

Thiết kế và tối ưu thực thi bộ giải mã cầu trên phần cứng chuyên dụng cho hệ thống thông tin vô tuyến MIMO

Nguyễn Minh Thường¹, Trần Xuân Nam²,
Nguyễn Đức Thắng², Vũ Tiến Anh², Trịnh Quang Kiên^{2*}

¹Viện KH-CN quân sự;

²Học viện Kỹ thuật quân sự .

*Email: kien.trinh@lqdtu.edu.vn

Nhận bài: 01/4/2022; Hoàn thiện: 08/5/2022; Chấp nhận đăng: 17/5/2022; Xuất bản: 28/6/2022.

DOI: <https://doi.org/10.54939/1859-1043.j.mst.80.2022.80-91>

TÓM TẮT

Bộ tách tín hiệu hợp lý cực đại (Maximum likelihood – ML) có thể đạt được tỷ lệ lỗi bit tốt nhất nhưng đòi hỏi độ phức tạp tính toán rất cao. Điều này làm cho thuật toán này không được áp dụng trong thực tế. Do đó, nhiều kiến trúc bộ giải mã đã được đề xuất để khắc phục độ phức tạp cao của bộ tách tín hiệu ML. Trong số đó, thuật toán giải mã cầu (Sphere decoder – SD) là một trong những cách tiếp cận hứa hẹn nhất cung cấp chất lượng gần như ML với khối lượng tính toán hợp lý. Bài báo này đề xuất một cách tiếp cận hiệu quả và có tính khả dụng cao cho thiết kế bộ giải mã cầu trên phần cứng có thể cấu hình lại (FPGA). Thiết kế được đánh giá là mang lại giá trị tiệm cận về chất lượng của phương pháp ước lượng hợp lý cực đại (ML) nhưng với độ phức tạp tính toán giảm đáng kể.

Từ khóa: MIMO; FPGA; Ghép không gian (SM); Bộ giải mã cầu (SD); Hợp lệ cực đại (ML).

1. MỞ ĐẦU

Trong các nghiên cứu gần đây, các hệ thống truyền thông MIMO đã được chứng minh có khả năng tăng dung lượng kênh và nâng cao độ tin cậy của kênh truyền vô tuyến. Ý tưởng chính của hệ thống MIMO là khả năng biến hiệu ứng lan truyền đa đường, vốn là một trở ngại trong giao tiếp vô tuyến thông thường, thành một lợi ích cho hệ thống [1]. Hiện nay, các kỹ thuật MIMO đã được chấp nhận như một tiêu chuẩn giao tiếp vô tuyến cho các hệ thống truyền thông không dây hiện đại như hệ thống thông tin di động 4G LTE, 5G, Wi-Fi, WiMAX, cho phép tăng thông lượng truyền dẫn bằng cách thực hiện các sửa đổi trong lớp PHY và MAC [2].

Với sự gia tăng của số lượng các ăng-ten phát và bậc điều chế, độ phức tạp của bộ giải mã hợp lệ cực đại tối ưu ML (Maximum Likelihood) tăng lên theo cấp số nhân khiến nó không còn phù hợp để triển khai phần cứng thời gian thực, đặc biệt là đối với các hệ thống MIMO cỡ lớn (Massive MIMO) [1, 3]. Giải mã cầu (Sphere Decoder – SD) là một trong các giải pháp tiếp cận chính để giảm bớt độ phức tạp tách tín hiệu trong các hệ thống ghép kênh phân chia theo không gian (Spatial Division Multiplexing – SDM) và đa truy cập phân chia theo không gian (Spatial Division Multiple Access – SDMA) trong khi vẫn duy trì được các đường cong BER của SD tiệm cận với đường cong BER của bộ giải mã ML. Giải mã cầu ban đầu được giới thiệu vào năm 1985 bởi Finke và Pohst [4], nó như là một kỹ thuật để giảm khối lượng tính toán khi tìm véc tơ có độ dài ngắn trong một lưới. SD lần đầu tiên được sử dụng trong truyền thông vào năm 1993 để giải mã mềm mã Golay bởi Viterbi và Bigleri [5]. Sau đó, một số lượng lớn các nghiên cứu về thuật toán giải mã cầu đã được nghiên cứu với mục đích đảm bảo các hệ số phẩm chất của bộ giải mã SD tiến gần tới chất lượng của bộ giải mã ML với độ phức tạp tính toán giảm đáng kể [4, 6-8].

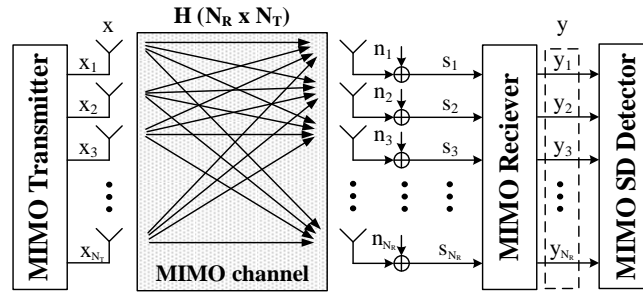
Một số các sơ đồ cây tìm kiếm được sử dụng trong giải mã cầu có thể kể đến như tìm kiếm cây theo chiều sâu, tìm kiếm cây theo chiều rộng [9-11] tìm kiếm theo chiều sâu kết hợp giữa chiều rộng và chiều sâu [12]. Một phương pháp khác là giảm độ phức tạp cho tìm kiếm theo độ sâu là thuật toán giải mã Schnorr-Euchner (SE) [13, 14]. Thuật toán này liệt kê và sắp xếp các

khoảng cách Euclid của chúng theo thứ tự tăng dần để tìm kiếm ưu tiên. Quá trình này có thể dừng lại khi giải pháp thỏa đáng được tìm thấy trong siêu cầu, do đó tránh tăng thêm khối lượng tính toán [5, 14]. Tuy nhiên, độ phức tạp tính toán của loại phương pháp giải mã này phụ thuộc vào giá trị được chọn của bán kính cầu và mức độ nhiễu cũng như điều kiện kênh truyền. Những yếu tố này dẫn đến thông lượng và độ trễ không thể đoán trước được và không phù hợp cho việc triển khai phần cứng. Do đó, trong nghiên cứu này, chúng tôi thực thi thiết kế đề xuất một kiến trúc cho bộ giải mã cầu để đánh giá chất lượng hệ thống và tài nguyên chiếm dụng. Tối ưu thiết kế cho hệ thống đảm bảo cân bằng giữa các tài nguyên sử dụng và hệ số phẩm chất BER, thông lượng hệ thống để mang lại hiệu quả thiết kế tối ưu. Với kiến trúc cho bộ giải mã cầu đề xuất, hệ số phẩm chất BER và thông lượng hệ thống có thể được tối ưu và điều chỉnh thông qua tham số khởi tạo cầu. Đóng góp của nghiên cứu là đã đề xuất và thực thi được một kiến trúc cho bộ giải mã cầu SD trên nền tảng phần cứng FPGA có thể đạt được tỉ lệ lỗi bit xấp xỉ của bộ tách ML với độ phức tạp phù hợp có thể được cài đặt trên các linh kiện có tài nguyên tương đương với FPGA Kintex 7 XC7k325 và Virtex UltraScale+ xcvy7p-flva2104-3-e-EVAL của hãng Xilinx.

