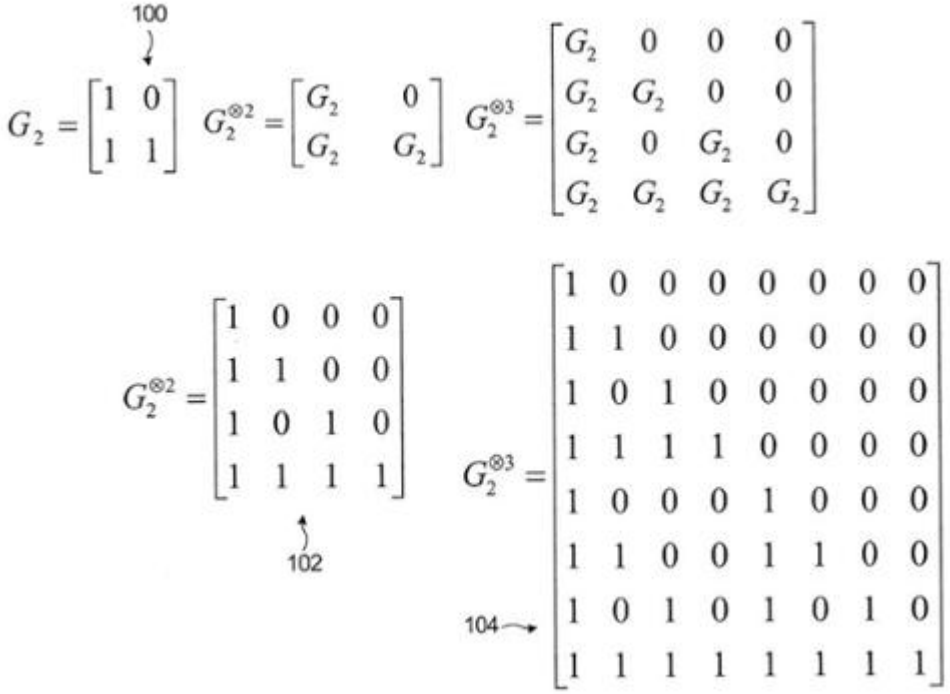
(12) BẢN MÔ TẢ SÁNG CHẾ THUỘC BẰNG ĐỘC QUYỀN SÁNG CHẾ
(19) **Cộng hòa xã hội chủ nghĩa Việt Nam (VN)**
CỤC SỞ HỮU TRÍ TUỆ
(11) **1-0034891**
(51)[8] **H04L 9/28**
(13) **B**

(21) 1-2018-03658
(22) 19/01/2017
(86) PCT/CN2017/071735 19/01/2017
(87) WO 2017/125046 27/07/2017
(30) 15/003,184 21/01/2016 US
(45) 27/03/2023 420
(43) 26/11/2018 368A
(73) HUAWEI TECHNOLOGIES CO., LTD. (CN)
Huawei Administration Building, Bantian,Longgang District Shenzhen, Guangdong 518129, P.R. China
(72) GE, Yiqun (CA); SHI, Wuxian (CA).
(74) Công ty TNHH một thành viên Sở hữu trí tuệ VCCI (VCCI-IP CO.,LTD)

(54) PHƯƠNG PHÁP TRUYỀN THÔNG ĐƯỢC CHO PHÉP SỬA LỖI, BỘ TRUYỀN VÀ BỘ THU

(57) Sáng chế đề cập đến phương pháp truyền thông được cho phép sửa lỗi, bộ truyền và bộ thu. Phương pháp mã hóa kết hợp các chuỗi các bit thông tin thứ nhất và thứ hai và các bít mã dư tuần hoàn (CRC) và các bít cố định vào trong vectơ đầu vào. Vectơ đầu vào được nhân với ma trận sinh cho mã cực để tạo ra từ mã ghép. Phương pháp giải mã thu từ mã như vậy và tạo ra vectơ được giải mã nhờ tạo ra các mức kế tiếp của cây quyết định. Đối với số lượng của các mức của cây quyết định thứ nhất, các đường vượt quá số lượng tối đa thứ nhất của các đường có thể nhất được loại bỏ. Đối với số lượng của các mức của cây quyết định thứ hai, các đường vượt quá số lượng tối đa thứ hai của các đường có thể nhất được loại bỏ. Trong một số trường hợp, phương pháp giải mã có thể được cải thiện hiệu suất khi so với một số phương pháp giải mã cho các từ mã không ghép.



Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến các kỹ thuật mã hóa cực và các bộ mã hóa và giải mã cho các mã cực, chẳng hạn như cho các ứng dụng truyền thông không dây.

Tình trạng kỹ thuật của sáng chế

Các mã cực dựa trên các ma trận tích số Kronecker. $G = F^{\otimes m} = F \otimes \cdots \otimes F$ là tích số Kronecker gấp m của ma trận hạt nhân F.

Bản chất kỹ thuật của sáng chế

Theo một khía cạnh, phương pháp được đề cập bao gồm bước xử lý K_1 bit thông tin để tạo ra mã phát hiện lỗi (EDC – Error-detecting code) u_1-bit, xử lý K_2 bit thông tin để tạo ra EDC u_2-bit, và bước tạo ra vectơ đầu vào bao gồm chuỗi các bit đầu vào thứ nhất, chuỗi các bit đầu vào thứ hai, và các bit cố định cho mã cực. Chuỗi các bit đầu vào thứ nhất bao gồm K_1 bit thông tin và EDC u_1-bit, chuỗi các bit đầu vào thứ hai bao gồm K_2 bit thông tin và EDC u_2-bit, và chuỗi các bit đầu vào thứ nhất xuất hiện trong vectơ đầu vào trước chuỗi các bit đầu vào thứ hai. Phương pháp này bao gồm bước nhân vectơ đầu vào với ma trận sinh cho mã cực để tạo ra từ mã thứ nhất, và bước truyền hoặc lưu trữ từ mã thứ nhất.

Một cách tùy chọn, ma trận sinh cho mã cực là ma trận tích số Kronecker gấp m $G_2^{\otimes m}$, mà tại đó $G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ hoặc $G_2 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$.

Một cách tùy chọn, EDC u_1-bit và EDC u_2-bit là các mã dư tuần hoàn (CRC – cyclic redundancy codes).

Một cách tùy chọn, phương pháp này còn bao gồm bước xử lý chuỗi K bit thông tin để tạo ra EDC u_3-bit. Vectơ đầu vào đã được tạo ra còn bao gồm chuỗi các bit đầu vào thứ ba, và chuỗi các bit đầu vào thứ ba bao gồm chuỗi K

bit thông tin và EDC u_3-bit. Chuỗi các bit đầu vào thứ hai xuất hiện trong vectơ đầu vào trước chuỗi các bit đầu vào thứ ba.

Một cách tùy chọn, K_1 bit thông tin là các bit thông tin của bản tin thứ nhất, và K_2 bit thông tin là các bit thông tin dẫn đầu của bản tin thứ hai.

Một cách tùy chọn, phương pháp này còn bao gồm bước mã hóa bản tin thứ hai với bộ mã hóa cực để tạo ra từ mã thứ hai.

Một cách tùy chọn, bản tin thứ hai chứa K_3 bit thông tin sau K_2 bit thông tin, và EDC u_3-bit được tạo ra nhờ kết hợp K_2 bit thông tin và K_3.

Một cách tùy chọn, phương pháp này còn bao gồm bước truyền từ mã thứ nhất trước khi truyền từ mã thứ hai.

Một cách tùy chọn, phương pháp này còn bao gồm bước truyền từ mã thứ nhất trong khoảng thời gian xấp xỉ truyền từ mã thứ hai.

Một cách tùy chọn, phương pháp này còn bao gồm bước truyền, tại mức công suất thứ nhất, từ mã thứ nhất; và bước truyền, tại mức công suất thứ hai, từ mã thứ hai. Mức công suất thứ nhất cao hơn mức công suất thứ hai.

Một cách tùy chọn, phương pháp được thực hiện tại trạm gốc thứ nhất truyền thông với thiết bị người dùng (UE). Phương pháp này còn bao gồm bước thu K_2 bit thông tin từ trạm gốc thứ hai truyền thông với thiết bị UE, và truyền từ mã thứ nhất bằng trạm gốc thứ nhất đến thiết bị UE.

Một cách tùy chọn, trạm gốc thứ nhất thu K_2 bit thông tin qua kết nối đường trục (backhaul) giữa trạm gốc thứ nhất và trạm gốc thứ hai.

Theo khía cạnh khác, phương pháp được đề xuất bao gồm bước thu từ thứ nhất, mà tại đó từ thu được thứ nhất dựa trên từ mã thứ nhất. Từ mã thứ nhất có các bit được tạo ra nhờ nhân vectơ đầu vào với ma trận sinh mã cực, vectơ đầu vào có chuỗi các bit đầu vào thứ nhất, chuỗi các bit đầu vào thứ hai, và các bit cố định cho mã cực. Chuỗi các bit đầu vào thứ nhất có K_1 bit thông tin và mã phát hiện lỗi (EDC) u_1-bit, chuỗi các bit đầu vào thứ hai có K_2 bit thông tin và EDC u_2-bit, và chuỗi các bit đầu vào thứ nhất xuất hiện trong vectơ đầu vào trước chuỗi các bit đầu vào thứ hai. Phương pháp sau đó bao gồm bước giải mã từ thu được thứ nhất.

Một cách tùy chọn, ma trận sinh mã cực là ma trận tích số Kronecker gấp m $G_2^{\otimes m}$, mà tại đó $G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ hoặc $G_2 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$.

Một cách tùy chọn, EDC u_1-bit và EDC u_2-bit là các mã dư tuần hoàn (CRC).

Một cách tùy chọn, bước giải mã từ thu được thứ nhất bao gồm việc xử lý từ thu được thứ nhất để tạo ra vectơ được giải mã thứ nhất. Vectơ được giải mã thứ nhất được tạo ra nhờ tạo ra các mức kế tiếp của cây quyết định nhị phân, mỗi mức tương ứng với quyết định về bit tương ứng, mà tại đó mỗi đường trong cây quyết định thể hiện các bit không cố định có thể được giải mã một phần và có khả năng tương ứng. Đối với $K_1 + u_1$ mức thứ nhất của cây quyết định sau nút gốc, khi số lượng các đường trong cây quyết định phát triển vượt ngưỡng L_1, gần như các đường L_1 có thể nhất được loại bỏ. Tại mức $K_1 + u_1$ của cây quyết định, EDC u_1-bit được thể hiện trong mỗi đường còn tồn tại tương ứng được sử dụng để xác định đường trong số các đường còn tồn tại thể hiện chuỗi K_1 bit được giải mã thứ nhất cho vectơ được giải mã thứ nhất, và tất cả các đường khác được loại bỏ. Đối với $K_2 + u_2$ mức thứ hai của cây quyết định, khi số lượng các đường trong cây quyết định phát triển vượt ngưỡng L_2, gần như các đường L_2 có thể nhất được loại bỏ. Tại mức $K_1 + u_1 + K_2 + u_2$ của cây quyết định, EDC u_2-bit được thể hiện trong mỗi đường còn tồn tại tương ứng được sử dụng để xác định đường trong số các đường còn tồn tại thể hiện chuỗi K_2 bit được giải mã thứ hai, và tất cả các đường khác được loại bỏ.

Một cách tùy chọn, phương pháp này còn bao gồm chuỗi K_2 bit được giải mã thứ hai trong vectơ được giải mã thứ nhất.

Một cách tùy chọn, vectơ đầu vào còn bao gồm chuỗi các bit đầu vào thứ ba chứa K_3 bit thông tin và EDC u_3-bit, và chuỗi các bit đầu vào thứ hai xuất hiện trong vectơ đầu vào trước chuỗi các bit đầu vào thứ ba. Phương pháp này còn bao gồm bước xử lý từ thu được thứ nhất để tạo ra vectơ được giải mã thứ nhất. Bước này bao gồm, đối với $K_3 + u_3$ mức thứ ba của cây quyết định, khi số lượng các đường trong cây quyết định phát triển vượt ngưỡng L_3, loại bỏ gần

như các đường L_3 có thể nhất. Tại mức $K_1 + u_1 + K_2 + u_2 + K_3 + u_3$ của cây quyết định, EDC u_3-bit được thể hiện trong mỗi đường còn tồn tại tương ứng được dùng để xác định đường trong số các đường còn tồn tại thể hiện chuỗi các bit được giải mã thứ ba K_3. Chuỗi các bit được giải mã thứ ba K_3 được bao gồm trong vectơ được giải mã thứ nhất.

Một cách tùy chọn, phương pháp này còn bao gồm bước thu từ thứ hai dựa trên từ mã thứ hai. Từ mã thứ hai bao gồm các bit được tạo ra bằng cách nhân vectơ đầu vào thứ hai với ma trận sinh mã cực. Vectơ đầu vào thứ hai bao gồm chuỗi các bit đầu vào thứ ba và các bit cố định cho mã cực. Chuỗi các bit đầu vào thứ ba bao gồm K_2 bit thông tin được theo sau bởi K_3 bit thông tin và EDC u_3-bit. Phương pháp này còn bao gồm bước dùng EDC u_2-bit, xác định xem K_2 bit thông tin có được giải mã thành công hay không. Nếu K_2 bit thông tin được giải mã thành công, từ thu được thứ hai được xử lý để tạo ra vectơ được giải mã thứ hai. Việc xử lý bao gồm sử dụng K_2 bit thông tin làm các bit khởi đầu K_2 cho vectơ được giải mã thứ hai. Các mức kế tiếp của cây quyết định nhị phân thứ hai được tạo ra, mỗi mức tương ứng với quyết định về bit tương ứng, mà tại đó mỗi đường trong cây quyết định thứ hai thể hiện phần chuỗi các bit không cố định có thể được giải mã và có khả năng tương ứng. K_2 bit thông tin được coi là các bit cố định. Đối với $K_3 + u_3$ mức của cây quyết định sau nút gốc, khi số lượng các đường trong cây quyết định phát triển vượt ngưỡng L_3, gần như các đường L_3 có thể nhất được loại bỏ. Tại mức $K_3 + u_3$ của cây quyết định, EDC u_3-bit được thể hiện trong mỗi đường còn tồn tại tương ứng được dùng để xác định đường trong số các đường còn tồn tại thể hiện chuỗi các bit được giải mã thứ ba K_3. Chuỗi các bit được giải mã thứ ba K_3 được bao gồm trong vectơ được giải mã thứ hai.

Một cách tùy chọn, ít nhất một trong các đường L_1, L_2, hoặc L_3 phụ thuộc vào mức công suất tại đó bản tin thứ nhất được truyền hoặc mức công suất tại đó bản tin thứ hai được truyền.

Một cách tùy chọn, bước giải mã từ thu được thứ nhất bao gồm việc xử lý từ thu được thứ nhất để tạo ra vectơ được giải mã. Vectơ được giải mã được

tạo ra nhờ tạo ra, ít nhất một phần song song, các mức kế tiếp của cây quyết định nhị phân thứ nhất và các mức kế tiếp của cây quyết định nhị phân thứ hai, mà tại đó mỗi đường trong mỗi trong số các cây quyết định thứ nhất và thứ hai thể hiện chuỗi các bit không cố định có thể được giải mã một phần và có khả năng tương ứng. Đối với $K_1 + u_1$ mức thứ nhất của cây quyết định thứ nhất sau nút gốc, khi số lượng các đường trong cây quyết định thứ nhất phát triển vượt ngưỡng L_1, gần như các đường L_1 có thể nhất được loại bỏ. Tại mức $K_1 + u_1$ của cây quyết định thứ nhất, EDC u_1-bit được thể hiện trong mỗi đường còn tồn tại tương ứng trong cây quyết định thứ nhất được dùng để xác định đường trong số các đường còn tồn tại thể hiện chuỗi K_1 bit được giải mã thứ nhất cho vectơ được giải mã. Đối với các mức thứ nhất R_1 của cây quyết định thứ hai sau nút gốc, khi số lượng các đường trong cây quyết định thứ hai phát triển vượt ngưỡng L_2, gần như các đường L_2 có thể nhất được loại bỏ. Đối với các mức R_2 tiếp theo của cây quyết định thứ hai, khi số lượng các đường trong cây quyết định phát triển vượt ngưỡng L_3, gần như các đường L_3 có thể nhất được loại bỏ. Tại mức $K_1 + u_1 + K_2 + u_2$ của cây quyết định thứ hai, EDC u_2-bit được thể hiện trong mỗi đường còn tồn tại tương ứng của cây quyết định thứ hai được dùng để xác định đường trong số các đường còn tồn tại thể hiện chuỗi K_2 bit thứ hai sau chuỗi K_1 bit thứ nhất cho vectơ được giải mã. R_1 bé hơn $K_1 + u_1$ và L_1 lớn hơn L_2.

Một cách tùy chọn, tại mức $K_1 + u_1$ của cây quyết định thứ hai, EDC u_1-bit được thể hiện trong mỗi đường còn tồn tại tương ứng của cây quyết định thứ hai được sử dụng để xác định xem một trong số các đường còn tồn tại của cây quyết định thứ hai có thể hiện việc giải mã chính xác chuỗi K_1 bit được giải mã thứ nhất cho vectơ được giải mã hay không. Nếu cây quyết định thứ nhất vẫn đang được tạo ra và nếu một trong số các đường còn tồn tại của cây quyết định thứ hai thể hiện việc giải mã chính xác của chuỗi K_1 bit được giải mã thứ nhất cho vectơ được giải mã, việc tạo ra cây quyết định thứ nhất được kết thúc.

Theo khía cạnh khác, thiết bị được đề xuất bao gồmbộ xử lý được tạo cấu hình để tạo ra từ mã thứ nhất bằng cách xử lý K_1 bit thông tin để tạo ra mã phát hiện lỗi (EDC) u_1-bit, xử lý K_2 bit thông tin để tạo ra EDC u_2-bit, tạo ra

vectơ đầu vào bao gồm chuỗi các bit đầu vào thứ nhất, chuỗi các bit đầu vào thứ hai, và các bit cố định cho mã cực. Chuỗi các bit đầu vào thứ nhất bao gồm K_1 bit thông tin và EDC u_1-bit, chuỗi các bit đầu vào thứ hai bao gồm K_2 bit thông tin và EDC u_2-bit, và chuỗi các bit đầu vào thứ nhất xuất hiện trong vectơ đầu vào trước chuỗi các bit đầu vào thứ hai. Vectơ đầu vào được nhân với ma trận sinh cho mã cực để tạo ra từ mã thứ nhất. Thiết bị còn bao gồm thiết bị truyền để truyền từ mã thứ nhất.

Theo khía cạnh khác nữa, thiết bị được đề xuất bao gồm bộ xử lý được tạo cấu hình để thực hiện phương pháp được mô tả ở trên hoặc dưới đây để tạo ra từ mã thứ nhất, và thiết bị truyền để truyền từ mã thứ nhất.

Theo khía cạnh khác nữa, thiết bị được đề xuất bao gồm thiết bị thu để thu từ thứ nhất và bộ xử lý được tạo cấu hình để giải mã từ thu được thứ nhất. Từ thu được thứ nhất dựa trên từ mã thứ nhất, mà tại đó từ mã thứ nhất bao gồm các bit được tạo ra nhờ nhân vectơ đầu vào với ma trận sinh mã cực. Vectơ đầu vào bao gồm chuỗi các bit đầu vào thứ nhất, chuỗi các bit đầu vào thứ hai, và các bit cố định cho mã cực. Chuỗi các bit đầu vào thứ nhất bao gồm K_1 bit thông tin và mã phát hiện lỗi (EDC) u_1-bit, chuỗi các bit đầu vào thứ hai bao gồm K_2 bit thông tin và EDC u_2-bit. Chuỗi các bit đầu vào thứ nhất xuất hiện trong vectơ đầu vào trước chuỗi các bit đầu vào thứ hai.

Theo khía cạnh khác nữa, thiết bị được đề xuất bao gồm thiết bị thu để thu từ thu được thứ nhất dựa trên từ mã thứ nhất, và bộ xử lý được tạo cấu hình để thực hiện phương pháp được mô tả ở trên hoặc dưới đây để giải mã từ thu được thứ nhất.

