8)	2508	2508	2508	2508	2508	1254	1254	1254	1254	1254
SCB9 (B9 + SB1)	1908	2244	2604	3684	2964	954	1122	1302	1842	1482
SCB10 (B10 + SB2)	924	1284	1656	2268	2136	462	642	828	1134	1068
SCB11 (SB5)	0	0	0	816	0	0	0	0	408	0
	Tỷ lệ 1/2									
SCB1 (B1 + SB3)	888	1212	1536	2280	1932	296	404	512	760	644
SCB2 (B2 + SB4)	1872	2160	2412	3432	2568	624	720	804	1144	856
SCB3 (B3)	2376	2376	2376	2376	2376	792	792	792	792	792
SCB4 (B4)	2388	2388	2388	2388	2388	796	796	796	796	796
SCB5 (B5)	2772	2772	2772	2772	2772	924	924	924	924	924
SCB6 (B6)	2472	2472	2472	2472	2472	824	824	824	824	824
SCB7 (B7)	2772	2772	2772	2772	2772	924	924	924	924	924
SCB8 (B8)	2508	2508	2508	2508	2508	836	836	836	836	836
SCB9 (B9 + SB1)	1908	2244	2604	3684	2964	636	748	868	1228	988
SCB10 (B10 + SB2)	924	1284	1656	2268	2136	308	428	552	756	712
SCB11 (SB5)	0	0	0	816	0	0	0	0	272	0
	Tỷ lệ 1/2									
SCB1 (B1 + SB3)	888	1212	1536	2280	1932	222	303	384	570	483
SCB2 (B2 + SB4)	1872	2160	2412	3432	2568	468	540	603	858	642
SCB3 (B3)	2376	2376	2376	2376	2376	594	594	594	594	594
SCB4 (B4)	2388	2388	2388	2388	2388	597	597	597	597	597
SCB5 (B5)	2772	2772	2772	2772	2772	693	693	693	693	693
SCB6 (B6)	2472	2472	2472	2472	2472	618	618	618	618	618
SCB7 (B7)	2772	2772	2772	2772	2772	693	693	693	693	693
SCB8 (B8)	2508	2508	2508	2508	2508	627	627	627	627	627
SCB9 (B9 + SB1)	1908	2244	2604	3684	2964	477	561	651	921	741
SCB10 (B10 + SB2)	924	1284	1656	2268	2136	231	321	414	567	534
SCB11 (SB5)	0	0	0	816	0	0	0	0	204	0

Ngoài ra, bảng sau biểu thị SOBL và SIBL nếu chế độ khối SCCC là 01 (tức là, khối đôi), chế độ khung RS là 01 (tức là, khung kép), và chế độ mở rộng khối SCCC là 01:

Bảng 23

Khối SCCC	SOBL					SIBL				
	Chế độ mở rộng được 00	Chế độ mở rộng được 01	Chế độ mở rộng được 10	Chế độ mở rộng được 11	Chế độ mở rộng được 11a	Chế độ mở rộng được 00	Chế độ mở rộng được 01	Chế độ mở rộng được 10	Chế độ mở rộng được 11	Chế độ mở rộng được 11a
						Tỷ lệ 1/2				
SCB1 (B1 + B6 + SB3)	3360	3684	4008	4752	4404	1680	1842	2004	2376	2202
SCB2 (B2 + B7 + SB4)	4644	4932	5184	6204	5340	2322	2466	2592	3102	2670
SCB3 (B3 + B8)	4884	4884	4884	4884	4884	2442	2442	2442	2442	2442
SCB4 (B4 + B9 + SB1)	4296	4632	4992	6072	5352	2148	2316	2496	3036	2676
SCB5 (B5 + B10 + SB2)	3696	4056	4428	5040	4908	1848	2028	2214	2520	2454
SCB6 (SB5)	0	0	0	816	0	0	0	0	408	0
						Tỷ lệ 1/2				
SCB1 (B1 + B6 + SB3)	3360	3684	4008	4752	4404	1120	1228	1336	1584	1468
SCB2 (B2 + B7 + SB4)	4644	4932	5184	6204	5340	1548	1644	1728	2068	1780
SCB3 (B3 + B8)	4884	4884	4884	4884	4884	1628	1628	1628	1628	1628
SCB4 (B4 + B9 + SB1)	4296	4632	4992	6072	5352	1432	1544	1664	2024	1784
SCB5 (B5 + B10 + SB2)	3696	4056	4428	5040	4908	1232	1352	1476	1680	1636
SCB6 (SB5)	0	0	0	816	0	0	0	0	272	0
						Tỷ lệ 1/2				
SCB1 (B1 + B6 + SB3)	3360	3684	4008	4752	4404	840	921	1002	1188	1101
SCB2 (B2 + B7 + SB4)	4644	4932	5184	6204	5340	1161	1233	1296	1551	1335
SCB3 (B3 + B8)	4884	4884	4884	4884	4884	1221	1221	1221	1221	1221
SCB4 (B4 + B9 + SB1)	4296	4632	4992	6072	5352	1074	1158	1248	1518	1338
SCB5 (B5 + B10 + SB2)	3696	4056	4428	5040	4908	924	1014	1107	1260	1227
SCB6 (SB5)	0	0	0	816	0	0	0	0	204	0

Như đã mô tả ở trên, PL, SOBL, và SIBL có thể có các kích thước khác nhau theo chế độ khối. Dữ liệu được bọc lỏ trong các bảng trên chỉ làm ví dụ và không được coi là giới hạn.

Khởi tạo

Như đã mô tả ở trên, nếu dữ liệu đã biết, tức là dữ liệu hướng dẫn có trong luồng, thì phải thực hiện khởi tạo. Tức là, trong hệ thống truyền nhận ATSC-M/H, bộ mã hóa dạng mắt cáo được khởi tạo theo chuỗi hướng dẫn cần được tạo ra và sau đó byte đã biết được xác định, sao cho bộ thu có thể nhận ra chuỗi hướng dẫn.

Trong định dạng nhóm của chế độ BEM 00, byte khởi tạo dạng mắt cáo được đặt trên mặt ranh giới giữa các răng cưa và byte đã biết được phân bố sau đó. Nếu thực hiện việc mã hóa dạng mắt cáo theo thứ tự phân đoạn từ trên xuống dưới và theo thứ tự byte từ trái sang phải, thì việc mã hóa dạng mắt cáo được thực hiện trên mặt ranh giới giữa các răng cưa được điền dữ liệu của một rãnh khác và theo đó giá trị bộ nhớ của bộ mã hóa dạng mắt cáo không thể dự đoán được trên mặt ranh giới giữa các răng cưa được điền dữ liệu của rãnh tiếp theo. Do đó, bộ mã hóa dạng mắt cáo phải được khởi tạo trên mọi mặt ranh giới của răng cưa. Như được thể hiện trên Fig.56 và Fig.57, byte khởi tạo có thể được phân bố trên mỗi ranh giới răng cưa của vùng đầu bao gồm các khối B1 và B2, và byte khởi tạo cũng có thể được phân bố trên mỗi ranh giới răng cưa của vùng đuôi bao gồm các khối từ SB1 đến SB4.

Nếu hai rãnh nhất định liên tiếp nhau ở chế độ BEM 00, dữ liệu hướng dẫn ngắn của mỗi vùng đầu/đuôi được đặt trên cùng phân đoạn và được nối liên tục, theo đó hoạt động như một phần hướng dẫn dài. Nếu hai rãnh BEM 00 liên tiếp nhau và dữ liệu hướng dẫn được ghép nối, thì chỉ 12 byte khởi tạo tối đa đầu tiên của phân đoạn trong đó có mặt dữ liệu hướng dẫn được sử dụng làm chế độ khởi tạo và byte khởi tạo hiện có trong vùng, tại đó các răng cưa được ghép khớp với nhau có thể được đưa vào như byte đã biết và có thể được mã hóa dạng mắt cáo.

Trừ 12 byte khởi tạo tối đa đầu tiên của phân đoạn, các byte khởi tạo trung gian hiện có trong vùng, tại đó các răng cưa được ghép khớp với nhau có thể được đưa vào làm byte đã biết hoặc byte khởi tạo theo liệu rãnh BEM 00 liên tiếp với cùng một rãnh hay rãnh khác. Tức là, hoạt động của bộ mã hóa dạng mắt cáo có thể được dồn kênh ở chế độ thường hoặc có thể được dồn kênh ở chế độ khởi tạo trong byte khởi tạo trung gian. Do số lượng các ký hiệu được tạo ra khác nhau theo chế độ trong đó bộ mã hóa dạng mắt cáo dồn kênh đầu vào, nên giá trị ký hiệu cần được sử dụng bởi bộ thu làm phần hướng dẫn có thể khác nhau. Để giảm thiểu sự rối loạn của bộ thu, dựa vào ký hiệu được tạo ra nhờ dồn kênh byte khởi tạo trung gian vào byte đã biết nếu phần hướng dẫn dài được tạo cấu hình bởi hai rãnh BEM 00 liên tiếp, có thể xác định giá trị byte khởi tạo trung gian để được sử dụng như chế độ khởi tạo nếu rãnh BEM 00 không liên tiếp với cùng rãnh. Tức là, giá trị

byte khởi tạo trung gian có thể được xác định để tạo ra cùng giá trị với giá trị ký hiệu phân hướng dẫn dài được tạo ra nếu các rãnh được ghép nối. Ở thời điểm này, hai giá trị ký hiệu đầu tiên của byte khởi tạo trung gian có thể khác với giá trị ký hiệu được tạo ra nếu các rãnh được ghép nối.

Như đã mô tả ở trên, phương pháp xử lý luồng của bộ truyền phát rộng dạng số được thực hiện sao cho chuỗi hướng dẫn dài có thể được tạo ra trên ranh giới giữa các rãnh liên tiếp.

Tức là, phương pháp xử lý luồng của bộ truyền bao gồm bước tạo cấu hình luồng, trong đó các rãnh gồm nhiều khối được đặt, và mã hóa và đan xen luồng để được đưa ra làm luồng vận chuyển.

Trong hoạt động tạo cấu hình luồng, nếu các rãnh của chế độ mở rộng khối 00, trong đó sử dụng tất cả các khối trong các rãnh tương ứng được đặt liên tục, thì dữ liệu đã biết có thể được đặt trong phân đoạn định trước của mỗi rãnh trong các rãnh liên kế sao cho chuỗi hướng dẫn dài được tạo ra trên ranh giới giữa các rãnh liên kế được ghép khớp với nhau trong cấu hình dạng răng cưa. Chế độ mở rộng khối 00 là chế độ, trong đó tất cả các khối bao gồm các khối B1 và B2 được sử dụng trong rãnh đó. Do đó, trên ranh giới của rãnh tiếp theo, răng cưa của rãnh trước và răng cưa của rãnh tiếp theo được ghép khớp với nhau. Trong trường hợp này, dữ liệu đã biết được đặt ở vị trí phân đoạn thích hợp của rãnh trước và vị trí phân đoạn thích hợp của rãnh tiếp theo, sao cho dữ liệu đã biết được nối với nhau trên các răng cưa của hai rãnh. Cụ thể, nếu dữ liệu đã biết được đặt trong khoảng phân đoạn thứ 130 của rãnh trước và phân đoạn thứ 15 của rãnh tiếp theo, theo cách tương ứng, thì dữ liệu đã biết được nối với nhau trên ranh giới để tạo ra chuỗi hướng dẫn dài đơn.

Như đã mô tả ở trên, nếu dữ liệu đã biết thứ nhất được đặt trong răng cưa của rãnh trước của các rãnh liên kế và dữ liệu đã biết thứ hai được đặt trong răng cưa của rãnh tiếp theo được nối luân phiên với nhau trên ranh giới, thì giá trị của dữ liệu đã biết thứ nhất và giá trị của dữ liệu đã biết thứ hai có thể là các giá trị định trước để tạo ra chuỗi hướng dẫn dài đã biết đối với bộ thu phát rộng dạng số.

Dữ liệu đã biết có thể được chèn vào để có cùng chuỗi như chuỗi hướng dẫn dài được sử dụng trong rãnh của chế độ mở rộng khối 01 trong đó một số khối của rãnh tương ứng được cung cấp cho một rãnh khác.

Fig.64 minh họa kết cấu luồng trước khi đan xen nếu chế độ mở rộng khối là 00, và Fig.65 minh họa kết cấu luồng sau khi đan xen nếu chế độ mở rộng khối 00.

Nếu dữ liệu đã biết được đặt trong định dạng của chuỗi hướng dẫn dài, thì mọi phần của dữ liệu đã biết không cần được khởi tạo. Do đó, trong trường hợp này, phương pháp có thể bao gồm bước khởi tạo bộ mã hóa dạng mắt cáo trước khi dữ liệu đã biết tương ứng với phần ban đầu của chuỗi hướng dẫn dài được mã hóa dạng mắt cáo.

Mặt khác, nếu các rãnh của các chế độ mở rộng khối khác nhau được đặt liên tục, thì dữ liệu đã biết không liên tục trên ranh giới. Do đó, trong hoạt động truyền, bộ mã hóa dạng mắt cáo có thể được khởi tạo trước khi đặt dữ liệu đã biết ở phần răng cưa trên ranh giới của các rãnh liên tiếp được mã hóa dạng mắt cáo.

Nếu dữ liệu đã biết được đặt trong ranh giới theo định dạng của chuỗi hướng dẫn dài và được truyền như đã mô tả ở trên, thì phương pháp xử lý luồng của bộ thu phát rộng dạng số có thể được thực hiện theo đó.

Tức là, phương pháp xử lý luồng của bộ thu phát rộng dạng số bao gồm các bước: nhận luồng vận chuyển được mã hóa và đan xen với các rãnh gồm nhiều khối được đặt liên tục, giải điều biến luồng vận chuyển nhận được, cân bằng luồng vận chuyển đã được giải điều biến, và giải mã dữ liệu di động mới từ luồng đã được cân bằng.

Mỗi rãnh của luồng vận chuyển có thể bao gồm ít nhất một trong số dữ liệu thường, dữ liệu di động hiện có, và dữ liệu di động mới.

Nếu các rãnh của chế độ mở rộng khối 00 trong đó sử dụng tất cả các khối trong rãnh tương ứng được đặt liên tục, thì luồng vận chuyển có thể có dữ liệu đã biết mà được đặt trong phân đoạn định trước của mỗi rãnh trong các rãnh liên kế sao cho chuỗi hướng dẫn dài được tạo ra trên ranh giới giữa các rãnh liên kế mà được ghép khớp với nhau trong kết cấu dạng răng cưa.

Như đã mô tả ở trên, dữ liệu đã biết được đặt trong ranh giới giữa rãnh trước và rãnh tiếp theo có thể được nối liên tục với nhau để tạo ra chuỗi hướng dẫn dài đã biết đối với bộ truyền phát rộng dạng số và bộ thu phát rộng dạng số.

Chuỗi hướng dẫn dài này có thể có cùng chuỗi như chuỗi hướng dẫn dài được sử dụng trong rãnh của chế độ mở rộng khối 01 trong đó một số khối trong rãnh tương ứng được cung cấp cho một rãnh khác.

Bộ thu phát rộng dạng số có thể biết liệu chuỗi hướng dẫn dài này có được sử dụng hay không bằng cách nhận diện chế độ mở rộng khối của mỗi rãnh.

Tức là, phương pháp xử lý luồng của bộ thu phát rộng dạng số còn có thể bao gồm các bước: giải mã dữ liệu truyền tín hiệu của mỗi rãnh và nhận diện chế độ mở rộng khối của mỗi rãnh. Cụ thể, chế độ mở rộng khối có thể được ghi lại trên TPC của mỗi rãnh.

Trong trường hợp này, thậm chí nếu đã nhận được một rãnh, thì bộ thu phát rộng dạng số có thể trì hoãn phát hiện và xử lý dữ liệu đã biết cho tới khi chế độ mở rộng khối của rãnh tiếp theo được nhận diện. Tức là, phương pháp có thể bao gồm, nếu bước mã hóa của dữ liệu truyền tín hiệu của một rãnh tiếp theo trong số các rãnh liên kế được hoàn thành và nó được nhận diện là chế độ mở rộng khối của rãnh tiếp theo là 00, thì phát hiện dữ liệu đã biết của phần rãnh cưa trên ranh giới của các rãnh liên kế như chuỗi hướng dẫn dài, và xử lý dữ liệu đã biết.

Theo một phương án ví dụ khác, dữ liệu truyền tín hiệu của mỗi rãnh có thể được thực hiện để thông báo thông tin về các rãnh xung quanh trước.

Trong trường hợp này, bộ thu phát rộng dạng số có thể giải mã dữ liệu truyền tín hiệu của rãnh trước trong số các rãnh liên kế và nhận diện các chế độ mở rộng khối của rãnh trước và rãnh tiếp theo.

Phương pháp xử lý luồng của bộ truyền phát rộng dạng số và bộ thu phát rộng dạng số có thể được thực hiện trong bộ truyền phát rộng dạng số và bộ thu phát rộng dạng số, chúng có các kết cấu khác nhau như được thể hiện trên các hình vẽ. Ví dụ, bộ thu phát rộng dạng số còn có thể bao gồm bộ phận phát hiện để phát hiện và xử lý dữ liệu đã biết, ngoài các thành phần cơ bản như bộ phận thu, bộ giải điều biến, bộ cân bằng, và bộ phận giải mã. Trong trường hợp này, nếu nhận diện được là đã nhận được hai rãnh của chế độ mở rộng khối 00, thì bộ phận phát hiện phát hiện dữ liệu hướng dẫn dài được đặt trong ranh giới của các rãnh và sử dụng dữ liệu hướng dẫn dài để hiệu chỉnh lỗi. Ngoài ra, bộ phận phát hiện có thể cung cấp kết quả phát hiện cho ít nhất một trong số bộ giải điều biến, bộ cân bằng, và bộ phận giải mã.

Vị trí dữ liệu hướng dẫn tính đến phần chẵn lẻ RS

Liên quan đến phân đoạn có giá trị phần chẵn lẻ RS đã được xác định, giá trị phần chẵn lẻ RS đã được tính toán phải được thay đổi như dữ liệu của phân đoạn được thay đổi

trong khi khởi tạo bộ mã hóa dạng mắt cáo, sao cho bộ thu không mắc lỗi và hoạt động bình thường. Trong trường hợp của gói trong đó có mặt byte khởi tạo dạng mắt cáo, 20 byte của phần chẵn lẻ RS không hệ thống của gói không đứng trước byte khởi tạo dạng mắt cáo. Byte khởi tạo dạng mắt cáo chỉ có mặt ở vị trí tại đó điều kiện này được thỏa mãn, và dữ liệu hướng dẫn có thể được tạo ra bởi byte khởi tạo này.

Như được thể hiện trên Fig.64 và Fig.65, để đặt byte khởi tạo dạng mắt cáo trước phần chẵn lẻ RS, vị trí phần chẵn lẻ RS được thay đổi khác với định dạng nhóm của rãnh BEM 01. Tức là, theo định dạng nhóm của rãnh BEM 01, phần chẵn lẻ RS được đặt trong 5 phân đoạn đầu tiên trong số 208 phân đoạn dữ liệu sau khi đan xen, nhưng, trong trường hợp của rãnh BEM 00, vị trí phần chẵn lẻ RS có thể thay đổi sao cho một phần dưới khối B2 được điền vào như được thể hiện trên Fig.64 và Fig.65.

Tính đến phần chẵn lẻ RS đã được thay đổi, dữ liệu hướng dẫn được phân bố trong rãnh BEM 00 được đặt sao cho phần hướng dẫn thứ nhất, thứ hai, và thứ ba được đặt lần lượt trong các phân đoạn thứ 7 và thứ 8, các phân đoạn thứ 20 và thứ 21, và các phân đoạn thứ 31 và thứ 32, trong các khối B1 và B2. Ngoài ra, các phần chẵn lẻ RS được thay đổi có thể được đặt trong các phân đoạn thứ 33 đến 37 của các khối B1 và B2. Ngoài ra, trong vùng đuôi, phần hướng dẫn thứ nhất, thứ hai, thứ ba, thứ tư, và thứ năm lần lượt có thể được đặt trong các phân đoạn thứ 134 và thứ 135, các phân đoạn thứ 150 và thứ 151, các phân đoạn thứ 163 và thứ 164, các phân đoạn thứ 176 và thứ 177, và các phân đoạn thứ 187 và thứ 188. Nếu hai rãnh BEM 00 liên kế nhau và tạo ra phần hướng dẫn dài được ghép nối, thì phần hướng dẫn thứ nhất của các khối B1 và B2 và phần hướng dẫn thứ ba của vùng đuôi được nối với nhau, phần hướng dẫn thứ hai của các khối B1 và B2 và phần hướng dẫn thứ tư của vùng đuôi được nối với nhau, và phần hướng dẫn thứ ba của các khối B1 và B2 và phần hướng dẫn thứ năm của vùng đuôi được nối với nhau.

Như đã mô tả ở trên, dữ liệu hướng dẫn được đặt theo các cách khác nhau và việc khởi tạo của chúng do đó được thực hiện.

Bộ thu phát rộng dạng số phát hiện dữ liệu hướng dẫn từ vị trí tại đó đặt dữ liệu hướng dẫn. Cụ thể, bộ phận phát hiện hoặc bộ giải mã truyền tín hiệu trên Fig.52 có thể phát hiện thông tin biểu thị vị trí của dữ liệu hướng dẫn. Do đó, dữ liệu hướng dẫn được phát hiện từ vị trí được nhận diện và lỗi có thể được hiệu chỉnh.

Các phương án ví dụ trên và ưu điểm chỉ nhằm làm ví dụ và không nhằm giới hạn phạm vi sáng chế. Các phương án ví dụ có thể được áp dụng một cách dễ dàng đối với các

loại thiết bị khác. Ngoài ra, phần mô tả của các phương án ví dụ nhằm mục đích minh họa, và không giới hạn phạm vi của các điểm yêu cầu bảo hộ, và nhiều biến đổi, cải biến, và thay đổi là rõ ràng đối với người có hiểu biết trung bình trong lĩnh vực kỹ thuật này.

YÊU CẦU BẢO HỘ

1. Phương pháp xử lý luồng của bộ truyền phát rộng dạng số, phương pháp này bao gồm các bước:

tạo ra luồng, trong đó các rãnh gồm nhiều khối được đặt liên tục; và

mã hóa và đan xen luồng,

trong đó, nếu các rãnh của chế độ mở rộng khối thứ nhất được đặt liên tục, thì bước tạo ra luồng bao gồm việc nối chuỗi hướng dẫn được đặt ở các vị trí định trước của các rãnh liên kề để tạo ra chuỗi hướng dẫn dài.

2. Phương pháp theo điểm 1, trong đó chuỗi hướng dẫn thứ nhất được đặt ở phần đuôi của rãnh trước của các rãnh liên kề và chuỗi hướng dẫn thứ hai được đặt ở phần đầu của rãnh kế tiếp của các rãnh liên kề được nối luân phiên với nhau trên ranh giới, và giá trị của chuỗi hướng dẫn thứ nhất và giá trị của chuỗi hướng dẫn thứ hai là các giá trị định trước để tạo ra chuỗi hướng dẫn dài mà đã được biết đối với bộ truyền phát rộng dạng số và bộ thu phát rộng dạng số.

3. Phương pháp theo điểm 1, trong đó chuỗi hướng dẫn bao gồm cùng chuỗi như chuỗi hướng dẫn dài mà được sử dụng trong rãnh của chế độ mở rộng khối thứ hai, trong đó một số khối của rãnh tương ứng được cung cấp cho một rãnh khác.

4. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 3, trong đó bước truyền bao gồm việc khởi tạo bộ mã hóa dạng mắt cáo trước khi chuỗi hướng dẫn tương ứng với phần ban đầu của chuỗi hướng dẫn dài được mã hóa dạng mắt cáo.

5. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 3, trong đó, nếu các rãnh của các chế độ mở rộng khối khác nhau được đặt liên tục, thì bước truyền bao gồm việc khởi tạo bộ mã hóa dạng mắt cáo trước khi mã hóa dạng mắt cáo chuỗi hướng dẫn được đặt ở phần rãnh của ranh giới giữa các rãnh được đặt liên tục.

6. Bộ truyền phát rộng dạng số bao gồm:

bộ phận tạo kết cấu luồng tạo ra luồng, trong đó các rãnh gồm nhiều khối được đặt; và

bộ phận kích thích mã hóa và đan xen luồng,

trong đó, nếu các rãnh của chế độ mở rộng khối thứ nhất được đặt liên tục, thì bộ phận tạo kết cấu luồng nổi chuỗi hướng dẫn được đặt ở các vị trí định trước của các rãnh liền kề để tạo ra chuỗi hướng dẫn dài.

7. Bộ truyền phát rộng dạng số theo điểm 6, trong đó chuỗi hướng dẫn thứ nhất được đặt ở phần đuôi của rãnh trước của các rãnh liền kề và chuỗi hướng dẫn thứ hai được đặt ở phần đầu của rãnh kế tiếp của các rãnh liền kề được nối luân phiên với nhau trên ranh giới, và giá trị của chuỗi hướng dẫn thứ nhất và giá trị của chuỗi hướng dẫn thứ hai là các giá trị định trước để tạo ra chuỗi hướng dẫn dài mà được biết là bộ truyền phát rộng dạng số và bộ thu phát rộng dạng số.

8. Phương pháp xử lý luồng của bộ thu phát rộng dạng số, phương pháp này bao gồm các bước:

nhận luồng vận chuyển;

giải điều biến luồng vận chuyển nhận được;

cân bằng luồng vận chuyển đã được giải điều biến; và

giải mã luồng được cân bằng,

trong đó mỗi rãnh của luồng vận chuyển bao gồm ít nhất một trong số dữ liệu thường và dữ liệu di động,

trong đó, nếu các rãnh của chế độ mở rộng khối thứ nhất được đặt liên tục, thì trong luồng vận chuyển, chuỗi hướng dẫn mà được đặt ở các vị trí định trước của các rãnh liền kề được nối để tạo ra chuỗi hướng dẫn dài.

9. Phương pháp theo điểm 8, trong đó chuỗi hướng dẫn thứ nhất được đặt ở phần đuôi của rãnh trước của các rãnh liền kề và chuỗi hướng dẫn thứ hai được đặt ở phần đầu của rãnh kế tiếp của các rãnh liền kề được nối luân phiên với nhau trên ranh giới, và giá trị của chuỗi hướng dẫn thứ nhất và giá trị của chuỗi hướng dẫn thứ hai là các giá trị định trước để tạo ra chuỗi hướng dẫn dài mà đã biết đối với bộ truyền phát rộng dạng số và bộ thu phát rộng dạng số.

10. Phương pháp theo điểm 8, trong đó chuỗi hướng dẫn bao gồm cùng chuỗi như chuỗi hướng dẫn dài mà được sử dụng trong rãnh của chế độ mở rộng khối thứ hai, trong đó một số khối của rãnh tương ứng được cung cấp cho một rãnh khác.

11. Phương pháp theo điểm bất kỳ trong số các điểm từ 8 đến 10, trong đó phương pháp này còn bao gồm bước giải mã dữ liệu truyền tín hiệu của mỗi rãnh và nhận dạng chế độ mở rộng khối của mỗi một trong số các rãnh.

12. Phương pháp theo điểm 11, trong đó, nếu bước giải mã dữ liệu truyền tín hiệu của rãnh kế tiếp của các rãnh liên kế được hoàn thành và nó được nhận diện là chế độ mở rộng khối của rãnh kế tiếp là chế độ mở rộng khối thứ nhất, thì phương pháp này còn bao gồm bước phát hiện chuỗi hướng dẫn được đặt trong phần răng cưa của ranh giới giữa các rãnh liên kế như chuỗi hướng dẫn dài và xử lý chuỗi hướng dẫn.

13. Phương pháp theo điểm bất kỳ trong số các điểm từ 8 đến 10, trong đó phương pháp này còn bao gồm bước giải mã dữ liệu truyền tín hiệu của rãnh trước của các rãnh liên kế và nhận diện các chế độ mở rộng khối của cả rãnh trước và rãnh kế tiếp.

14. Bộ thu phát rộng dạng số bao gồm:

bộ phận thu mà nhận luồng vận chuyển được mã hóa và đan xen nếu các rãnh gồm nhiều khối được đặt;

bộ giải điều biến giải điều biến luồng vận chuyển nhận được;

bộ cân bằng cân bằng luồng vận chuyển đã được giải điều biến; và

bộ phận giải mã giải mã luồng được cân bằng,

trong đó mỗi rãnh của luồng vận chuyển bao gồm ít nhất một trong số dữ liệu thường và dữ liệu di động,

trong đó, nếu các rãnh của chế độ mở rộng khối thứ nhất được đặt liên tục, thì trong luồng vận chuyển, chuỗi hướng dẫn mà được đặt ở các vị trí định trước của các rãnh liên kế được nối để tạo ra chuỗi hướng dẫn dài.

15. Bộ thu phát rộng dạng số theo điểm 14, trong đó chuỗi hướng dẫn thứ nhất được đặt trong phần đuôi của rãnh trước của các rãnh liên kế và chuỗi hướng dẫn thứ hai được đặt trong phần đầu của rãnh kế tiếp của các rãnh liên kế được nối luân phiên với nhau trên ranh giới, và giá trị của chuỗi hướng dẫn thứ nhất và giá trị của chuỗi hướng dẫn thứ hai là các giá trị định trước để tạo ra chuỗi hướng dẫn dài mà được biết đối với bộ truyền phát rộng dạng số và bộ thu phát rộng dạng số.