Phần còn lại của bài báo được tổ chức như sau: Phần 2 trình bày mô hình hệ thống chung và định dạng tín hiệu tương ứng. Phần 3 trình bày thực thi trên phần cứng chuyên dụng thuật toán giải mã cầu. Phần 4 kết luận bài báo.

2. NỘI DUNG CẦN GIẢI QUYẾT

2.1. Mô hình hệ thống



Hình 1. Sơ đồ khối của các hệ thống MIMO tổng quát.

Xem xét một hệ thống MIMO với N_T anten phát và N_R anten thu như trong hình 1. Kênh MIMO được đặc trưng bởi ma trận kênh phức $\mathbf{H} = (h_{ij})^{N_R \times N_T} \in \mathbb{C}^{N_R \times N_T}$, các phần tử của \mathbf{H} có phân bố với phương sai đơn vị và kỳ vọng bằng không. Các thông số này mô tả độ suy hao và lệch pha đối với mỗi đường dẫn từ một ăng-ten phát đến một ăng-ten thu; chúng được giả định là đã biết trước một cách hoàn hảo (có thể thông qua giai đoạn ước lượng kênh). Đối với quá trình truyền dẫn, các phần tử x_i của véc-tơ tín hiệu phức $\mathbf{x} = (x_i)^{N_T \times 1} \in \Omega \subset \mathbb{C}^{N_T \times 1}$ được gửi đồng thời qua N_T anten phát, trong đó Ω là tập hợp của chòm sao điều chế tín hiệu. Do đó, véc-tơ tín hiệu phức nhận được $\mathbf{y} = (y_i)^{N_R \times 1} \in \mathbb{C}^{N_R \times 1}$ có thể được biểu thị bằng công thức:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \quad (1)$$

trong đó, $\mathbf{n} = (n_i)^{N_R \times 1} \sim CN(0, \sigma^2 \mathbf{I})$ là một véc-tơ tạp âm Gauss phức trắng cộng tính (AWGN). Bộ giải mã ML thực hiện tìm kiếm theo phương pháp vét cạn tất cả các véc-tơ ký hiệu có thể có trong tập $\mathbf{S}^{N_T \times 1}$ để thu được véc-tơ phát với cự ly Euclid nhỏ nhất:

$$\hat{\mathbf{x}}_{ML} = \arg \min_{\mathbf{x} \in \Omega} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \quad (2)$$

Do đó, độ phức tạp tính toán của bộ giải mã ML tăng lên theo hàm mũ của bậc điều chế tín hiệu M và số lượng các ăng-ten thu trong hệ thống. Bộ giải mã cầu SD đơn giản hóa bộ giải mã ML bằng việc hạn chế các điểm tìm kiếm của SD để giảm độ phức tạp tính toán theo hướng chỉ so sánh những điểm tín hiệu nằm bên trong siêu cầu với bán kính xác định trước được hình thành xung quanh véc-tơ tín hiệu nhận được, tức là:

$$\hat{\mathbf{x}}_{SD} = \arg \min_{\mathbf{x} \in \mathcal{S}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \quad (3)$$

trong đó, $\mathcal{S} \subset \mathbb{C}^{N_T \times 1} : \|\mathbf{y} - \mathbf{H}\mathbf{x}\| \leq r_{sph}$ là tập hợp tất cả các điểm nằm trên lưới thỏa mãn khoảng cách của nó tới \mathbf{y} luôn nhỏ hơn bán kính r_{sph} của siêu cầu. Việc chọn giá trị của r_{sph} về cơ bản là quan trọng có ý nghĩa quyết định trực tiếp đến độ phức tạp tính toán của SD và hiệu suất BER. Để giảm hơn nữa số lượng tính toán của SD, phương trình (3) có thể được chuyển đổi thành một dạng khác tương đương nhờ biến đổi QRD cho ma trận kênh \mathbf{H} , khi đó $\mathbf{H} = \mathbf{Q}\mathbf{R}$ trong đó ma trận \mathbf{Q} là ma trận đơn nhất có kích thước là $N_R \times N_R$ và $\mathbf{Q}\mathbf{Q}^H = \mathbf{I}$ trong khi \mathbf{R} là ma trận tam giác trên $N_R \times N_T$. Thay thế \mathbf{H} bằng $\mathbf{Q}\mathbf{R}$ và sau khi biến đổi, biểu thức (1) trở thành:

$$\tilde{\mathbf{y}} = \mathbf{R}\mathbf{x} + \mathbf{Q}^H \mathbf{n} \text{ với } \tilde{\mathbf{y}} = \mathbf{Q}^H \mathbf{y} \quad (4)$$

Lưu ý rằng $\mathbf{Q}^H \mathbf{n}$ có cùng thống kê với \mathbf{n} , nên phương trình (3) được biến đổi về dạng tương đương:

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathcal{S}} \|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{x}\|^2 \text{ khi } \hat{\mathbf{y}} = \mathbf{R}\mathbf{x} \quad (5)$$

Phương trình (5) có thể được tính toán thông qua hàm giá trị như sau:

$$D(\tilde{\mathbf{y}}, \hat{\mathbf{y}}) = \|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{x}\|^2 \leq r_{sph}^2 \quad (6)$$

Vì ma trận \mathbf{R} là tam giác trên nên hàm giá trị $D(\tilde{\mathbf{y}}, \hat{\mathbf{y}})$ cũng là khoảng cách Euclide một phần có thể được tính toán đệ quy từ một ăng ten phát này đến một ăng ten phát khác:

$$D_m(\tilde{\mathbf{y}}, \hat{\mathbf{y}}) \triangleq \sum_{i=m}^{N_R} (\tilde{y}_i - \sum_j^{N_T} R_{ij} x_j)^2 \quad (7)$$

$$D(\tilde{\mathbf{y}}, \hat{\mathbf{y}}) = D_1(\tilde{\mathbf{y}}, \hat{\mathbf{y}}) \quad (8)$$

$$D_{m-1}(\tilde{\mathbf{y}}, \hat{\mathbf{y}}) = D_m(\tilde{\mathbf{y}}, \hat{\mathbf{y}}) + \left(\tilde{y}_{m-1} - \sum_{i=m-1}^{N_T} R_{m-1,i} x_i \right)^2 \quad (9)$$

với \tilde{y}_{m-1} là phần tử thứ $(m-1)$ của vector tín hiệu thu được sau khi nó được nhân với \mathbf{Q}^H , $R_{i,j}$ là phần tử của ma trận \mathbf{R} thuộc hàng thứ i và cột thứ j và hàm giá trị $D_m(\tilde{\mathbf{y}}, \hat{\mathbf{y}})$ là khoảng cách Euclid một phần của symbol \mathbf{x} tại mức tìm kiếm m . Đối với tất cả các véc-tơ ký hiệu phát thỏa mãn $x_j \in \left\{ \mathbf{x} \in \mathcal{S}^{N_T \times 1} \subset \mathbb{C}^{N_T \times 1} : \|\mathbf{R}\mathbf{x} - \mathbf{y}\| \leq r_{sph} \right\}$, khởi tạo $D_{N_R+1}(\tilde{\mathbf{y}}, \hat{\mathbf{y}}) = 0$

$$D_{m-1}(\tilde{\mathbf{y}}, \hat{\mathbf{y}}) \leq r_{sph_{m-1}}^2 - D_m(\tilde{\mathbf{y}}, \hat{\mathbf{y}})$$

trong đó $r_{sph_{m-1}}^2 = r_{sph}^2 - \sum_{i=m+1}^{N_R} D_i(\tilde{\mathbf{y}}, \hat{\mathbf{y}})$ (10)