Mô tả vắn tắt các hình vẽ

Các phương án của sáng chế sẽ được mô tả chi tiết dựa vào các hình vẽ kèm theo, trong đó:

Fig.1 là sơ đồ thể hiện cách ma trận tích số Kronecker có thể được tạo ra từ ma trận hạt nhân;

Fig.2 là sơ đồ thể hiện ví dụ sử dụng ma trận sinh mã cực để tạo ra các từ mã và sơ đồ minh họa của bộ mã hóa cực ví dụ;

Fig.3 là lưu đồ thể hiện phương pháp mã hóa thông tin sử dụng mã cực;

Fig.4 là sơ đồ minh họa của bộ mã hóa cực ví dụ;

Fig.5 là sơ đồ thể hiện một phần của cây quyết định ví dụ dùng trong bộ giải mã cực kiểu danh sách;

Fig.6 là hình minh họa của hai chuỗi bit đầu vào đang được mã hóa độc lập thành hai từ mã cho việc truyền;

Fig.7 là hình minh họa của hai chuỗi bit đầu vào đang được ghép và mã hóa thành từ mã kết hợp cho việc truyền;

Fig.8 là lưu đồ của phương pháp mã hóa cực để tạo ra từ mã kết hợp như được thể hiện trên Fig.7;

Fig.9 là đồ thị thể hiện các giá trị tính toán của độ tin cậy của một số kênh tổng hợp của mã cực để truyền từ mã ví dụ được mã hóa theo phương pháp của Fig.8;

Fig.10 là hình minh họa của ba chuỗi bit đầu vào đang được ghép và mã hóa thành từ mã kết hợp cho việc truyền;

Fig.11 là sơ đồ thể hiện cây quyết định ví dụ để giải mã từ mã được tạo ra bằng mã hóa ghép cực;

Fig.12A là lưu đồ của phương pháp giải mã cực;

Fig.12B là lưu đồ của phương pháp giải mã cực mà bao gồm bước tạo ra cây quyết định như được thể hiện trên Fig.11;

Fig.13 là lưu đồ của phương pháp giải mã cực mà tạo ra các cây quyết định song song;

Fig.14A là hình minh họa của bốn chuỗi bit đầu vào để mã hóa thành từ mã kết hợp;

Fig.14B là hình minh họa việc giải mã tuần tự từ mã mà đã được mã hóa dựa trên chuỗi các bit đầu vào của Fig.14A;

Fig.14C là hình minh họa của việc giải mã song song của từ mã mà đã được mã hóa dựa trên chuỗi các bit đầu vào của Fig.14A;

Fig.14D là hình minh họa của việc giải mã song song, với việc kết thúc sớm, của từ mã mà đã được mã hóa dựa trên chuỗi các bit đầu vào của Fig.14A;

Fig.15 là hình minh họa của hai chuỗi các bit đầu vào đang được mã hóa thành hai từ mã liên tiếp cho việc truyền, chuỗi thứ nhất bao gồm tập hợp con của các bit đầu vào được sao chép từ chuỗi thứ hai;

Fig.16 là sơ đồ minh họa hai trạm gốc truyền thông với thiết bị người dùng (UE);

Fig.17 là lưu đồ của phương pháp giải mã cực của bản tin thứ hai dựa trên từ mã thứ hai được mã hóa với phương pháp được minh họa trên Fig.14;

Fig.18A là sơ đồ minh họa thiết bị để mã hóa và truyền từ mã; và

Fig.18B là sơ đồ minh họa thiết bị để thu và giải mã từ mã.

Mô tả chi tiết sáng chế

Fig.1 thể hiện cách ma trận tích số Kronecker có thể được tạo ra từ ma trận hạt nhân G_2 100, mà tại đó $G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. Được thể hiện trên Fig.1 là ma trận tích số Kronecker gấp 2 $G_2^{\otimes 2}$ 102 và ma trận tích số Kronecker gấp 3 $G_2^{\otimes 3}$ 104. Cách tiếp cận này có thể được tiếp tục để tạo ra ma trận tích số Kronecker gấp m $G_2^{\otimes m}$.

Mã cực có thể được tạo ra từ ma trận tích số Kronecker dựa trên ma trận G_2. Đối với mã cực có các từ mã có độ dài $N = 2^m$, ma trận sinh cho mã cực là $G_2^{\otimes m}$. Ví dụ dùng ma trận tích số Kronecker $G_2^{\otimes 3}$ để tạo ra các từ mã có độ dài 8 được mô tả trên Fig.2. Từ mã x được tạo ra bằng tích số của vectơ đầu vào u làm vectơ hàng và ma trận tích số Kronecker $G_2^{\otimes 3}$ 204 như được chỉ báo tại 200. (Ngoài ra, từ mã x có thể được tạo ra bằng tích số của ma trận tích số Kronecker $G_2^{T\otimes 3}$ và vectơ đầu vào u làm vectơ cột.) Vectơ đầu vào u là tập hợp của các bit cố định và các bit thông tin. Ví dụ cụ thể, N=8, vì thế vectơ đầu vào u là vectơ 8 bit, và từ mã x là vectơ 8-bit. Vectơ đầu vào có các bit cố định tại các vị trí 0, 1, 2, và 4, và có các bit thông tin tại các vị trí 3, 5, 6, và 7. Việc thực hiện ví dụ của bộ mã hóa mà tạo ra các từ mã được chỉ báo tại 212, mà tại đó các bit cố định đều được thiết đặt thành 0. Trong bộ mã hóa 212 ví dụ, biểu tượng

cộng vòng tròn thể hiện môđun 2 thêm vào. Chẳng hạn như trên Fig.2, N=8 bit vectơ đầu vào được tạo ra từ các bit thông tin K=4 và (N − K) = 4 bit cố định. Các mã của dạng này được gọi là các mã cực và bộ mã hóa được gọi là bộ mã hóa cực. Các bộ giải mã để giải mã các mã cực được gọi là các bộ giải mã cực.

Trong bài báo "Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels" của E. Arıkan, tạp chí khoa học IEEE Transactions on Information Theory, tập 55, số 7 (tháng 7 năm 2009) [Arıkan], lý thuyết đề cập đến "phân cực kênh" của các mã cực đã được chứng minh trong phần IV. Phân cực kênh là hoạt động mà nó tạo ra N kênh "tổng hợp" từ N bản sao độc lập của kênh bộ nhớ rời rạc đầu vào nhị phân (B-DMC) sao cho, với các giá trị gia tăng của N, các kênh tổng hợp mới được phân cực nghĩa là thông tin chung của chúng là hoặc gần bằng 0 (các kênh nhiễu hoàn toàn) hoặc gần bằng 1 (các kênh không nhiễu hoàn toàn). Nói cách khác, một số vị trí bit của vectơ đầu vào được cung cấp cho bộ mã hóa sẽ trải qua kênh nhiễu hoàn toàn, nghĩa là, có độ tin cậy tương đối thấp/khả năng thấp để được giải mã chính xác khi được xem xét độc lập với các kênh tổng hợp khác. Một số vị trí bit của vectơ đầu vào được cung cấp cho bộ mã hóa sẽ trải qua kênh rất sạch, nghĩa là, có độ tin cậy cao/khả năng cao để được giải mã chính xác khi được xem xét độc lập với các kênh tổng hợp khác. Trong một số trường hợp, độ tin cậy của kênh tổng hợp khi được xem xét độc lập với các kênh tổng hợp khác có thể được gọi là "công suất" của kênh tổng hợp. Ví dụ cụ thể của mã cực được mô tả ở trên trong đó mã dựa trên tích số Kronecker gấp m của ma trận cụ thể G_2. Việc dùng ma trận sinh này dẫn đến phân cực kênh. Tổng quát hơn, ma trận sinh bất kỳ mà tạo ra hiệu ứng phân cực kênh sẽ được gọi ở đây là ma trận sinh mã cực.

Trong cấu trúc mã cực, mọi nỗ lực được thực hiện để đặt các bit thông tin ở các vị trí "tin cậy" hơn của vectơ đầu vào, và để các bit cố định (ví dụ, các bit đã được biết bởi cả hai bộ mã hóa và bộ giải mã) ở các vị trí "không tin cậy" hơn của vectơ đầu vào. Tuy nhiên, khi thông tin được truyền qua kênh vật lý, độ tin cậy của các vị trí bit ban đầu cũng là chức năng của các đặc tính của kênh vật

lý, chẳng hạn như tỷ số tín hiệu trên nhiễu (SNR) và tỷ số lỗi bit của kênh vật lý. Trong hầu hết các ứng dụng, các bit cố định có thể được thiết đặt giá trị bất kỳ miễn là chuỗi các bit cố định được biết bởi cả hai bộ mã hóa và bộ giải mã. Trong các ứng dụng thông thường, các bit cố định đều được thiết đặt thành 0.

Các bit mã phát hiện lỗi (EDC) có thể được bao gồm trong vectơ đầu vào để hỗ trợ giải mã. Theo phương án ví dụ, mã kiểm tra dự phòng tuần hoàn (CRC) được dùng làm mã EDC. Tuy nhiên, nên được hiểu rằng các mã EDC khác có thể cũng được dùng trong một số phương án, chẳng hạn như kiểm tra tổng Fletcher, và các mã sửa lỗi đó có thể được dùng làm các EDC. Để mô tả đơn giản, các phương án được mô tả trong đặc điểm kỹ thuật này sẽ được mô tả dùng mã CRC làm mã EDC.

Các bit CRC được tạo ra dựa trên các bit thông tin đang được truyền. Các bit CRC thường được đặt tại các vị trí đáng tin cậy hơn trong vectơ đầu vào, mặc dù các bit CRC có thể còn được đặt tại các vị trí khác trong vectơ đầu vào. Các bit CRC có thể được thêm vào để cải thiện hiệu suất mã cực cho các độ dài từ mã từ ngắn đến trung bình. Đối với bộ mã hóa cực từ mã độ dài rất dài, các bit CRC có thể không được yêu cầu để cải thiện hơn nữa độ tin cậy mã cực, nhưng được bao gồm để hỗ trợ giải mã. Trong khi mã hóa, vectơ đầu vào N-bit được tạo ra từ các bit thông tin K, CRC u-bit, và các bit cố định (N − K − u). Ví dụ được mô tả trên Fig.3. Bắt đầu với các bit thông tin K 300, CRC u-bit được thêm vào tại 302 để tạo ra, tại 304, tập hợp của các bit đầu vào bao gồm các bit thông tin K và CRC u-bit. Tại 306, các bit cố định (N − K − u) được chèn vào để tạo ra vectơ đầu vào N-bit, tại 308, với các bit thông tin K, CRC u-bit, và các bit cố định (N − K − u), mà tại đó N là công suất của 2. Vectơ đầu vào 308 sau đó được nhân với ma trận tích số Kronecker bao gồm ma trận sinh cho mã cực tại 310 để tạo ra từ mã N-bit 312.

Ví dụ việc thực hiện của bộ mã hóa cực bao gồm các bit CRC được mô tả theo sơ đồ trên Fig.4, mà tại đó biểu tượng cộng vòng tròn thể hiện môđun 2 thêm vào. Bộ mã hóa tạo ra từ mã 404 từ vectơ đầu vào 402. Trong ví dụ được thể hiện, N=16, K=8, u=2, và có sáu bit cố định cho tỷ lệ mã là 0,5. Các bit cố

định, mà đã được minh họa là đã được thiết đặt thành 0, được chèn vào tại các vị trí 0, 1, 2, 4, 9, và 12 của vectơ đầu vào 402. Các bit thông tin, được chỉ báo là Info[0] qua Info[7], được cung cấp tại các vị trí 3, 5, 6, 7, 8, 10, 11, và 13, một cách tương ứng, của vectơ đầu vào 402. Hai bit CRC, được chỉ báo là CRC[0] và CRC[1], lần lượt được cung cấp tại các vị trí 14 và 15 của vectơ đầu vào 402.

Vectơ đầu vào được dùng bởi bộ mã hóa để tạo ra từ mã đôi khi gọi là bản tin. Từ mã có thể được truyền qua kênh, và bộ thu có thể, theo lượt, thu từ thu được. Do các hiệu ứng kênh chẳng hạn như từ mã được truyền bị nhiễu, từ thu được có thể không giống với từ mã được truyền. Bộ giải mã cố gắng để giải mã từ thu được để xác định các bit thông tin trong bản tin được truyền ban đầu.

Trong khi giải mã từ mã được mã hóa từ vectơ đầu vào, các vị trí và giá trị của các bit cố định trong vectơ đầu vào được coi là đã biết. Để mô tả đơn giản, các bit của vectơ đầu vào mà không được biết bởi bộ giải mã từ trước sẽ được gọi là các bit "không xác định". Chẳng hạn như, các bit thông tin và các bit CRC là các bit không xác định. Đặc tính của một số bộ giải mã cực đó là các bit không xác định được giải mã tuần tự, chẳng hạn như thuật toán giải mã cực có thể dựa trên triệt tiêu liên tiếp. Một khi quyết định cụ thể đã được thực hiện về cách mà bit không xác định được giải mã như thế nào, bit đó không, trong đó các bộ giải mã cực, có cơ hội được thay đổi hoặc được hiệu chỉnh, và bộ giải mã chuyển sang giải mã bit không xác định tiếp theo. Nói cách khác, không có "quay trở lại". Bit mà đã được thiết đặt tại bước i không thể được thay đổi tại bước $j > i$. Ngoài ra, sự nhận biết của giá trị của các bit cố định tiếp theo không được tính đến, nghĩa là, các bit cố định tiếp theo, mặc dù được biết bởi bộ giải mã, sẽ không giúp giải mã bit không xác định hiện tại.

Theo Arıkan, thuật toán triệt tiêu liên tiếp được mô tả để giải mã các mã cực. Loại khác của thuật toán giải mã cực với hiệu quả không gian lớn hơn và độ phức tạp thời gian thấp hơn, được gọi là bộ giải mã danh sách, được mô tả trong "List Decoding of Polar Codes" bởi Tal và Vardy, Proceedings of the 2011 IEEE International Symposium on Information Theory, trang 1–5 (tháng 7, 2011). Trong bộ giải mã danh sách, các mức kế tiếp của cây quyết định nhị phân

được tạo ra, mỗi mức tương ứng với quyết định về bit không xác định tương ứng. Mỗi đường trong cây quyết định từ nút gốc đến các nút lá thể hiện chuỗi các bit không xác định có thể được giải mã một phần và có khả năng tương ứng. Cây quyết định được tạo ra theo cách theo chiều rộng thứ nhất. Trong quá trình tạo ra cây quyết định, tại mỗi mức của cây quyết định mà tại đó số lượng các đường phát triển vượt ngưỡng thiết đặt L, các đường L có khả năng cao nhất được nhận dạng, và các đường còn lại được loại bỏ. Nếu từ mã bao gồm các bit CRC được mã hóa cho các bit thông tin trước, khi cây quyết định được tạo ra, mỗi trong số các đường tồn tại tương ứng với các bit thông tin được giải mã được kiểm tra đối với các bit CRC được thể hiện trong mỗi trong số các đường tồn tại. Bộ giải mã sau đó xuất ra dưới dạng vectơ được giải mã các bit thông tin trong đường tồn tại mà vượt qua kiểm tra CRC. Nếu nhiều hơn hai đường vượt qua kiểm tra CRC, bộ giải mã lựa chọn để đưa ra đường mà vượt qua kiểm tra CRC và có khả năng cao nhất, đó là, các đường tồn tại mà vượt qua kiểm tra CRC với khả năng cao nhất theo số liệu. Nếu không đường nào vượt qua kiểm tra CRC, hoặc nếu từ mã không bao gồm các bit CRC được mã hóa, bộ giải mã lựa chọn đường đầu ra mà có khả năng cao nhất đó là, các đường tồn tại với khả năng cao nhất theo số liệu.

Fig.5 là sơ đồ thể hiện một phần của cây quyết định ví dụ được dùng trong bộ giải mã cực kiểu danh sách, mà tại đó giá trị của L được thiết đặt thành 4. Năm mức 502, 504, 506, 508, 510 của cây quyết định được minh họa, mà nó được gọi lần lượt là các mức 0, 1, 2, 3, và 4. Mặc dù năm mức được minh họa, nên được hiểu rằng cây quyết định để giải mã M bit không xác định sẽ có M + 1 mức. Các nút con của nút gốc 520 thể hiện các lựa chọn có thể cho bit không xác định thứ nhất, và các nút con tiếp theo thể hiện các lựa chọn có thể cho các bit tiếp theo. Nút lá 530a, chẳng hạn như, thể hiện phần chuỗi bit không xác định có thể được giải mã sau đây: 0, 1, 0, 0. Ở mức 3 (508), số lượng các đường có thể là lớn hơn L, vì thế các đường L có khả năng cao nhất được xác định, và các đường còn lại được loại bỏ. Tại mức 4 (510), số lượng các đường có thể lại lớn hơn L, vì thế các đường L có khả năng cao nhất được xác định, và các đường

còn lại sẽ lại được loại bỏ. Trong ví dụ được thể hiện, các đường kết thúc trong các nút lá 530a, 530b, 530c, và 530d thể hiện các đường khả năng cao nhất. Mỗi trong số các đường khả năng cao nhất này là được vẽ với các nét đậm. Các đường kết thúc trong các nút lá 540a, 540b, 540c, 540d là các đường khả năng thấp hơn mà nó sẽ được loại bỏ.

Trong một số ứng dụng truyền thông, các từ mã được truyền qua kênh theo các đơn vị gọi là các khối. Các khối có thể được truyền tuần tự hoặc song song. Một số kích thước khối ví dụ là 1024 bit (1K), 2048 bit (2K), 4096 bit (4K), và 8192 bit (8K), mặc dù các kích thước khối khác là có thể. Trong một số ứng dụng truyền thông, yêu cầu các kích thước khối không vượt quá các kích thước nhất định để xử lý các vấn đề độ trễ.

Fig.6 là hình minh họa của hai chuỗi các bit đầu vào 650, 652 được mã hóa độc lập thành hai từ mã 622, 624 cho việc truyền. Chuỗi các bit đầu vào thứ nhất 650 bao gồm K_1 bit thông tin 602 và CRC u_1-bit 612 được tính toán dựa trên K_1 bit thông tin 602. Chuỗi các bit đầu vào thứ hai 652 bao gồm K_2 bit thông tin 604 và CRC u_2-bit 614 được tính toán dựa trên K_2 bit thông tin 604.

Chuỗi các bit đầu vào thứ nhất 650 được xử lý bởi quy trình mã hóa cực để tạo ra từ mã thứ nhất 622 có độ dài N. Chuỗi các bit đầu vào thứ hai 652 được xử lý bởi quy trình mã hóa cực để tạo ra từ mã thứ hai 624 có độ dài N. Quy trình mã hóa cực cho chuỗi các bit đầu vào thứ nhất 650 bao gồm bước 640 chèn các bit cố định vào trong chuỗi các bit đầu vào thứ nhất 650 để tạo ra vectơ đầu vào thứ nhất 690 có độ dài N, và sau đó nhân 660 vectơ đầu vào thứ nhất 690 với ma trận sinh mã cực, như được mô tả ở trên với liên quan đến Fig.3, các bước 306 và 310. Quy trình mã hóa cực cho chuỗi các bit đầu vào thứ hai 652 bao gồm việc áp dụng chuỗi các hoạt động tương tự liên quan đến chuỗi các bit đầu vào thứ hai 652. Đó là, bước 642 chèn các bit cố định vào trong chuỗi các bit đầu vào thứ hai 652 được thực hiện để tạo ra vectơ đầu vào thứ hai 692 có độ dài N, và sau đó nhân 662 vectơ đầu vào thứ hai 692 với ma trận sinh mã cực. Các bit cố định cần phải không được chèn vào trong các vị trí tương tự trong vectơ đầu vào thứ nhất và vectơ đầu vào thứ hai. Chuỗi các bit đầu vào thứ nhất

650 và chuỗi các bit đầu vào thứ hai 652 được mã hóa độc lập với nhau, đó là, các quy trình mã hóa cực cho mỗi chuỗi các bit đầu vào không phụ thuộc vào nhau.

Fig.7 là hình minh họa của hai chuỗi các bit đầu vào 750, 752 đang được mã hóa thành từ mã kết hợp 720 cho việc truyền. Chuỗi các bit đầu vào thứ nhất 750 bao gồm K_1 bit thông tin 702 và CRC u_1-bit 712 được tính toán dựa trên K_1 bit thông tin 702. Chuỗi các bit đầu vào thứ hai 752 bao gồm K_2 bit thông tin 704 và CRC u_2-bit 714 được tính toán dựa trên K_2 bit thông tin 704. Mặc dù CRC u_1-bit 712 được thể hiện được thêm vào K_1 bit thông tin 702, và CRC u_2-bit 714 được thể hiện được thêm vào K_2 bit thông tin 704, theo một số phương án, các bit CRC và các bit thông tin trong mỗi trong số các chuỗi 750, 752 có thể được sắp xếp theo thứ tự khác nhau. Chuỗi các bit đầu vào thứ nhất 750 và chuỗi các bit đầu vào thứ hai 752 được ghép để tạo ra chuỗi 754 có kích thước $K_1+u_1+K_2+u_2$.

Việc ghép chuỗi 754 được xử lý bởi quy trình mã hóa cực để tạo ra từ mã 720 có độ dài N. Quy trình mã hóa cực bao gồm bước 740 chèn các bit cố định vào chuỗi ghép 754 để tạo ra vectơ đầu vào 790 có độ dài N, và sau đó nhân 760 vectơ đầu vào với ma trận sinh mã cực, như được mô tả dưới đây liên quan đến Fig.8, bước 808. Nên được hiểu rằng giá trị của N thể hiện độ dài của từ mã 720 được minh họa trên Fig.7 có thể khác với giá trị của N được đề cập trong phần mô tả ở trên của Fig.6. Chẳng hạn như, nếu mỗi trong số các chuỗi các bit đầu vào thứ nhất 650 và chuỗi các bit đầu vào thứ hai 652 trên Fig.6 có độ dài tương đương với chuỗi các bit đầu vào thứ nhất 750 và chuỗi các bit đầu vào thứ hai 752 trên Fig.7, từ mã 720 sẽ dài gấp đôi mỗi trong số các từ mã thứ nhất 622 và từ mã thứ hai 624 trên Fig.6.