FIG. 1

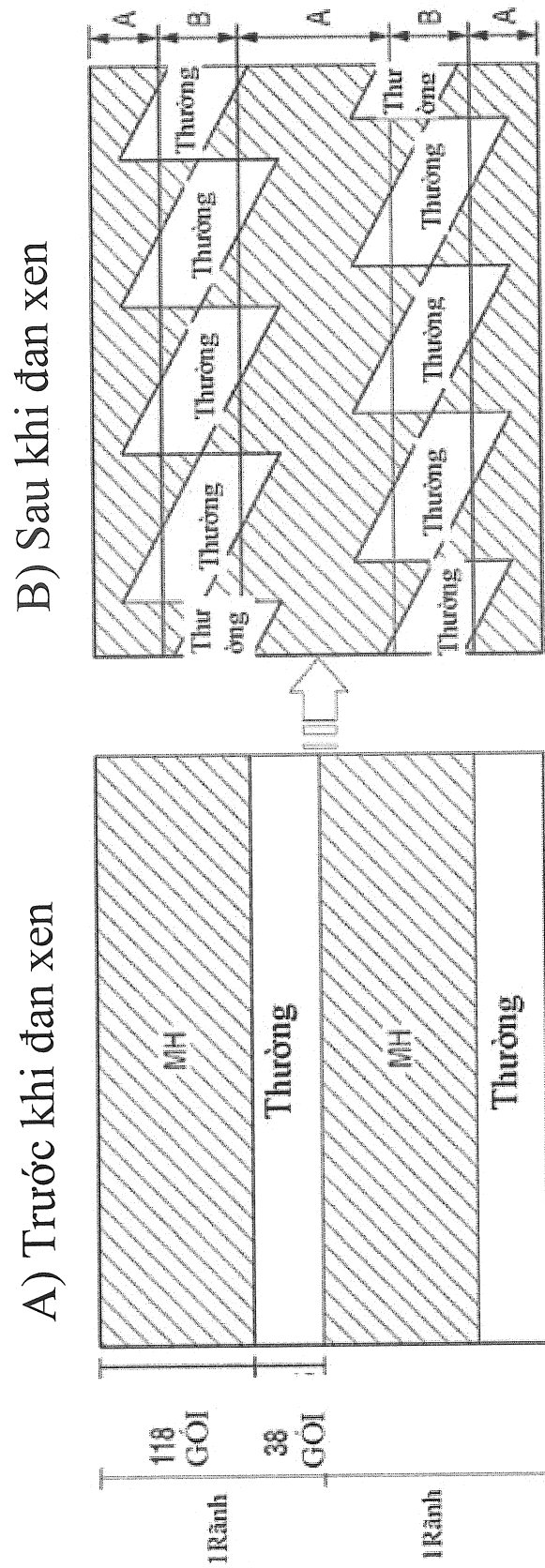


FIG. 2

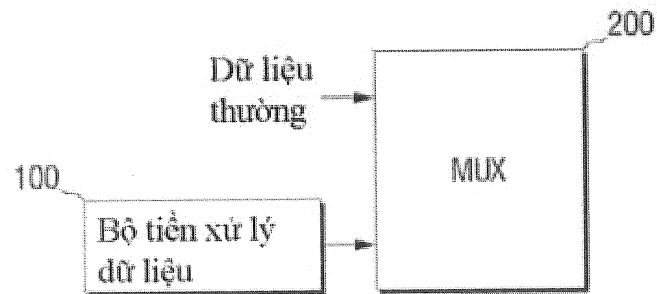


FIG. 3

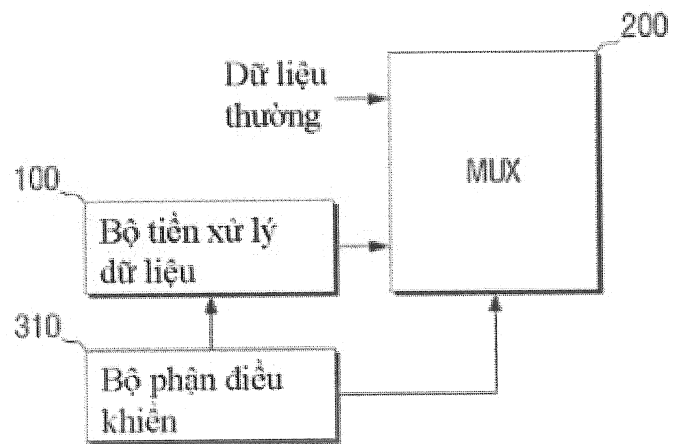


FIG. 5

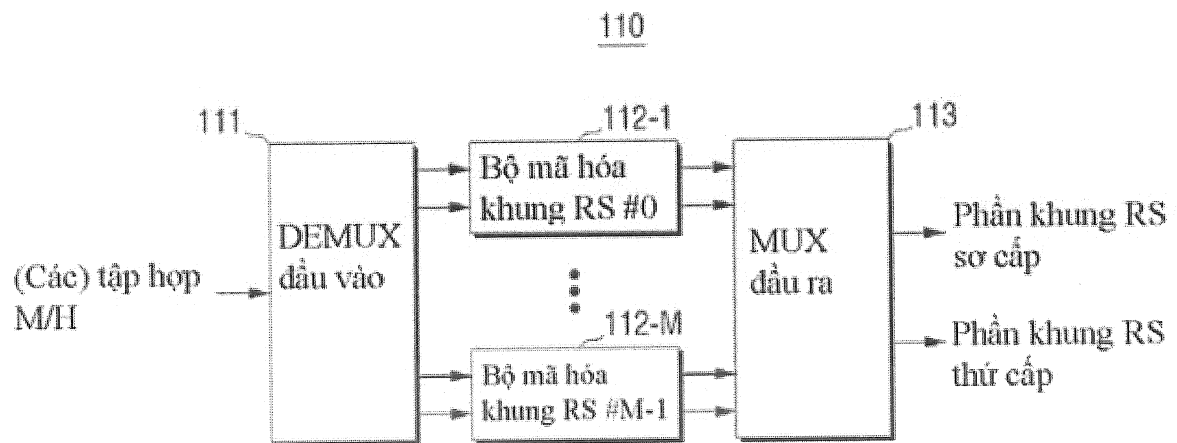


FIG. 6

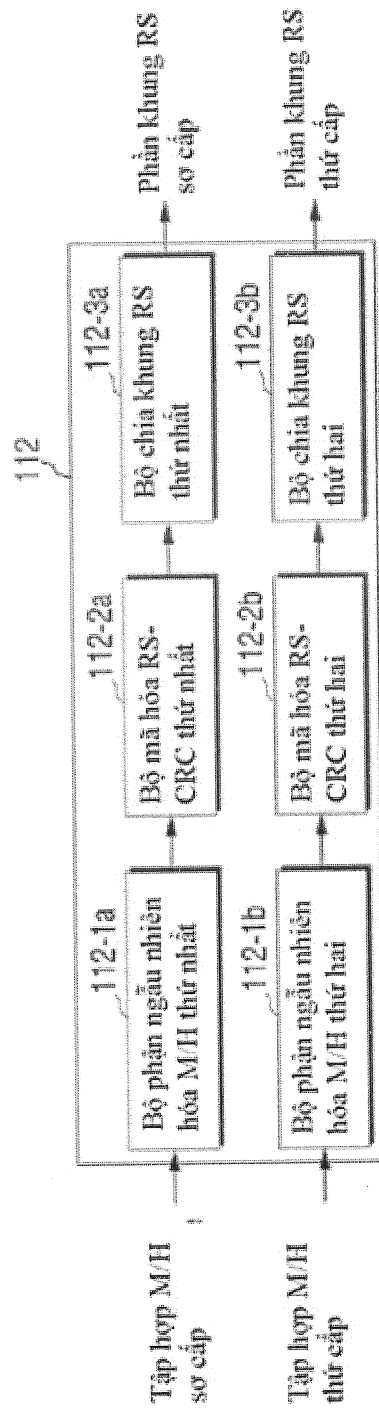


FIG. 7

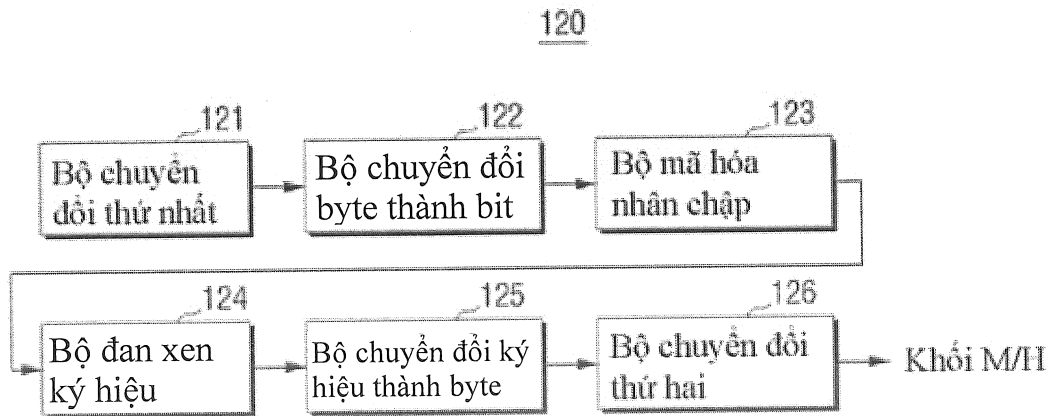


FIG. 8

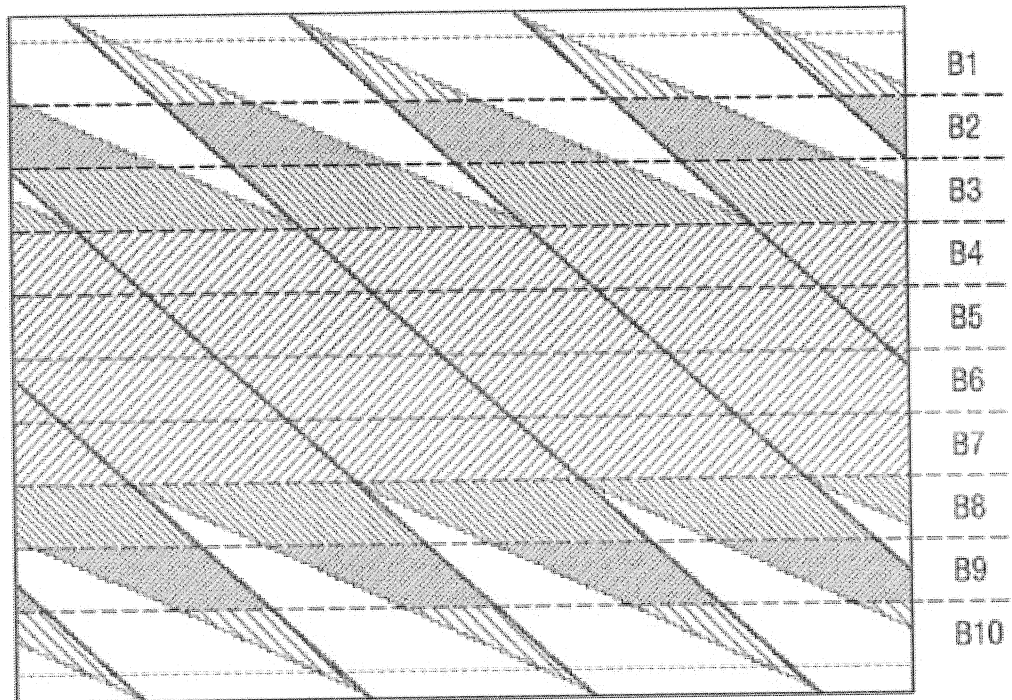


FIG. 9

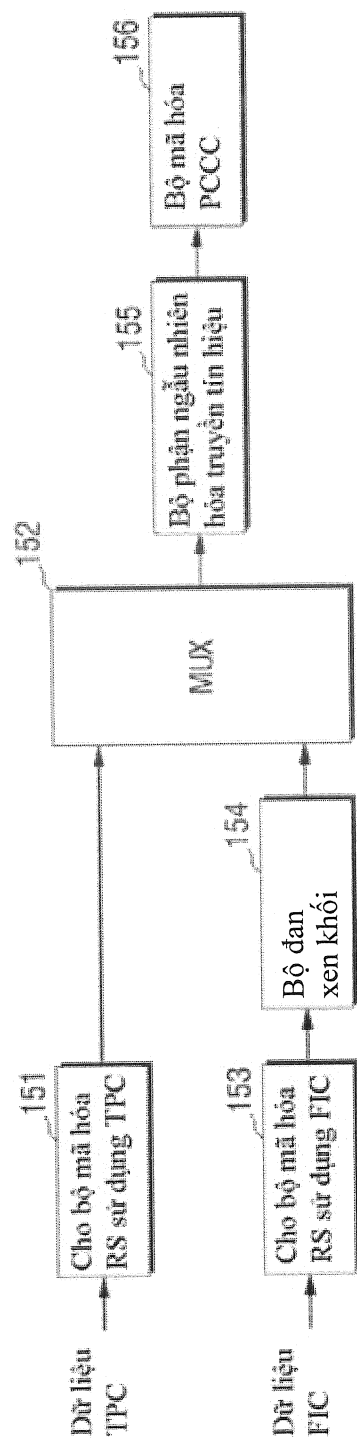


FIG. 10

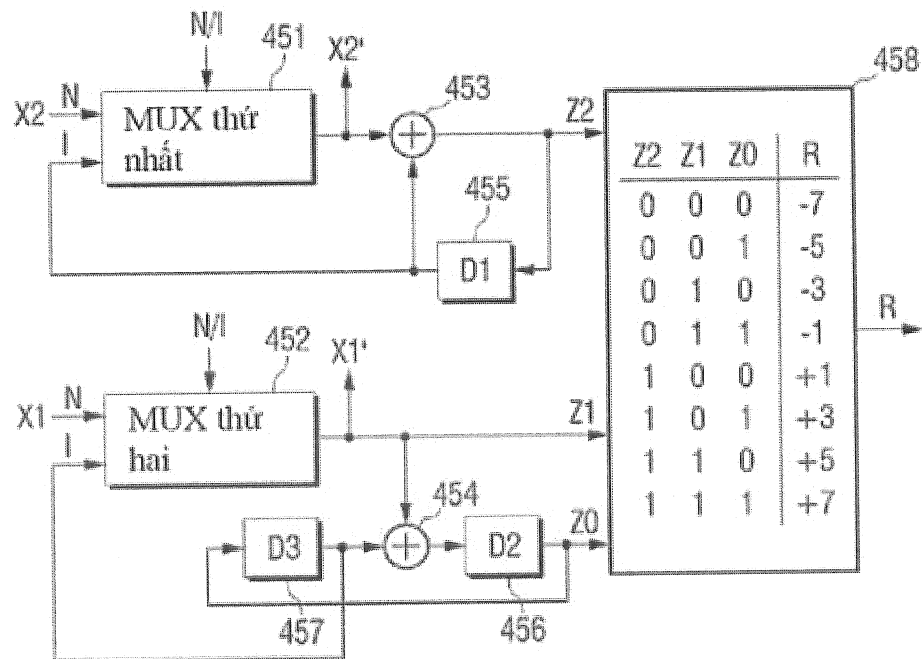


FIG. 11

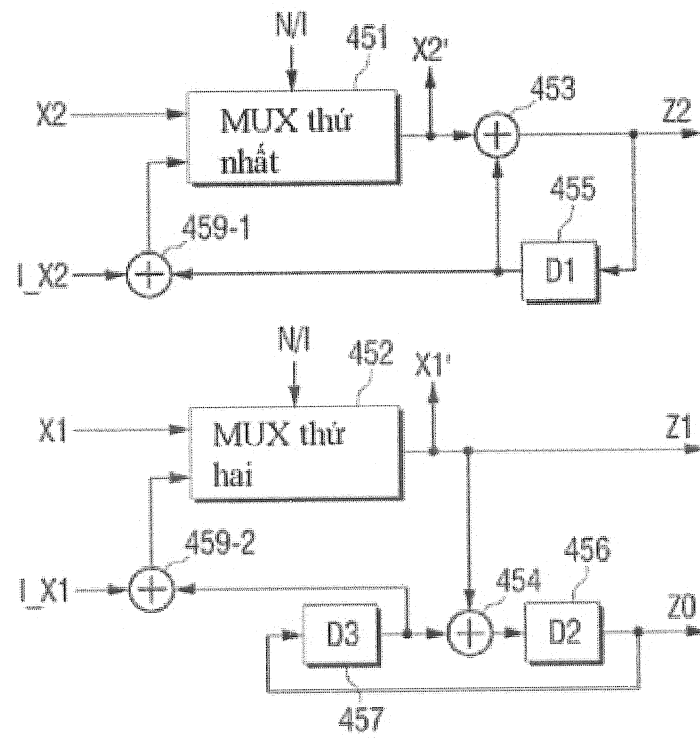


FIG. 12

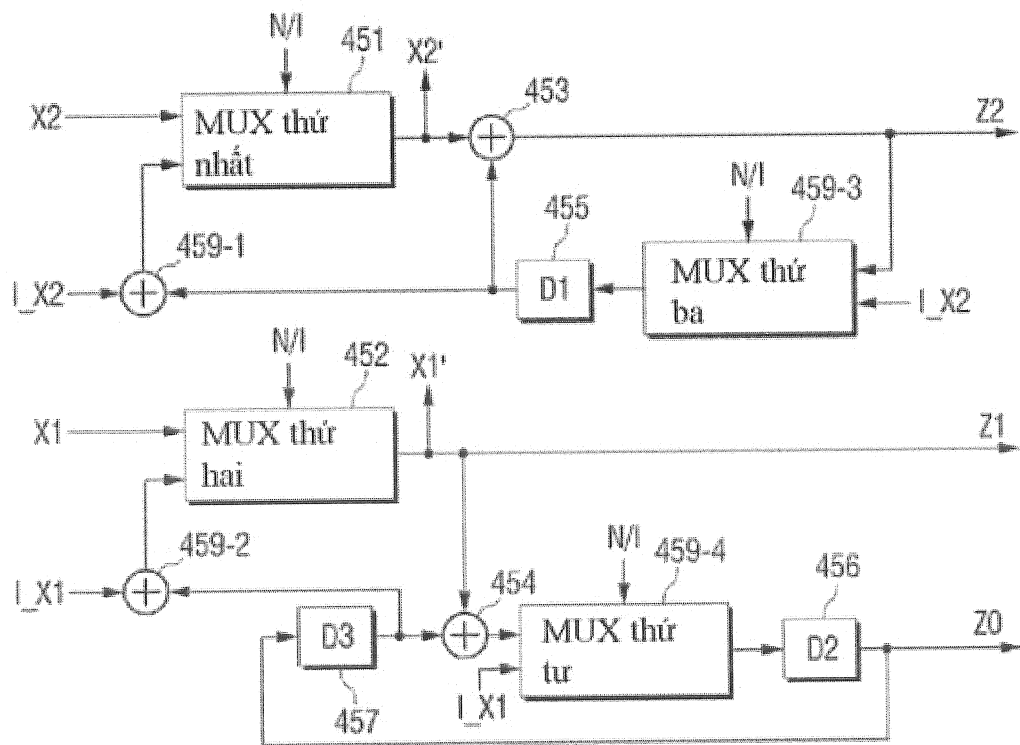


FIG. 13

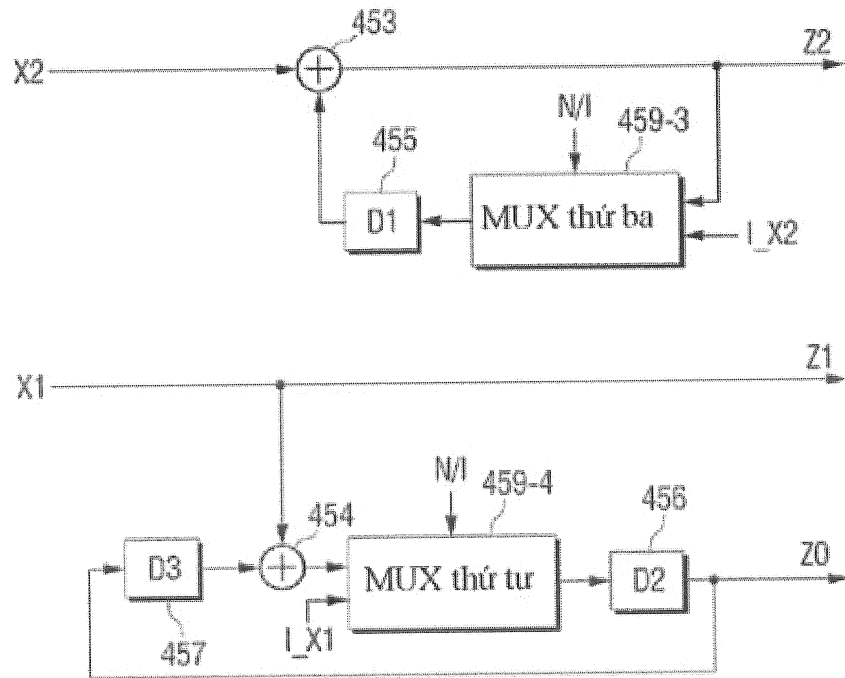


FIG. 14

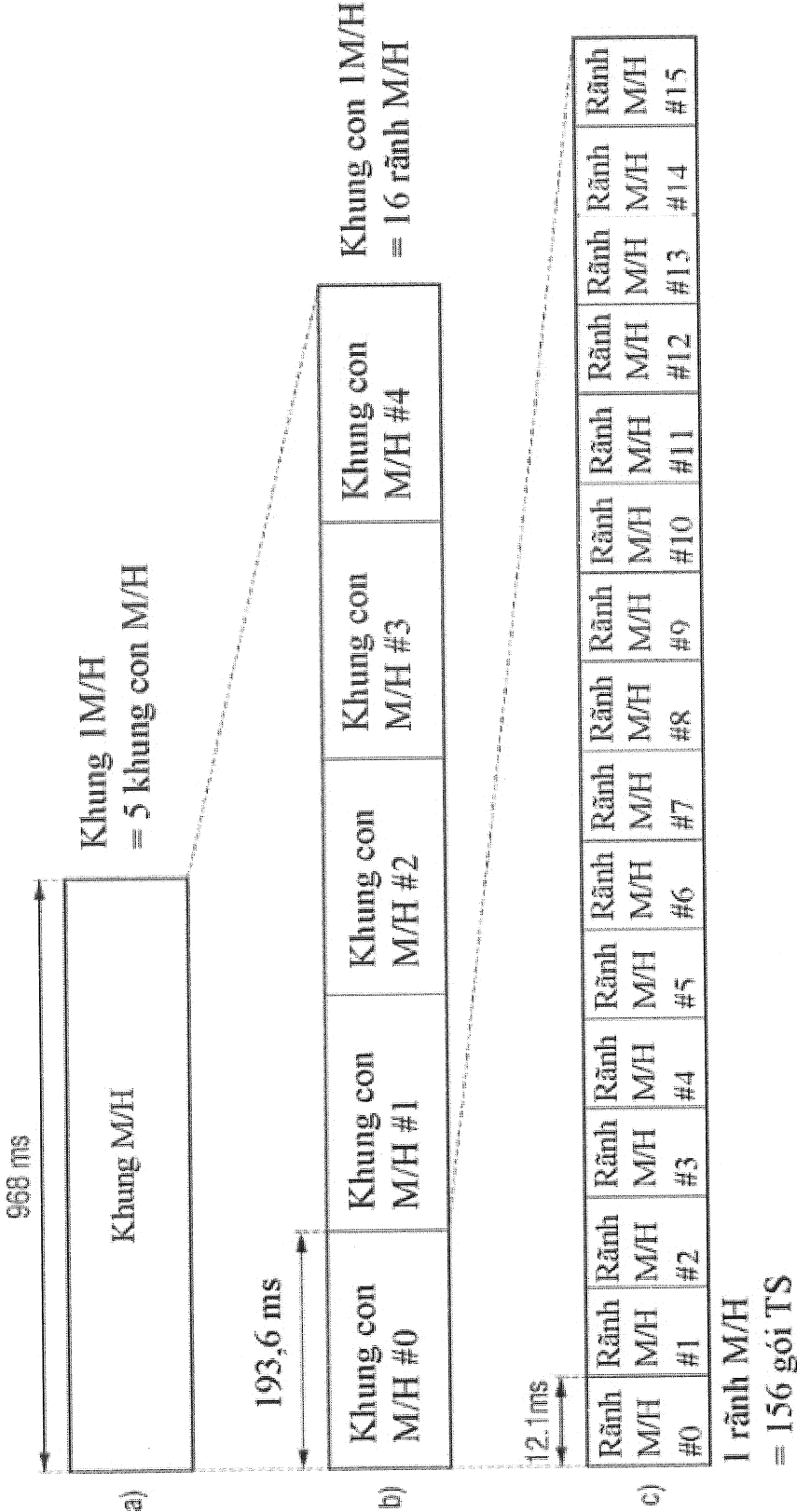


FIG. 15

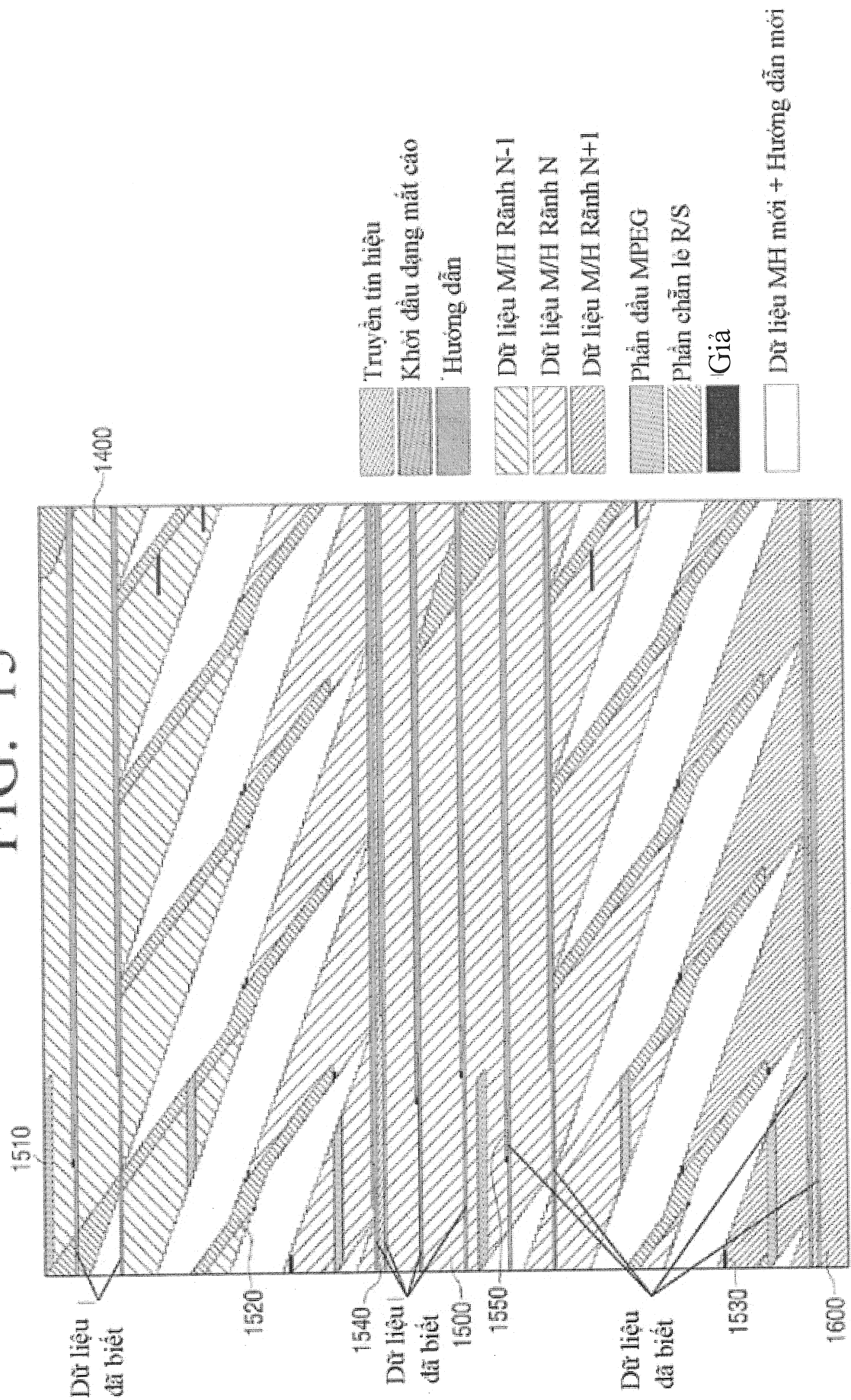


FIG. 16

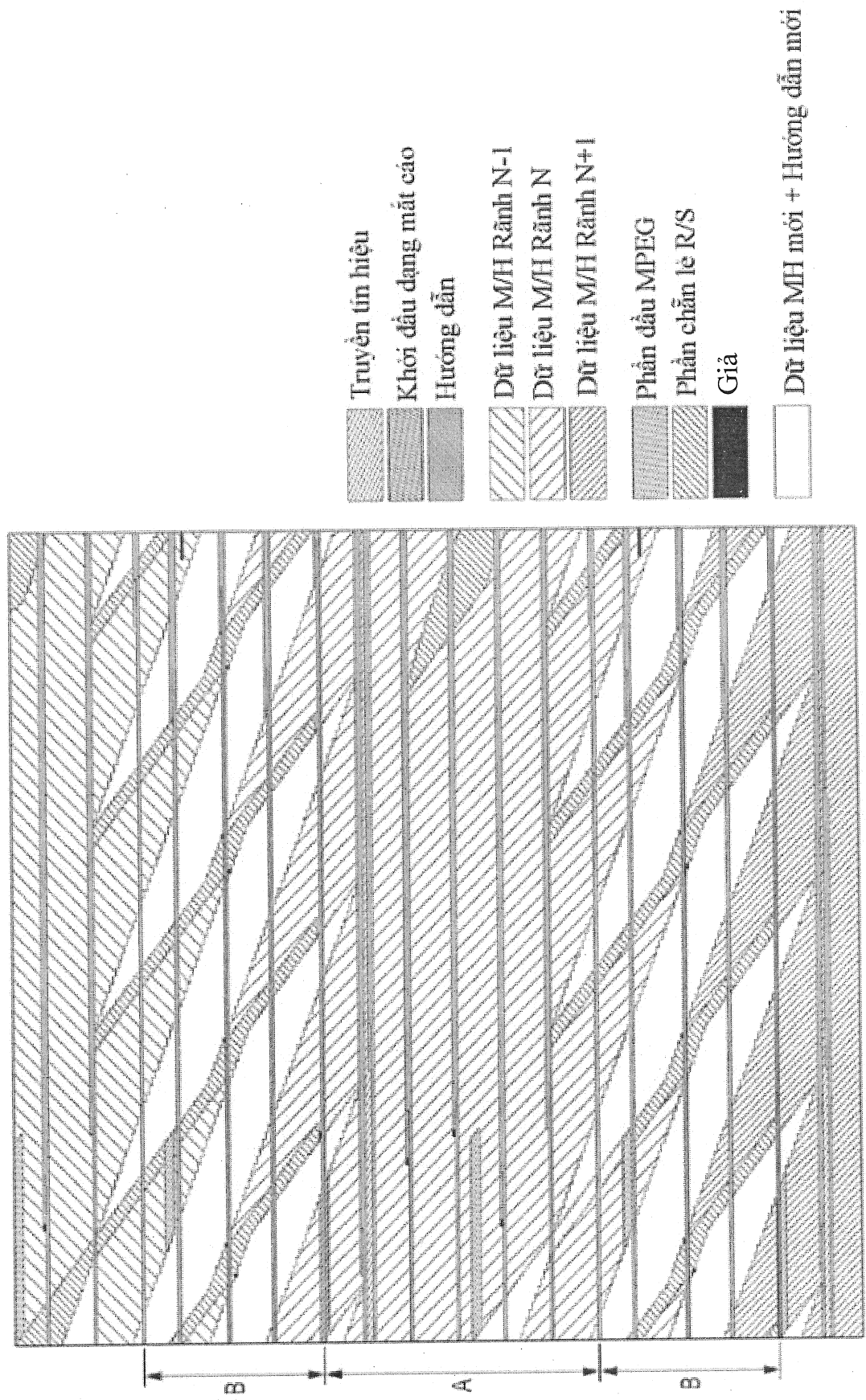


FIG. 17

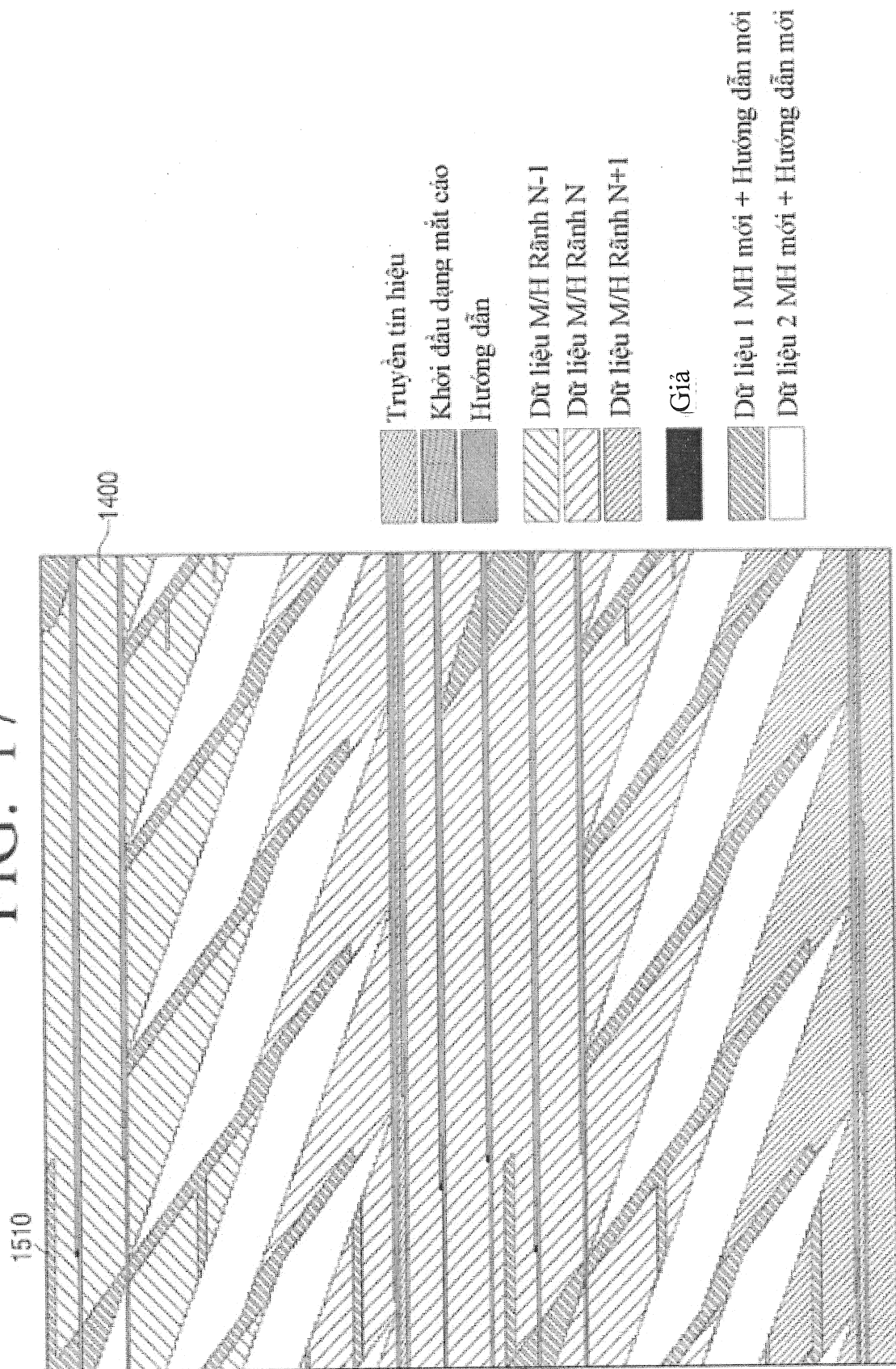


FIG. 18

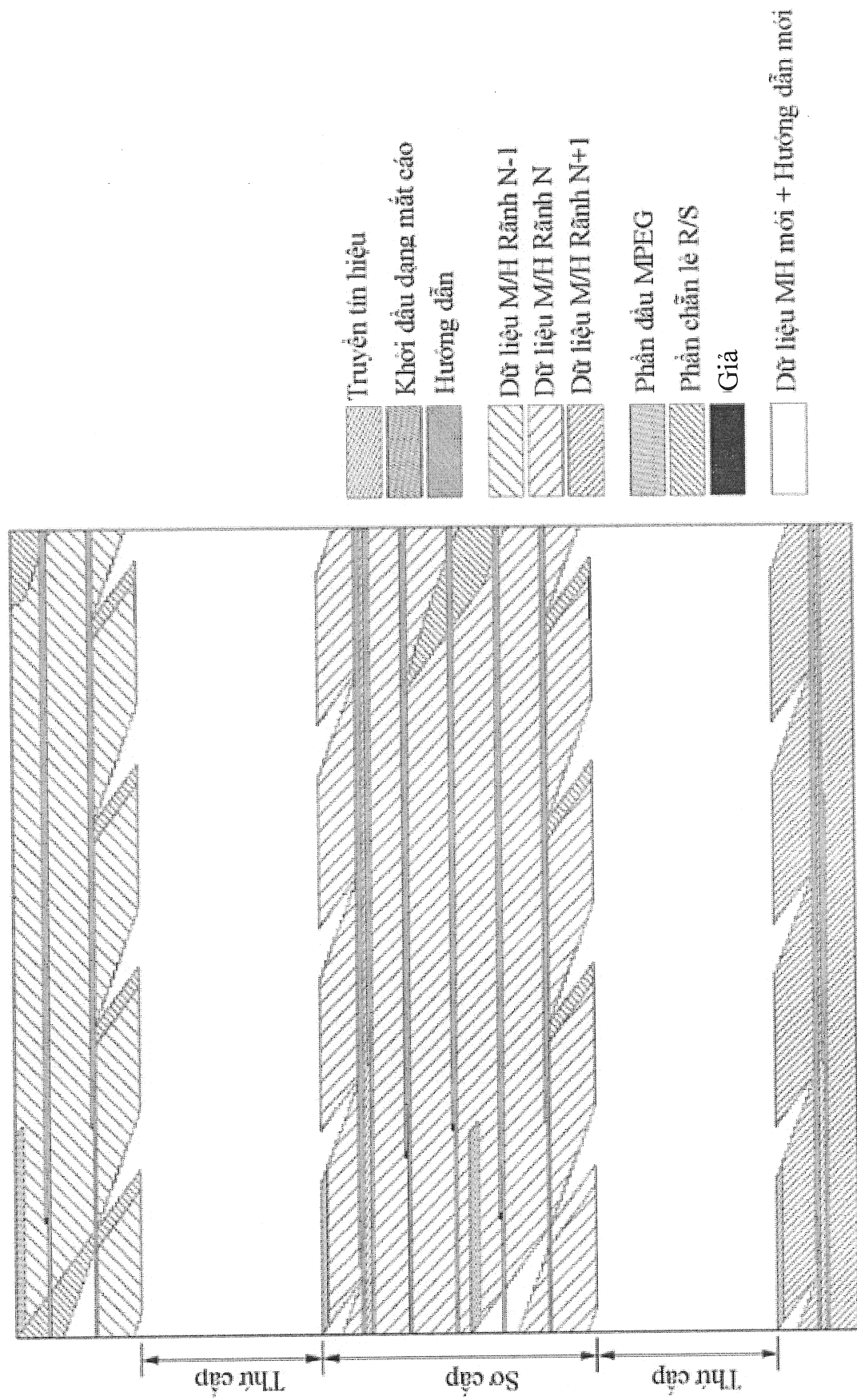


FIG. 19

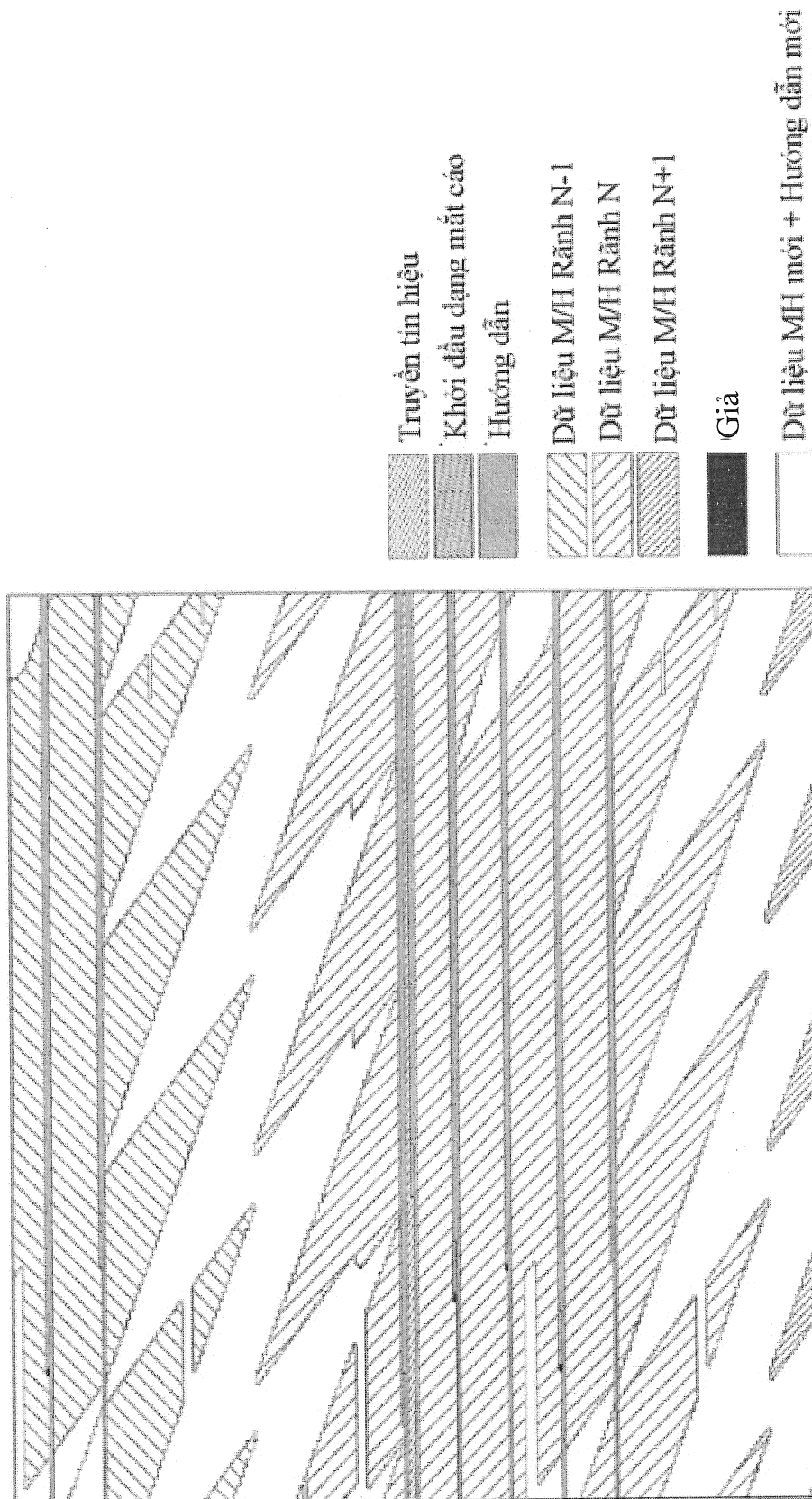


FIG. 20