Đối với việc triển khai phần cứng, việc thực hiện phân rã giá trị thực (RVD) của H cũng hiệu

quả, điều này giúp đơn giản hóa việc tính toán khoảng cách Euclid. Phép phân tích giá trị thực tách phương trình kênh (1) thành một biểu diễn giá trị thực mới như sau [15]:

$$\begin{bmatrix} \Re(\mathbf{y}) \\ \Im(\mathbf{y}) \end{bmatrix} = \begin{bmatrix} \Re(\mathbf{H}) & -\Im(\mathbf{H}) \\ \Im(\mathbf{H}) & \Re(\mathbf{H}) \end{bmatrix} \begin{bmatrix} \Re(\mathbf{x}) \\ \Im(\mathbf{x}) \end{bmatrix} + \begin{bmatrix} \Re(\mathbf{n}) \\ \Im(\mathbf{n}) \end{bmatrix} \quad (11)$$

với $\Re(\cdot)$, $\Im(\cdot)$ tương ứng biểu diễn phần thực và phần ảo của véc-tơ phức. Hơn nữa, chúng ta có thể giải phương trình (1) thông qua phương trình (11) với các bước như trên, và với việc biến đổi tập hợp chòm sao phức thành tập số nguyên như sau:

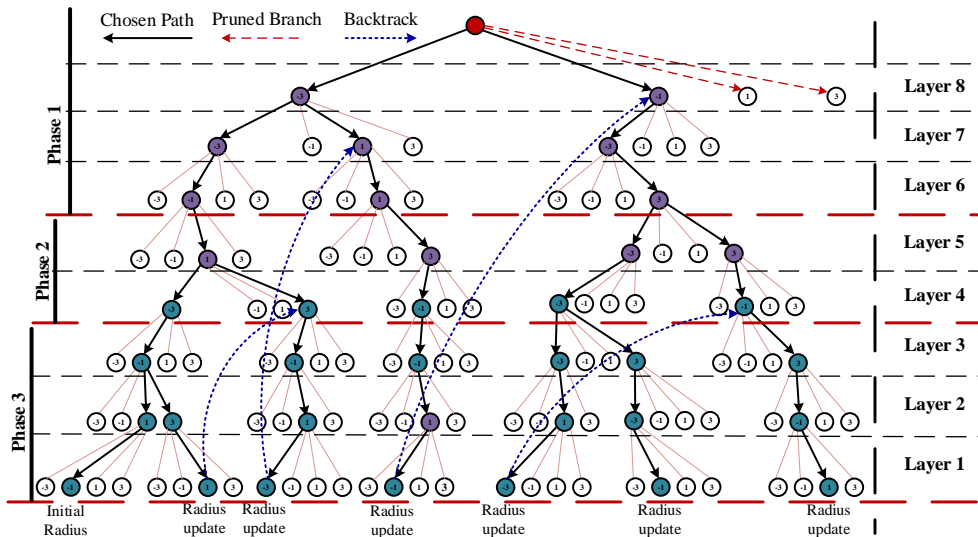
$$S_r = \{-\sqrt{M} + 1, \dots, \sqrt{M} + 1\} \quad (12)$$

trong đó, M là bậc điều chế. Sau đó, QRD có thể được thực hiện nói chung dựa trên phương trình kênh tăng cường trong (12). Kích thước của ma trận kênh tương đương (\mathbf{H}), ma trận đơn nhất (\mathbf{Q}) và ma trận tam giác trên (\mathbf{R}) lần lượt được biến đổi thành $2N_R \times 2N_T$, $2N_R \times 2N_R$, $2N_T \times 2N_T$. Số lượng cấp độ tìm kiếm cây thay đổi thành $2N_R$. Trong phần sau, sẽ trình bày thực thi thuật toán giải mã cầu trên phần cứng chuyên dụng.

2.2. Thực thi trên phần cứng chuyên dụng thuật toán giải mã cầu

Trong phần này, chúng tôi mô tả việc triển khai VHDL của hệ thống 4x4 MIMO với điều chế 16-QAM sử dụng thuật toán giải mã cầu. Toàn bộ thiết kế được mô tả dùng ngôn ngữ VHDL và được mô phỏng về mặt chức năng dùng phần mềm Xilinx Vivado 2016.4. Dữ liệu đầu vào được tạo ra trên Matlab, với mức độ ngẫu nhiên gần như lý tưởng (tỷ lệ bit 0/1 là 50/50). Sau đó, các tín hiệu 16-QAM được ánh xạ và có thể được cộng thêm tạp âm nhiễu trắng AWGN. Bộ dữ liệu gồm một triệu mẫu đầu vào cho phép này cũng giúp ta đánh giá sát thực nhất về chức năng và chất lượng của hệ thống.

2.2.1. Thiết kế các khối tính toán cơ bản trong khối giải mã cầu



Hình 2. Cây giải mã đối với thuật toán giải mã cầu.

Toàn bộ chức năng tính toán giải mã cầu được thực hiện bằng một khối chức năng duy nhất, gọi là CELL với sơ đồ trình bày trên hình 3, hình 4. Thiết kế như vậy đảm bảo được tính đơn giản, thống nhất cũng như khả năng mở rộng của thiết kế về sau. Tất cả các lớp tìm kiếm (gọi tắt là lớp) của quá trình giải mã SD sẽ được thực hiện lần lượt bằng CELL. Toàn bộ quá trình tính toán sẽ được chia thành ba pha chính. Pha thứ nhất sẽ tính 3 lớp đầu tiên, pha tiếp theo sẽ tính

tiếp 2 lớp và pha cuối cùng sẽ tính 3 lớp còn lại (hình 2). Giữa các pha tính toán, dữ liệu trung gian sẽ được lưu tại các thanh ghi để chờ xử lý lần lượt. Bằng cách tính toán này mà tài nguyên được sử dụng một cách hiệu quả hơn trong khi vẫn bảo đảm về mặt chức năng.

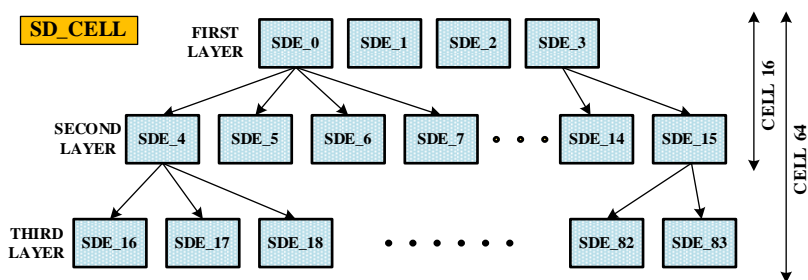
Khối tính toán cơ bản CELL

Phần tử tính toán cơ bản SD-Element (SDE) thực hiện tính giá trị bán kính cầu mới, đó là phép tính cơ bản nhất của thuật toán giải mã cầu. Theo lý thuyết đã trình bày ở phần trước ta có công thức tính được trình bày phía dưới. Phần tử tính toán cơ bản phải thực hiện phép tính này bằng tài nguyên phần cứng có sẵn trên FPGA.