Theo một số phương án, K_1 và K_2 là bằng nhau và u_1 và u_2 là bằng nhau. Theo các phương án này, mã hóa cực được đề cập ở đây như là mã hóa ghép cực đối xứng, hoặc ngoài ra như là mã hóa cực kết hợp khối đối xứng. Từ mã 720 được đề cập ở đây như là từ mã ghép đối xứng, hoặc ngoài ra như là từ mã kết hợp khối đối xứng. Mặc dù Fig.7 minh họa hai chuỗi các bit đầu vào 750, 752

đang được ghép và mã hóa, nên được hiểu rằng theo các phương án khác nhiều chuỗi các bit đầu vào hơn có thể được ghép và mã hóa để tạo ra từ mã 720.

Fig.8 là lưu đồ của phương pháp mã hóa cực để tạo ra từ mã như được thể hiện trên Fig.7. Vì mục đích tổng quát, các bước của Fig.8 đã được minh họa là tạo ra các EDC, mặc dù đã được giải thích trước đó, nên được hiểu rằng nhiều phương án sẽ dùng các CRC làm dạng cụ thể của EDC. Tại bước 800, K_1 bit thông tin được xử lý để tạo ra EDC u_1-bit, và tại bước 802, K_2 bit thông tin được xử lý để tạo ra EDC u_2-bit. Theo một số phương án, K_1 bằng K_1.

Phương pháp sau đó thực hiện bước 804, mà tại đó vectơ đầu vào có kích thước N, mà tại đó N là công suất của 2, được tạo ra nhờ kết hợp toàn bộ phần sau đây: chuỗi các bit đầu vào thứ nhất mà bao gồm K_1 bit thông tin và EDC u_1-bit, chuỗi các bit đầu vào thứ hai mà bao gồm K_2 bit thông tin và EDC u_2-bit, và các bit cố định cho mã cực. Vectơ đầu vào được tạo ra để cho chuỗi các bit đầu vào thứ nhất xuất hiện trong vectơ đầu vào trước chuỗi các bit đầu vào thứ hai. Các vị trí của các bit EDC u_1-bit trong chuỗi các bit đầu vào thứ nhất, các vị trí của các bit EDC u_2-bit trong chuỗi các bit đầu vào thứ hai, và các vị trí của các bit cố định là các lựa chọn thiết kế mà phải được biết bởi cả hai bộ mã hóa và bộ giải mã. Theo một số phương án, các vị trí của EDC u_1-bit và EDC u_2-bit được xác định từ trước hoặc được lựa chọn dựa trên một hoặc các tính toán của độ tin cậy truyền của các kênh tổng hợp của mã cực tương ứng với các vị trí bit đơn lẻ của vectơ đầu vào. Các tính toán có thể được thực hiện dựa trên một hoặc các giả định về các đặc tính kênh vật lý dự kiến, chẳng hạn như các SNR được giả định và/hoặc các xác suất xóa được giả định trong mô hình kênh BEC. Như vậy, các vị trí bit được lựa chọn dựa trên các tính toán này có thể không tối ưu cho kênh vật lý thực tế đang được dùng để truyền thông. Theo một số phương án, các EDC được đặt tại các vị trí bit đơn lẻ của vectơ đầu vào mà tương ứng với các kênh tổng hợp đạt tiêu chí độ tin cậy. Chẳng hạn như, theo một số phương án, một hoặc các EDC được đặt tại các vị trí bit đơn lẻ của vectơ đầu vào mà tương ứng với các kênh tổng hợp được tính toán để có ít nhất một mức độ tin cậy cụ thể theo một hoặc các điều kiện kênh vật lý được giả định.

Theo một số phương án, các bit thông tin còn được đặt tại các vị trí bit đơn lẻ của vectơ đầu vào mà đạt tiêu chí độ tin cậy. Chẳng hạn như, các bit thông tin có thể được đặt tại các vị trí bit đơn lẻ của vectơ đầu vào mà tương ứng với các kênh tổng hợp được tính toán để có độ tin cậy cao nhất còn lại theo một hoặc các điều kiện kênh vật lý giả định sau khi các bit EDC đã được đặt. Các bit cố định có thể được đặt tại các vị trí bit đơn lẻ của vectơ đầu vào mà được tính toán để có mức độ tin cậy thấp theo các điều kiện kênh vật lý giả định.

Tại bước 808, vectơ đầu vào được nhân với ma trận sinh mã cực để tạo ra từ mã có độ dài N. Cuối cùng, từ mã hoặc được truyền qua kênh vật lý hoặc lưu trữ tại bước 810.

Fig.9 là đồ thị thể hiện các giá trị tính toán của độ tin cậy của một số kênh tổng hợp của mã cực để truyền ví dụ từ mã được mã hóa theo phương pháp của Fig.8, mà tại đó CRC được dùng làm EDC. Cho ví dụ này, để tính toán độ tin cậy của các kênh tổng hợp, hiệu suất của kênh vật lý trong khi truyền từ mã ví dụ được mô hình hóa là kênh không nhớ xóa nhị phân (BEC) với xác suất xóa là 0,25. (Mô hình kênh BEC như trên có thể được coi như phép tính xấp xỉ của mô hình kênh tạp âm Gausse trắng cộng sinh (AWGN).) Từ mã ví dụ có độ dài $N=4096$, và $K_1+u_1=K_2+u_2=1024$, dẫn đến tỷ lệ mã là 0,5.

Trục x của đồ thị của Fig.9 thể hiện mỗi vị trí bit thông tin riêng lẻ (không cố định) trong vectơ đầu vào được dùng để tạo ra từ mã ví dụ, và trục y của đồ thị thể hiện độ tin cậy truyền được tính toán của kênh tổng hợp của mã cực tương ứng với vị trí bit riêng lẻ đã cho trong vectơ đầu vào. Như được giải thích trước đó, độ tin cậy của kênh tổng hợp có thể còn được gọi là "khả năng" của kênh tổng hợp.

Một điểm được đánh dấu trên đồ thị của Fig.9 tại mỗi vị trí bit của bit không cố định trong vectơ đầu vào. Mỗi trong số các điểm được đánh dấu thể hiện độ tin cậy truyền được tính toán của kênh tổng hợp của mã cực cho vị trí bit tương ứng của kênh tổng hợp trong vectơ đầu vào. Nhóm các điểm được đánh dấu thứ nhất 902 chứa 1024 (ví dụ, K_1+u_1) điểm tương ứng với các vị trí của K_1 bit thông tin và các bit CRC u_1 trong vectơ đầu vào. Nhóm các điểm được đánh

dấu thứ hai 904 chứa 1024 (ví dụ, K_2+u_2) điểm tương ứng với các vị trí của K_2 bit thông tin và các bit CRC u_2 trong vectơ đầu vào qua 4096-bit (ví dụ, $(K_1+u_1)/R_1 + (K_2+u_2)/R_2$, mà tại đó R_1 và R_2 thể hiện tỷ lệ mã bằng 0,5) từ mã

Trong ví dụ được mô tả trong Fig.9, các vị trí bit không cố định trong vectơ đầu vào được lựa chọn nhờ tính toán độ tin cậy của mỗi trong số các vị trí bit trong vectơ đầu vào, và sau đó xác định 2048 vị trí bit đáng tin cậy nhất. 2048 vị trí bit đáng tin cậy nhất sau đó được chia tuần tự thành nhóm các vị trí bit thứ nhất tương ứng với nhóm các điểm được đánh dấu thứ nhất 902 và nhóm các vị trí bit thứ hai tương ứng với nhóm các điểm được đánh dấu thứ hai 904. K_1 bit thông tin và các bit CRC u_1 được đặt tại nhóm các vị trí bit thứ nhất. K_2 bit thông tin và các bit CRC u_2 được đặt tại nhóm các vị trí bit thứ hai. Cụ thể hơn, ví dụ cụ thể được minh họa, các bit CRC u_1 được đặt tại các vị trí bit tương ứng với các điểm được đánh dấu 912, và các bit CRC u_2 được đặt tại các vị trí bit tương ứng với các điểm vẽ 914, bởi vì các vị trí bit này thỏa mãn tiêu chí độ tin cậy mong muốn, cụ thể là chúng nằm trong số các điểm được đánh dấu với các độ tin cậy được tính toán cao nhất. Các bit cố định được đặt tại các vị trí bit còn lại trong vectơ đầu vào. Như được thể hiện trên Fig.9, nhóm các điểm vẽ thứ nhất 902 và nhóm các điểm vẽ thứ hai 904 được phân bố không đồng đều dọc theo trục x. Đó là bởi vì, do nguyên lý phân cực kênh được mô tả trước đó, các kênh tổng hợp tại các vị trí bit ngày càng cao dẫn đến có độ tin cậy cao hơn. Nói cách khác, độ tin cậy được phân bố không bằng nhau giữa các vị trí bit của vectơ đầu vào. Trong ví dụ được đề cập, các kênh tổng hợp tương ứng với nhóm các điểm vẽ thứ nhất 902 có độ tin cậy trung bình là 28,7%, trong khi các kênh tổng hợp tương ứng với nhóm các điểm vẽ thứ hai 904 có độ tin cậy trung bình là 71,3%.

Như được lưu ý ở trên, nhiều hơn hai chuỗi bit đầu vào có thể được ghép và sau đó được mã hóa thành từ mã kết hợp. Fig.10 là hình minh họa của ba chuỗi bit đầu vào 1050, 1052, 1054 đang được ghép thành chuỗi ghép 1056 và được mã hóa thành từ mã kết hợp 1020 cho việc truyền. Chuỗi các bit đầu vào thứ nhất 1050 bao gồm K_1 bit thông tin 1002 và CRC u_1-bit 1012 được tính

toán dựa trên K_1 bit thông tin 1002. Chuỗi bit đầu vào thứ hai 1052 bao gồm K_2 bit thông tin 1004 và CRC u_2-bit 1014 được tính toán dựa trên K_2 bit thông tin 1004. Chuỗi bit đầu vào thứ ba 1054 bao gồm K_3 bit thông tin 1006 và CRC u_3-bit 1016 được tính toán dựa trên K_3 bit thông tin 1006. Ba chuỗi các bit đầu vào 1050, 1052, 1054 được ghép để tạo ra chuỗi 1056 có kích thước $K_1+u_1+K_2+u_2+K_3+u_3$.

Việc ghép chuỗi 1056 được xử lý bởi quy trình mã hóa cực để tạo ra từ mã 1020 có độ dài N. Quy trình mã hóa cực bao gồm bước 1040 là chèn các bit cố định vào trong chuỗi ghép 1056 để tạo ra vectơ đầu vào 1090 có độ dài N, và sau đó nhân 1060 vectơ đầu vào 1090 với ma trận sinh mã cực, như được mô tả ở trên, chẳng hạn như với liên quan đến Fig.8, bước 808.

Theo một số phương án, không phải tất cả K_1, K_2, và K_3 là bằng nhau. Theo các phương án này, mã hóa cực được đề cập ở đây là mã hóa ghép cực không đối xứng, hoặc theo cách khác là mã hóa cực kết hợp khối không đối xứng. Từ mã 1020 được tạo ra được đề cập ở đây là từ mã ghép không đối xứng, hoặc còn là từ mã kết hợp khối không đối xứng. Theo một số phương án, K_1 khác K_2. Theo một số phương án, K_1, K_2, và K_3 đều khác nhau. Theo một số phương án, $K_1 + K_2$ bằng K_3 và $u_1 + u_2$ bằng u_3. Theo phương án ví dụ mà trong đó $K_1 + K_2$ bằng K_3, chuỗi các bit đầu vào thứ nhất 1050 và chuỗi các bit đầu vào thứ hai 1052 được tạo ra nhờ chia chuỗi $K_1 + K_2$ bit thông tin trước khi tính toán các CRC 1012 và 1014.

Trong một số trường hợp, mã hóa ghép cực không đối xứng có thể tạo điều kiện phân bố các bit CRC tại các vị trí bit trong vectơ đầu vào mà thỏa mãn tiêu chí độ tin cậy tốt hơn mã hóa ghép cực đối xứng. Mặc dù Fig.10 minh họa ba chuỗi bit đầu vào 1050, 1052, 1054 đang được ghép và mã hóa, nên được hiểu rằng theo các phương án khác ít hoặc nhiều chuỗi bit đầu vào hơn có thể được ghép và được mã hóa cho đầu ra như là từ mã 1020.

Fig.11 là sơ đồ thể hiện cây quyết định ví dụ để giải mã từ mã ghép, mà tại đó từ mã được mã hóa tuần tự nhờ kết hợp chuỗi K_1 bit thông tin với CRC u_1-bit và chuỗi K_2 bit thông tin với CRC u_2-bit. Cây quyết định có $K_1 + u_1 + K_2$

$+ u_2 + 1$ mức, mà nó được gọi là các mức 0 qua $K_1 + u_1 + K_2 + u_2$, một cách tương ứng. Cây quyết định được tạo ra theo cách theo chiều rộng thứ nhất, với mỗi mức của cây quyết định tương ứng với quyết định về bit không xác định tương ứng. Mỗi đường trong cây quyết định từ nút gốc đến các nút lá thể hiện phần chuỗi các bit không xác định có thể được giải mã và có khả năng tương ứng.

Trong quá trình tạo ra cây quyết định, đối với mỗi trong số $K_1 + u_1 + 1$ mức thứ nhất 1150 bắt đầu với nút gốc 1102, khi số lượng các đường có thể phát triển lớn hơn $L_1 = 32$, các đường L_1 có khả năng cao nhất được xác định, và các đường còn lại được loại bỏ. Tại mức $K_1 + u_1$ (1110), chuỗi K_1 bit thông tin thể hiện trong mỗi trong số các đường tồn tại đã được ước tính đối với CRC u_1-bit trong đường đó. Nút cuối cùng trong đường tồn tại mà vượt qua kiểm tra CRC này được thể hiện là nút 1112. Theo đó, đường từ nút gốc 1102 đến nút 1112 chứa chuỗi K_1 bit thông tin được giải mã. Các đường khác tại mức $K_1 + u_1$ (1110) được loại bỏ.

Đối với mỗi trong số $K_2 + u_2$ mức tiếp theo 1160 bắt đầu với nút 1112, khi số lượng các đường có thể phát triển lớn hơn $L_2 = 16$, các đường L_2 có khả năng cao nhất được xác định, và các đường còn lại được loại bỏ. Tại mức $K_1 + u_1 + K_2 + u_2$ (1120), chuỗi K_2 bit thông tin thể hiện trong mỗi trong số các đường tồn tại đã được ước tính đối với CRC u_2-bit trong đường đó. Nút cuối cùng trong đường tồn tại mà vượt qua kiểm tra CRC này được thể hiện là nút 1122. Theo đó, đường từ nút 1112 đến nút 1122 chứa chuỗi K_2 bit thông tin được giải mã.

Nên được hiểu rằng các giá trị của L_1 và L_2 được minh họa trong Fig.11 chỉ là ví dụ, và các giá trị khác có thể được chọn để giải mã từ mã ghép. Theo một số phương án, L_1 và L_2 là bằng nhau. Theo một số phương án, L_2 bé hơn L_1. Việc sử dụng giá trị của L_2 mà bé hơn giá trị của L_1 có thể làm giảm tính toán và các yêu cầu khoảng trống để tạo ra cây quyết định, và do đó có thể cải thiện hiệu suất, chẳng hạn như thông lượng, của bộ giải mã mà tạo ra cây quyết định. Bởi vì độ tin cậy của các kênh tổng hợp tương ứng với các vị trí bit đơn lẻ có xu hướng gia tăng theo độ dài của vectơ đầu vào được mã hóa, một số giá trị của L_2

mà bé hơn giá trị của L_1 có thể cải thiện thông lượng của bộ giải mã mà tạo ra cây quyết định với tác động tối thiểu trên tỷ lệ lỗi khối tổng thể của bộ giải mã.

Fig.12A là lưu đồ thể hiện phương pháp 1200 để giải mã cực. Vì mục đích tổng quát, các bước của Fig.12A đã được minh họa là bao gồm từ thu được sử dụng các EDC, mặc dù đã được giải thích trước đó, nên được hiểu rằng nhiều phương án sẽ dùng các CRC làm dạng cụ thể của EDC. Tại bước 1250, từ thứ nhất được thu. Từ thu được thứ nhất dựa trên từ mã thứ nhất, mà tại đó từ mã thứ nhất chứa các bit được tạo ra nhờ nhân vectơ đầu vào với ma trận sinh mã cực, mà tại đó vectơ đầu vào chứa chuỗi các bit đầu vào thứ nhất, chuỗi các bit đầu vào thứ hai, và các bit cố định cho mã cực. Chuỗi các bit đầu vào thứ nhất được dùng để tạo ra từ mã chứa K_1 bit thông tin và EDC u_1-bit, và chuỗi các bit đầu vào thứ hai chứa K_2 bit thông tin và EDC u_2-bit. Chuỗi các bit đầu vào thứ nhất xuất hiện trong vectơ đầu vào trước chuỗi các bit đầu vào thứ hai. Các vị trí của các bit của EDC u_1-bit trong chuỗi các bit đầu vào thứ nhất, các vị trí của các bit của EDC u_2-bit trong chuỗi các bit đầu vào thứ hai, và các vị trí của các bit cố định được biết bởi cả hai bộ mã hóa và bộ giải mã. Tại bước 1252, từ thu được thứ nhất được giải mã.

Fig.12B là lưu đồ của phương pháp 1202 để giải mã cực từ thu được thứ nhất của Fig.12A, phương pháp này bao gồm bước tạo ra cây quyết định như được thể hiện trên Fig.11, mà tại đó các CRC được dùng làm dạng cụ thể của EDC.

Tại bước 1204, các mức kế tiếp của cây quyết định nhị phân được tạo ra, mỗi mức tương ứng với quyết định về bit tương ứng, mà tại đó mỗi đường trong cây quyết định thể hiện phần chuỗi các bit không cố định có thể được giải mã và có khả năng tương ứng.

Trong quá trình tạo ra cây quyết định, tại bước 1206, đối với $K_1 + u_1$ mức thứ nhất của cây quyết định sau nút gốc, khi số lượng các đường trong cây quyết định phát triển vượt ngưỡng L_1, gần như các đường L_1 có thể nhất được loại bỏ. Tại bước 1208, khi mức $K_1 + u_1$ của cây quyết định đã được tạo ra, CRC u_1-bit được thể hiện trong mỗi đường còn tồn tại tương ứng được dùng để xác

định đường trong số các đường còn tồn tại thể hiện chuỗi K_1 bit được giải mã thứ nhất để bao gồm trong vectơ được giải mã, và tất cả các đường khác được loại bỏ.

Trong bước 1210, đối với $K_2 + u_2$ mức thứ hai của cây quyết định, khi số lượng các đường trong cây quyết định phát triển vượt ngưỡng L_2, gần như các đường L_2 có thể nhất được loại bỏ. Tại bước 1212, khi mức $K_1 + u_1 + K_2 + u_2$ của cây quyết định đã được tạo ra, CRC u_2-bit được thể hiện trong mỗi đường còn tồn tại tương ứng được dùng để xác định đường trong số các đường còn tồn tại thể hiện chuỗi K_2 bit được giải mã thứ hai để bao gồm trong vectơ được giải mã, và tất cả các đường khác được loại bỏ.

Theo một số phương án, L_1 và L_2 là bằng nhau. Theo một số phương án, L_1 là lớn hơn L_2. Theo một số phương án, K_1 bằng K_2.

Theo một số phương án, mà ở đó vectơ đầu vào được sử dụng để mã hóa từ mã còn được bao gồm chuỗi các bit đầu vào thứ ba có K_3 bit thông tin và u_3-bit CRC, và mà ở đó chuỗi bit đầu vào thứ hai xuất hiện trong vectơ đầu vào trước chuỗi bit đầu vào thứ ba, phương pháp giải mã có thể tiếp tục tạo ra cây quyết định. Đối với $K_3 + u_3$ mức thứ ba của cây quyết định, khi số lượng các đường trong cây quyết định phát triển vượt ngưỡng L_3, gần như các đường L_3 có thể nhất được loại bỏ. Khi mức $K_1 + u_1 + K_2 + u_2 + K_3 + u_3$ của cây quyết định đã được tạo ra, u_3-bit CRC thể hiện trong mỗi đường còn tồn tại tương ứng được dùng để xác định đường trong số các đường còn tồn tại thể hiện chuỗi thứ ba của K_3 bit được giải mã để bao gồm trong vectơ được giải mã.

Theo một số phương án, L_1 bằng L_2 và L_2 là lớn hơn L_3. Theo một số phương án, L_1 là lớn hơn L_2 và L_2 là lớn hơn L_3.

Một số phương án về bộ giải mã đưa ra vectơ được giải mã ngay khi từ thu được đã được giải mã hoàn toàn. Các phương án khác đưa ra mỗi phần của vectơ được giải mã ngay khi mỗi phần đã được xác định. Chẳng hạn như, chuỗi K_1 bit được giải mã thứ nhất có thể được đưa ra sau khi mức $K_1 + u_1$ của cây quyết định đã được tạo ra và kiểm tra CRC cho mức đó đã được thực hiện. Tương tự, chuỗi K_2 bit được giải mã thứ hai của có thể được xuất ra sau khi mức

$K_1 + u_1 + K_2 + u_2$ của cây quyết định đã được tạo ra và kiểm tra CRC cho mức đó đã được thực hiện.