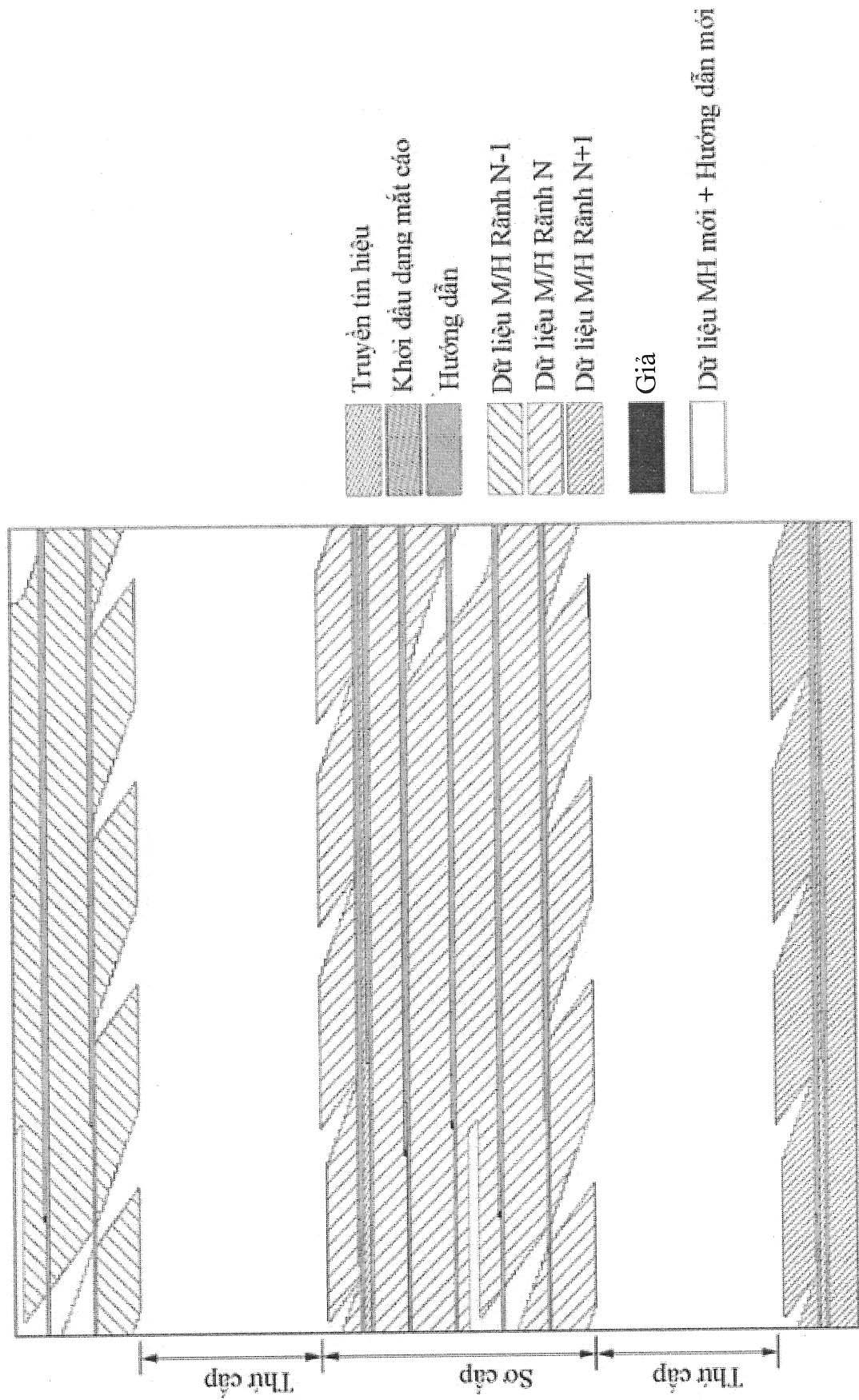


FIG. 21

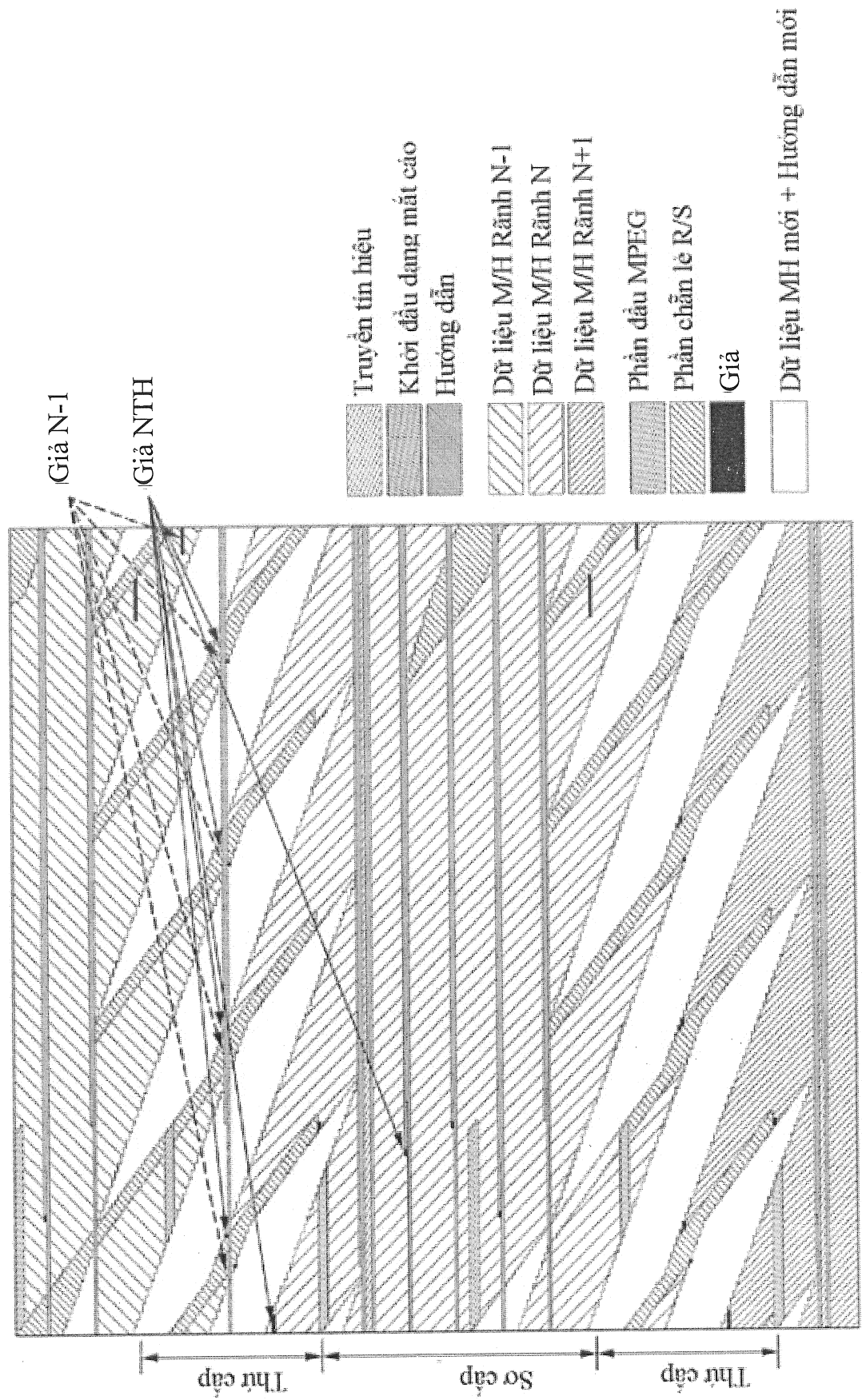


FIG. 22

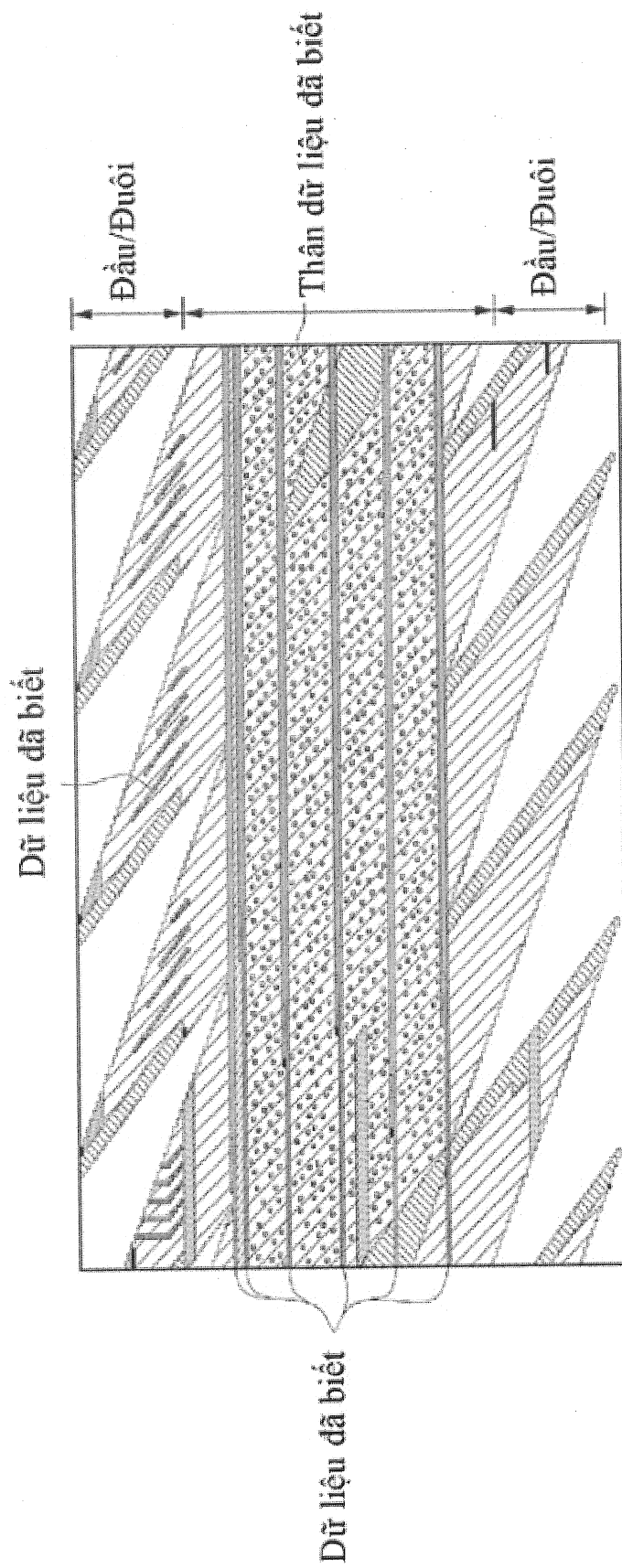


FIG. 23

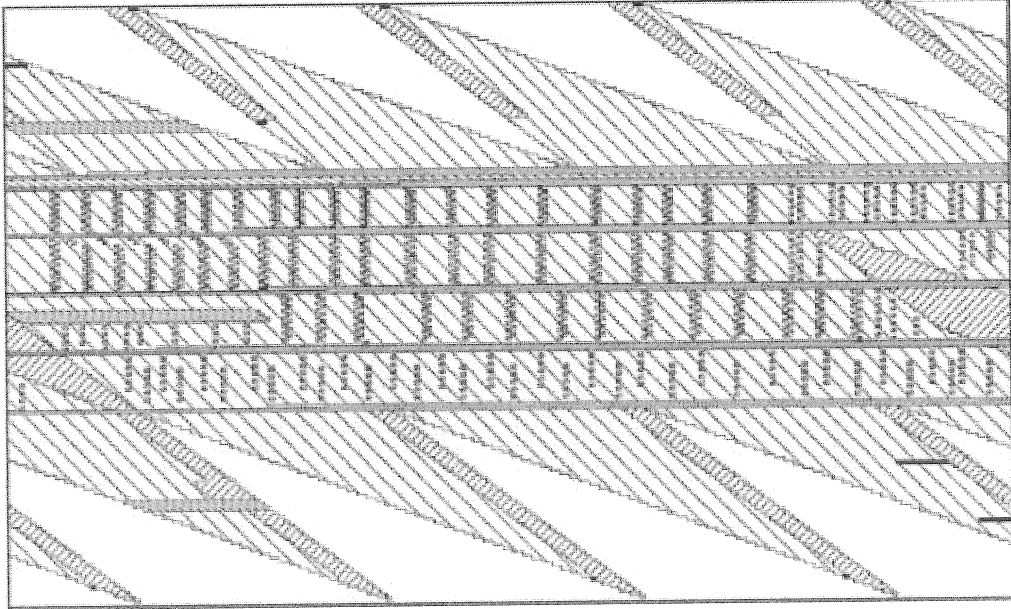


FIG. 24

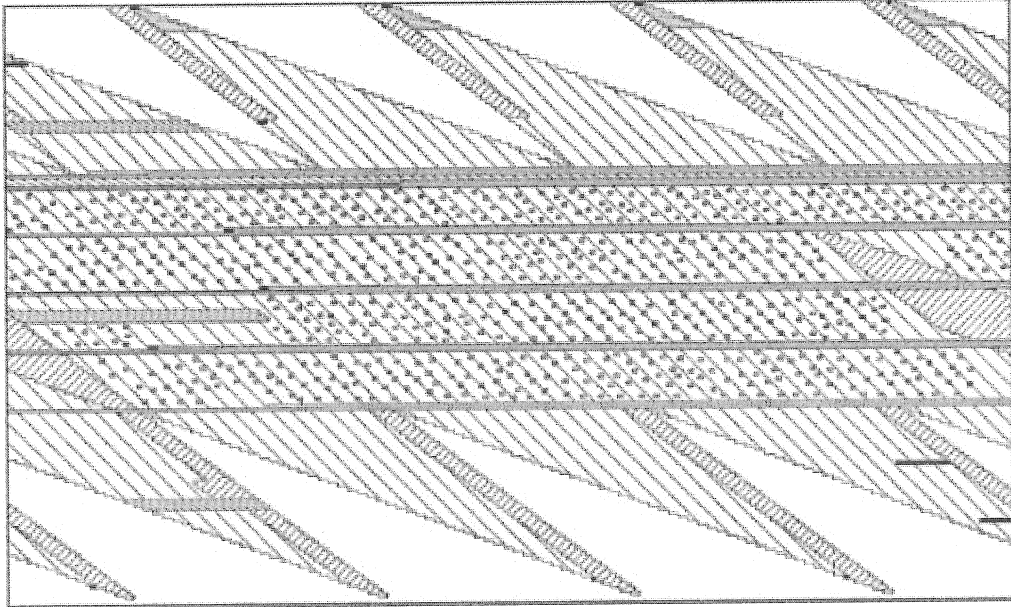


FIG. 25

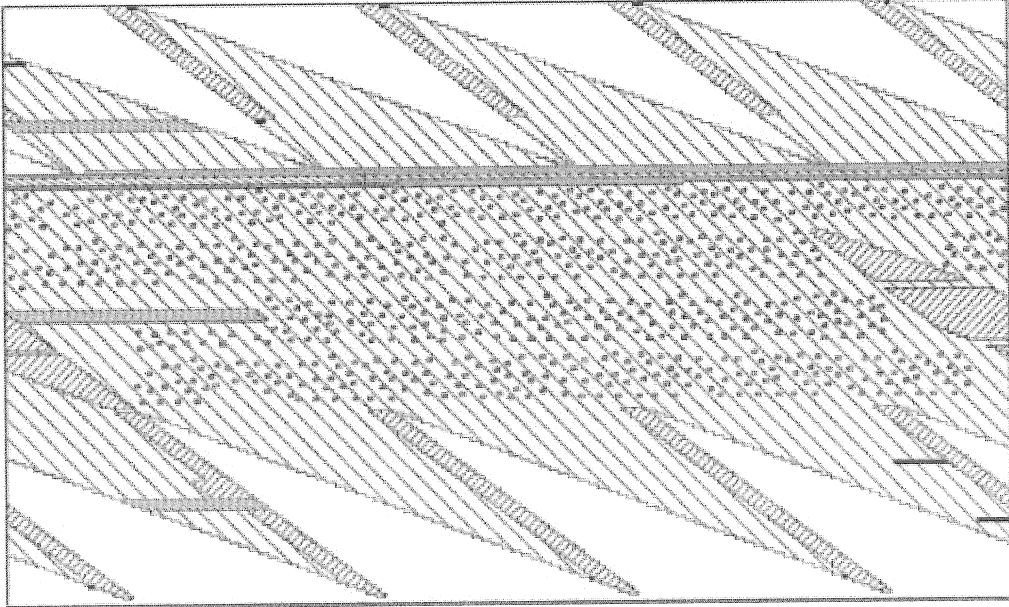


FIG. 26

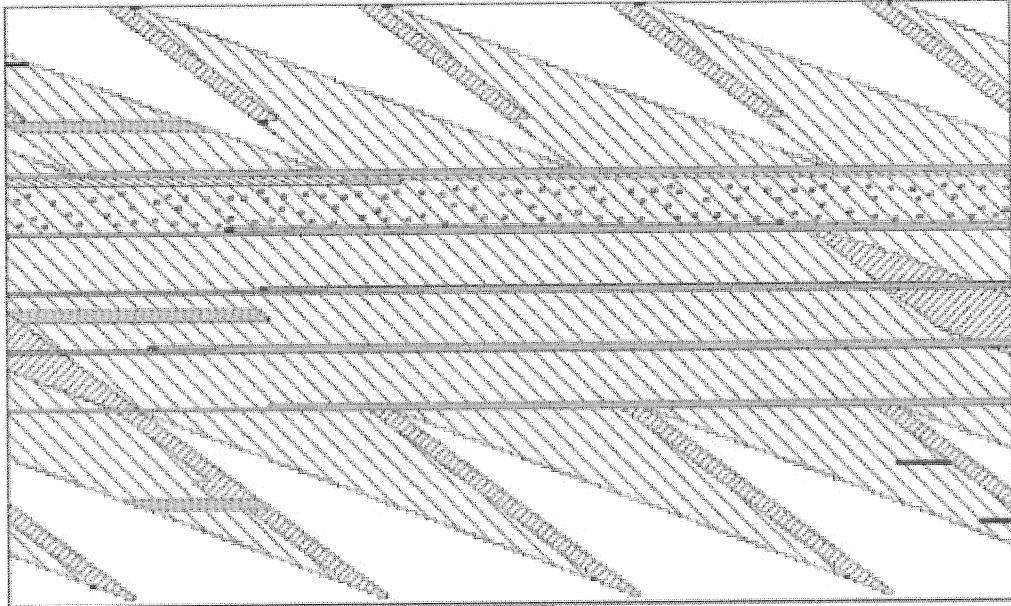


FIG. 27

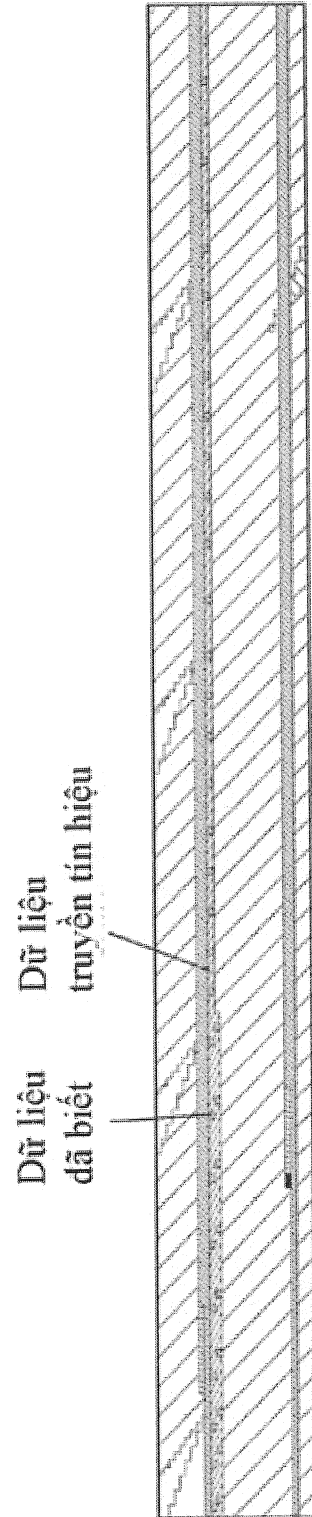


FIG. 28

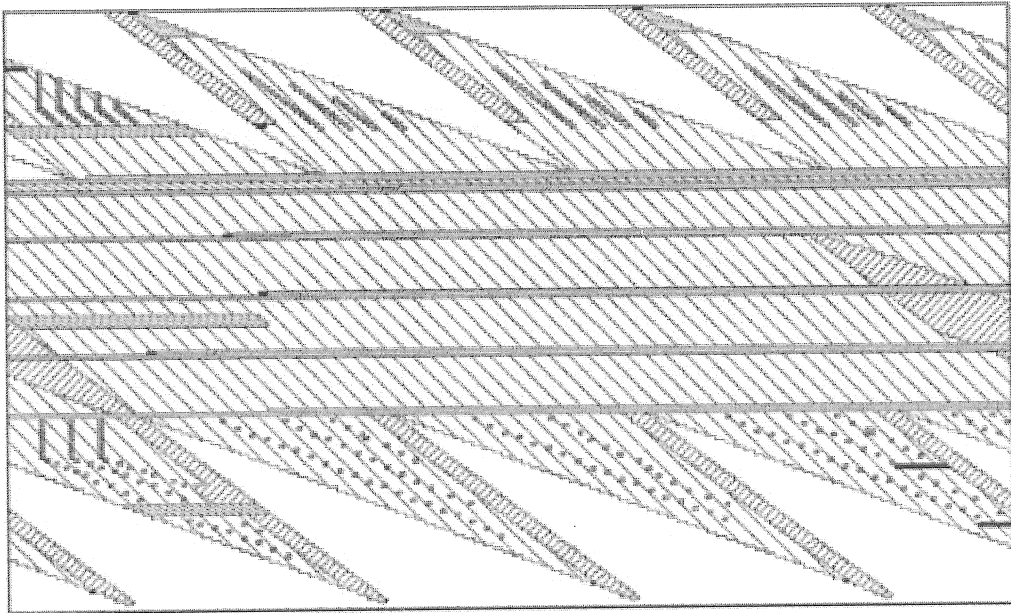


FIG. 29

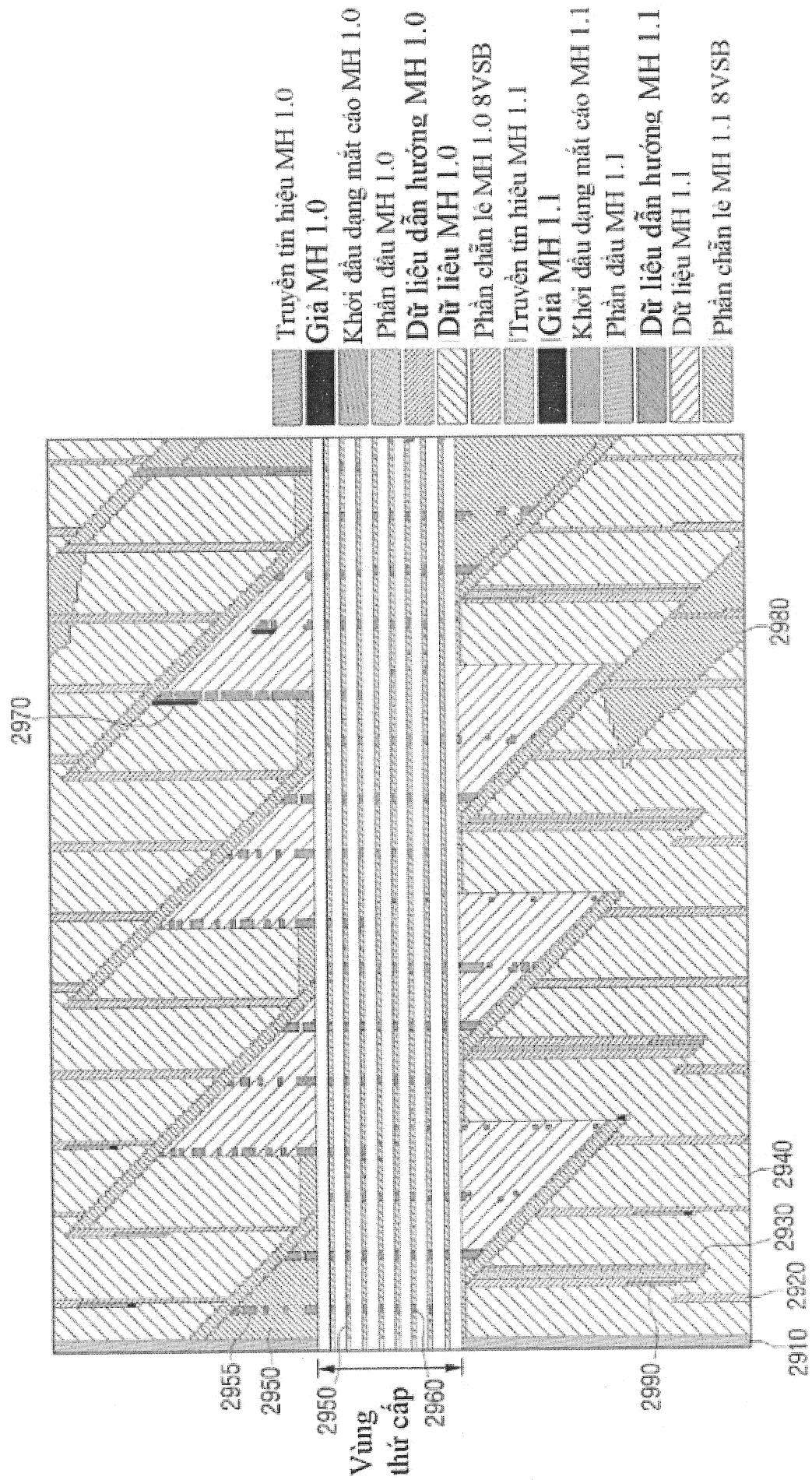


FIG. 30

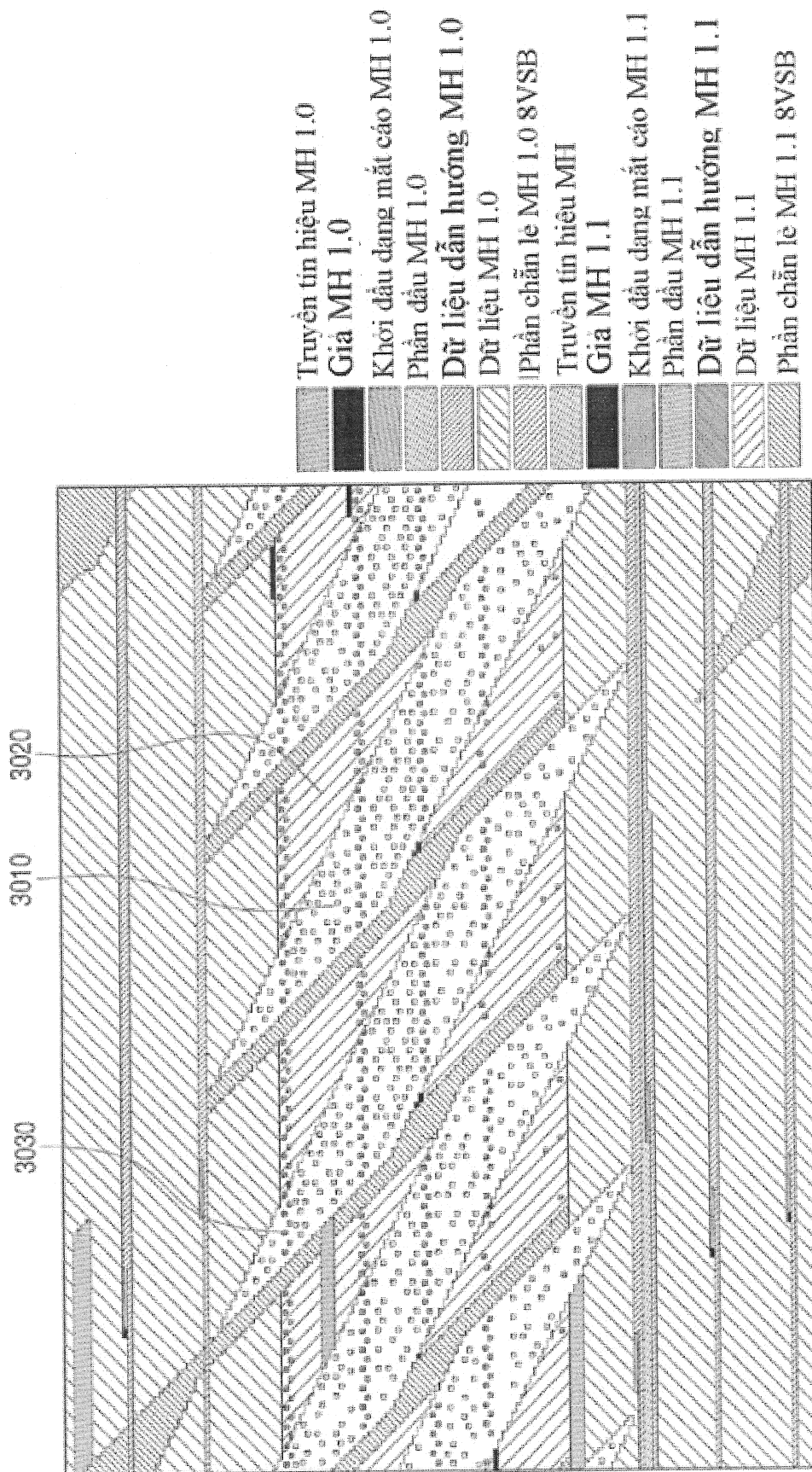


FIG. 31

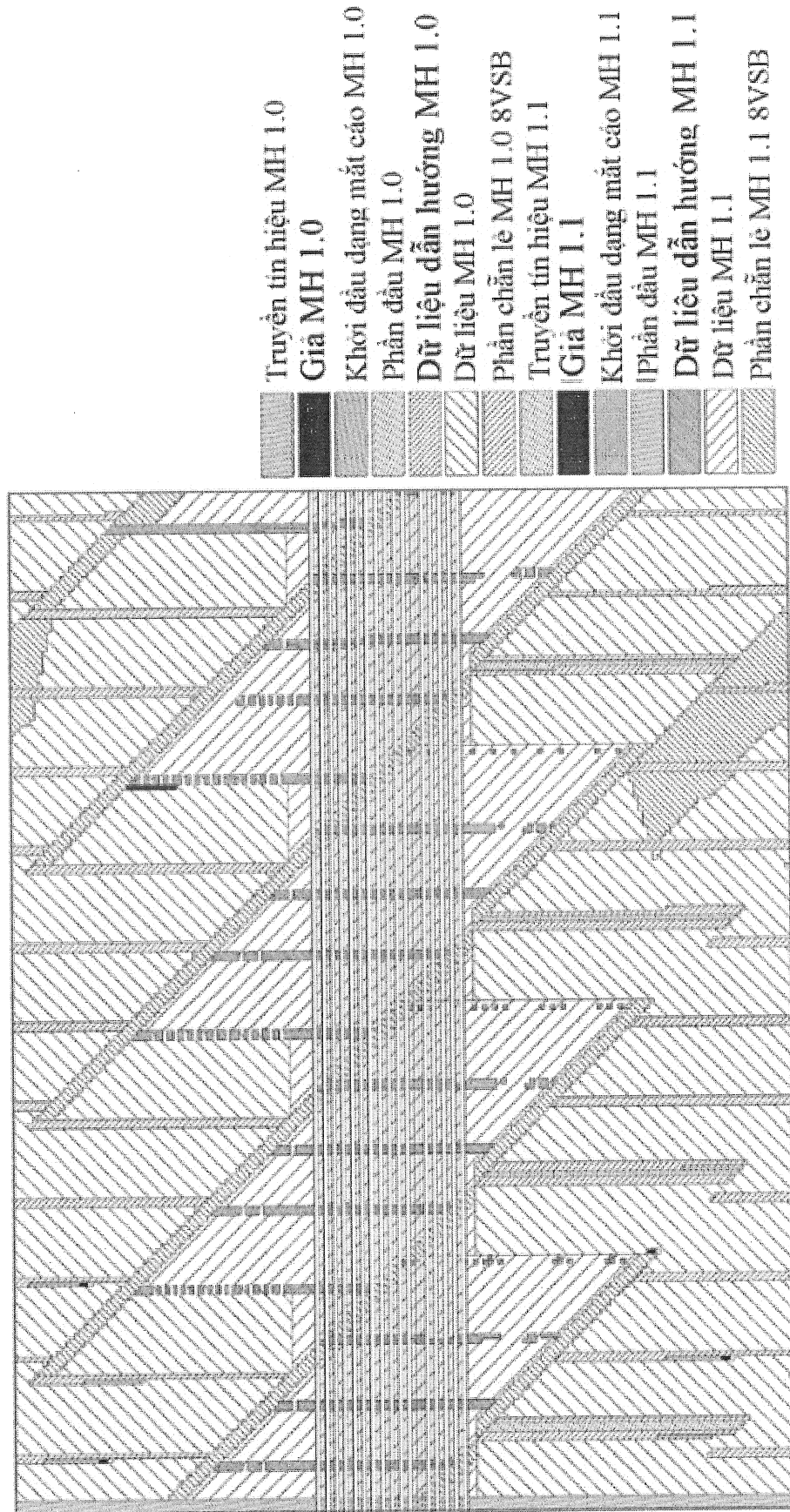


FIG. 32

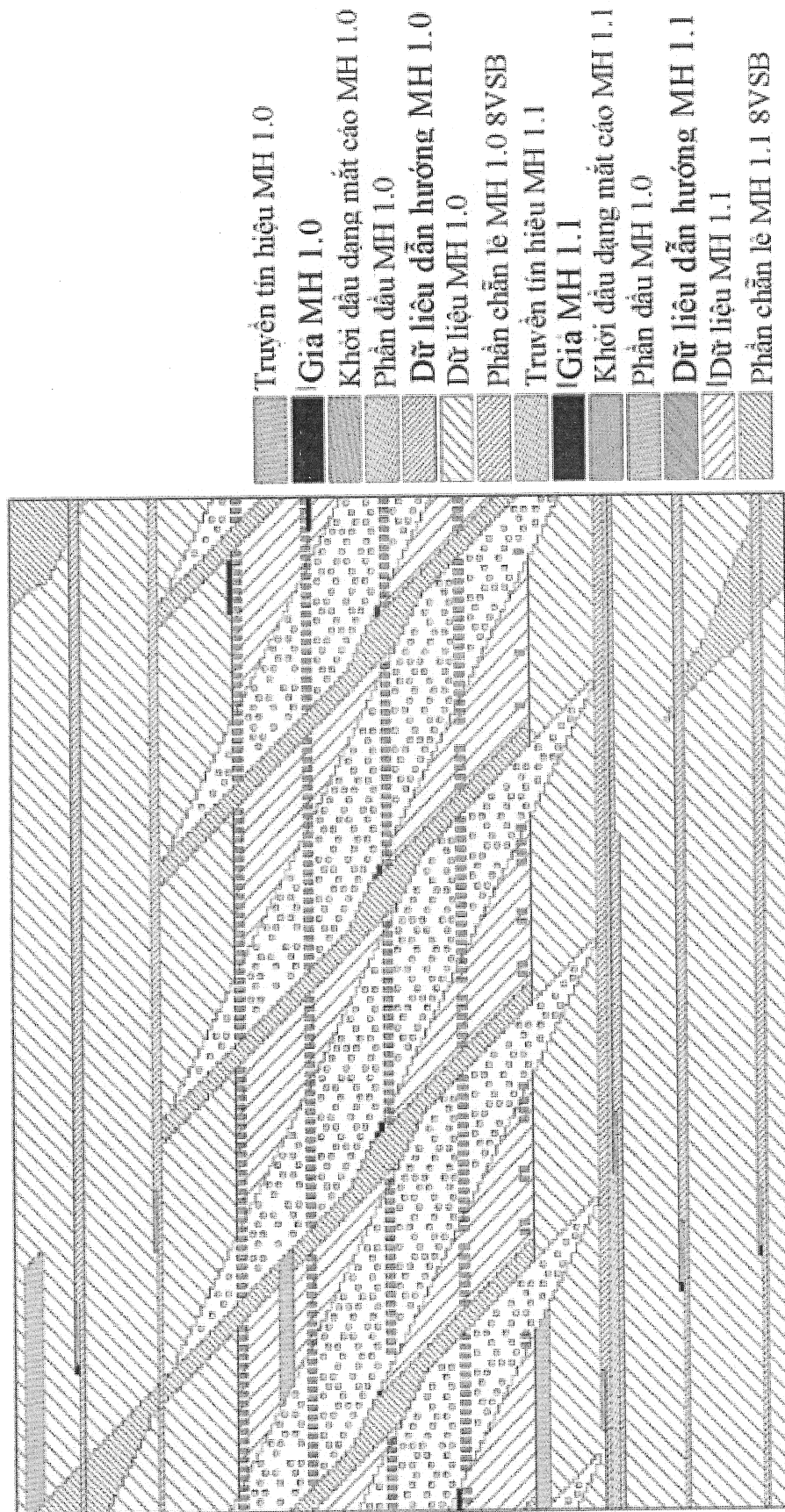


FIG. 33

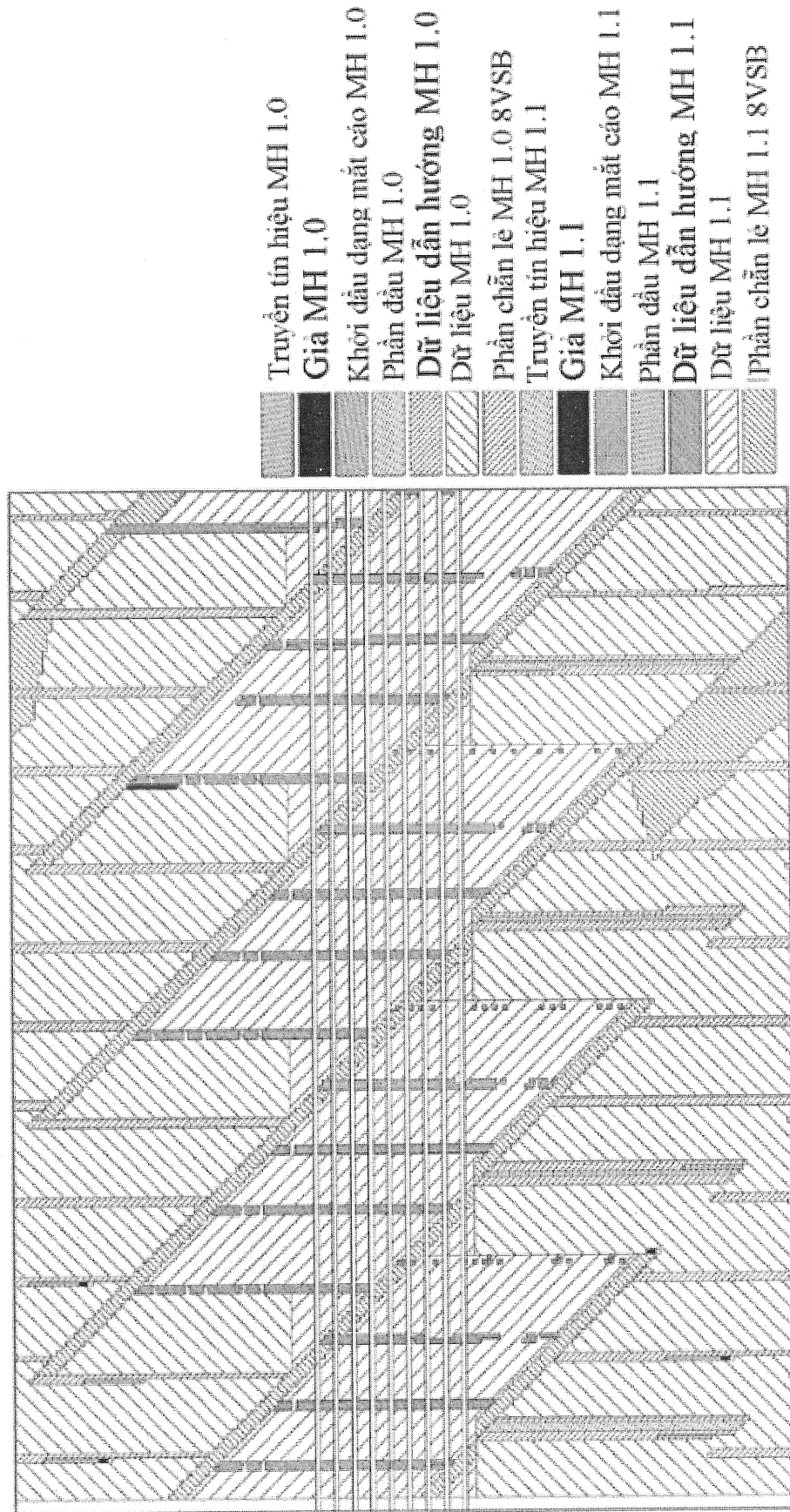


FIG. 34

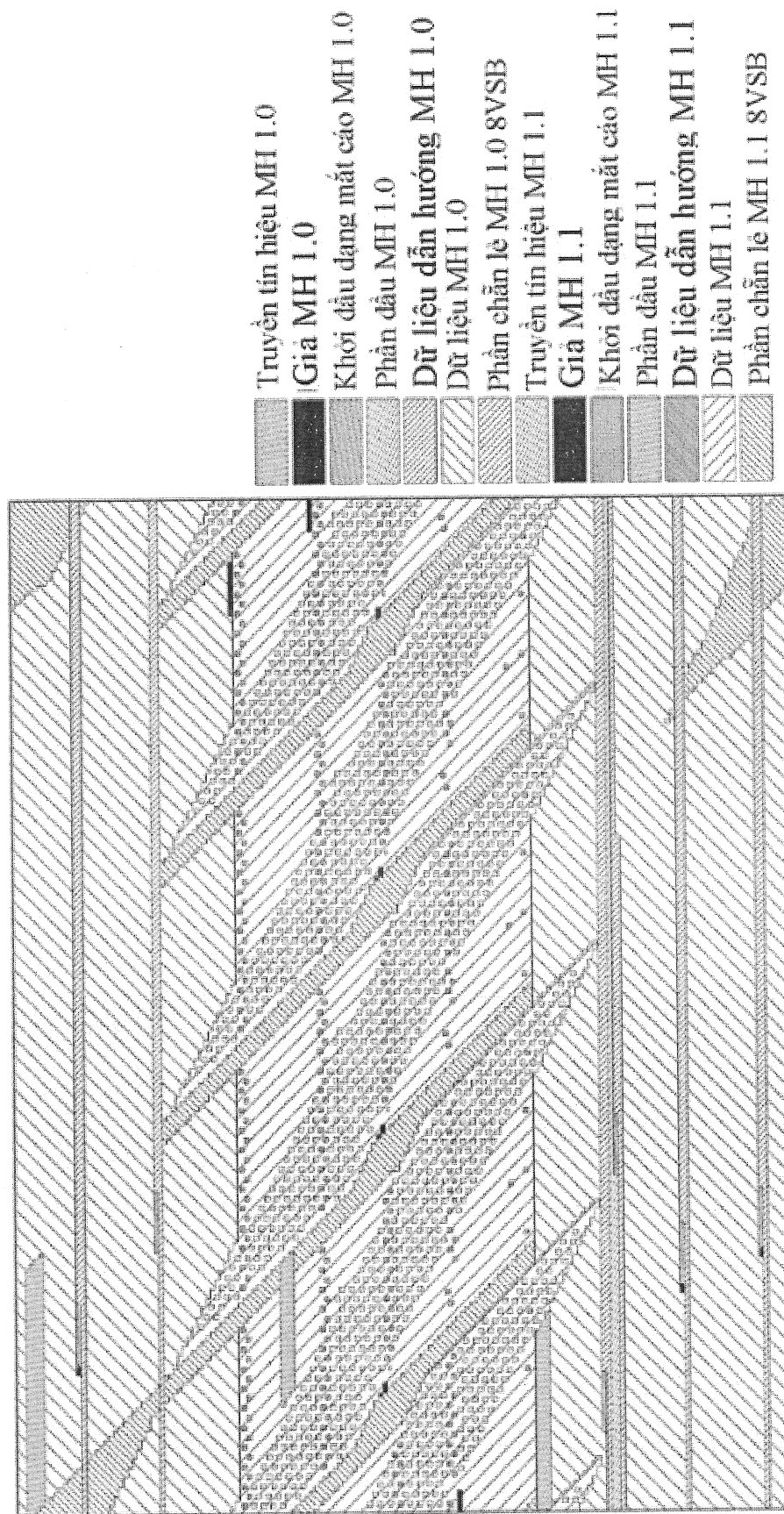
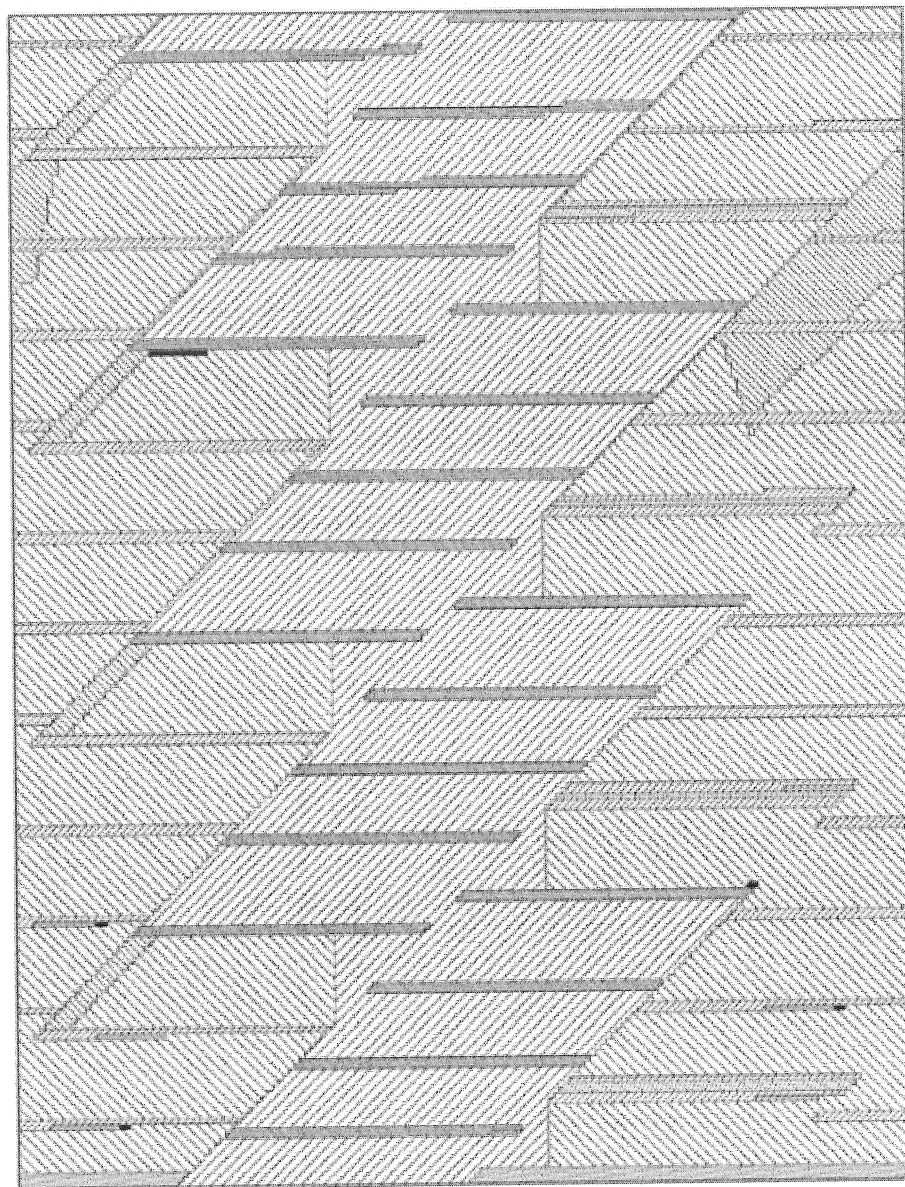


FIG. 35



- Truyền tín hiệu MH 1.0
- Giả MH 1.0
- Khởi đầu dạng mặt cào MH 1.0
- Phân đầu MH 1.0
- Dữ liệu dẫn hướng MH 1.0
- Dữ liệu MH 1.0
- Phản chấn lè MH 1.0 8VSB