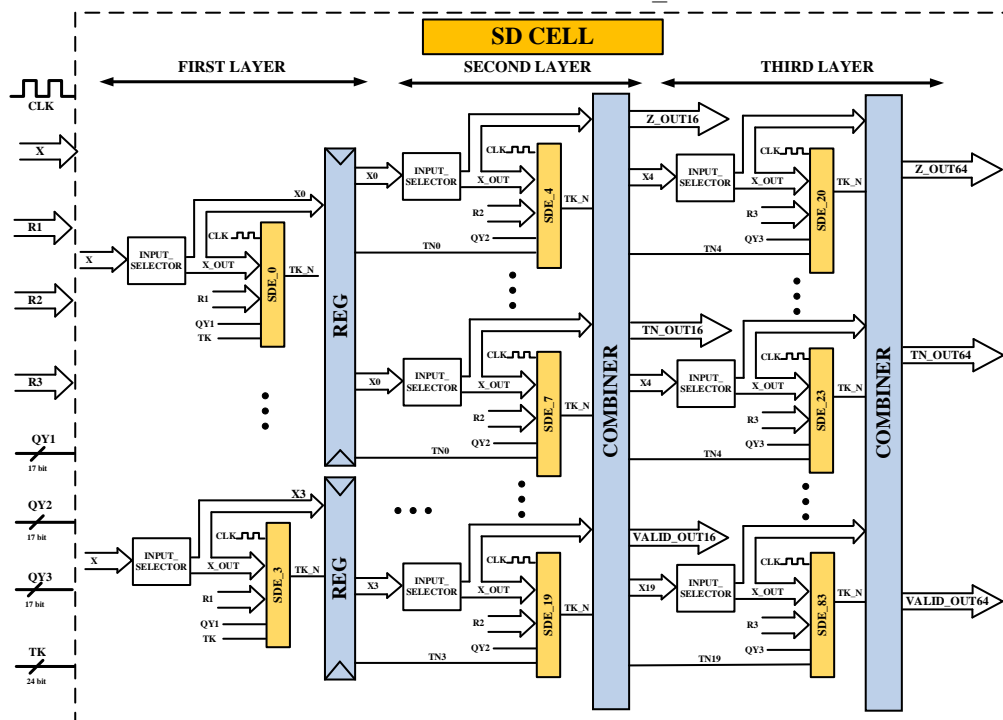
$$r_{m-1}^2 = r_m^2 - |e_m(\hat{x}_m)|^2 \tag{13}$$

Nhằm mục đích dễ dàng áp dụng phép tính trên phần cứng (tránh các phép toán khai căn) và thống nhất tên gọi của các tập dữ liệu đầu vào, công thức trên được biến đổi tương đương như sau:

$$r_{m-1}^2 = r_m^2 - \left(\sum_{i=1}^8 x_i R_{m,i} - (Q'y)_m \right)^2 \tag{14}$$



Hình 3. Sơ đồ khối của khối SD_CELL.



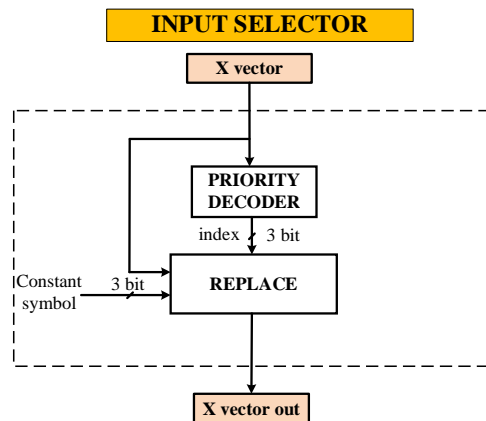
Hình 4. Cấu trúc phần cứng khối CELL.

trong đó $m = 1 \div 8$ là thứ tự các lớp. Để đảm nhiệm tính toán giải mã đồng thời cho từ 2 đến 3 lớp liên tiếp, theo sơ đồ giải mã hình cây chúng ta cần tổng cộng 84 khối SDE, 4 khối cho lớp đầu tiên, 16 khối cho lớp thứ hai và 64 khối cho lớp thứ 3 như trình bày trên hình 3. SD CELL là sự ghép nối các khối SDE lại với nhau theo đúng sơ đồ giải mã hình cây, trong đó sự tổ hợp các dữ liệu đầu ra của từng khối SDE tạo thành các luồng dữ liệu được sử dụng, đồng thời xác định các tổ hợp nghiệm đúng bằng cách xét các giá trị bán kính cầu mới từ các SDE.

Dữ liệu đầu vào của khối CELL là các tập dữ liệu cần thiết để tính toán cho ba lớp liên tiếp bao gồm: vector nghiệm ban đầu $\mathbf{x} = \{x_1, x_2, \dots, x_8\}$; ba vector $\mathbf{R}_m = \{R_{m1}, R_{m2}, \dots, R_{m8}\}$ nằm trong ma trận R và ba giá trị $Q'y_m$ tương ứng với ba lớp cần tính toán giải mã; giá trị bán kính cầu bình phương r_m^2 , trong trường hợp này chúng tôi ký hiệu là TK. Dữ liệu đầu ra của khối CELL là tập 16 bộ nghiệm cho tính toán 2 lớp và tập 64 bộ nghiệm cho tính toán 3 lớp. Trong đó mỗi bộ nghiệm bao gồm: vector nghiệm x , giá trị bán kính cầu mới bình phương r_{m-1}^2 và bit hợp lệ (*valid bit*) để xác nhận nghiệm đúng. Từng thành phần trong mỗi bộ nghiệm được tập hợp thành các tập dữ liệu lớn hơn. Cụ thể, véc-tơ nghiệm x của các bộ nghiệm được tập hợp thành ma trận hai chiều Z_OUT có kích thước $8 \times N$, trong đó N là số lượng véc-tơ \mathbf{x} chứa trong ma trận, mỗi cột tương ứng với một véc-tơ nghiệm \mathbf{x} . Các giá trị bán kính cầu mới bình phương r_{m-1}^2 chứa trong mảng một chiều TN_OUT kích thước $1 \times N$, trong đó, N tương ứng với số bộ nghiệm, tương tự như vậy các bit valid của mỗi bộ nghiệm tập hợp thành một vector $VALID_OUT$ có độ rộng N bit. Theo cách bố trí này, trong trường hợp tính toán cho hai lớp đầu ra của CELL sẽ là $Z_OUT16, TN_OUT16, VALID_OUT16$ ứng với 16 bộ nghiệm và với trường hợp tính toán cho ba lớp ta có $Z_OUT64, TN_OUT64, VALID_OUT64$.