Theo một số phương án, một số bước giải mã được thực hiện song song. Fig.13 là lưu đồ của phương pháp 1300 để giải mã cực mà tạo ra các cây quyết định song song, có khả năng giảm độ trễ cần thiết để giải mã hoàn toàn từ thu được so với phương pháp của Fig.12B. Phương pháp được minh họa xử lý từ thu được dựa trên từ mã của kiểu tương tự được mô tả với liên quan đến bước 1250 trên Fig.12A được thu, mà tại đó các CRC được dùng làm dạng cụ thể của EDC.

Tại bước 1304, từ thu được được xử lý nhờ tạo ra các cây quyết định nhị phân thứ nhất và thứ hai. Các cây quyết định nhị phân thứ nhất và thứ hai được tạo ra ít nhất một phần song song. Cụ thể là, ít nhất một phần của các bước 1306 đến 1308 và các bước 1310 đến 1314 được mô tả dưới đây được thực hiện theo cách chồng chéo ít nhất một phần theo thời gian.

Trong quá trình tạo ra cây quyết định thứ nhất, tại bước 1306, đối với $K_1 + u_1$ mức thứ nhất của cây quyết định thứ nhất sau nút gốc của cây quyết định thứ nhất, khi số lượng các đường trong cây quyết định thứ nhất phát triển vượt ngưỡng L_1, gần như các đường L_1 có thể nhất được loại bỏ. Tại bước 1308, khi mức $K_1 + u_1$ của cây quyết định đã được tạo ra, CRC u_1-bit được thể hiện trong mỗi đường còn tồn tại tương ứng được dùng để xác định đường trong số các đường còn tồn tại thể hiện chuỗi K_1 bit được giải mã thứ nhất để bao gồm trong vectơ được giải mã.

Trong quá trình tạo ra cây quyết định thứ hai, tại bước 1310, đối với các mức thứ nhất R_1 của cây quyết định thứ hai sau nút gốc của cây quyết định thứ hai, khi số lượng các đường trong cây quyết định thứ hai phát triển vượt ngưỡng L_2, gần như các đường L_2 có thể nhất được loại bỏ. Tại bước 1312, đối với các mức R_2 tiếp theo của cây quyết định thứ hai, khi số lượng các đường trong cây quyết định phát triển vượt ngưỡng L_3, gần như các đường L_3 có thể nhất được loại bỏ. Theo một số phương án, tại mức $K_1 + u_1$, CRC u_1-bit được thể hiện trong mỗi đường tồn tại của cây quyết định thứ hai được dùng để loại bỏ các đường tồn tại không vượt qua kiểm tra CRC với CRC u_1-bit, miễn là ít nhất một

đường tồn tại vượt qua kiểm tra CRC. Tại bước 1314, khi mức $K_1 + u_1 + K_2 + u_2$ của cây quyết định thứ hai đã được tạo ra, CRC u_2-bit được thể hiện trong mỗi đường còn tồn tại tương ứng của cây quyết định thứ hai được dùng để xác định đường trong số các đường còn tồn tại thể hiện chuỗi K_2 bit thứ hai để bao gồm trong vectơ được giải mã sau chuỗi K_1 bit thứ nhất.

Theo phương pháp được thể hiện trên Fig.13, R_1 nhỏ hơn $K_1 + u_1$ và L_1 lớn hơn L_2. Theo đó, đối với ít nhất một phần khởi tạo của quá trình tạo ra cây quyết định thứ hai, các đường trong cây quyết định thứ hai được giữ lại làm các đường tồn tại ít hơn các đường được giữ lại làm các đường tồn tại khi tạo ra cây quyết định thứ nhất. Điều này cho phép các mức khởi đầu của cây quyết định thứ hai được tạo ra nhanh hơn các mức tương tự của cây quyết định thứ nhất.

Theo một số phương án, L_2 bé hơn L_3. Đó là, quá trình tạo ra cây quyết định thứ hai có thể được cho bắt đầu bằng các đường tồn tại được giữ lại ít hơn tại các mức khởi tạo của cây quyết định thứ hai và sau đó "tăng lượng" để giữ lại nhiều đường tồn tại tại ở các mức về sau. Theo một số phương án, R_2 bằng $(K_2 + u_2 - R_1)$. Đó là, "tăng lượng" xảy ra đối với các mức của cây quyết định thứ hai mà thể hiện các quyết định đối với K_2 bit thông tin và CRC u_2-bit.

Theo một số phương án, "tăng lượng" được thực hiện trong nhiều giai đoạn. Chẳng hạn như, quá trình tạo ra cây quyết định thứ hai có thể bắt đầu với L=2, sau đó chuyển sang L=4, sau đó chuyển sang L=8. Theo một số phương án, bộ giải mã cực tạo ra nhiều hơn hai cây quyết định song song, với các cây quyết định có các cấu hình khác nhau của các giai đoạn "tăng lượng".

Các Fig.14A đến Fig.14D dùng để minh họa một số cách trong đó giải mã "tăng lượng" được thực hiện trong các giai đoạn có thể cải thiện độ trễ xử lý của bộ giải mã. Fig.14 là hình minh họa của chuỗi ghép 1460 của các bit đầu vào bao gồm bốn chuỗi bit đầu vào 1450, 1542, 1544, và 1456. Chuỗi bit đầu vào thứ nhất 1450 bao gồm K_1 bit thông tin 1402 và CRC u_1-bit 1412 được tính toán dựa trên K_1 bit thông tin 1402. Chuỗi các bit đầu vào thứ hai 1452 bao gồm K_2 bit thông tin 1404 và CRC u_2-bit 1414 được tính toán dựa trên K_2 bit thông tin 1404. Chuỗi bit đầu vào thứ ba 1454 bao gồm K_3 bit thông tin 1406 và CRC

u_3-bit 1416 được tính toán dựa trên K_3 bit thông tin 1406. Chuỗi bit đầu vào thứ tư 1456 bao gồm K_4 bit thông tin 1408 và CRC u_4-bit 1418 được tính toán dựa trên K_4 bit thông tin 1408. Chuỗi ghép 1460 có thể được mã hóa dùng kỹ thuật mã hóa cực, chẳng hạn như được mô tả ở trên với liên quan đến Fig.8, thành một từ mã kết hợp (không được thể hiện) để lưu trữ hoặc truyền.

Fig.14B minh họa ví dụ giải mã tuần tự từ mã mà đã được mã hóa nhờ ghép chuỗi các bit đầu vào 1460 của Fig.14A. Trong quá trình giải mã, kỹ thuật giải mã cực kiểu danh sách, chẳng hạn như được mô tả ở trên với liên quan đến Fig.12, được dùng. Các mức kế tiếp của cây quyết định nhị phân được tạo ra, mỗi mức tương ứng với quyết định về bit tương ứng, mà tại đó mỗi đường trong cây quyết định thể hiện phần chuỗi các bit không cố định có thể được giải mã và có khả năng tương ứng.

Trong giai đoạn thứ nhất 1422, K_1 bit thông tin 1402 được giải mã nhờ tạo ra cây quyết định nhị phân với ngưỡng bề rộng L=8. Đó là, đối với $K_1 + u_1$ mức thứ nhất của cây quyết định sau nút gốc, khi số lượng các đường trong cây quyết định phát triển vượt ngưỡng L=8, gần như các đường L có thể nhất được loại bỏ. Khi mức $K_1 + u_1$ của cây quyết định đã được tạo ra, CRC u_1-bit được thể hiện trong mỗi đường còn tồn tại tương ứng được dùng để xác định đường trong số các đường còn tồn tại thể hiện K_1 bit được giải mã để bao gồm trong vectơ được giải mã, và tất cả các đường khác được loại bỏ.

Trong giai đoạn tiếp theo 1424, bộ giải mã tiếp tục cách tương tự để giải mã K_2 bit thông tin 1404 nhờ tạo ra các mức kế tiếp của cây quyết định nhị phân với ngưỡng bề rộng L=4. Bộ giải mã sau đó tiếp tục cách tương tự trong giai đoạn tiếp theo 1426 để giải mã K_3 bit thông tin 1406 nhờ tạo ra các mức kế tiếp của cây quyết định nhị phân với ngưỡng bề rộng L=4. Cuối cùng, bộ giải mã sau đó tiếp tục cách tương tự trong giai đoạn tiếp theo 1428 để giải mã K_4 bit thông tin 1408 nhờ tạo ra các mức kế tiếp của cây quyết định nhị phân với ngưỡng bề rộng L=2. Trong ví dụ được minh họa của việc giải mã tuần tự, độ trễ xử lý 1470 để giải mã toàn bộ các bit thông tin được mã hóa bao gồm thời gian để thực hiện các giai đoạn 1422, 1424, 1426, và 1428 theo chuỗi. Nên được hiểu

rằng các giá trị cụ thể của L được lựa chọn cho các giai đoạn 1422, 1424, 1426, và 1428 là các lựa chọn thiết kế, và các giá trị khác có thể được lựa chọn, chẳng hạn như dựa trên các đặc tính độ chính xác và/hoặc độ trễ mong muốn cho bộ giải mã.

Fig.14C là hình minh họa việc giải mã, nhờ tạo ra các cây quyết định song song, của từ mã mà đã được mã hóa dựa trên chuỗi các bit đầu vào của Fig.14A. Trong quá trình giải mã, kỹ thuật giải mã cực kiểu danh sách song song, chẳng hạn như được mô tả ở trên liên quan đến Fig.13, được sử dụng. Các mức kế tiếp của bốn cây quyết định nhị phân được tạo ra song song, mỗi mức tương ứng với quyết định về bit tương ứng, mà tại đó mỗi đường trong cây quyết định thể hiện phần chuỗi bit không cố định có thể được giải mã và có khả năng tương ứng.

Trong tập hợp các hoạt động thứ nhất 1432, K_1 bit thông tin 1402 được giải mã nhờ tạo ra cây quyết định nhị phân thứ nhất với ngưỡng bề rộng L=8. Đó là, đối với $K_1 + u_1$ mức thứ nhất của cây quyết định sau nút gốc, khi số lượng các đường trong cây quyết định thứ nhất phát triển vượt ngưỡng L=8, tất cả các đường ngoại trừ các đường L có thể nhất được loại bỏ. Khi mức $K_1 + u_1$ của cây quyết định thứ nhất đã được tạo ra, CRC u_1-bit được thể hiện trong mỗi đường còn tồn tại tương ứng được dùng để xác định đường trong số các đường còn tồn tại thể hiện K_1 bit được giải mã để bao gồm trong vectơ được giải mã.

Trong tập hợp các hoạt động thứ hai 1434, K_1 bit thông tin 1402 được giải mã trong giai đoạn tăng lượng 1480 nhờ tạo ra cây quyết định nhị phân thứ hai với ngưỡng bề rộng mà tại đó L bé hơn 4 đối với ít nhất một phần của $K_1 + u_1$ mức thứ nhất của cây quyết định sau nút gốc. Quá trình tạo ra cây quyết định nhị phân thứ hai tiếp tục sau khi CRC u_1-bit cho K_1 bit thông tin được kiểm tra và đường tồn tại được lựa chọn. K_2 bit thông tin 1404 được giải mã tiếp theo với cây quyết định thứ hai dùng ngưỡng bề rộng L=4. Khi mức $K_1 + u_1 + K_2 + u_2$ của cây quyết định thứ hai đã được tạo ra, CRC u_2-bit được thể hiện trong mỗi đường còn tồn tại tương ứng được dùng để xác định đường trong số các đường còn tồn tại thể hiện K_2 bit được giải mã để bao gồm trong vectơ được giải mã.

Trong tập hợp thứ ba của các hoạt động 1436, K_1 bit thông tin 1402 và K_2 bit thông tin 1404 được giải mã trong giai đoạn tăng lượng 1482 nhờ tạo ra cây quyết định nhị phân thứ ba với ngưỡng bề rộng mà tại đó L bé hơn 4 đối với ít nhất một phần của $K_1 + u_1 + K_2 + u_2$ mức thứ nhất của cây quyết định sau nút gốc. Quá trình tạo ra cây quyết định nhị phân thứ ba tiếp tục sau khi CRC u_2-bit cho K_2 bit thông tin được kiểm tra và đường tồn tại được lựa chọn. K_3 bit thông tin 1406 được giải mã tiếp theo với cây quyết định thứ ba dùng ngưỡng bề rộng L=4. Khi mức $K_1 + u_1 + K_2 + u_2 + K_3 + u_3$ của cây quyết định thứ hai đã được tạo ra, CRC u_3-bit được thể hiện trong mỗi đường còn tồn tại tương ứng được dùng để xác định đường trong số các đường còn tồn tại thể hiện K_3 bit được giải mã để bao gồm trong vectơ được giải mã.

Trong tập hợp thứ tư của các hoạt động 1438, K_1 bit thông tin 1402, K_2 bit thông tin 1404, và K_3 bit thông tin 1406 được giải mã trong giai đoạn tăng lượng 1484 nhờ tạo ra cây quyết định nhị phân thứ tư với ngưỡng bề rộng mà tại đó L bé hơn 2 đối với ít nhất một phần của $K_1 + u_1 + K_2 + u_2 + K_3 + u_3$ mức thứ nhất của cây quyết định sau nút gốc. Quá trình tạo ra cây quyết định nhị phân thứ tư tiếp tục sau khi CRC u_3-bit cho K_3 bit thông tin được kiểm tra và đường tồn tại được lựa chọn. K_4 bit thông tin 1408 được giải mã tiếp theo với cây quyết định thứ ba dùng ngưỡng bề rộng L=2. Khi mức $K_1 + u_1 + K_2 + u_2 + K_3 + u_3 + K_4 + u_4$ của cây quyết định thứ hai đã được tạo ra, CRC u_4-bit được thể hiện trong mỗi đường còn tồn tại tương ứng được dùng để xác định đường trong số các đường còn tồn tại thể hiện K_4 bit được giải mã để bao gồm trong vectơ được giải mã.

Trong ví dụ được minh họa của việc giải mã song song, độ trễ xử lý 1472 để giải mã toàn bộ các bit thông tin được mã hóa bao gồm thời gian để thực hiện các hoạt động 1432, 1434, 1436, và 1438 song song. Trong ví dụ được minh họa, độ trễ xử lý 1472 ít hơn độ trễ xử lý 1470 được thể hiện trên Fig.14B. Nên được hiểu rằng các giá trị cụ thể của L được lựa chọn cho các hoạt động 1432, 1434, 1436, và 1438 là các lựa chọn thiết kế, và các giá trị khác có thể được lựa chọn, chẳng hạn như dựa trên các đặc tính độ chính xác và/hoặc độ trễ

mong muốn cho bộ giải mã. Cụ thể là, nhiều giá trị của L có thể được dùng trong các giai đoạn tăng lượng 1480, 1482, và 1484. Chẳng hạn như, trong giai đoạn tăng lượng 1482, giá trị của L=1 có thể được sử dụng ban đầu, mà chuyển tiếp tại một thời điểm nào đó trong quá trình tạo ra cây quyết định thứ ba thành L=2, và sau đó tại điểm sau này trong quá trình tạo ra cây quyết định thứ ba thành L=4.

Fig.14D minh họa sự thay đổi của kỹ thuật giải mã song song được thể hiện trên Fig.14C trong đó việc kết thúc sớm có thể được sử dụng để có khả năng cải thiện hiệu suất, chẳng hạn như nhờ cải thiện thông lượng và/hoặc công suất tiêu thụ. Kỹ thuật giải mã cực kiểu danh sách song song được dùng để giải mã từ mã mà đã được mã hóa dựa trên chuỗi bit đầu vào của Fig.14A. Trong quá trình giải mã, các mức kế tiếp của bốn cây quyết định nhị phân được tạo ra song song bằng các luồng riêng biệt.

Các luồng thứ nhất, thứ hai, thứ ba, và thứ tư lần lượt thực hiện tập hợp thứ nhất của các hoạt động 1932, tập hợp thứ hai của các hoạt động 1934, tập hợp thứ ba của các hoạt động 1936, và tập hợp thứ tư của các hoạt động 1938. Nếu việc kết thúc sớm không được thực hiện, các tập hợp của các hoạt động 1932, 1934, 1936, 1938 tiến hành như được mô tả ở trên lần lượt liên quan đến các hoạt động 1432, 1434, 1436, và 1438, của Fig.14C.

Tuy nhiên, trong một số ví dụ, việc kết thúc sớm của một số chuỗi có thể khả thi. Đó là bởi vì một số các bit thông tin được giải mã bằng nhiều hơn một luồng, một số luồng thực hiện giải mã kiểu danh sách cho các bit thông tin tương tự tại các mức độ phức tạp khác nhau. Chẳng hạn như, trong ví dụ được minh họa, luồng thứ nhất cố gắng để giải mã K_1 bit thông tin 1402 dùng ngưỡng bề rộng L=8. Các cố gắng còn được thực hiện để giải mã K_1 bit thông tin 1402 bằng luồng thứ hai sử dụng L=4, bằng luồng thứ ba sử dụng L=2, và bằng luồng thứ tư sử dụng L=1. Luồng cụ thể có thể được kết thúc sớm bằng một trong số các luồng được đánh số cao hơn nếu một trong số các luồng được đánh số cao hơn bất kỳ thực hiện kiểm tra CRC mà chỉ báo rằng các bit thông tin đang được giải mã bằng luồng cụ thể đã được giải mã thành công.

Chẳng hạn như, nếu luồng thứ tư thực hiện kiểm tra CRC 1996 và xác định rằng K_1 bit thông tin 1402 đã được giải mã thành công bằng luồng thứ tư, luồng thứ tư có thể làm luồng thứ nhất kết thúc tại thời điểm 1946. Nếu luồng thứ ba thực hiện kiểm tra CRC 1994 và xác định rằng K_1 bit thông tin 1402 đã được giải mã thành công bằng luồng thứ ba, luồng thứ ba có thể làm luồng thứ nhất kết thúc tại thời điểm 1944. Nếu luồng thứ hai thực hiện kiểm tra CRC 1992 và xác định rằng K_1 bit thông tin 1402 đã được giải mã thành công bằng luồng thứ hai, luồng thứ hai có thể làm luồng thứ nhất kết thúc tại thời điểm 1942. Nếu kết thúc sớm không được thực hiện, các bit được giải mã để bao gồm trong vectơ được giải mã tương ứng với K_1 bit thông tin 1402 không được xác định cho đến khi kiểm tra CRC 1990 đã được hoàn thành bằng luồng thứ nhất.

Mặc dù các ví dụ cụ thể của việc kết thúc sớm được thể hiện trong Fig.14D, nên được hiểu tổng quát hơn rằng, luồng được đánh số thấp hơn bất kỳ có thể được kết thúc sớm bằng luồng được đánh số cao hơn bất kỳ mà vượt qua kiểm tra CRC cho các bit thông tin đang được giải mã bằng luồng được đánh số thấp hơn với chi phí thấp hơn, chẳng hạn như do sử dụng ngưỡng bề rộng L nhỏ hơn. Trong một số trường hợp, việc kết thúc sớm có thể cải thiện độ trễ xử lý tổng thể 1972 và/hoặc giải mã thông lượng bởi vì một số luồng được kết thúc sớm. Trong một số trường hợp, việc kết thúc sớm có thể còn giảm công suất tiêu thụ trung bình tổng thể của quá trình giải mã nhờ giảm toàn bộ số lượng các tính toán cần thiết để giải mã từ mã được cho.

Theo một số phương án, các bit thông tin đã biết có thể được lặp lại trên nhiều bản tin đang được mã hóa thành nhiều từ mã. Nói cách khác, một loại ghép các bit thông tin khác bao gồm việc ghép các bit thông tin trên nhiều bản tin, theo đó dẫn đến việc ghép các bit thông tin trên nhiều từ mã. Nhờ lặp lại các bit thông tin đã biết, nó có thể giảm tỷ lệ lỗi khi truyền thông qua kênh, cải thiện hiệu suất của bộ giải mã, và/hoặc giảm độ phức tạp của bộ giải mã.

Fig.15 là ví dụ cụ thể của phương án này, mô tả hình minh họa của hai chuỗi bit đầu vào 1554, 1552 đang được mã hóa thành hai từ mã 1530, 1532 để truyền, chuỗi thứ nhất 1554 bao gồm tập hợp con của các bit đầu vào 1520 từ

chuỗi thứ hai 1552. Vì mục đích của việc giải mã, từ thu được dựa trên từ mã 1532 được giải mã sau từ thu được dựa trên từ mã 1530. Tuy nhiên, các từ mã 1530, 1532 có thể không cần phải được truyền liên tiếp, và các từ thu được dựa trên các từ mã 1530, 1532 có thể không cần phải được thu liên tiếp. Theo một số phương án, từ mã thứ nhất 1530 được truyền trước khi truyền từ mã thứ hai 1532. Theo một số phương án, từ mã thứ nhất 1530 được truyền đồng thời với, hoặc nếu không thì trong khoảng thời gian xấp xỉ với, từ mã thứ hai 1532.