- Truyền tín hiệu MH 1.1
- Giả MH 1.1
- Khởi đầu dạng mặt cào MH 1.1
- Phân đầu MH 1.1
- Dữ liệu dẫn hướng MH 1.1
- Dữ liệu MH 1.1
- Phản chấn lè MH 1.1 8VSB

FIG. 36

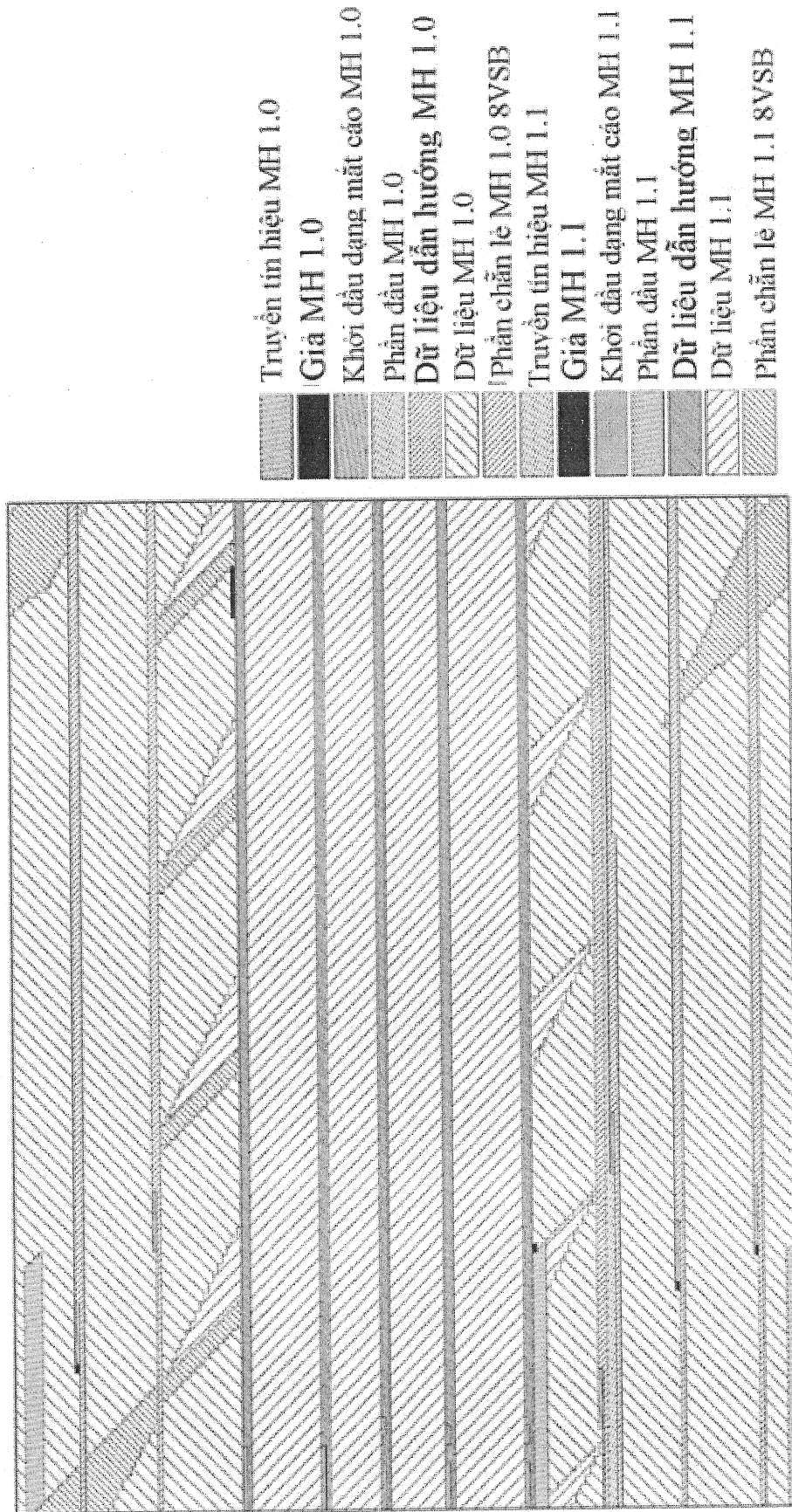


FIG. 37

[illegible]

FIG. 38

[illegible]

FIG. 39

[illegible]

FIG. 40

Di động ATSC 1.1	
Di động ATSC 1.1	
Di động ATSC 1.1	
Di động ATSC 1.1	
Thường	
Thường	
Thường	
• • •	
Thường	
Thường	
Thường	
Di động ATSC 1.1	
Di động ATSC 1.1	
Di động ATSC 1.1	
Di động ATSC 1.1	

a) Chế độ 1

Di động ATSC 1.1
Di động ATSC 1.1
Di động ATSC 1.1
Di động ATSC 1.1
Di động ATSC 1.1
Di động ATSC 1.1
Di động ATSC 1.1
Di động ATSC 1.1
Thường
Thường
• • •
Thường
Thường
Di động ATSC 1.1
Di động ATSC 1.1
Di động ATSC 1.1
Di động ATSC 1.1
Di động ATSC 1.1
Di động ATSC 1.1
Di động ATSC 1.1
Di động ATSC 1.1

b) Chế độ 2

Di động ATSC 1.1	12
Di động ATSC 1.1	
• • •	
Di động ATSC 1.1	
Di động ATSC 1.1	
Thường	
• • •	
Thường	12
Di động ATSC 1.1	
Di động ATSC 1.1	