Khối lựa chọn đầu vào INPUT_SELECTOR

Khối CELL thực hiện tính toán giải mã dựa trên hoạt động tính toán giá trị bán kính cầu mới của khối SDE. Như đã trình bày ở phần trước khối phần tử tính toán cơ bản SDE chỉ thực hiện chức năng tính toán giá trị bán kính cầu mới để đưa vào lớp tiếp theo, nó không phân biệt việc tính toán cho lớp nào trong toàn bộ quá trình giải mã. Bản chất quá trình giải mã cầu áp dụng trong thiết kế này là từ các tổ hợp nghiệm đã biết trước để tiến hành lựa chọn ra nghiệm đúng bằng cách thử tính khoảng cách Euclid của tổ hợp nghiệm đó. Do đó, ta cần một khối chọn nghiệm để xác định lớp cần tính toán và tổ hợp nghiệm đầu vào để đẩy vào SDE tính toán bán kính cầu mới. Chức năng này được thực hiện bởi khối INPUT_SELECTOR.



Hình 5. Cấu trúc của INPUT_SELECTOR.

Trong thiết kế này chúng tôi có quy ước rằng những lớp nào chưa được tính toán giải mã thì phần tử x_i trong véc-tơ nghiệm \mathbf{x} tương ứng được đặt bằng 0. Ví dụ, ta đang thực hiện tính toán tại lớp 6, véc-tơ nghiệm đưa đến sẽ có dạng là $\mathbf{x} = \{0,0,0,0,0,0,1,3\}$, dựa vào quy ước này, INPUT_SELECTOR sẽ phát hiện giá trị $x_i = 0$ đầu tiên khi quét x_i từ x_8 đến x_1 để xác định lớp

cần tính toán qua chỉ số i . Trong ví dụ này khối INPUT SELECTOR xác định được $x_6 = 0$ và xác định lớp cần tính toán là lớp 6. Tiếp theo khối INPUT SELECTOR sẽ gán giá trị nằm trong tập $\{\pm 1, \pm 3\}$ cho x_6 , cụ thể là giá trị nào tùy thuộc vào vị trí khối INPUT SELECTOR đi cùng với SDE nào trong sơ đồ hình cây. Nếu ở vị trí 0 theo hình 3 thì x_6 nhận giá trị -3 và véc-tơ nghiệm \mathbf{x} tại lớp 6 đưa đến khối SDE tương ứng và đưa đến tầng sau là $x_{16} = \{0,0,0,0,0, -3,1,3\}$. Cấu trúc của INPUT SELECTOR được trình bày ở hình 5.

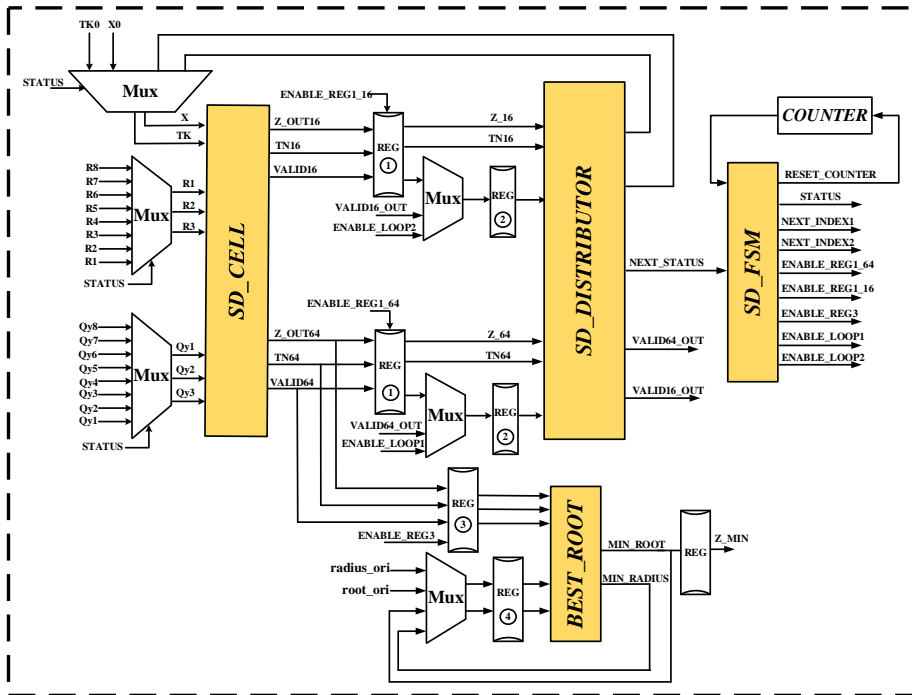
Sau khi nhận được véc-tơ nghiệm \mathbf{x} từ INPUT SELECTOR, khối SDE tính toán giá trị bán kính cầu mới của nghiệm \mathbf{x} theo lý thuyết giải mã. Véc-tơ nghiệm \mathbf{x} được xác nhận là nghiệm đúng khi khoảng cách Euclid của nó nhỏ hơn bán kính cầu ban đầu, điều này có nghĩa là bán kính cầu mới phải dương. Do vậy, việc xác định nghiệm đúng được thực hiện bằng cách xét dấu bán kính cầu mới, nếu dương thì bit VALID của nghiệm \mathbf{x} đặt bằng 1, ngược lại đặt bằng 0. Mỗi đơn vị tính toán bao gồm INPUT SELECTOR và SDE gọi là SDU (*Sphere Decoder Unit*), dữ liệu đầu ra của mỗi SDU là một bộ nghiệm bao gồm véc-tơ \mathbf{x} , r_{m-1}^2 và bit VALID. Số lượng SDU bằng số lượng SDE trong cấu trúc của CELL chi tiết đến từng lớp. Dữ liệu đầu ra của các khối SDU trong từng lớp được chốt vào thanh ghi trước khi đưa vào SDU của lớp tiếp theo. Ngoài ra tại lớp thứ hai và lớp thứ ba các dữ liệu này được kết hợp thành các dạng dữ liệu đã được xác định trước để đẩy ra ngoài.

Trong quá trình tính toán giải mã của CELL, các dữ liệu đầu vào như véc-tơ nghiệm \mathbf{x} ban đầu, các véc-tơ \mathbf{R}_m , $\mathbf{Q}'\mathbf{y}_m$ được đẩy vào song song và đồng thời với nhau. Các khối INPUT SELECTOR là các khối tổ hợp, như vậy chỉ cần 3 xung nhịp clock để cấp đủ các thành phần tính toán khoảng cách Euclid cho các SDE ở các lớp. Việc tính toán khoảng cách Euclid sẽ được hoàn thành xong trước phép tính giá trị bán kính cầu mới, công việc còn lại là đợi giá trị bán kính cầu mới được tính xong một cách lần lượt để hoàn tất giải mã. Mỗi khối SDE cần 4 xung nhịp clock để hoàn tất tính toán, theo cách tính thông thường thì để hoàn tất tính toán cho 3 lớp chúng ta cần 12 xung nhịp clock. Tuy nhiên, bằng cách nạp dữ liệu song song và tính trước khoảng cách Euclid chúng ta cần 4 xung nhịp để hoàn tất lớp đầu tiên, 5 xung nhịp để hoàn tất lớp thứ hai và 6 xung nhịp để hoàn tất lớp thứ 3. Điều này giúp giảm thời gian trễ hệ thống xuống còn một nửa và tăng gấp đôi băng thông.