Chuỗi các bit đầu vào thứ nhất 1554 bao gồm chuỗi các bit đầu vào ban đầu 1550, chuỗi ban đầu bao gồm K_1 bit thông tin 1502 và CRC u_1-bit 1512 được tính toán dựa trên K_1 bit thông tin 1502. Chuỗi các bit đầu vào thứ nhất 1554 còn bao gồm các bit thông tin được sao chép 1520. Các bit thông tin được sao chép 1520 là bản sao của tập hợp con 1506 của chuỗi các bit thông tin thứ hai 1504. Các bit thông tin được sao chép 1520 bao gồm K_2 bit thông tin. Cuối cùng, chuỗi các bit đầu vào thứ nhất còn bao gồm CRC u_2-bit 1522 được tính toán dựa trên K_2 bit thông tin.

Chuỗi các bit đầu vào thứ hai 1552 bao gồm $K_2 + K_3$ bit thông tin 1504 và CRC u_3-bit 1514 được tính toán dựa trên $K_2 + K_3$ bit thông tin 1504. K_2 bit thông tin dẫn đầu 1506 trong số $K_2 + K_3$ bit thông tin 1504 giống với các bit thông tin được sao chép 1520. K_3 bit thông tin 1508 xuất hiện sau K_2 bit thông tin dẫn đầu 1506 trong $K_2 + K_3$ bit thông tin 1504.

Nói cách khác, K_2 bit thông tin dẫn đầu của chuỗi bit đầu vào thứ hai 1552 đã được sao chép vào chuỗi các bit đầu vào thứ nhất 1554. Bộ mã hóa mà nó sao chép tập hợp con của các bit thông tin để được mã hóa thành từ mã kế tiếp vào chuỗi các bit thông tin để được mã hóa thành từ mã kế trước được đề cập ở đây là bộ mã hóa cực cửa sổ trượt. Các mã được mã hóa với bộ mã hóa được đề cập ở đây là các mã cực cửa sổ trượt.

Chuỗi các bit đầu vào thứ nhất 1554 được xử lý bởi quy trình mã hóa cực để tạo ra từ mã thứ nhất 1530 có độ dài N. Chuỗi các bit đầu vào thứ hai 1552 được xử lý bởi quy trình mã hóa cực để tạo ra từ mã thứ hai 1532 có độ dài N. Quy trình mã hóa cực đối với chuỗi các bit đầu vào thứ nhất 1554 bao gồm

bước 1540 chèn các bit cố định vào chuỗi các bit đầu vào thứ nhất 1554 để tạo ra vectơ đầu vào thứ nhất 1590 có độ dài N, và sau đó nhân 1560 vectơ đầu vào thứ nhất 1590 với ma trận sinh mã cực để tạo ra từ mã thứ nhất 1530. Quy trình mã hóa cực cho chuỗi các bit đầu vào thứ hai 1552 bao gồm bước 1542 chèn các bit cố định vào chuỗi các bit đầu vào thứ hai 1552 để tạo ra vectơ đầu vào thứ hai 1592 có độ dài N, và sau đó nhân 1562 vectơ đầu vào thứ hai 1592 với ma trận sinh mã cực để tạo ra từ mã thứ hai 1532.

Theo phương án ví dụ, K_1 bằng $K_2 + K_3$ và u_1 bằng u_3. Theo phương án này, các bit thông tin được mã hóa thành từ mã 1530 nhiều hơn từ mã 1532, dẫn đến tỷ lệ mã hóa thấp hơn đối với từ mã 1530. Để bù lại, theo một số phương án, mức công suất tại đó từ mã 1530 được truyền là tương đối cao hơn mức công suất gốc và/hoặc tương đối với mức công suất tại đó từ mã 1532 được truyền. Bởi vì một số các bit thông tin được mã hóa thành từ mã 1532 là còn được mã hóa thành từ mã 1530, tỷ lệ lỗi đối với SNR được cho đối với từ mã 1532 sau quy trình giải mã có thể được cải thiện, như được giải thích dưới đây với liên quan đến Fig.17. Do tỷ lệ lỗi được cải thiện, theo một số phương án, mức công suất tại đó từ mã 1532 được truyền là tương đối thấp hơn mức công suất gốc và/hoặc tương đối với mức công suất tại đó từ mã 1530 được truyền.

Theo một số phương án, từ mã thứ nhất 1530 và từ mã thứ hai 1532 được tạo ra bằng thiết bị giống nhau, chẳng hạn như trạm gốc truyền thông với thiết bị người dùng (UE). Tuy nhiên, từ mã thứ nhất 1530 và từ mã thứ hai 1532 có thể còn được tạo ra bằng các thiết bị khác nhau. Fig.16 là sơ đồ minh họa của trạm gốc thứ nhất 1610 và trạm gốc thứ hai 1612 truyền thông với thiết bị UE 1602, mà tại đó trạm gốc thứ nhất 1610 tạo ra từ mã thứ nhất và trạm gốc thứ hai 1612 tạo ra từ mã thứ hai.

Trong ví dụ dùng trường hợp được minh họa trong Fig.16, thiết bị UE 1602 được đặt trong khối thứ nhất 1680 và gần ranh giới của khối thứ hai 1682. Trạm gốc thứ nhất 1610 và trạm gốc thứ hai 1612 kết hợp để tạo thuận lợi để chuyển giao thiết bị UE 1602 đến khối thứ hai 1682. Trạm gốc thứ nhất 1610 nhằm mục đích truyền ít nhất một chuỗi K_1 bit thông tin thứ nhất đến thiết bị UE

1602 trong việc truyền thứ nhất 1620 bao gồm từ mã thứ nhất. Trạm gốc thứ hai 1612 nhằm mục đích truyền chuỗi $K_2 + K_3$ bit thông tin thứ hai đến thiết bị UE 1602 trong việc truyền thứ hai 1622 bao gồm từ mã thứ hai.

Trạm gốc thứ hai 1612 truyền, qua kết nối đường trục đến trạm gốc thứ nhất 1610, K_2 bit thông tin dẫn đầu trong số chuỗi $K_2 + K_3$ bit thông tin. Trạm gốc thứ nhất 1610 sau đó mã hóa K_1 và K_2 bit thông tin thành từ mã thứ nhất theo cách được mô tả ở trên với liên quan đến từ mã thứ nhất 1530 của Fig.15, và sau đó truyền từ mã thứ nhất đến UE. Trạm gốc thứ hai mã hóa $K_2 + K_3$ bit thông tin thành từ mã thứ hai theo cách được minh họa với liên quan đến từ mã thứ hai 1532 của Fig.15, và sau đó truyền từ mã thứ hai đến UE.

Quay sang bước giải mã, giả sử rằng các từ thu được thứ nhất và thứ hai dựa trên các từ mã mã cực cửa sổ trượt thứ nhất và thứ hai, một cách tương ứng, được thu, mà tại đó từ mã thứ hai kế tiếp từ mã thứ nhất. Từ thu được thứ nhất có thể được giải mã bằng các phương pháp được giải thích ở trên và được minh họa, chẳng hạn như, trên các Fig.12 và Fig.13. Fig.17 là lưu đồ của phương pháp 1700 giải mã cực cửa sổ trượt của từ thu được thứ hai. Để mô tả đơn giản, phương pháp 1700 sẽ được mô tả theo giả định rằng phương pháp 1700 tiếp tục sau khi phương pháp 1202 được giải thích ở trên với liên quan đến Fig.12B đã được thực hiện.

Tại bước 1702, từ thu được thứ hai dựa trên từ mã thứ hai được thu, từ mã thứ hai chứa các bit được tạo ra nhờ nhân vectơ đầu vào thứ hai với ma trận sinh mã cực, mà tại đó vectơ đầu vào thứ hai chứa chuỗi các bit đầu vào thứ ba và các bit cố định cho mã cực. Chuỗi các bit đầu vào thứ ba được dùng để tạo ra từ mã thứ hai chứa K_2 bit thông tin được theo sau bởi K_3 bit thông tin và CRC u_3-bit. Các vị trí của các bit CRC u_3-bit trong chuỗi các bit đầu vào thứ ba và các vị trí của các bit cố định được biết bởi cả hai bộ mã hóa và bộ giải mã.

Tại bước 1704, CRC u_2-bit được dùng để xác định xem K_2 bit thông tin trong từ thu được thứ nhất có được giải mã thành công hay không. Nếu K_2 bit thông tin trong từ thu được thứ nhất không được giải mã thành công, từ thu được thứ nhất sau đó được loại bỏ và từ thu được thứ hai được giải mã với kỹ thuật

giải mã cực thông thường, chẳng hạn như được mô tả trong các bước 1204, 1206, và 1208 của Fig.12. Tại bước 1706, nếu K_2 bit thông tin trong từ thu được thứ nhất được giải mã thành công, K_2 bit thông tin sau đó được bao gồm làm K_2 bit khởi đầu cho vectơ được giải mã thứ hai.

Tại bước 1708, các mức kế tiếp của cây quyết định nhị phân thứ hai được tạo ra, mỗi mức tương ứng với quyết định về bit tương ứng, mà tại đó mỗi đường trong cây quyết định thứ hai thể hiện phần chuỗi các bit không cố định có thể được giải mã và có khả năng tương ứng. Trong quá trình tạo ra cây quyết định nhị phân thứ hai, K_2 bit thông tin được coi là các bit cố định.

Trong quá trình tạo ra cây quyết định nhị phân thứ hai, tại bước 1710, đối với các mức của cây quyết định thứ hai sau nút gốc, khi số lượng các đường trong cây quyết định phát triển vượt ngưỡng L_3, gần như các đường L_3 có thể nhất được loại bỏ. Tại bước 1712, khi mức $K_3 + u_3$ của cây quyết định thứ hai đã được tạo ra, CRC u_3-bit được thể hiện trong mỗi đường còn tồn tại tương ứng được dùng để xác định đường trong số các đường còn tồn tại thể hiện chuỗi K_3 bit được giải mã thứ ba, mà nó được bao gồm sau đó tại bước 1714 làm K_3 bit sau K_2 bit khởi đầu trong vectơ được giải mã thứ hai. Tất cả các đường khác được loại bỏ.

Vì kỹ thuật giải mã từ thu được thứ hai sử dụng thông tin từ từ thu được thứ nhất, bộ giải mã được đề cập ở đây là bộ giải mã cực cửa sổ trượt.

Bởi vì kỹ thuật giải mã cực cửa sổ trượt được mô tả có thể xử lý K_2 bit thông tin là các bit cố định, và bởi vì các vị trí bit tiếp theo dọc theo vectơ đầu vào có xu hướng tương ứng với các kênh tổng hợp có độ tin cậy cao hơn, Trong một số trường hợp giá trị L_3 nhỏ hơn có thể được dùng trong bộ giải mã cực cửa sổ trượt (so sánh với một số bộ giải mã cực mà không sử dụng thông tin từ từ thu được thứ nhất) mà không làm giảm đáng kể tỷ số lỗi khối của bộ giải mã. Trong một số trường hợp, các giá trị L_3 nhỏ hơn có thể cải thiện độ trễ giải mã và/hoặc cải thiện hiệu suất giải mã. Theo một số phương án, L_1 là lớn hơn L_2. Theo một số phương án, L_1 là lớn hơn L_3. Theo một số phương án, giá trị khác nhau của ít nhất một trong các đường L_1, L_2, và/hoặc L_3 có thể được dùng trong

bộ giải mã dựa trên mức công suất tại đó từ mã thứ nhất được truyền và/hoặc mức công suất tại đó từ mã thứ hai được truyền.

Bây giờ, chuyển sang các thiết bị ví dụ để thực hiện các phương pháp được mô tả ở trên, Fig.18A là sơ đồ minh họa của thiết bị 1800 để mã hóa và truyền từ mã. Thiết bị 1800 bao gồm bộ mã hóa 1804 ghép cặp với thiết bị truyền 1806. Theo phương án được minh họa, thiết bị truyền 1806 có anten 1808 để truyền các tín hiệu qua kênh không dây. Theo một số phương án, thiết bị truyền 1806 bao gồm bộ điều biến, bộ khuyến đại, và/hoặc các thành phần khác của chuỗi truyền tần số radio (RF). Bộ mã hóa 1804 thu đầu vào 1802 bao gồm các bit thông tin và được tạo cấu hình để thực hiện phương pháp được mô tả ở trên để mã hóa các bit thông tin thành từ mã, mà nó được cung cấp đến thiết bị truyền 1806 để truyền bằng anten 1808.

Fig.18B là sơ đồ minh họa của thiết bị 1810 để thu và giải mã từ thu được. Thiết bị 1810 bao gồm thiết bị thu 1814 ghép cặp với bộ giải mã 1416. Theo phương án được minh họa, thiết bị thu 1814 có anten 1812 để thu các tín hiệu từ kênh không dây. Theo một số phương án, thiết bị thu 1814 bao gồm bộ giải điều biến, bộ khuyến đại, và/hoặc các thành phần khác của chuỗi thu tần số radio (RF). Thiết bị thu 1814 thu tín hiệu mang từ thu dựa trên từ mã bằng anten 1812. Từ thu được được cung cấp đến bộ giải mã 1816. Bộ giải mã 1816 được tạo cấu hình để thực hiện phương pháp được mô tả ở trên để giải mã từ thu được thành vectơ đầu ra bao gồm các bit thông tin, mà nó được cung cấp là đầu ra 1818 từ bộ giải mã.

Theo một số phương án, phương tiện có thể đọc được bằng máy tính không mất dữ liệu khi mất nguồn bao gồm các chỉ dẫn cho việc thực hiện bằng bộ xử lý có thể được bố trí để điều khiển hoạt động của bộ mã hóa 1804 trên Fig.18A hoặc bộ giải mã 1816 trên Fig.18B, và/hoặc nếu không thì để điều khiển việc thực hiện các phương pháp được mô tả ở trên. Theo một số phương án, bộ xử lý được điều khiển có thể là thành phần của nền tảng phần cứng máy tính có mục đích thông thường. Theo các phương án khác, bộ xử lý có thể là thành phần của nền tảng phần cứng có mục đích chuyên dụng. Chẳng hạn như,

bộ xử lý có thể là bộ xử lý nhúng, và các chỉ dẫn có thể được cung cấp là chương trình cơ sở. Một số phương án có thể được thực hiện nhờ chỉ dùng phần cứng. Theo một số phương án, các chỉ dẫn cho việc thực hiện bằng bộ xử lý có thể được nhúng trong biểu mẫu tích số phần mềm. Tích số phần mềm có thể được lưu trữ trong phương tiện lưu trữ không cấp điện hoặc không tạm thời, mà nó có thể là, chẳng hạn như, đĩa CD chứa dữ liệu chỉ đọc (CD-ROM), ổ cứng gắn nhanh cổng USB, hoặc ổ cứng tháo rời được.

Phần mô tả ở trên về một số phương án được cung cấp để cho phép người có hiểu biết trung bình trong lĩnh vực tạo ra hoặc sử dụng thiết bị, phương pháp, hoặc phương tiện có thể đọc bởi bộ xử lý theo sáng chế. Các cải biến khác nhau của các phương án này sẽ được dễ dàng thực hiện bởi người có hiểu biết trung bình trong lĩnh vực, và các nguyên lý chung của các phương pháp và các thiết bị được mô tả ở đây có thể được áp dụng vào các phương án khác. Như vậy, sáng chế không bị giới hạn bởi các phương án được trình bày ở đây mà được chấp nhận phạm vi bảo hộ rộng nhất phù hợp với các nguyên lý và các đặc điểm mới được bộc lộ ở đây.

YÊU CẦU BẢO HỘ

1. Phương pháp truyền thông được cho phép sửa lỗi, bao gồm:

mã hóa cực vectơ đầu vào để đưa ra từ mã thứ nhất, vectơ đầu vào bao gồm các bit đầu vào thứ nhất, các bit đầu vào thứ hai, và các bit đóng băng cho mã Cực, trong đó:

các bit đầu vào thứ nhất bao gồm K_1 bit thông tin và mã sửa lỗi (ECC-error-correcting code) bit u_1 được tạo ra từ K_1 bit thông tin,

các bit đầu vào thứ hai bao gồm K_2 bit thông tin và ECC bit u_2 được tạo ra từ K_2 bit thông tin, và

các bit đầu vào thứ nhất xuất hiện trong vectơ đầu vào trước các bit đầu vào thứ hai; và

truyền từ mã thứ nhất qua kênh vật lý.

2. Phương pháp theo điểm 1, trong đó ít nhất một trong số ECC bit u_1 hoặc ECC bit u_2 nằm trong các vị trí bit của vectơ đầu vào mà thỏa mãn tiêu chí độ tin cậy mong muốn.

3. Phương pháp theo điểm 1, trong đó mã hóa cực vectơ đầu vào bao gồm nhân vectơ đầu vào với ma trận tích Kronecker gấp m $\mathbf{G}_2^{\otimes m}$, trong đó

$$\mathbf{G}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \text{ hoặc } \mathbf{G}_2 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}.$$

4. Phương pháp theo điểm 1, trong đó vectơ đầu vào còn bao gồm các bit đầu vào thứ ba, các bit đầu vào thứ ba này bao gồm K_3 bit thông tin và ECC bit u_3 được tạo ra từ K bit thông tin, và các bit đầu vào thứ hai xuất hiện trong vectơ đầu vào trước chuỗi bit đầu vào thứ ba.

5. Phương pháp truyền thông được cho phép sửa lỗi, bao gồm:

thu từ mã thứ nhất được tạo từ mã hóa cực vectơ đầu vào, vectơ đầu vào bao gồm các bit đầu vào thứ nhất, các bit đầu vào thứ hai, và các bit đóng băng cho mã cực, trong đó các bit đầu vào thứ nhất bao gồm K_1 bit thông tin và mã

sửa lỗi (ECC) bit u_1 được tạo ra từ K_1 bit thông tin, các bit đầu vào thứ hai bao gồm K_2 bit thông tin và ECC bit u_2 được tạo ra từ K_2 bit thông tin, và các bit đầu vào thứ nhất xuất hiện trong vectơ đầu vào trước các bit đầu vào thứ hai; và

giải mã từ mã thứ nhất để đưa ra vectơ đầu vào được giải mã.

6. Phương pháp theo điểm 5, trong đó ít nhất một trong số ECC bit u_1 hoặc ECC bit u_2 được đặt tại các vị trí bit của vectơ đầu vào mà thỏa mãn tiêu chí độ tin cậy muốn có.

7. Phương pháp theo điểm 5, trong đó vectơ đầu vào còn bao gồm các bit đầu vào thứ ba, các bit đầu vào thứ ba này bao gồm K_3 bit thông tin và ECC bit u_3 được tạo ra từ K_3 bit thông tin, và các bit đầu vào thứ hai xuất hiện trong vectơ đầu vào trước các bit đầu vào thứ ba.

8. Bộ truyền dùng cho truyền thông được cho phép sửa lỗi, bao gồm:

bộ xử lý; và

bộ nhớ được ghép nối với bộ xử lý, bộ nhớ này lưu trữ các lệnh mà, khi được thực thi bởi bộ xử lý, khiến cho bộ xử lý:

mã hóa cực vectơ đầu vào để đưa ra từ mã thứ nhất, vectơ đầu vào bao gồm các bit đầu vào thứ nhất, các bit đầu vào thứ hai, và các bit đóng băng cho mã cực, trong đó

các bit đầu vào thứ nhất bao gồm K_1 bit thông tin và mã sửa lỗi (ECC) bit u_1 được tạo ra từ K_1 bit thông tin,

các bit đầu vào thứ hai bao gồm K_2 bit thông tin và ECC bit u_2 được tạo ra từ K_2 bit thông tin, và

các bit đầu vào thứ nhất xuất hiện trong vectơ đầu vào trước các bit đầu vào thứ hai; và

truyền từ mã thứ nhất qua kênh vật lý.

9. Bộ truyền theo điểm 8, trong đó ít nhất một trong số ECC bit u_1 hoặc ECC bit u_2 nằm trong các vị trí bit của vectơ đầu vào mà thỏa mãn tiêu chí độ tin cậy mong muốn.

10. Bộ truyền theo điểm 8, trong đó các lệnh khiến bộ xử lý mã hóa cực vectơ đầu vào bằng cách nhân vectơ đầu vào với ma trận tích Kronecker gấp m $\mathbf{G}_2^{\otimes m}$, trong đó:

$$\mathbf{G}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \text{ hoặc } \mathbf{G}_2 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}.$$

11. Bộ truyền theo điểm 8, trong đó vectơ đầu vào còn bao gồm các bit đầu vào thứ ba, các bit đầu vào thứ ba này bao gồm K_3 bit thông tin và ECC bit u_3 được tạo ra từ K bit thông tin, và các bit đầu vào thứ hai xuất hiện trong vectơ đầu vào trước chuỗi bit đầu vào thứ ba.

12. Bộ thu dùng cho truyền thông được cho phép sửa lỗi, bao gồm:

bộ xử lý; và

bộ nhớ được ghép nối với bộ xử lý, bộ nhớ này lưu trữ các lệnh mà, khi được thực thi bởi bộ xử lý, khiến cho bộ xử lý:

thu từ mã thứ nhất được tạo từ mã hóa cực vectơ đầu vào, vectơ đầu vào bao gồm các bit đầu vào thứ nhất, các bit đầu vào thứ hai, và các bit đóng băng cho mã Cực, trong đó các bit đầu vào thứ nhất bao gồm K_1 bit thông tin và mã sửa lỗi (ECC) bit u_1 được tạo ra từ K_1 bit thông tin, các bit đầu vào thứ hai bao gồm K_2 bit thông tin và ECC bit u_2 được tạo ra từ K_2 bit thông tin, và các bit đầu vào thứ nhất xuất hiện trong vectơ đầu vào trước các bit đầu vào thứ hai; và

giải mã từ mã thứ nhất để đưa ra vectơ đầu vào được giải mã.