• • •	
Di động ATSC 1.1	
Di động ATSC 1.1	

c) Chế độ 3

FIG. 41

Loại 1
Loại 0
Loại 0
Loại 0
Loại 1
Loại 0
Loại 0
Loại 0
Loại 1
Loại 0
Loại 0
Loại 0

FIG. 42

ex 1	Loại 0	Loại 1-1	Loại 1-2	Loại 1-3	Loại 1-4	Loại 0	Loại 1-1	Loại 1-2	Loại 1-3	Loại 1-4	Loại 0	Loại 1-2	Loại 1-3	Loại 1-4	Loại 0
ex 2	Loại 1-4	Loại 0	Loại 1-4	Loại 0	Loại 1-4	Loại 0	Loại 1-4	Loại 0	Loại 1-4	Loại 0	Loại 1-4	Loại 0	Loại 1-4	Loại 0	Loại 0
ex 3	Loại 1-4	Loại 0	Loại 1-3	Loại 0	Loại 1-4	Loại 0	Loại 1-3	Loại 0	Loại 1-4	Loại 0	Loại 1-4	Loại 0	Loại 1-3	Loại 1-3	Loại 0
ex 4	Loại 1-4	Loại 1-2	Loại 1-3	Loại 0	Loại 1-4	Loại 1-2	Loại 1-3	Loại 0	Loại 1-4	Loại 1-2	Loại 1-3	Loại 0	Loại 1-4	Loại 1-2	Loại 0
ex 5	Loại 1-4	Loại 1-2	Loại 1-3	Loại 1-1	Loại 1-4	Loại 1-2	Loại 1-3	Loại 1-1	Loại 1-4	Loại 1-2	Loại 1-3	Loại 1-1	Loại 1-4	Loại 1-2	Loại 1-1
ex 6	Loại 1-4	Loại 1-4	Loại 1-4	Loại 1-4	Loại 1-4	Loại 1-4	Loại 1-4	Loại 1-4	Loại 1-4	Loại 1-4	Loại 1-4	Loại 1-4	Loại 1-4	Loại 1-4	Loại 1-4