2.2.2. Cấu trúc hệ thống khối giải mã cầu

Cấu trúc tổng thể của khối giải mã cầu bao gồm khối tính toán giải mã CELL và các khối chức năng khác để thực hiện thuật toán giải mã trên phần cứng đã được đề xuất ở phần trước. Quá trình giải mã bao gồm 8 lớp, trong khi đó, chúng ta chỉ sử dụng một khối tính toán duy nhất là CELL, mà CELL chỉ tính toán được tối đa ba lớp liên tiếp do những hạn chế về tài nguyên phần cứng. Vì vậy, ta cần tái sử dụng khối CELL để thực hiện toàn bộ 8 lớp giải mã, để tái sử dụng được CELL chúng ta cần sử dụng đến các khối chức năng khác thực hiện phân phối nghiệm (SD DISTRIBUTOR), so sánh (BEST ROOT) và quản lý trạng thái (SD FSM). Cấu trúc chi tiết của khối giải mã cầu được trình bày ở Hình 6.

Nguyên lý hoạt động của mạch như sau: vì thiết kế thực hiện theo cấu trúc 3-2-3 (hình 3) nên sau khi thực hiện 3 lớp đầu (pha thứ nhất) (lúc này sẽ tính được 3 phần tử trong véc-tơ nghiệm \mathbf{x}) sẽ thu được tối đa là $4^3 = 64$ nghiệm thỏa mãn nằm trong cầu. Tiếp theo mỗi nghiệm trên được đưa tới pha thứ 2 để tính toán 2 lớp tiếp theo (xác định giá trị của hai phần tử tiếp theo trong vector \mathbf{x}), sau đó, mỗi nghiệm thu được ở pha thứ 2 được đưa thẳng xuống pha thứ 3 để tính toán 3 lớp cuối cùng và so sánh tìm ra nghiệm tốt nhất mà không tính hết các nghiệm luôn ở pha thứ hai, điều này giúp ta tiết kiệm được bộ nhớ để lưu trữ nghiệm. Quá trình này được lặp lại cho đến khi các nghiệm thỏa mãn ở lớp 2 được đẩy xuống và tính toán xong ở lớp 3. Tại mỗi vòng lặp, nghiệm tối ưu của vòng lặp này được so sánh với nghiệm tốt nhất của các vòng lặp trước đó, nếu tốt hơn thì giữ lại còn không sẽ bị loại bỏ. Cuối cùng ta sẽ thu được nghiệm tốt nhất trong tất cả các nghiệm thỏa mãn bán kính cầu đã đưa ra.

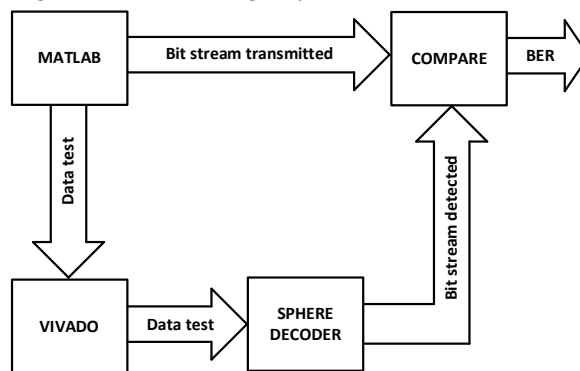


Hình 6. Cấu trúc phần cứng bộ giải mã cầu.

3. MÔ PHỎNG LOGIC VÀ THỰC THI TRÊN FPGA

3.1. Mô phỏng logic và kiểm tra chức năng

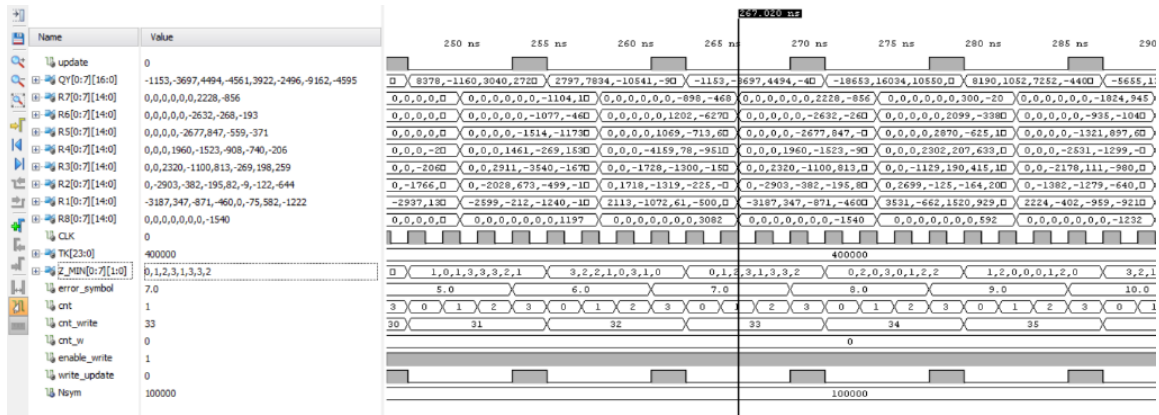
Để kiểm tra chức năng của thiết kế, chúng tôi sử dụng phần mềm Matlab để tạo ra dữ liệu kiểm tra và sử dụng trình mô phỏng của VIVADO để mô phỏng chức năng phần cứng của khối giải mã cầu. Matlab mô phỏng kênh truyền MIMO trong điều kiện có tạp âm Gauss tạo ra luồng bit truyền và tập dữ liệu đầu vào của khối giải mã bao gồm ma trận \mathbf{R} , ma trận \mathbf{Q}'_y . Chúng tôi lưu các dữ liệu đầu vào thành các file dữ liệu và sử dụng thư viện TextIO của VHDL để đọc vào trình mô phỏng và nạp vào khối giải mã cầu. Kết quả giải mã từ khối giải mã cầu được ghi vào file và được phân tích bằng Matlab để đánh giá tỷ lệ lỗi bit.



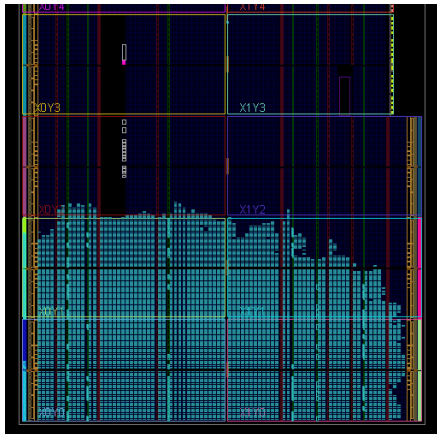
Hình 7. Mô hình kiểm tra chức năng.

Cụ thể chúng tôi tạo ra 1.000.000 tập dữ liệu kiểm tra tương ứng với 1.000.000 symbol được truyền. Dữ liệu được chuyển thành nhị phân trước khi đưa vào mô phỏng logic, các symbol kết quả của quá trình giải mã được chuyển thành luồng bit để so sánh với chuỗi bit truyền ban đầu. Với 1.000.000 symbol chúng tôi có 16.000.000 bit kiểm tra, đủ lớn để đánh giá chức năng của khối giải mã.