13. Bộ thu theo điểm 12, trong đó ít nhất một trong số ECC bit u_1 hoặc ECC bit u_2 được đặt tại các vị trí bit của vectơ đầu vào mà thỏa mãn tiêu chí độ tin cậy muốn có.

14. Bộ thu theo điểm 12, trong đó vectơ đầu vào còn bao gồm các bit đầu vào thứ ba, các bit đầu vào thứ ba này bao gồm K_3 bit thông tin và ECC bit u_3 được

tạo ra từ K_3 bit thông tin, và các bit đầu vào thứ hai xuất hiện trong vectơ đầu vào trước các bit đầu vào thứ ba.

100

$$G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad G_2^{\otimes 2} = \begin{bmatrix} G_2 & 0 \\ G_2 & G_2 \end{bmatrix} \quad G_2^{\otimes 3} = \begin{bmatrix} G_2 & 0 & 0 & 0 \\ G_2 & G_2 & 0 & 0 \\ G_2 & 0 & G_2 & 0 \\ G_2 & G_2 & G_2 & G_2 \end{bmatrix}$$

$$G_2^{\otimes 2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

102

$$G_2^{\otimes 3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

104

FIG. 1

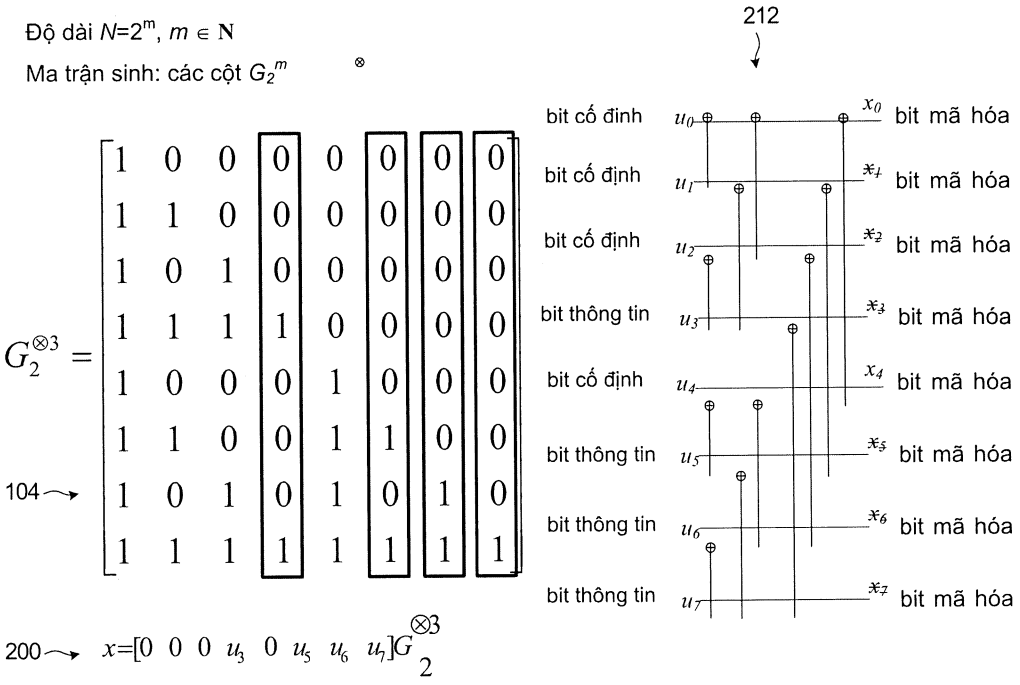

Độ dài $N=2^m$, $m \in \mathbf{N}$
Ma trận sinh: các cột G_2^m
212
$G_2^{\otimes 3} =$
104
200
$x=[0\ 0\ 0\ u_3\ 0\ u_5\ u_6\ u_7]G_2^{\otimes 3}$
bit cố đinh
bit cố định
bit cố định
bit thông tin
bit cố định
bit thông tin
bit thông tin
bit thông tin
bit mã hóa

FIG. 2

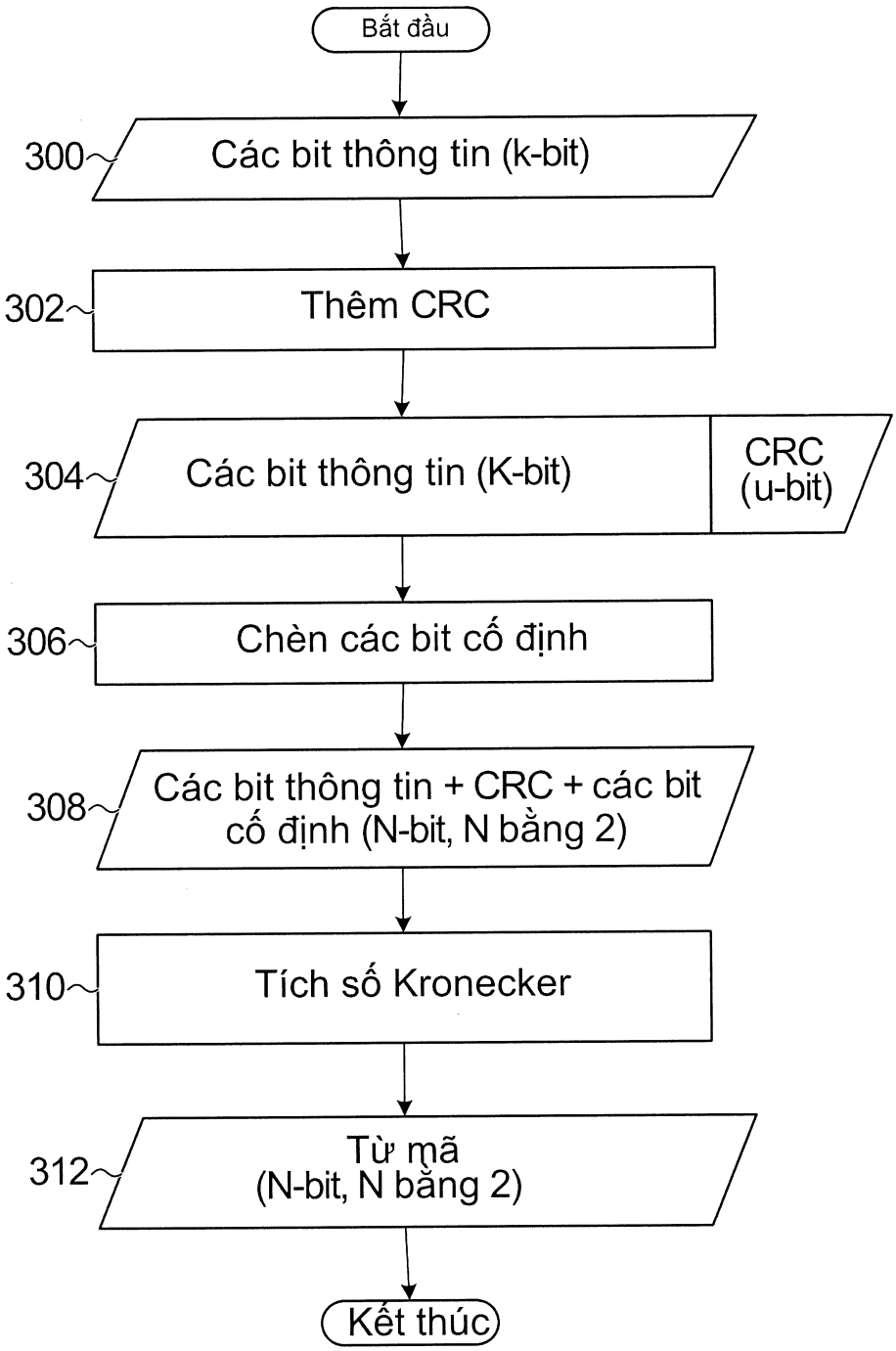

Bắt đầu
300
Các bit thông tin (k-bit)
302
Thêm CRC
304
Các bit thông tin (K-bit)
CRC
(u-bit)
306
Chèn các bit cố định
308
Các bit thông tin + CRC + các bit cố định (N-bit, N bằng 2)
310
Tích số Kronecker
312
Từ mã
(N-bit, N bằng 2)
Kết thúc

FIG. 3

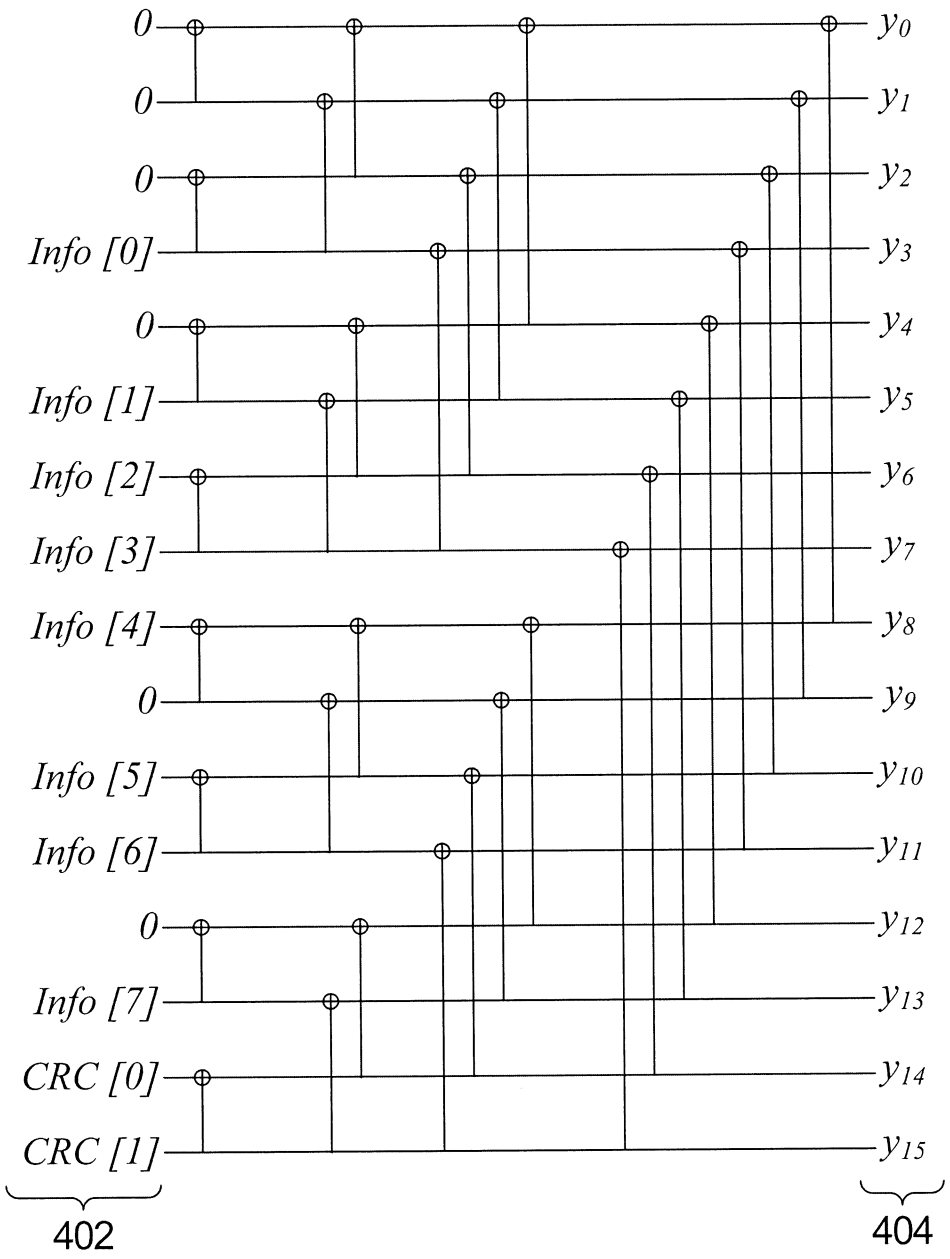

0
0
0
Info [0]
0
Info [1]
Info [2]
Info [3]
Info [4]
0
Info [5]
Info [6]
0
Info [7]
CRC [0]
CRC [1]
y0
y1
y2
y3
y4
y5
y6
y7
y8
y9
y10
y11
y12
y13
y14
y15
402
404

FIG. 4

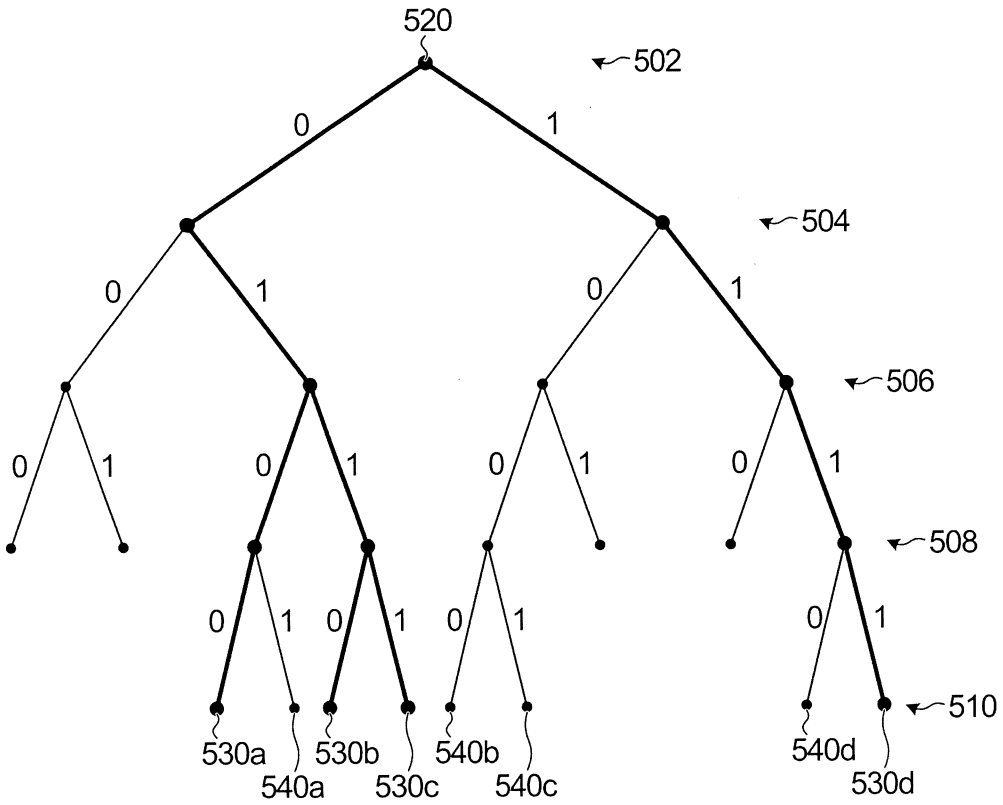

520
502
504
506
508
510
530a
540a
530b
530c
540b
540c
540d
530d

FIG. 5

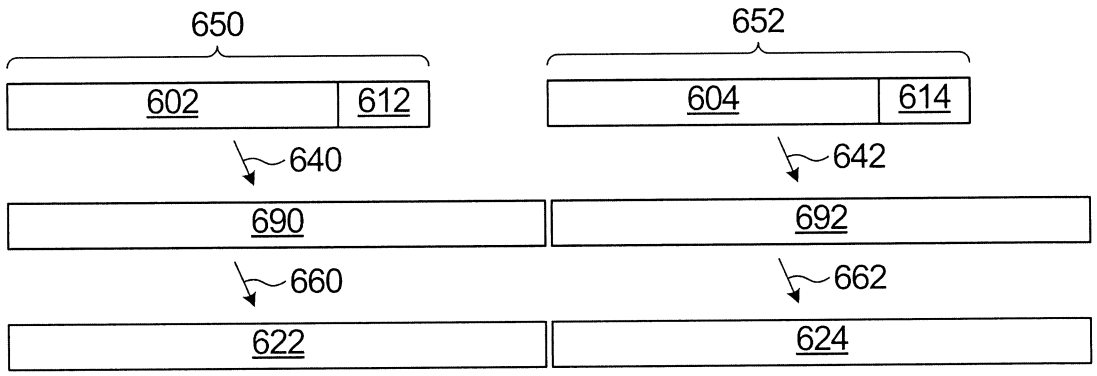

650
652
602
612
604
614
640
642
690
692
660
662
622
624

FIG. 6

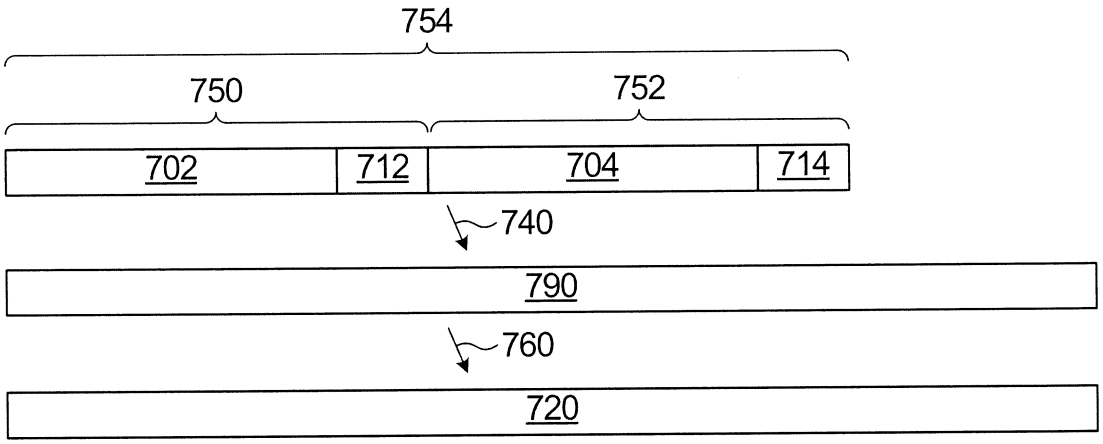

754
750
752
702
712
704
714
740
790
760
720

FIG. 7

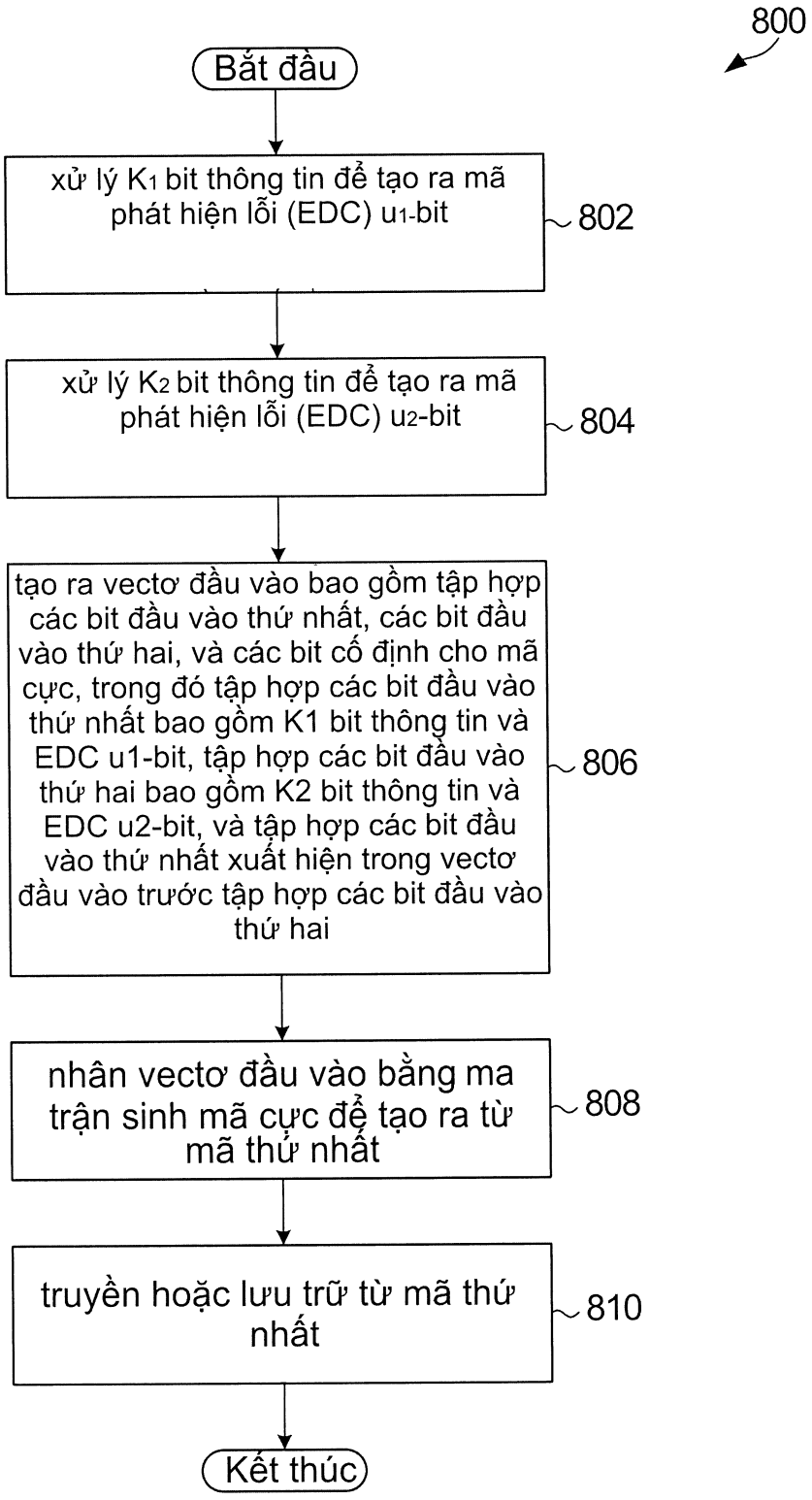

800
Bắt đầu
xử lý K1 bit thông tin để tạo ra mã phát hiện lỗi (EDC) u1-bit
802
xử lý K2 bit thông tin để tạo ra mã phát hiện lỗi (EDC) u2-bit
804
tạo ra vectơ đầu vào bao gồm tập hợp các bit đầu vào thứ nhất, các bit đầu vào thứ hai, và các bit cố định cho mã cực, trong đó tập hợp các bit đầu vào thứ nhất bao gồm K1 bit thông tin và EDC u1-bit, tập hợp các bit đầu vào thứ hai bao gồm K2 bit thông tin và EDC u2-bit, và tập hợp các bit đầu vào thứ nhất xuất hiện trong vectơ đầu vào trước tập hợp các bit đầu vào thứ hai
806
nhân vectơ đầu vào bằng ma trận sinh mã cực để tạo ra từ mã thứ nhất
808
truyền hoặc lưu trữ từ mã thứ nhất
810
Kết thúc