FIG. 43

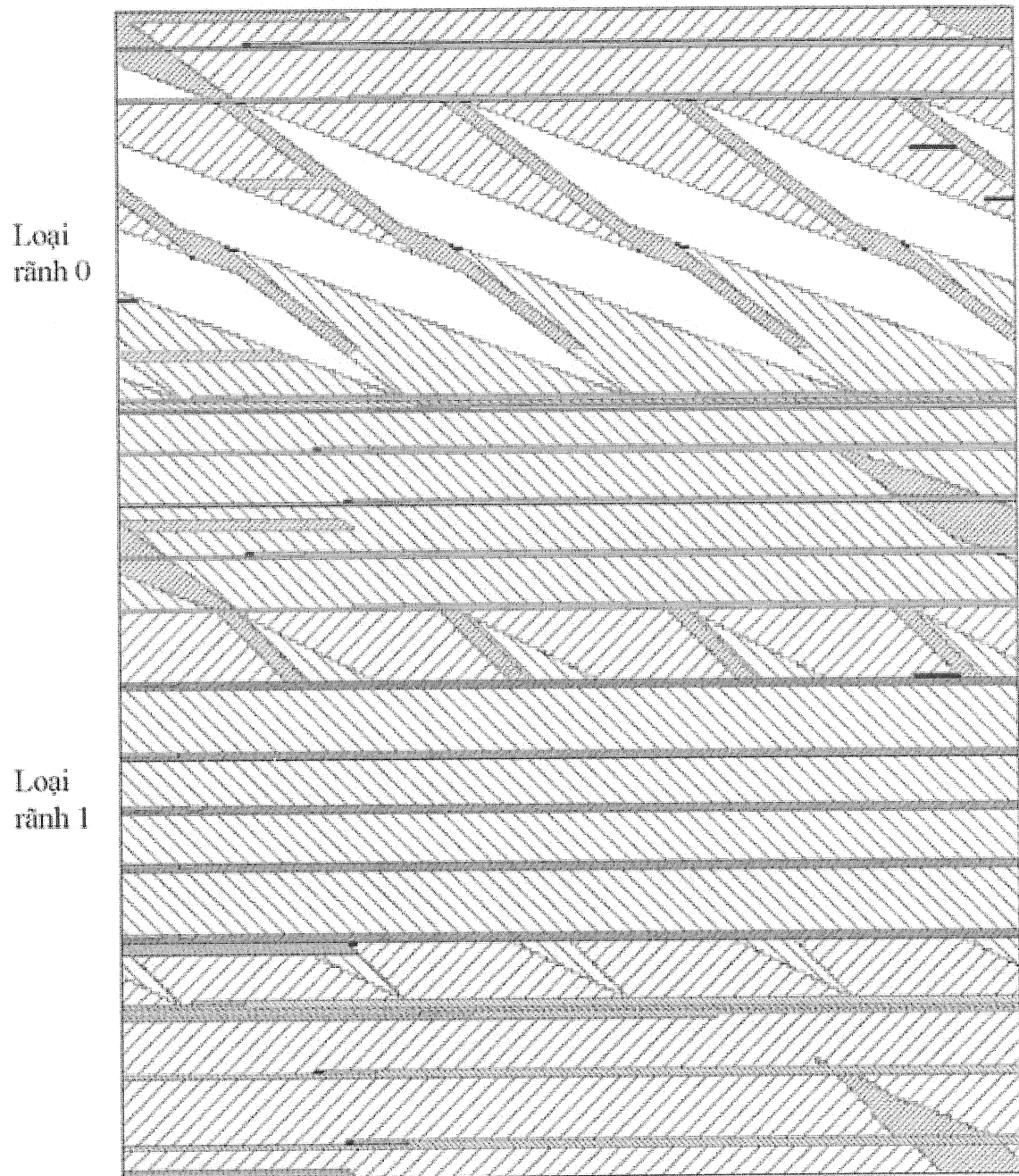


FIG. 44

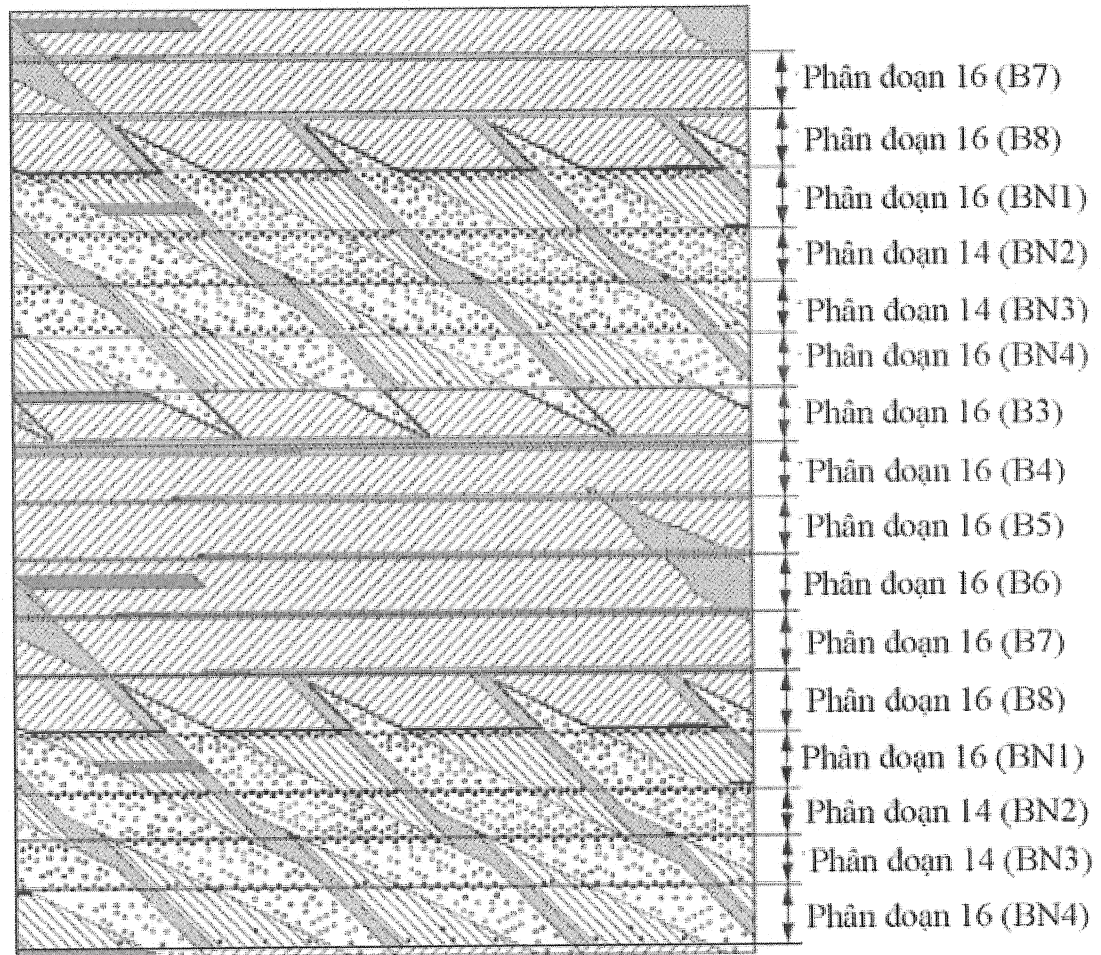


FIG. 45

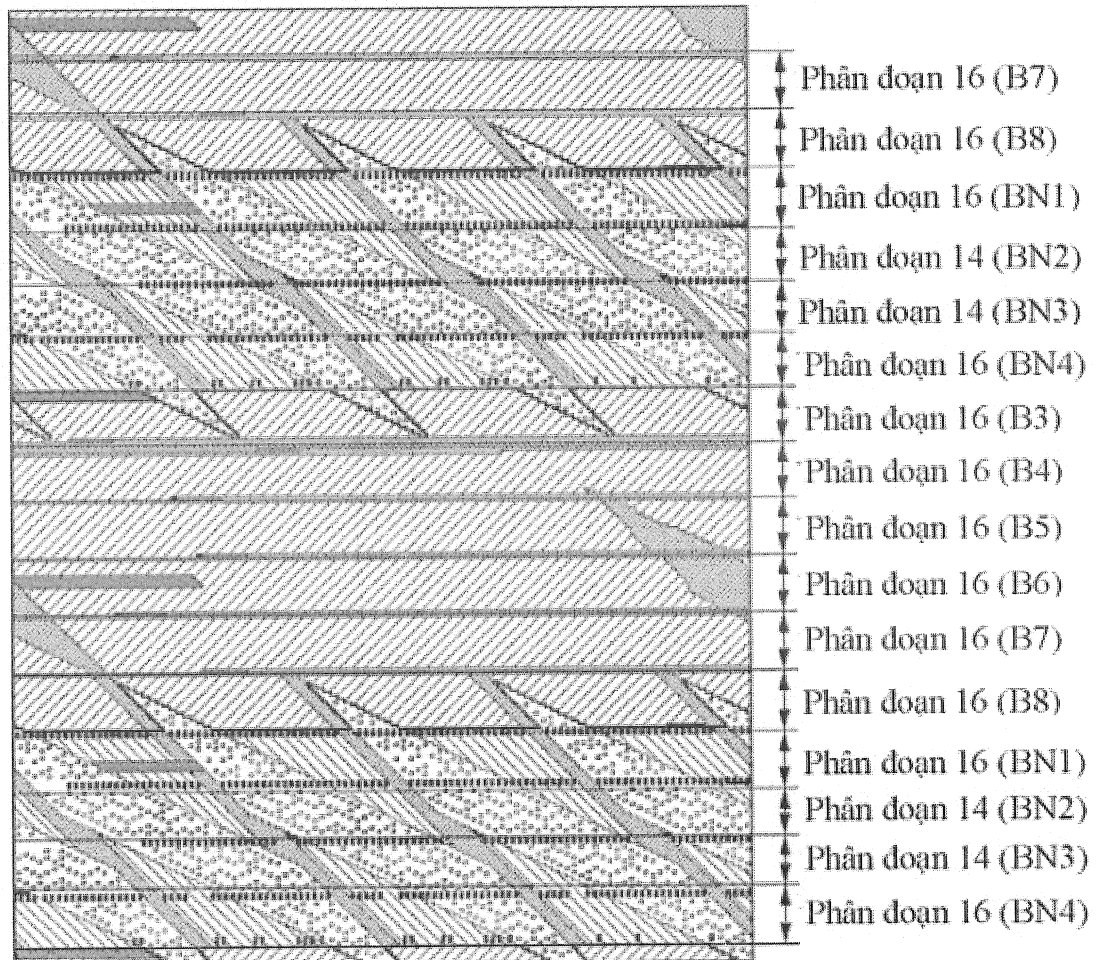


FIG. 46

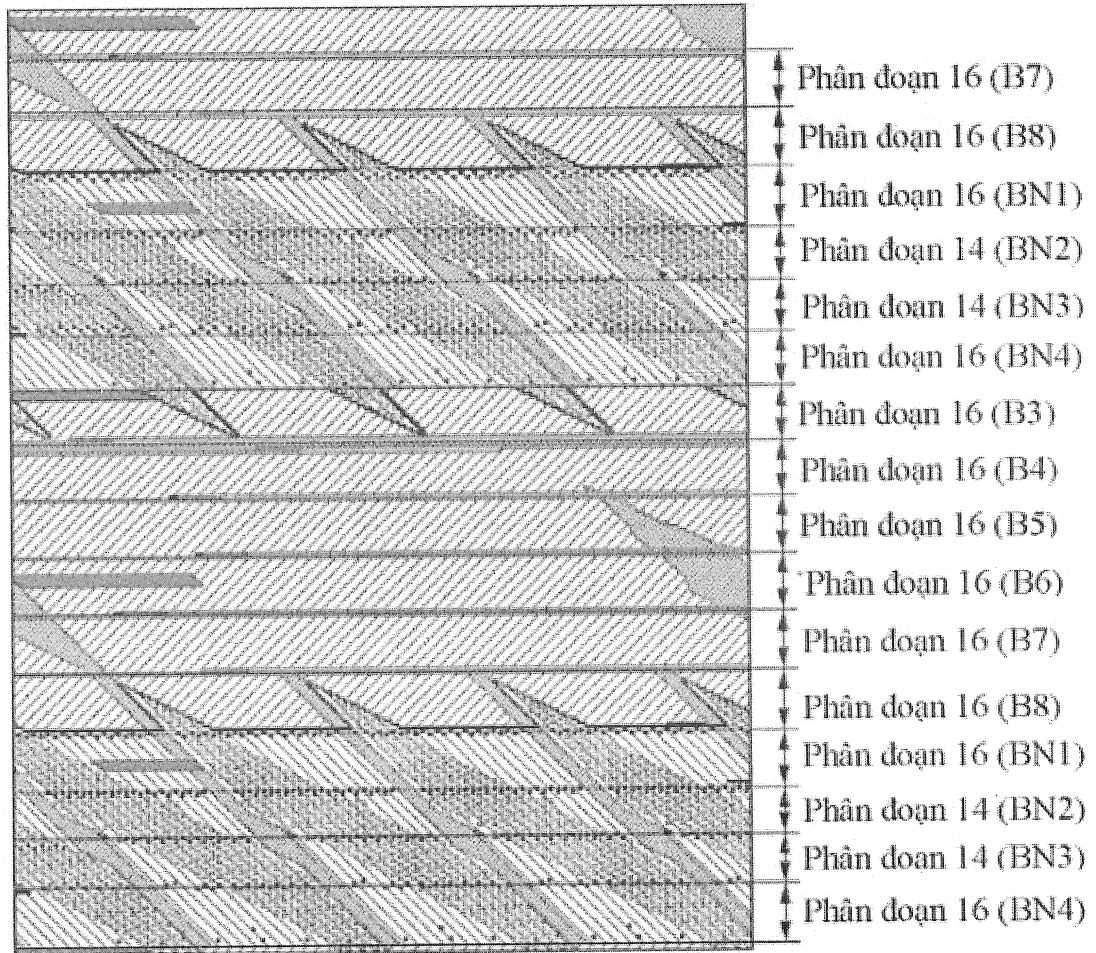


FIG. 47

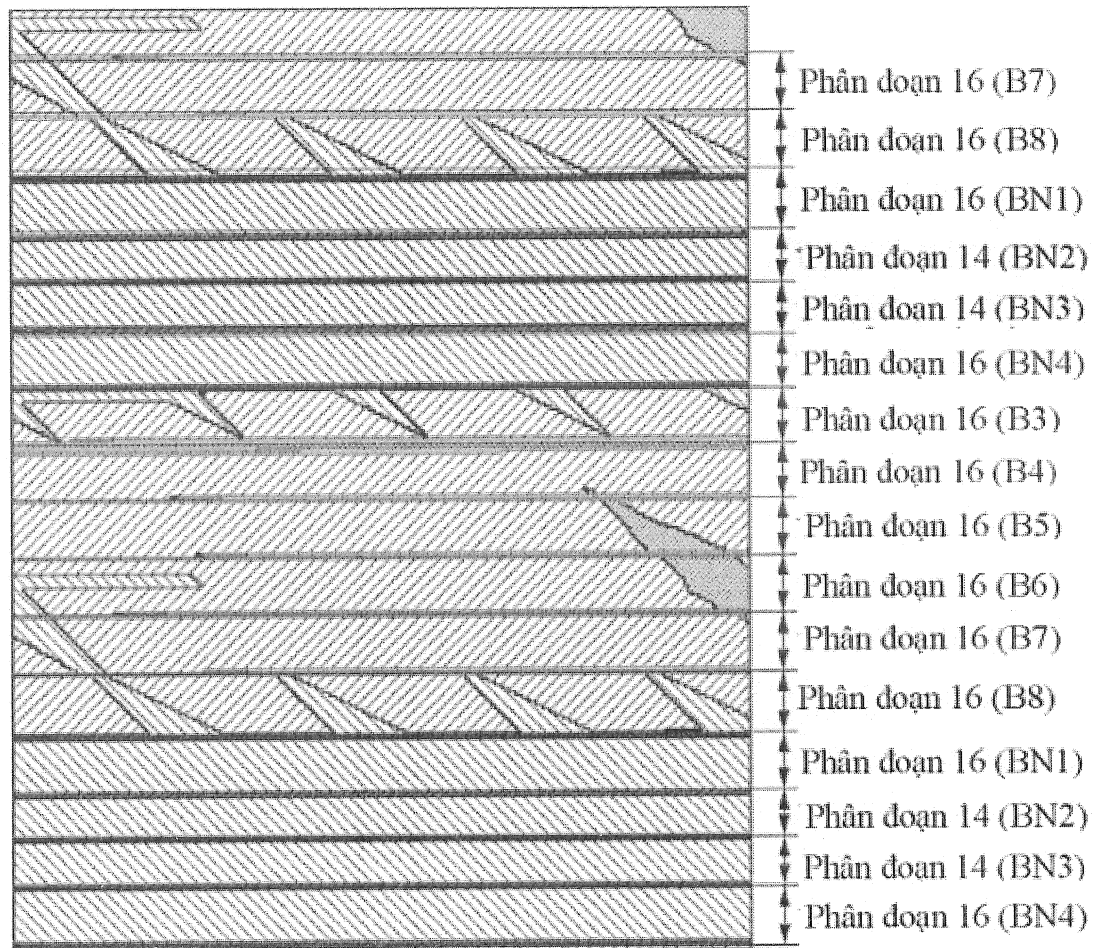


FIG. 48

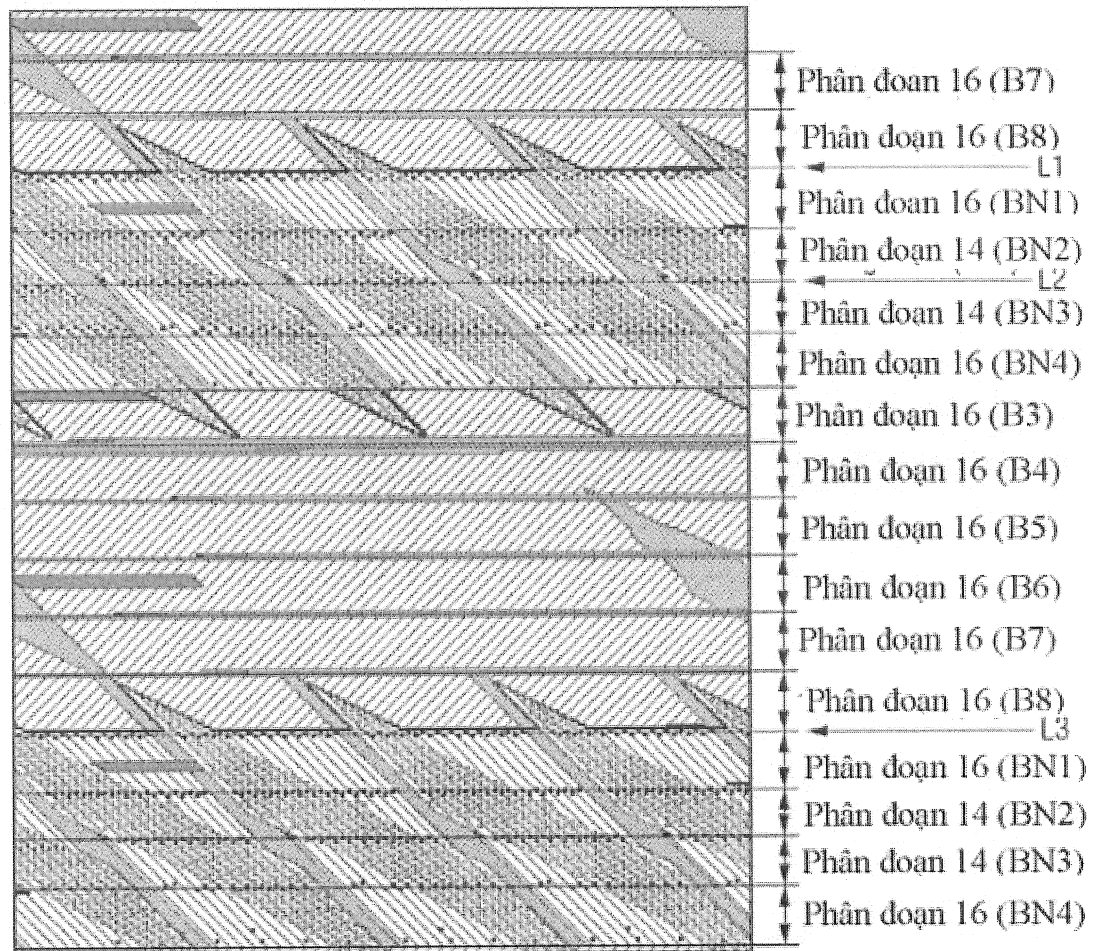


FIG. 49

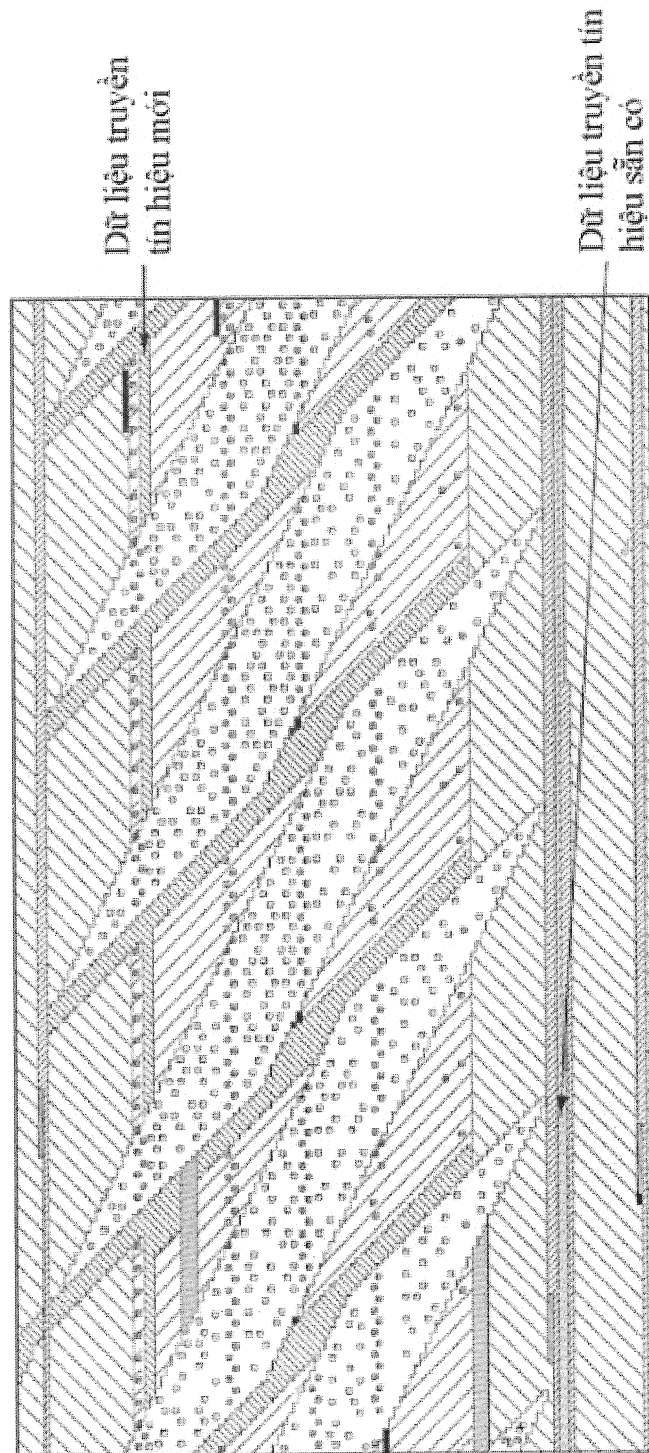


FIG. 50

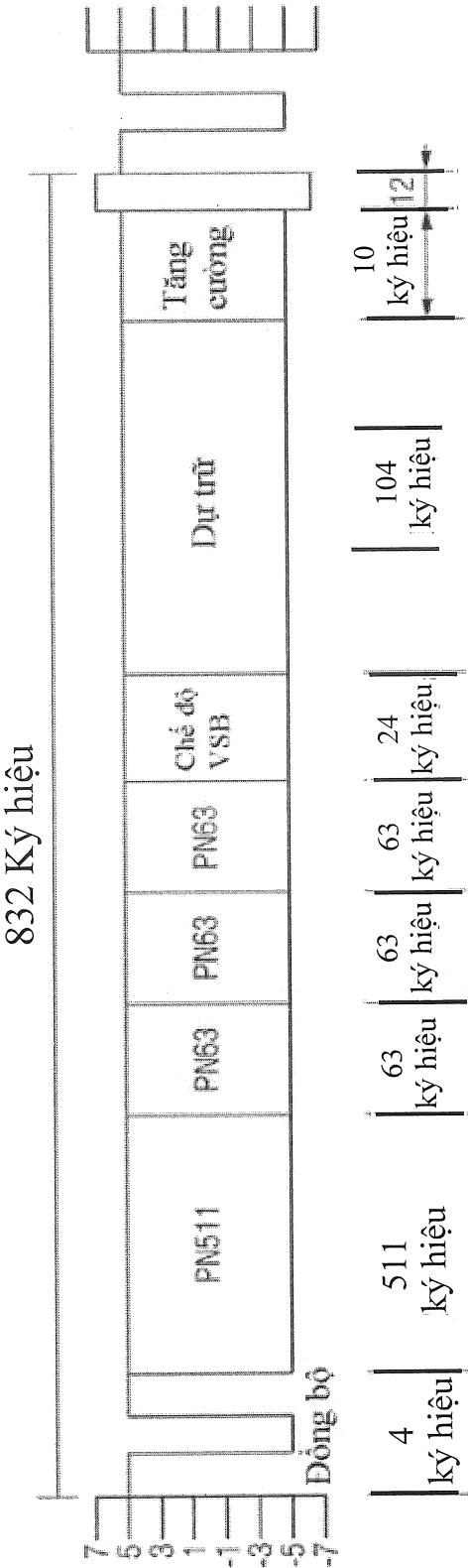


FIG. 51

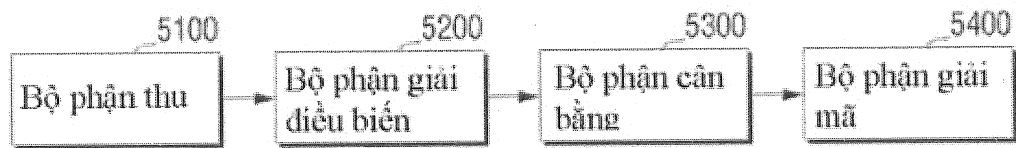


FIG. 53

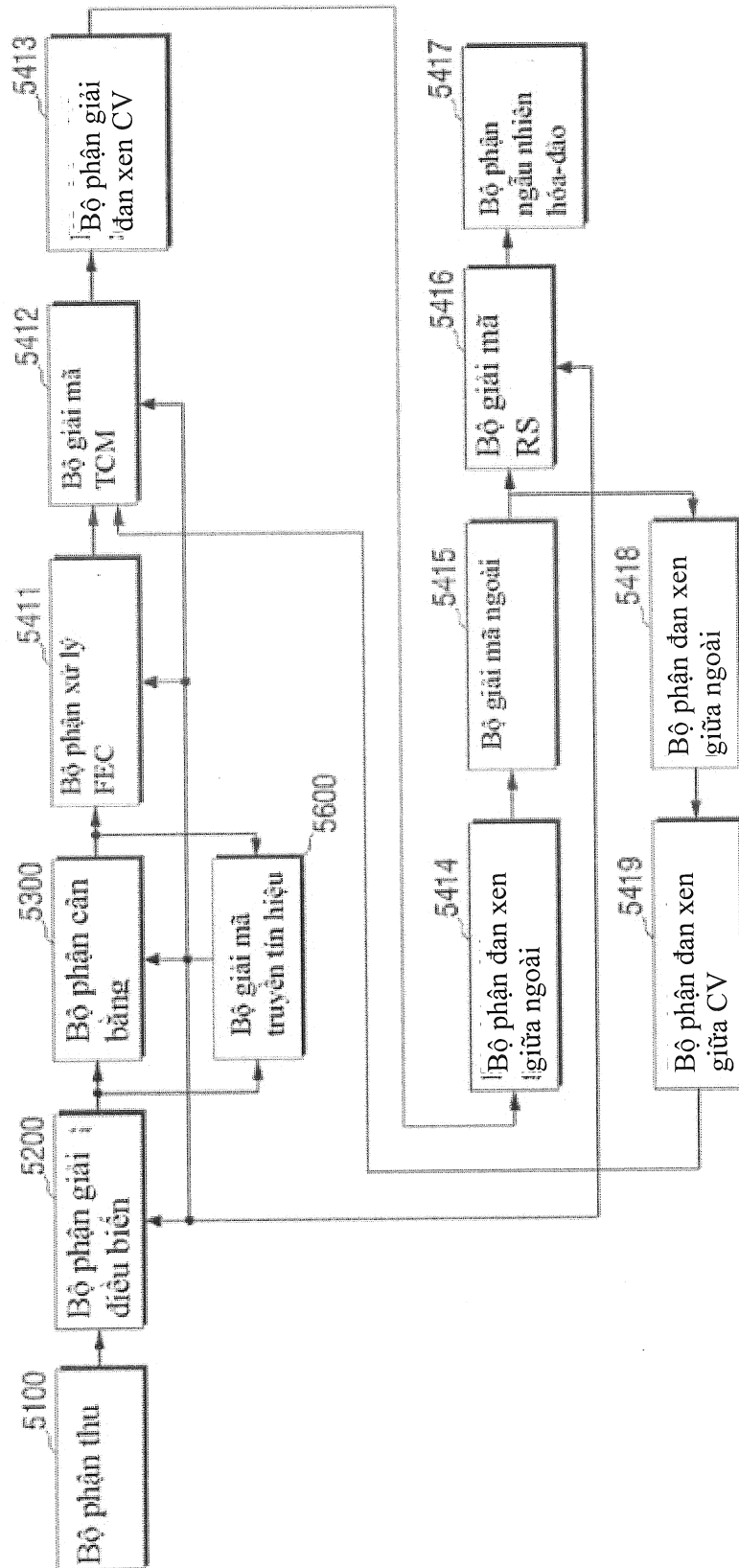


FIG. 54