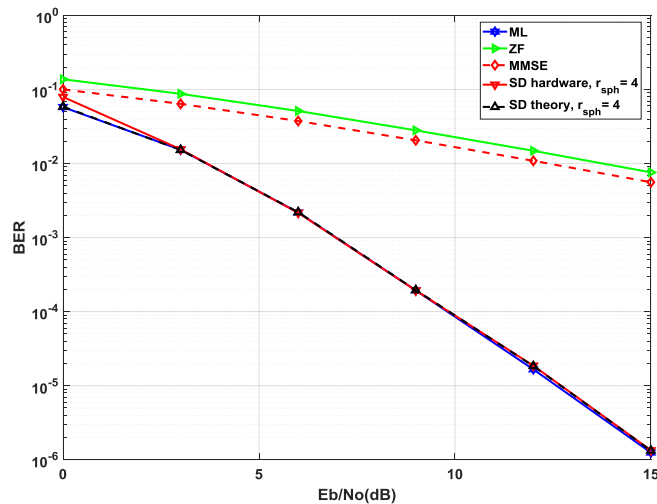
Sau khi chạy mô phỏng với 1.000.000 tập dữ liệu chúng tôi thu được kết quả như hình 10. Với bán kính cầu lựa chọn $r_{sph} = 4$, bộ giải mã SD đường đặc tuyến BER luôn tốt hơn với đường BER của các bộ giải mã tuyến tính ZF và MMSE. Với tỉ số Eb/No lần lượt là 3 dB, 6 dB, 19 dB, 12 dB, 15 dB, giá trị BER của bộ tách ZF tương ứng gấp khoảng 6, 24, 145, 799, 5.680 lần giá trị BER của bộ giải mã SD được thực thi với kiến trúc đề xuất trên FPGA. Tương tự với bộ giải mã MMSE, BER của nó gấp khoảng 4, 17, 106, 585, 4.176 lần so với BER của bộ giải mã SD đề xuất. Như vậy, khi tỉ số Eb/No càng tăng, tỉ lệ lỗi BER của giải mã SD được thực thi trên kiến trúc đề xuất càng giảm nhiều so với bộ giải mã ZF và MMSE. Bên cạnh đó, kết quả khảo sát cho thấy bộ giải mã cầu phân cứng được thực thi trên kiến trúc đề xuất cho đường BER xấp xỉ với BER của bộ giải mã cầu lý thuyết được mô phỏng bằng Matlab với cùng điều kiện kiểm tra và tiệm cận với đường BER của bộ giải mã hợp lý cực đại khi tỉ số Eb/No lớn hơn 3 dB. Như vậy, có thể khẳng định thiết kế bộ giải mã cầu phân cứng đã thực hiện đúng chức năng giải cầu lý thuyết.



Hình 8. Đồ thị thời gian mô phỏng trên Vivado.



Hình 9. Kết quả thực thi trên FPGA xc7k325tfg676-3.



Hình 10. Kết quả mô phỏng logic.

3.2. Kết quả tổng hợp trên FPGA

Thiết kế khối giải mã cầu được tổng hợp trên chip FPGA Kintex 7 xc7k325tfg676-3 (*) (tầm trung) và Virtex UltraScale+ xcvy7p-flva2104-3-e-EVAL (**) (cao cấp) sử dụng phần mềm Vivado 2016.4. Với độ rộng bit của các dữ liệu đầu vào được thể hiện ở bảng 1, tài nguyên chiếm dụng của thiết kế trên các chip được thể hiện ở bảng 2.

Bảng 1. Độ rộng bit của các tín hiệu được sử dụng trong khối giải mã cầu.

Tín hiệu	Wordlength
R	15 bit
Qy	17 bit
x	3 bit
r_{sph}^2	36 bit

Bảng 2. Tài nguyên chiếm dụng của thiết kế ((*)-xc7k325tfg676-3, (**) xcvy7p-flva2104-3-e-EVAL).

Tài nguyên	Ước lượng		Có sẵn		Hiệu suất %	
	(*)	(**)	(*)	(**)	(*)	(**)
LUT	80.814	54.365	203.800	78.160	39,65	6,90
FF	23.256	23.153	407.600	1.576.320	5,71	1,47
DSP	84	84	840	4.560	10,00	1,84
BUFG	1	1	32	1.200	3,13	0,08

Tốc độ clock tối đa mà thiết kế có thể đạt được trên chip FPGA Kintex 7 XC7k325 là xấp xỉ 200 MHz, và trên Virtex UltraScale+ xcvy7p-flva2104-3-e-EVAL là xấp xỉ 400 MHz.

3.3. Đánh giá thông lượng của thiết kế

Bảng 3. Khảo sát thời gian trung bình để giải mã một symbol.

Bán kính cầu r_{sph}	Số chu kỳ trung bình	Thời gian giải mã T=5ns
2	90.27	0.45 us
3	175.15	0.874 us
3.5	288.62	1.443 us
4	427.08	2.1354 us
5	933.314	4.666 us

Thiết kế khối giải mã cầu trên phần cứng FPGA đã đáp ứng được yêu cầu về mặt chức năng so với lý thuyết giải mã cầu. Tuy nhiên, về mặt thông lượng thì thiết kế này tỏ ra không xác định mà phụ thuộc vào mức độ và đặc điểm của tín hiệu đầu vào, vì tính chất xét hết tất cả các nghiệm thỏa mãn nằm trong cầu trong quá trình giải mã nên thời gian để giải một symbol phụ thuộc vào sự phân bố số lượng nghiệm. Số nghiệm thỏa mãn càng nhiều thì thời gian giải mã càng lớn, đặc biệt là ở các lớp đầu tiên. Số lượng nghiệm tồn tại ở từng lớp có phân bố thống kê nhất định và phụ thuộc vào E_b/N_0 và bán kính cầu ban đầu r_{sph} . Bảng dưới đây khảo sát số chu kỳ clock trung bình để hoàn tất giải mã một symbol với $E_b/N_0 = 15$ dB.

Thông thường của thiết kế không cố định, phụ thuộc mạnh vào điều kiện kênh truyền (E_b/N_0) và bán kính cầu ban đầu r_{sph} . Nếu ta xét trong trường hợp $E_b/N_0 = 15$ dB và bán kính cầu ban đầu là 2 thì thông lượng trung bình của bộ giải mã áp dụng cho hệ thống 16-QAM 4x4 MIMO là:

$$T_{kintex} = \frac{f_{clk}}{90,27} \times 16bit = \frac{200.10^6}{90,27} \times 16 - bit = 35,44(Mbps)$$

$$T_{Virtex} = \frac{f_{clk}}{90,27} \times 16bit = \frac{400.10^6}{90,27} \times 16 - bit = 70,88(Mbps)$$

3.4. Thảo luận về thiết kế phần cứng của hệ thống

Bảng 4. So sánh đánh giá tài nguyên sử dụng cho bộ giải mã cầu (SD) thực thi trên FPGA.