FIG. 8

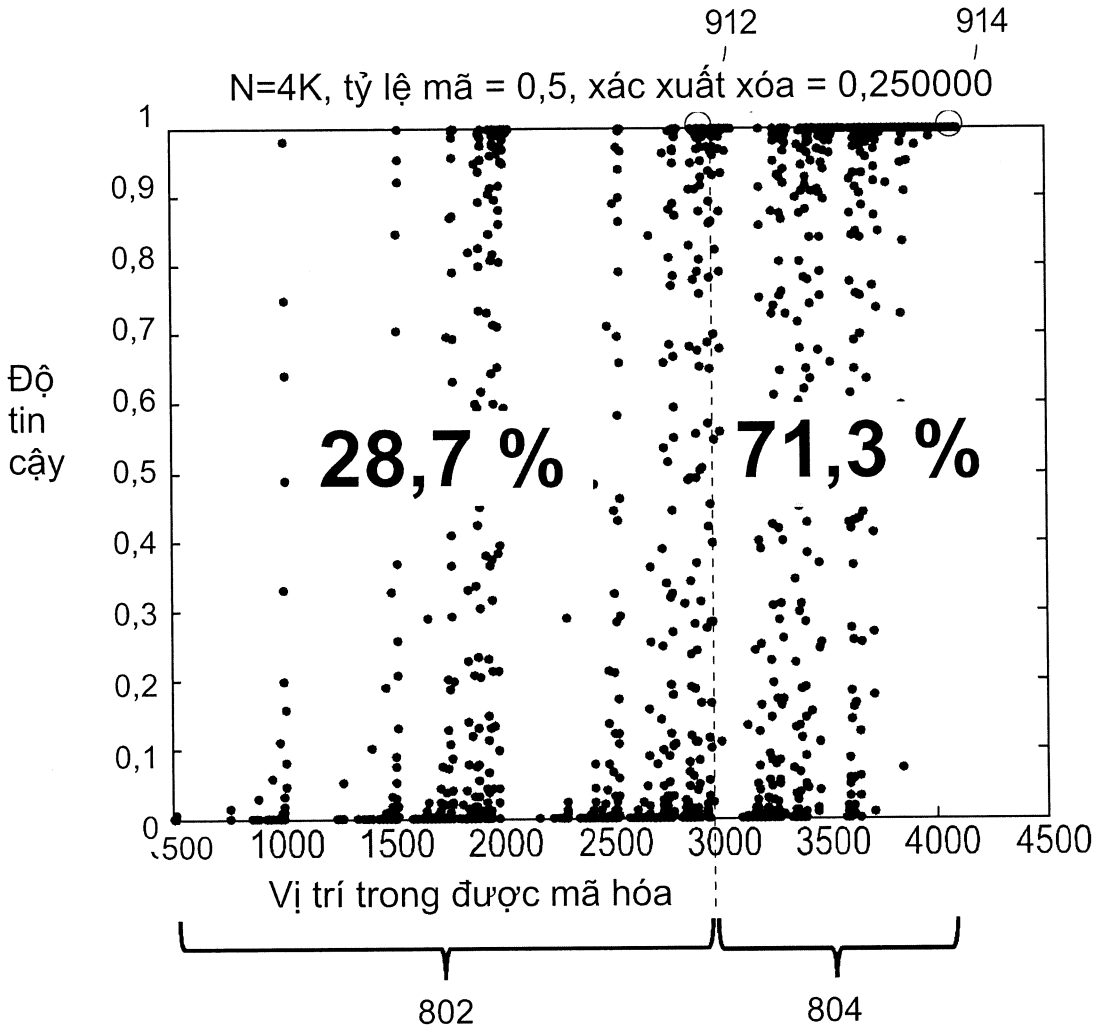

912
914
N=4K, tỷ lệ mã = 0,5, xác xuất xóa = 0,250000
Độ tin cậy
28,7 %
71,3 %
1
0,9
0,8
0,7
0,6
0,5
0,4
0,3
0,2
0,1
0
500
1000
1500
2000
2500
3000
3500
4000
4500
Vị trí trong được mã hóa
802
804

FIG. 9

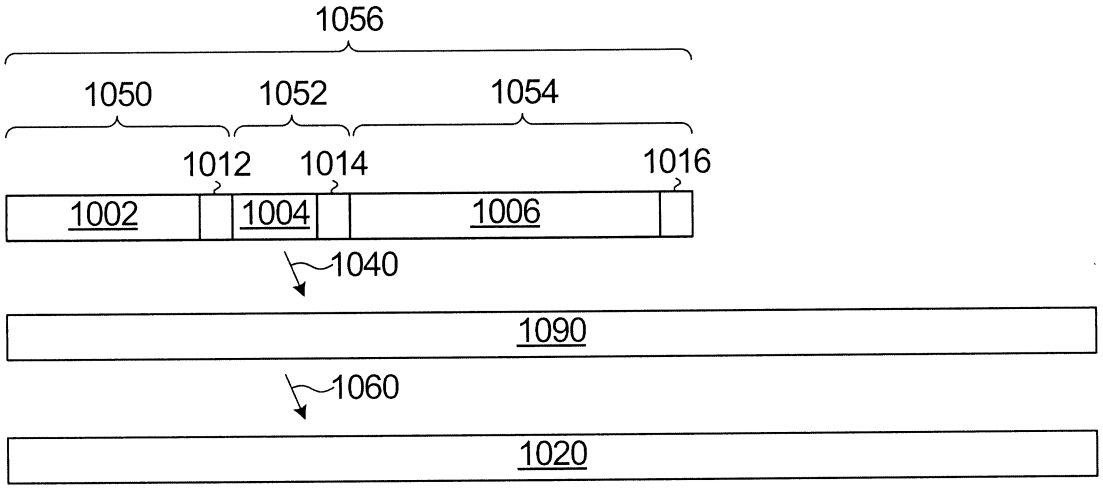

1056
1050
1052
1054
1012
1014
1016
1002
1004
1006
1040
1090
1060
1020

FIG. 10

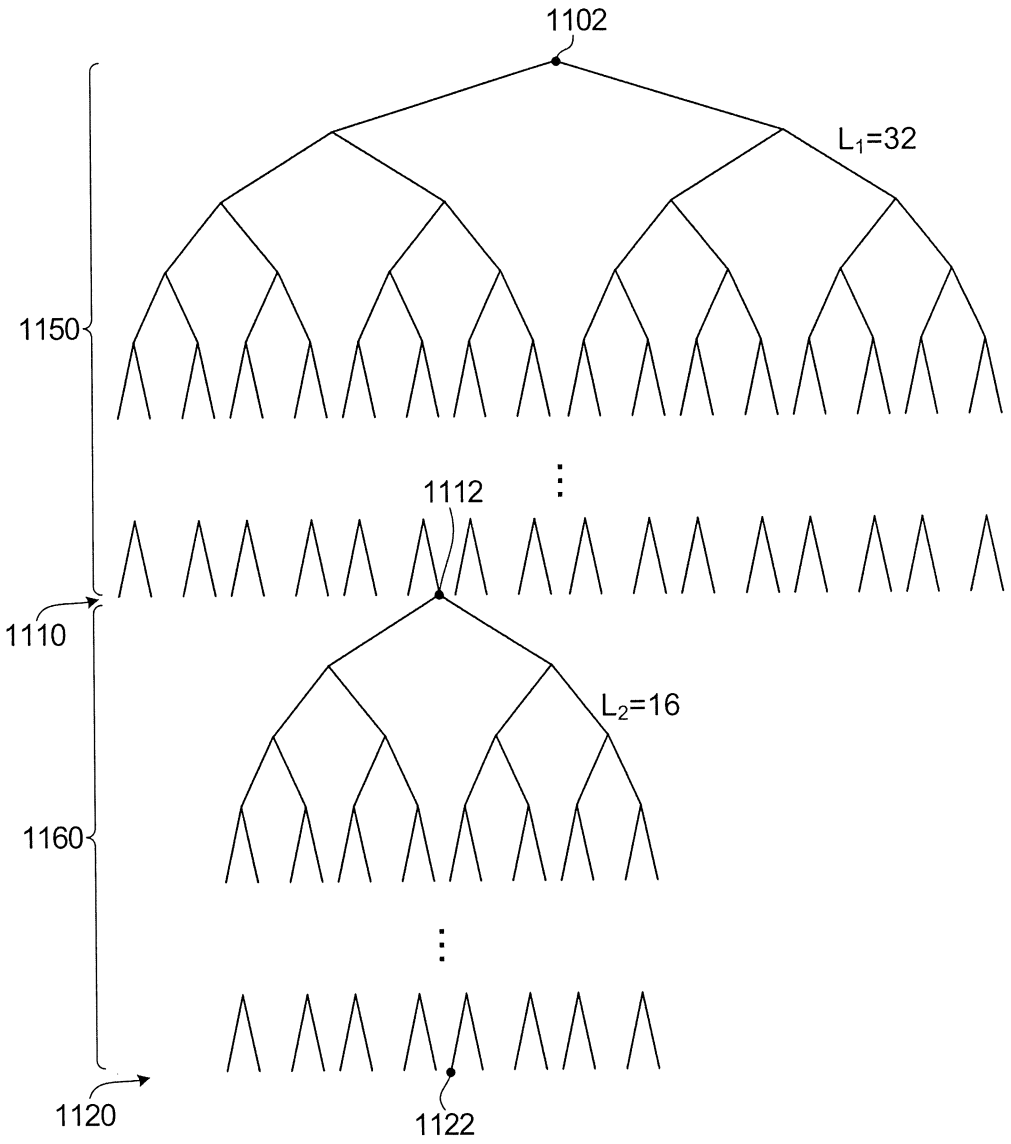


FIG. 11

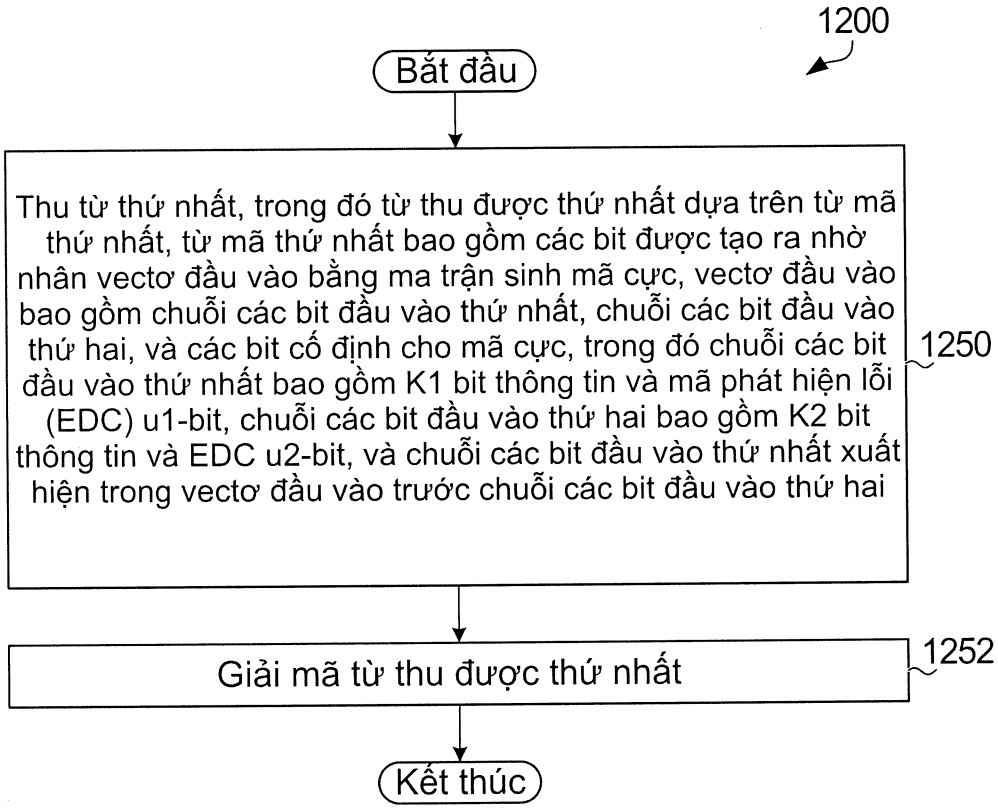

1200
Bắt đầu
Thu từ thứ nhất, trong đó từ thu được thứ nhất dựa trên từ mã thứ nhất, từ mã thứ nhất bao gồm các bit được tạo ra nhờ nhân vectơ đầu vào bằng ma trận sinh mã cực, vectơ đầu vào bao gồm chuỗi các bit đầu vào thứ nhất, chuỗi các bit đầu vào thứ hai, và các bit cố định cho mã cực, trong đó chuỗi các bit đầu vào thứ nhất bao gồm K1 bit thông tin và mã phát hiện lỗi (EDC) u1-bit, chuỗi các bit đầu vào thứ hai bao gồm K2 bit thông tin và EDC u2-bit, và chuỗi các bit đầu vào thứ nhất xuất hiện trong vectơ đầu vào trước chuỗi các bit đầu vào thứ hai
1250
Giải mã từ thu được thứ nhất
1252
Kết thúc

FIG. 12A

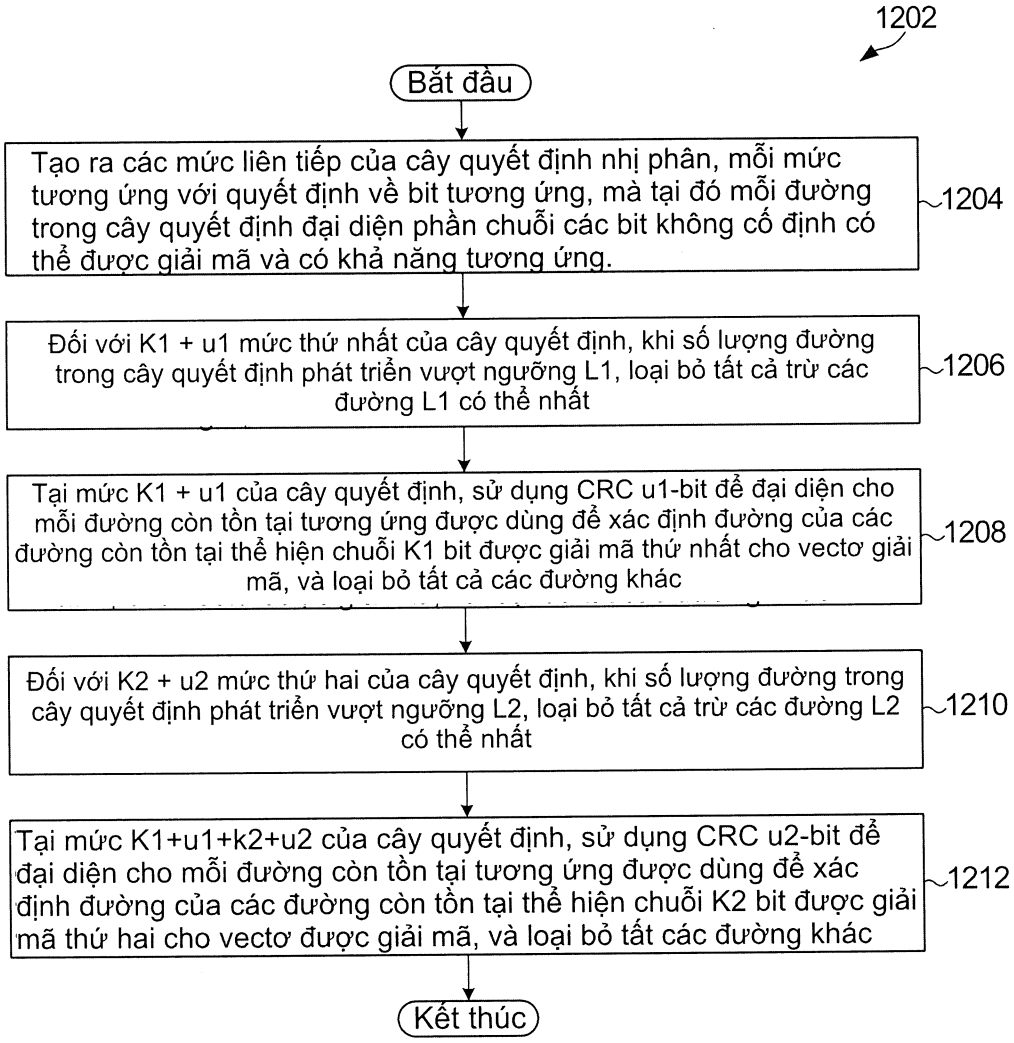

1202
Bắt đầu
Tạo ra các mức liên tiếp của cây quyết định nhị phân, mỗi mức tương ứng với quyết định về bit tương ứng, mà tại đó mỗi đường trong cây quyết định đại diện phần chuỗi các bit không cố định có thể được giải mã và có khả năng tương ứng.
1204
Đối với K1 + u1 mức thứ nhất của cây quyết định, khi số lượng đường trong cây quyết định phát triển vượt ngưỡng L1, loại bỏ tất cả trừ các đường L1 có thể nhất
1206
Tại mức K1 + u1 của cây quyết định, sử dụng CRC u1-bit để đại diện cho mỗi đường còn tồn tại tương ứng được dùng để xác định đường của các đường còn tồn tại thể hiện chuỗi K1 bit được giải mã thứ nhất cho vectơ giải mã, và loại bỏ tất cả các đường khác
1208
Đối với K2 + u2 mức thứ hai của cây quyết định, khi số lượng đường trong cây quyết định phát triển vượt ngưỡng L2, loại bỏ tất cả trừ các đường L2 có thể nhất
1210
Tại mức K1+u1+k2+u2 của cây quyết định, sử dụng CRC u2-bit để đại diện cho mỗi đường còn tồn tại tương ứng được dùng để xác định đường của các đường còn tồn tại thể hiện chuỗi K2 bit được giải mã thứ hai cho vectơ được giải mã, và loại bỏ tất các đường khác
1212
Kết thúc