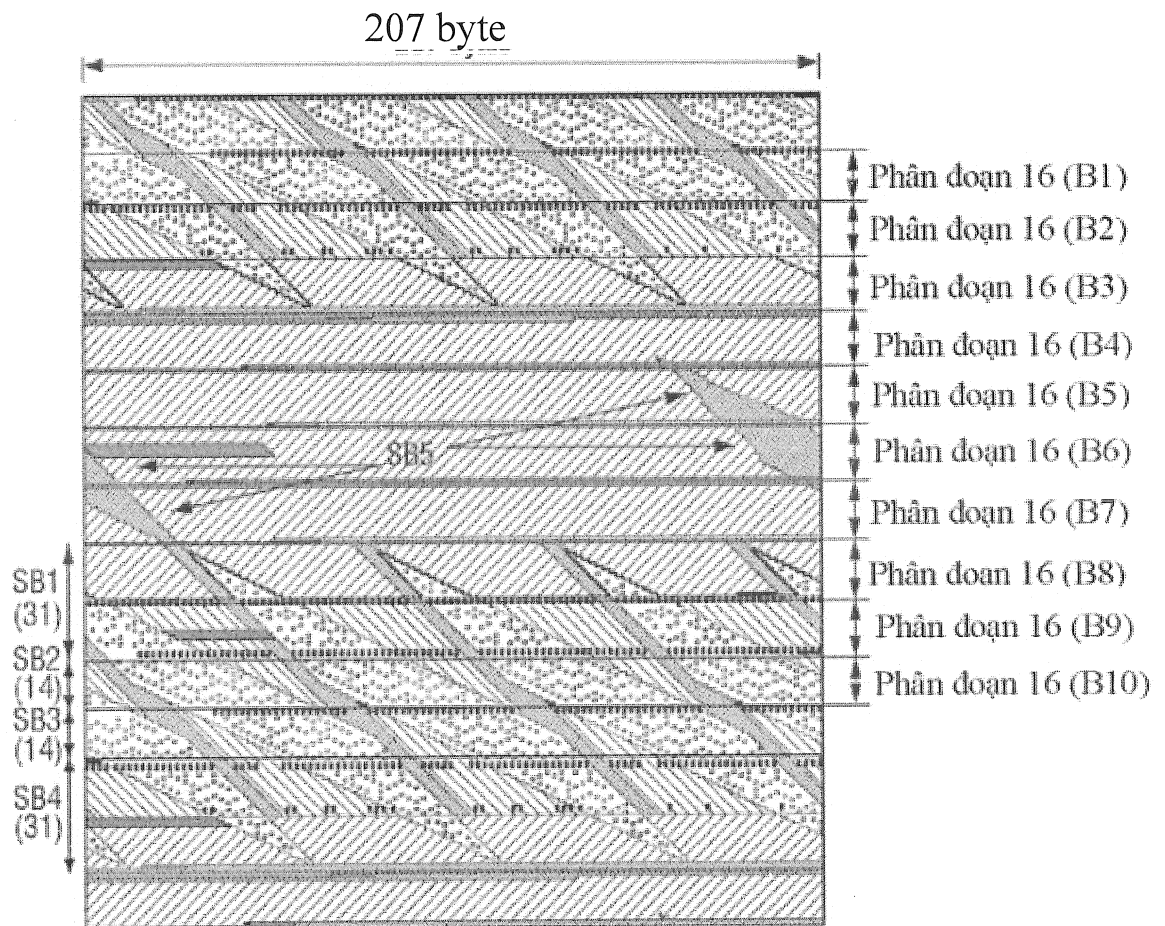


FIG. 55

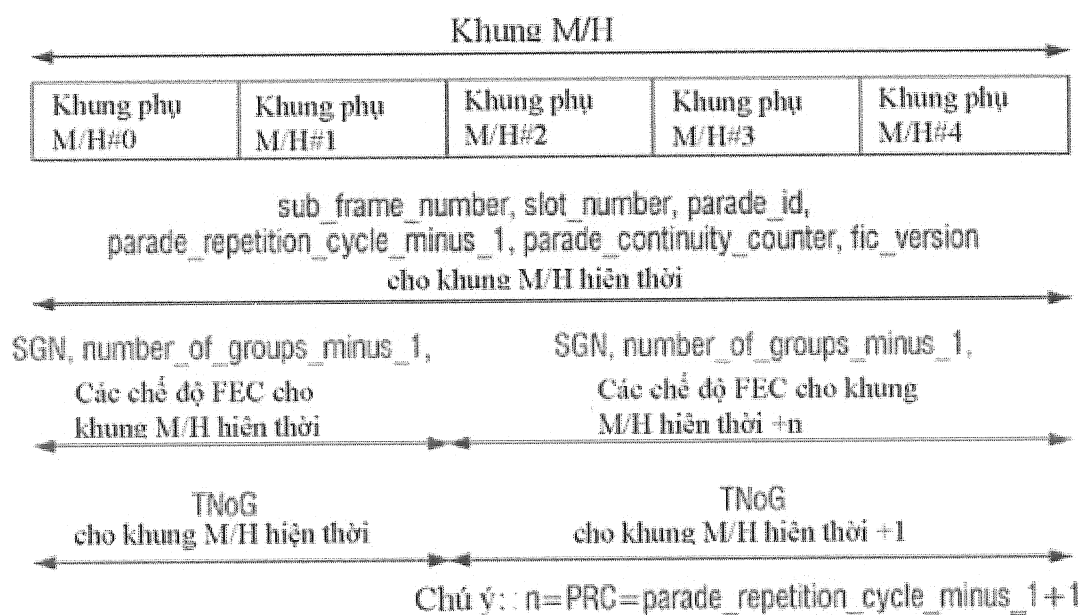


FIG. 56

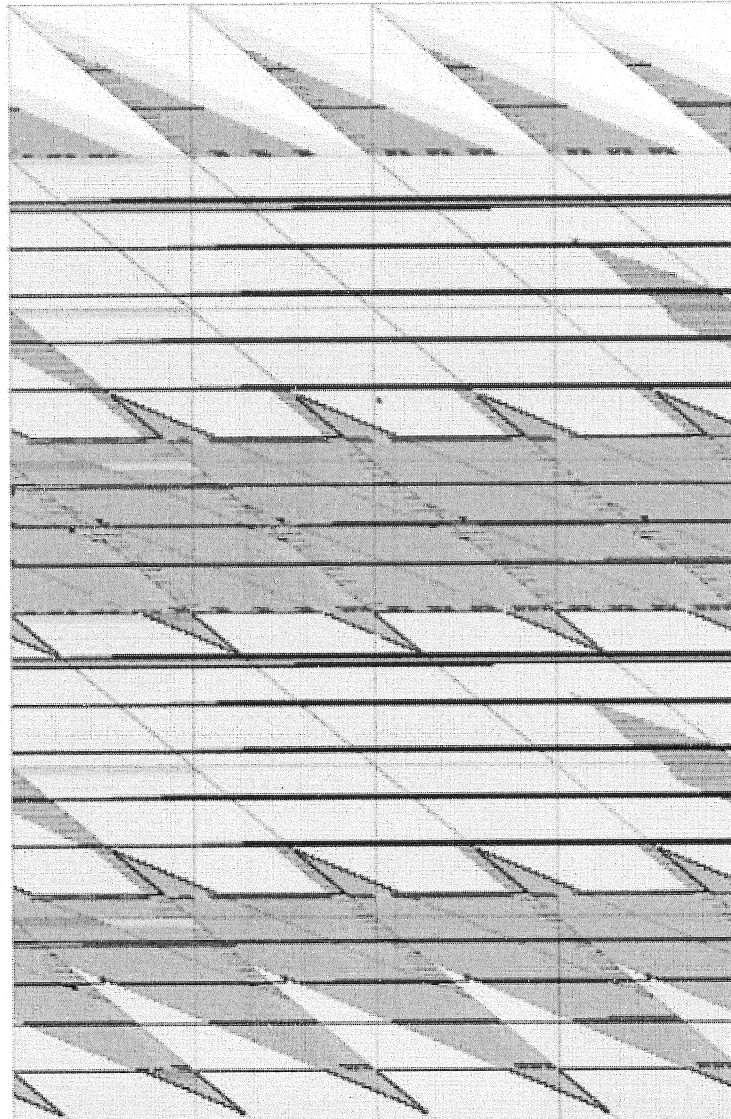


FIG. 57

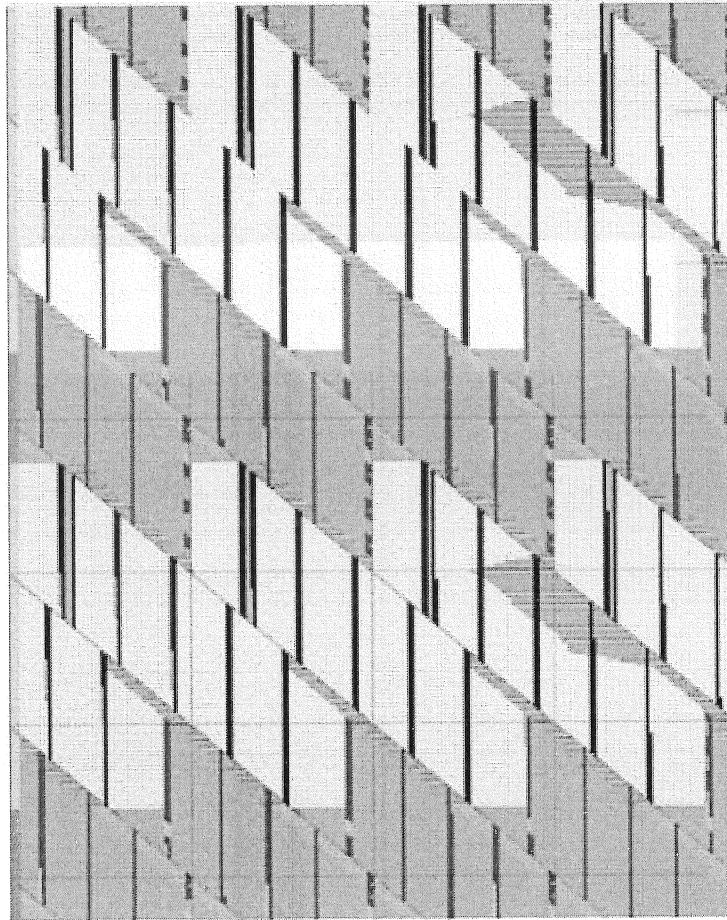


FIG. 58

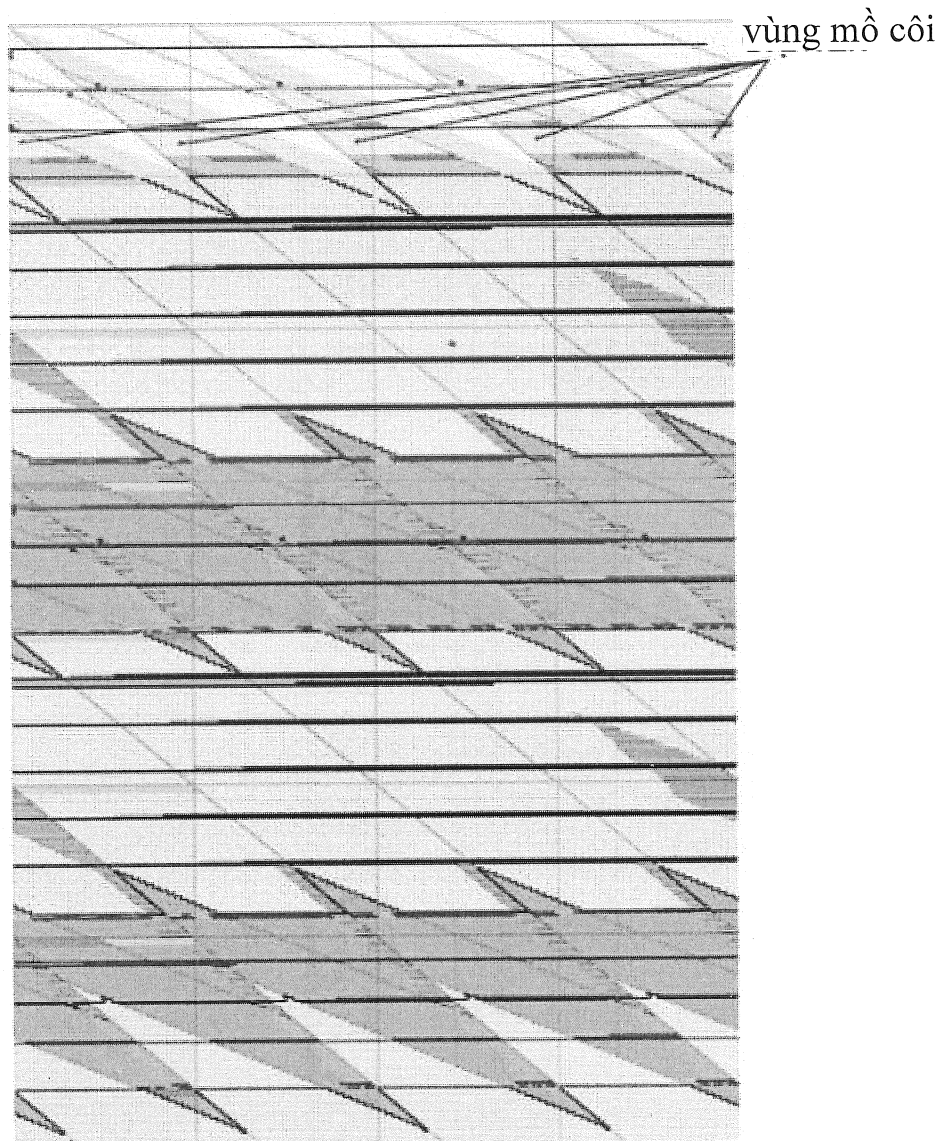


FIG. 59

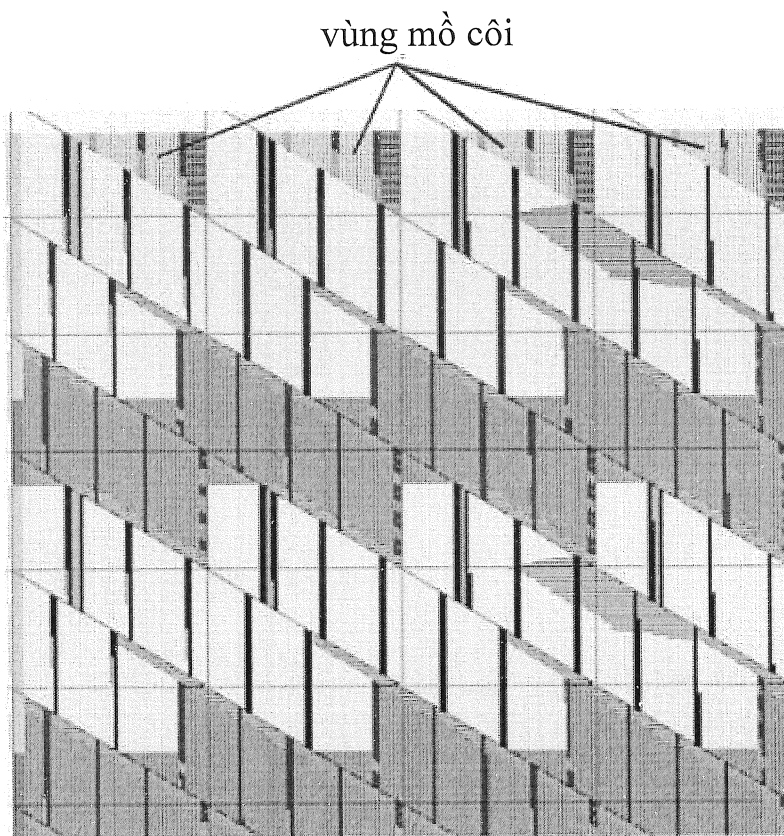


FIG. 60

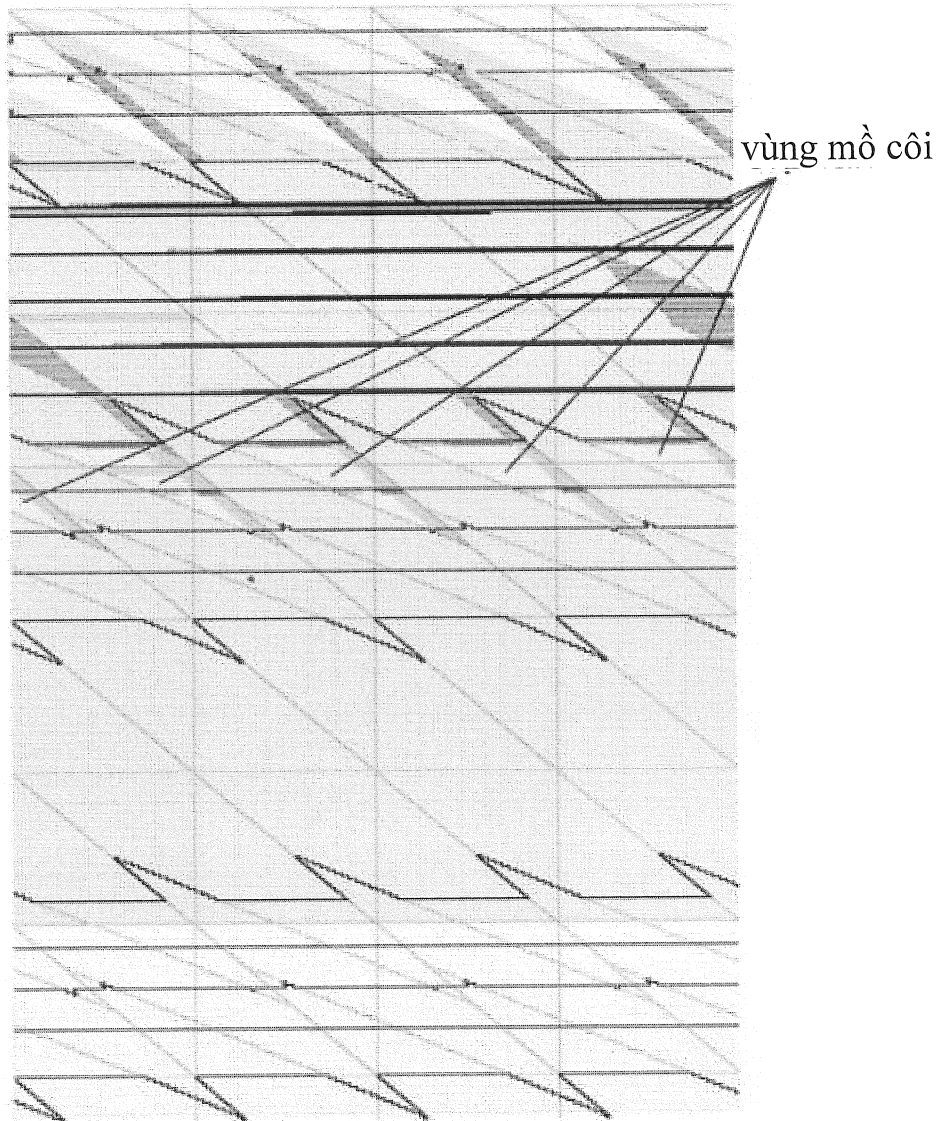


FIG. 61

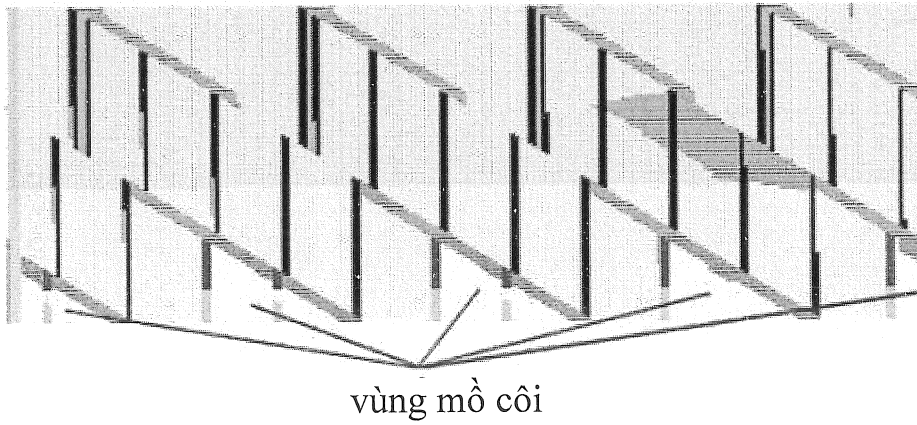


FIG. 62

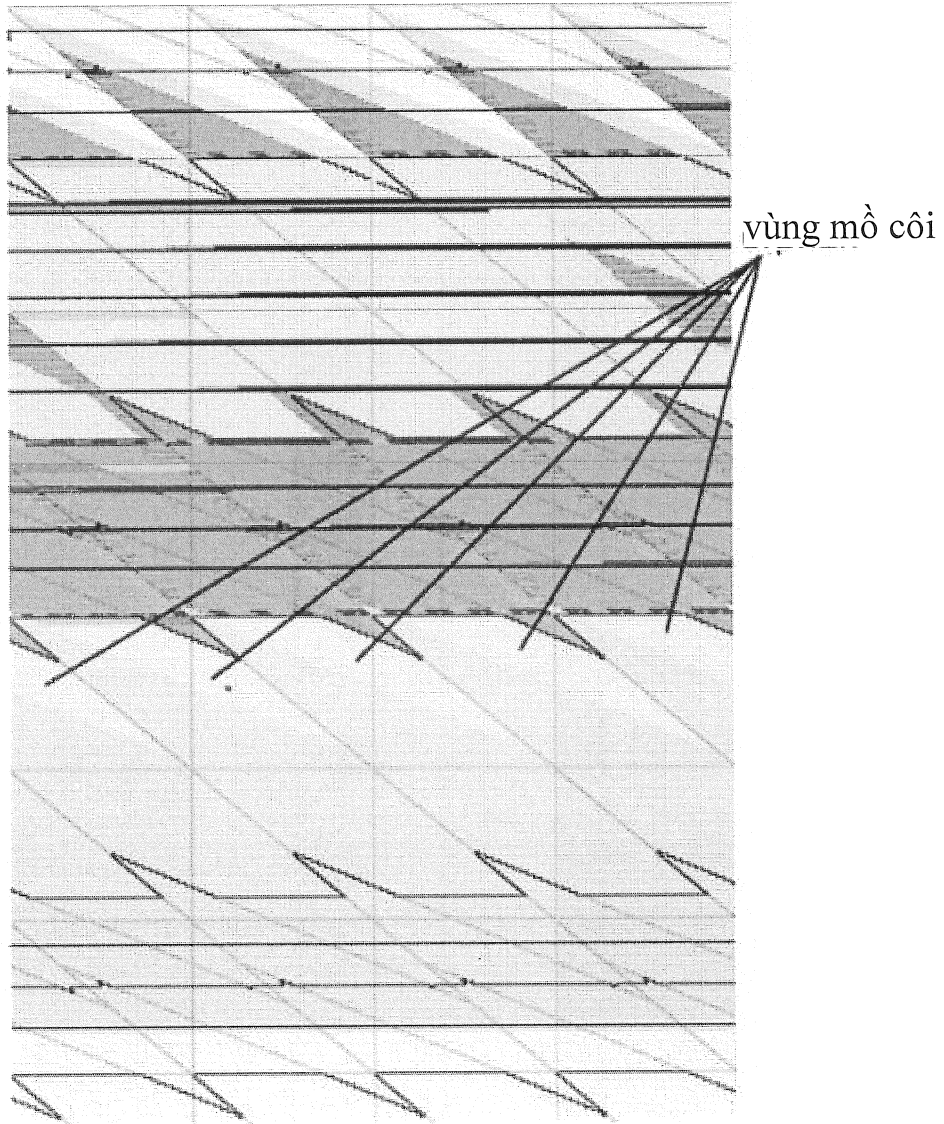


FIG. 63

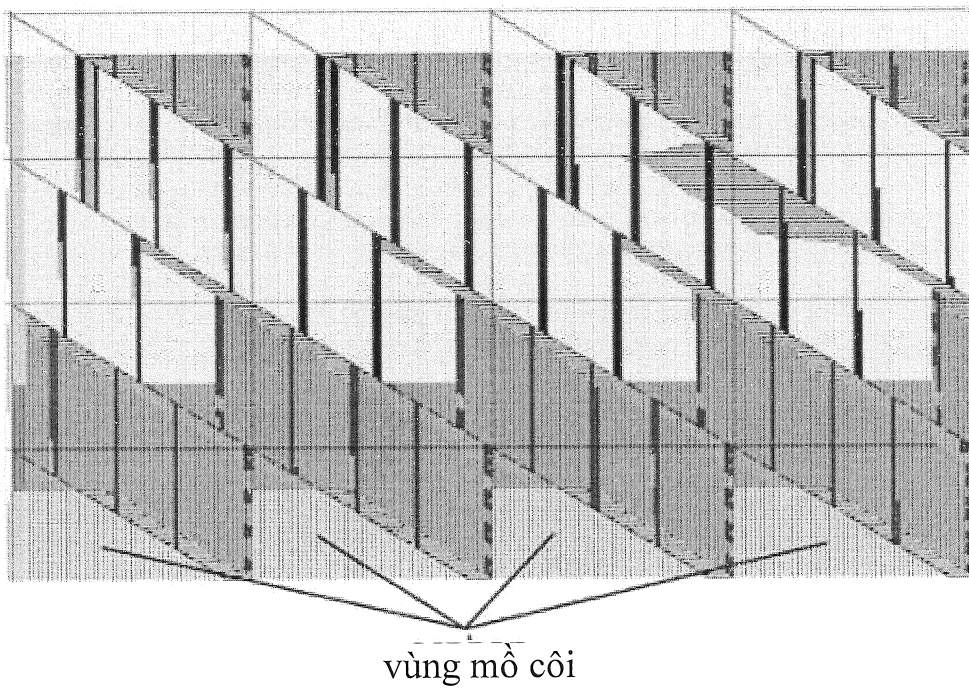


FIG. 64

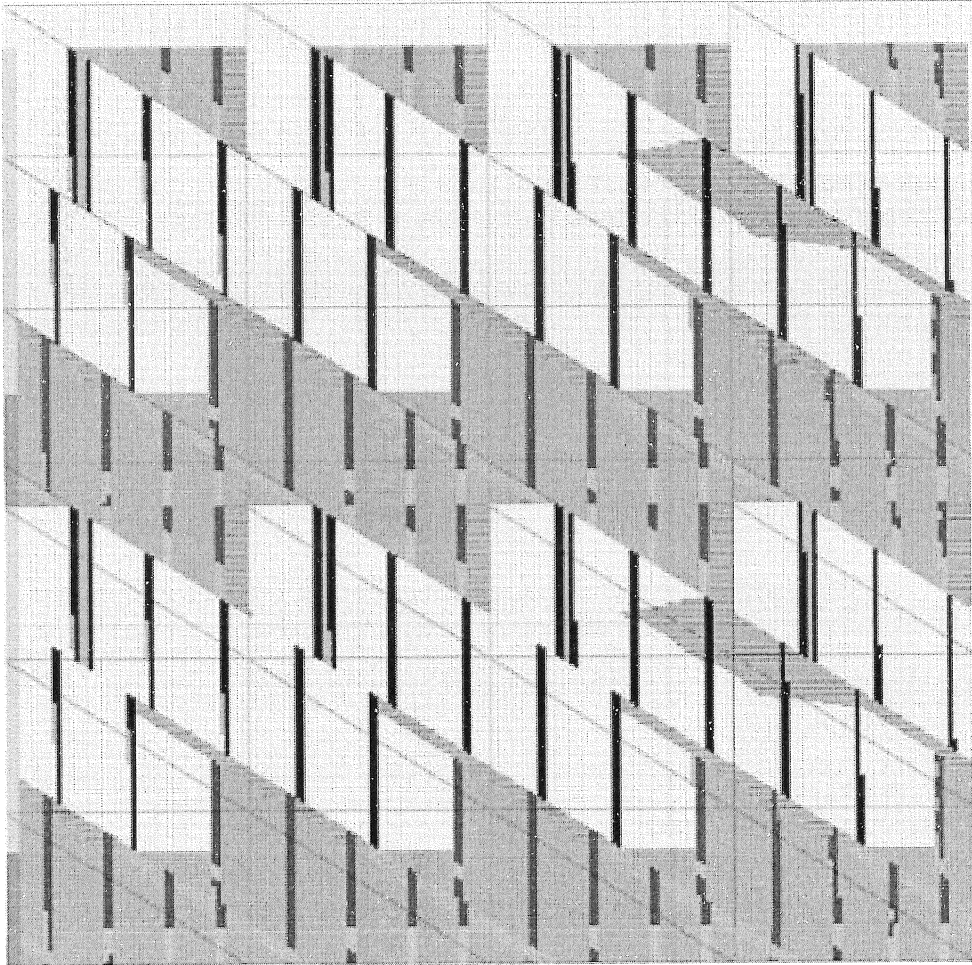


FIG. 65