Tài nguyên	(*)	(**)	[2]
Mô hình	4x4	4x4	2x2
Điều chế	16-QAM	16-QAM	QPSK
LUT	80.814	54.365	119.562
FF	23.256	23.153	17.858
DSP	84	84	280
BUFG	1	1	16

Công trình [2], tập tin hiệu đầu vào hệ thống sử dụng điều chế QPSK và mô hình MIMO 2x2 có cấu trúc tín hiệu và xử lý không phức tạp bằng tập tin hiệu sử dụng 16-QAM và mô hình MIMO 4x4 được sử dụng cho kiến trúc thực thi được sử dụng trong nghiên cứu đề xuất. Tuy nhiên, qua bảng 4, tài nguyên chiếm dụng của kiến trúc đề xuất nhỏ hơn đáng kể so với tài nguyên chiếm dụng được công bố trong công trình [2]. Nên có thể thấy rằng kiến trúc đề xuất và kết quả thực thi trên phần cứng đề xuất cho kết quả tối ưu hơn so với kiến trúc và kết quả thực thi được đưa ra trong công trình [2]. Ta thấy rằng, với việc sử dụng một chip FPGA tầm trung Kintex 7 XC7k325 thì tài nguyên sử dụng cho thiết kế chiếm lên đến 40% tổng tài nguyên của chip, xung nhịp của hệ thống tối đa là 200 MHz làm cho thông lượng đạt được khoảng 35 Mbps. Với chip Virtex Ultra Scale + với tốc độ clock gấp đôi có thể đạt được khoảng 70 Mbps. Từ bảng 2 cho thấy khối giải mã cầu với một khối tính toán duy nhất chỉ chiếm gần 7% tài nguyên trên Virtex UltraScale+. Do đó, để cải thiện thông lượng chúng ta có thể xây dựng 4 luồng giải mã song song với 4 khối tính toán CELL mà vẫn bảo đảm được về mặt tài nguyên. Với tần số clock tối đa có thể đạt được trên chip Virtex UltraScale+ là 400 MHz thông lượng của khối giải mã đạt được xấp xỉ 280 Mbps. Tuy nhiên, các giải pháp mẫu chốt có thể không nằm ở vấn đề tài nguyên mà sẽ nằm ở việc cải tiến vi kiến trúc của thiết kế, ví dụ để thực thi hoàn toàn theo mô hình đường ống, mặt khác cần có các giải pháp về thuật toán để giảm bớt sự phức tạp tính toán và tăng tốc giải mã với định hướng thực thi cho phần cứng.

4. KẾT LUẬN

Bộ giải mã cầu và các thuật toán tìm kiếm dạng cây tương tự cung cấp sự cân bằng hiệu suất-độ phức tạp phù hợp với việc triển khai phần cứng của hệ thống truyền thông MIMO vô tuyến. Trong bài báo này, chúng tôi đã đề xuất kiến trúc và thực thi bộ giải mã SD trên nền tảng công nghệ phần cứng FPGA với chất lượng cho đường cong của hệ số phẩm chất BER xấp xỉ với bộ giải mã ML. Kiến trúc đề xuất và thực thi cho bộ giải mã cầu được tối ưu với tài nguyên chiếm dụng khoảng 40% tài nguyên chip cho FPGA Kintex 7 XC7k325 và 7% với chip Virtex UltraScale+. Với tài nguyên chiếm dụng, kiến trúc đề xuất có thể thực hiện tăng luồng xử lý song song nhằm nâng cao thông lượng xử lý dữ liệu hệ thống.

TÀI LIỆU THAM KHẢO

- [1]. T. X. Nam, L. M. Tuấn, *Xử lý tín hiệu không gian thời gian*, NXB Khoa học và kỹ thuật, (2013).
- [2]. Zekry, Abdelhalim, "FPGA Implementation of Sphere Detector for Spatial Multiplexing MIMO System," International Journal of Electronics and Telecommunications, vol. 65, pp. 245–252, (2019).
- [3]. M. O. Damen, H. E. Gamal, and G. Caire, "On maximum likelihood detection and the search for the closest lattice point," IEEE Trans. Inform. Theory, vol. 49, pp. 2389–2402, (2003).

- [4]. U. Fincke, M. Pohst, "Improved methods for calculating vectors of short length," Mathematics of Computation, (1985).
- [5]. Biglieri, E. Viterbo and E., "A universal decoding algorithm for lattice codes", Colloque GRETSI, vol. 14, pp. 611–614, (1993).
- [6]. D. Wubben, R. Bohnke, V. Kuhn, and K.-D. Kammeyer, "MMSE extension of V-BLAST based on sorted QR decomposition," in Proc. IEEE 58th Vehicular Technology Conference (VTC), vol. 1, no. 1, pp. 508–512, (2003).
- [7]. M. Pohst, "On the computation of lattice vectors of minimal length, successive minima," SIGSAM Bull., vol. 15, no. 1, pp. 37-44, (1981).
- [8]. X. Jun, G. Diyuan and W. Zengye, "Research of Improved Sphere Decoding Algorithm," 2019 Chinese Control And Decision Conference (CCDC), pp. 1043-1047, (2019).
- [9]. P. Tsai, W. Chen, X. Lin and M. Huang, "A 4×4 64-QAM reduced-complexity K-best MIMO detector up to 1.5Gbps," Proceedings of 2010 IEEE International Symposium on Circuits and Systems, pp. 3953-3956, (2010).
- [10]. Kang, B. Shim and I., "Sphere Decoding With a Probabilistic Tree Pruning," IEEE Transactions on Signal Processing,, vol. 56, pp. 4867-4878, (2008).
- [11]. K.-W. Wong, C.-Y. Tsui, R. S.-K. Cheng, and W.-H. Mow, "A VLSI Architecture of a K-Best Lattice Decoding Algorithm for MIMO Channels," in Proc. IEEE International Symposium on Circuits and Systems (ISCAS), vol. 3, no. 1, pp. 273–276, (2002).
- [12]. Vikalo, B. Hassibi and H., "On the expected complexity of sphere decoding," in Proc. Thirty-Fifth Asilomar Conference on Signals, Systems and Computers, vol. 2, pp. 1051–1055, (2001).
- [13]. C. P. Schnorr and M. Euchner, "Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems," Math.Program, vol. 66, pp. 181–191, (1994).
- [14]. Nilsson, Z. Guo and P., "Reduced complexity Schnorr-Euchner decoding algorithms for MIMO systems," IEEE Communications Letters, vol. 8, pp. 286–288, (2004).
- [15]. Ibrahim A, Bello, Basel Halak, Mohammed El-Hajjar, Mark Zwolinski, "VLSI Implementation of a Fully-Pipelined K-Best MIMODetector with Successive Interference Cancellation," in Circuits Systems and Signal Processing, (2019).

ABSTRACT

Design and evaluation of sphere decoder accelerator on reconfiguration hardware

The Maximum likelihood (ML) detection can achieve the best bit error rate but requires very high computational complexity. The latter makes this algorithm is not practically applicable. Many decoder architectures hence have been proposed to overcome the ML high complexity. The sphere decoding (SD) algorithm is one of the most promising approaches that offer quasi-ML performance with a reasonable computing workload. This paper proposes an efficient and practical approach for a sphere decoder design on reconfigurable hardware (FPGA). The design is evaluated to yield a quality approximation of the Maximum Likelihood (ML) method but with significantly reduced computational complexity.

Keywords: MIMO; FPGA; SM; SD; ML.