FIG. 12B

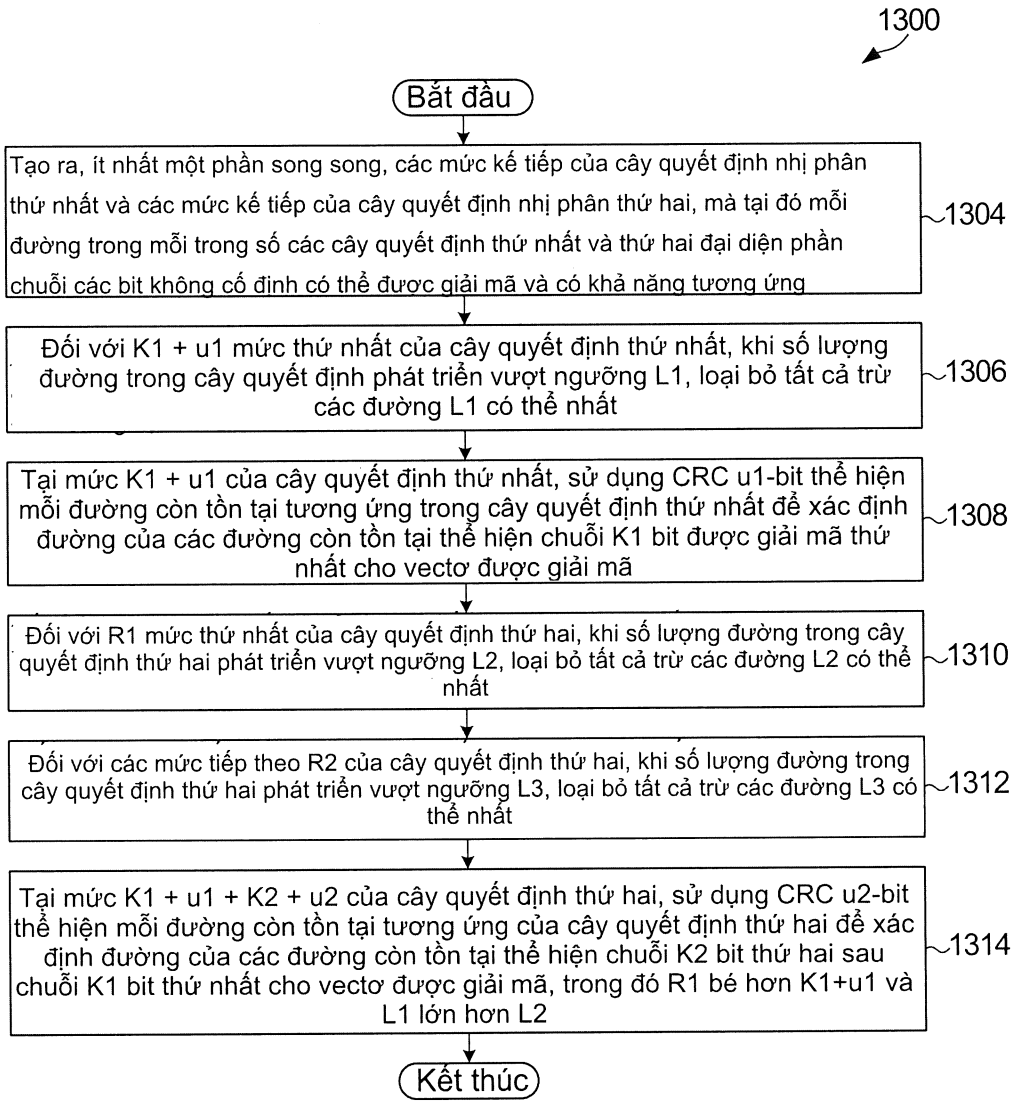

1300
Bắt đầu
Tạo ra, ít nhất một phần song song, các mức kế tiếp của cây quyết định nhị phân thứ nhất và các mức kế tiếp của cây quyết định nhị phân thứ hai, mà tại đó mỗi đường trong mỗi trong số các cây quyết định thứ nhất và thứ hai đại diện phần chuỗi các bit không cố định có thể được giải mã và có khả năng tương ứng
1304
Đối với K1 + u1 mức thứ nhất của cây quyết định thứ nhất, khi số lượng đường trong cây quyết định phát triển vượt ngưỡng L1, loại bỏ tất cả trừ các đường L1 có thể nhất
1306
Tại mức K1 + u1 của cây quyết định thứ nhất, sử dụng CRC u1-bit thể hiện mỗi đường còn tồn tại tương ứng trong cây quyết định thứ nhất để xác định đường của các đường còn tồn tại thể hiện chuỗi K1 bit được giải mã thứ nhất cho vectơ được giải mã
1308
Đối với R1 mức thứ nhất của cây quyết định thứ hai, khi số lượng đường trong cây quyết định thứ hai phát triển vượt ngưỡng L2, loại bỏ tất cả trừ các đường L2 có thể nhất
1310
Đối với các mức tiếp theo R2 của cây quyết định thứ hai, khi số lượng đường trong cây quyết định thứ hai phát triển vượt ngưỡng L3, loại bỏ tất cả trừ các đường L3 có thể nhất
1312
Tại mức K1 + u1 + K2 + u2 của cây quyết định thứ hai, sử dụng CRC u2-bit thể hiện mỗi đường còn tồn tại tương ứng của cây quyết định thứ hai để xác định đường của các đường còn tồn tại thể hiện chuỗi K2 bit thứ hai sau chuỗi K1 bit thứ nhất cho vectơ được giải mã, trong đó R1 bé hơn K1+u1 và L1 lớn hơn L2
1314
Kết thúc

FIG. 13

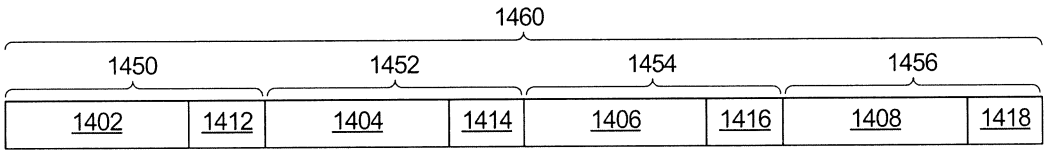


FIG. 14A

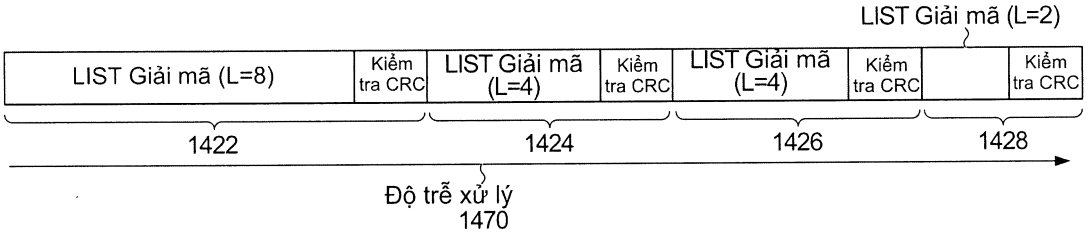


FIG. 14B

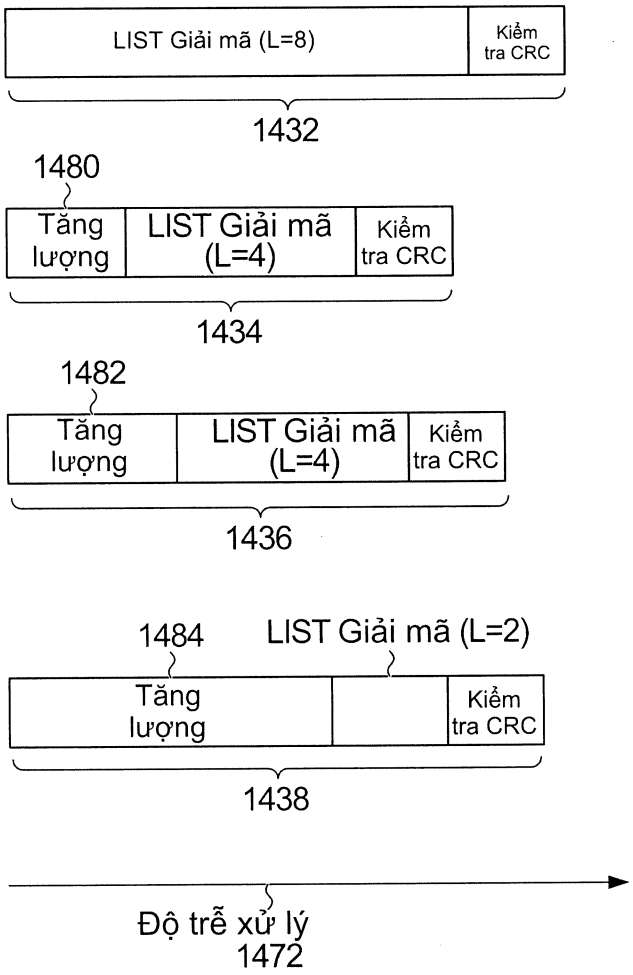

LIST Giải mã (L=8)
Kiểm tra CRC
1432
1480
Tăng lượng
LIST Giải mã (L=4)
Kiểm tra CRC
1434
1482
Tăng lượng
LIST Giải mã (L=4)
Kiểm tra CRC
1436
1484
LIST Giải mã (L=2)
Tăng lượng
Kiểm tra CRC
1438
Độ trễ xử lý
1472

FIG. 14C

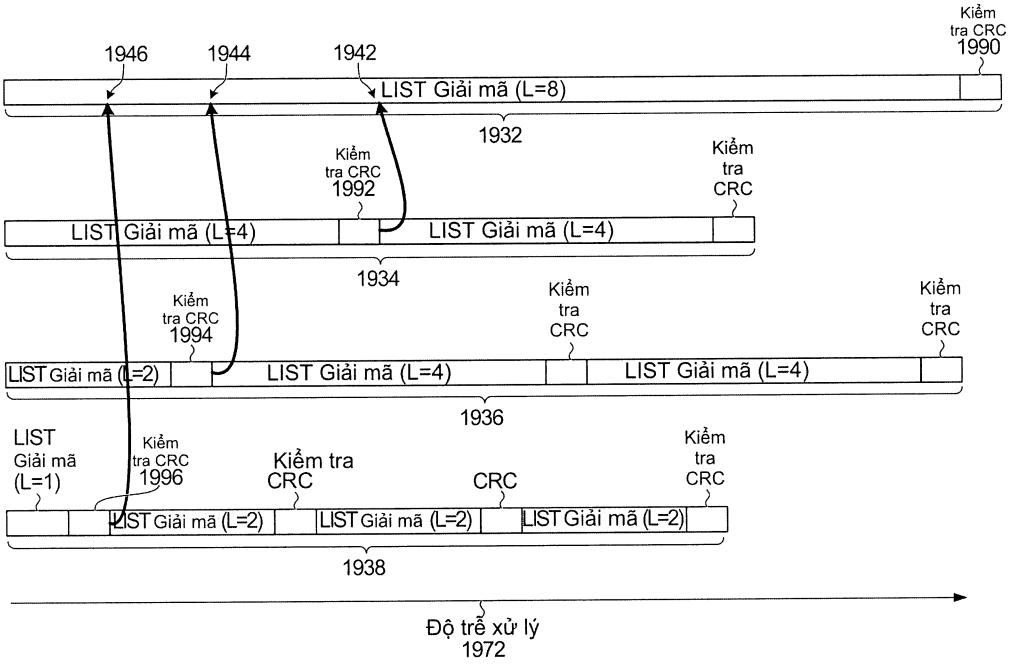

1946
1944
1942
Kiểm tra CRC 1990
LIST Giải mã (L=8)
1932
Kiểm tra CRC 1992
Kiểm tra CRC
LIST Giải mã (L=4)
LIST Giải mã (L=4)
1934
Kiểm tra CRC 1994
Kiểm tra CRC
Kiểm tra CRC
LIST Giải mã (L=2)
LIST Giải mã (L=4)
LIST Giải mã (L=4)
1936
LIST Giải mã (L=1)
Kiểm tra CRC 1996
Kiểm tra CRC
CRC
Kiểm tra CRC
LIST Giải mã (L=2)
LIST Giải mã (L=2)
LIST Giải mã (L=2)
1938
Độ trễ xử lý 1972

FIG. 14D

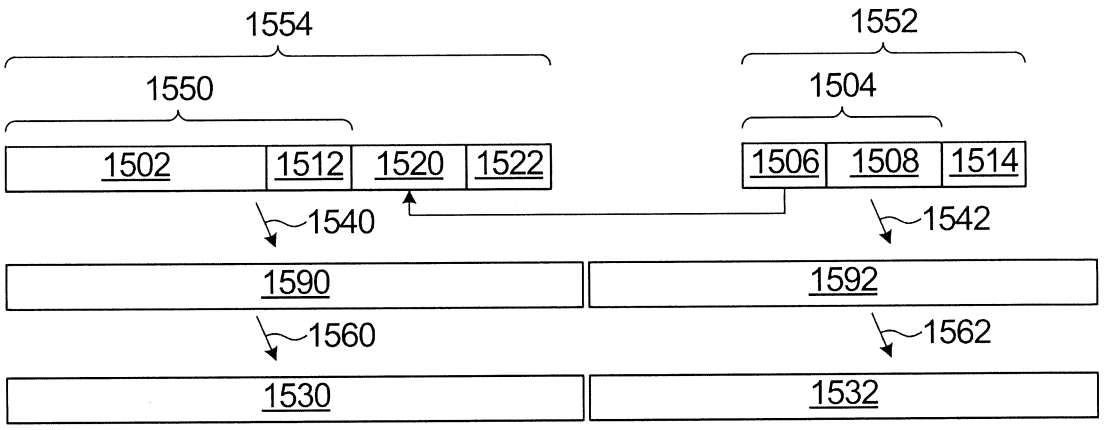

1554
1552
1550
1504
1502
1512
1520
1522
1506
1508
1514
1540
1542
1590
1592
1560
1562
1530
1532

FIG. 15

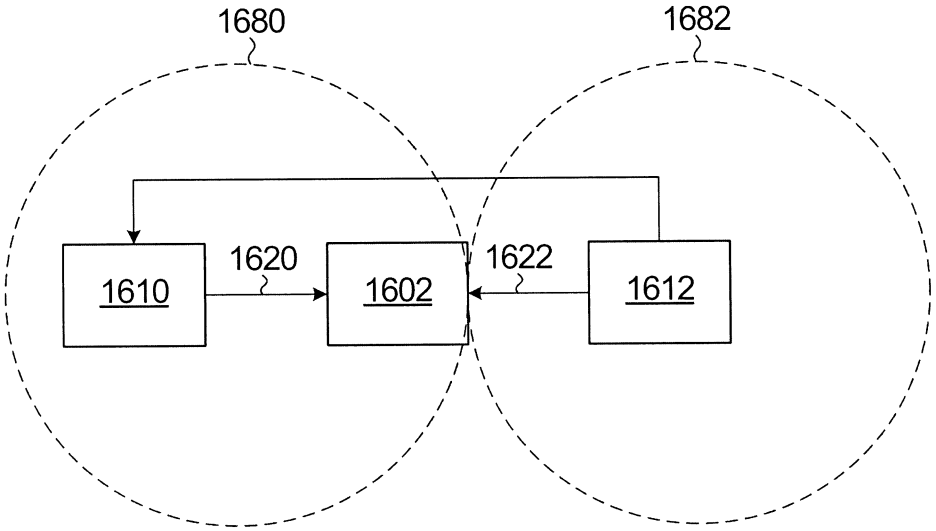

1680
1682
1610
1620
1602
1622
1612

FIG. 16

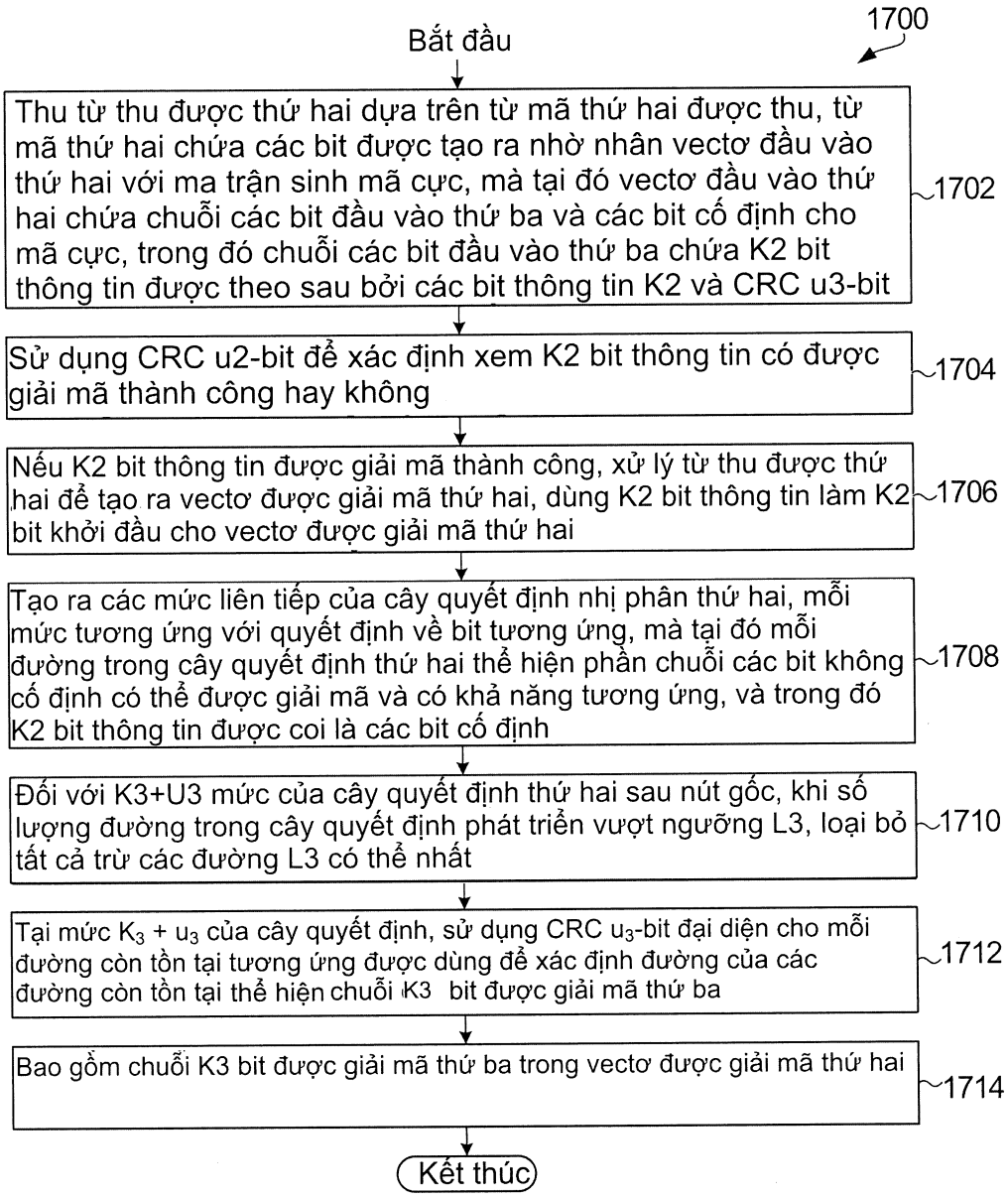

1700
Bắt đầu
Thu từ thu được thứ hai dựa trên từ mã thứ hai được thu, từ mã thứ hai chứa các bit được tạo ra nhờ nhân vectơ đầu vào thứ hai với ma trận sinh mã cực, mà tại đó vectơ đầu vào thứ hai chứa chuỗi các bit đầu vào thứ ba và các bit cố định cho mã cực, trong đó chuỗi các bit đầu vào thứ ba chứa K2 bit thông tin được theo sau bởi các bit thông tin K2 và CRC u3-bit
1702
Sử dụng CRC u2-bit để xác định xem K2 bit thông tin có được giải mã thành công hay không
1704
Nếu K2 bit thông tin được giải mã thành công, xử lý từ thu được thứ hai để tạo ra vectơ được giải mã thứ hai, dùng K2 bit thông tin làm K2 bit khởi đầu cho vectơ được giải mã thứ hai
1706
Tạo ra các mức liên tiếp của cây quyết định nhị phân thứ hai, mỗi mức tương ứng với quyết định về bit tương ứng, mà tại đó mỗi đường trong cây quyết định thứ hai thể hiện phần chuỗi các bit không cố định có thể được giải mã và có khả năng tương ứng, và trong đó K2 bit thông tin được coi là các bit cố định
1708
Đối với K3+U3 mức của cây quyết định thứ hai sau nút gốc, khi số lượng đường trong cây quyết định phát triển vượt ngưỡng L3, loại bỏ tất cả trừ các đường L3 có thể nhất
1710
Tại mức $K_3 + u_3$ của cây quyết định, sử dụng CRC u_3-bit đại diện cho mỗi đường còn tồn tại tương ứng được dùng để xác định đường của các đường còn tồn tại thể hiện chuỗi K3 bit được giải mã thứ ba
1712
Bao gồm chuỗi K3 bit được giải mã thứ ba trong vectơ được giải mã thứ hai
1714
Kết thúc

FIG. 17

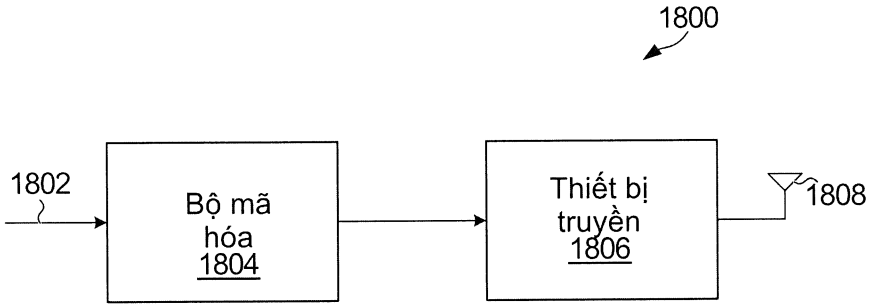


FIG. 18A

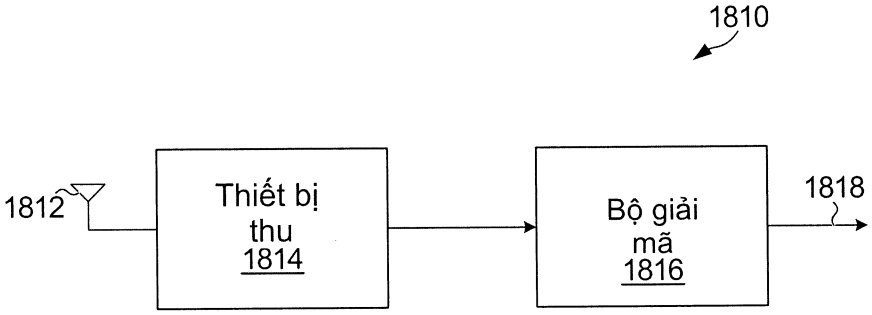


FIG. 18B