

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA

TRẦN THỊ BÍCH NGỌC

NGHIÊN CỨU VÀ THIẾT KẾ BỘ GIẢI MÃ  
KIỂM TRA CHẤM LỖ MẬT ĐỘ THẤP (LDPC)  
TRONG HỆ THỐNG THÔNG TIN THẾ HỆ MỚI

Ngành: Kỹ thuật điện tử

Mã số ngành: 9520203

TÓM TẮT LUẬN ÁN TIẾN SĨ

TP. HỒ CHÍ MINH - NĂM 2024

Công trình được hoàn thành tại **Trường Đại học Bách Khoa – ĐHQG-HCM**

Người hướng dẫn 1: PGS.TS. Hoàng Trang

Người hướng dẫn 2: TS. Nguyễn Lý Thiên Trường

Phản biện độc lập 1:

Phản biện độc lập 2:

Phản biện 1:

Phản biện 2:

Phản biện 3:

Luận án sẽ được bảo vệ trước Hội đồng đánh giá luận án họp tại

.....  
.....  
vào lúc          giờ          ngày          tháng          năm

Có thể tìm hiểu luận án tại thư viện:

- Thư viện Trường Đại học Bách Khoa – ĐHQG-HCM
- Thư viện Đại học Quốc gia Tp.HCM
- Thư viện Khoa học Tổng hợp Tp.HCM

# GIỚI THIỆU

## 1.1 Tính cấp thiết của luận án

Mã kiểm tra chẵn lẻ mật độ thấp (Low-Density Parity-Check code-LDPC) là một trong những mã đã được sử dụng trong phương pháp FEC. Mã LDPC có những ưu điểm như cách thức giải mã không quá phức tạp [2, 3], tối ưu về diện tích thiết kế phần cứng, tốc độ xử lý cao và hiệu quả về năng lượng [4-6]. Vì vậy, chúng được sử dụng rộng rãi trong các chuẩn truyền thông như 802.11ad (WiGig) [7], 802.11n (Wi-Fi) [8], 802.16e (WiMAX) [9], 802.15.3c (WPAN) [10], DVB-S2 [11, 12], trong mạng thông tin quang, trong bộ lưu trữ số [13-15] và cho mạng thông tin di động 5G [16, 17].

Bộ giải mã LDPC được thực hiện dựa trên thuật toán giải mã lập (Message-Passing (MP)). Thuật toán giải mã lập cơ sở là thuật toán Belief-Propagation (BP) [24]. Thuật toán này có hiệu suất sửa lỗi rất tốt nhưng thiết kế phần cứng vô cùng phức tạp. Chính vì vậy, thuật toán Min-Sum (MS) đã được cải tiến từ thuật toán BP bằng cách sử dụng phương pháp tính toán xấp xỉ trong quá trình xử lý các nút kiểm tra nghĩa là trong tính toán, thuật toán này chỉ cần sử dụng phép cộng và phép so sánh. Thiết kế phần cứng của bộ giải mã MS đơn giản hơn nhưng hiệu suất giải mã lại kém hơn so với bộ giải mã BP [25]. Do đó, nhiều nghiên cứu đã công bố với mục đích tăng hiệu suất sửa lỗi của thuật toán MS bằng các cách tác động lên quá trình xử lý các nút kiểm tra, quá trình xử lý các nút biến.

Nhằm cải thiện hiệu suất giải mã, có nhiều nghiên cứu đã đề xuất tác động vào quá trình xử lý các nút kiểm tra bằng cách sử dụng hệ số chuẩn hóa hay hệ số offset [24, 26]. Các nghiên cứu này cho thấy hiệu suất giải mã được cải thiện nhưng còn cách xa hiệu suất giải mã BP. Để tăng kết quả sửa lỗi hơn nữa, các nghiên cứu [27-33] đã đồng thời sử dụng hai hệ số chuẩn hóa hay hai hệ số offset tác động lên quá trình xử lý các nút kiểm tra. Kết quả giải mã được cải thiện hơn nhưng làm tăng tài nguyên phần cứng cho khối xử lý các nút kiểm

tra. Từ khía cạnh phần cứng, quá trình xử lý các nút kiểm tra của thuật toán MS là tìm hai giá trị cực tiểu của các thông tin ngõ vào ( $min1, min2$ ) và vị trí của giá trị cực tiểu thứ nhất. Nhằm tiết kiệm giảm tài nguyên phần cứng, các nghiên cứu [34-36] chỉ tìm giá trị cực tiểu  $min1$ , giá trị cực tiểu thứ 2 được lấy xấp xỉ từ giá trị  $min1$  bằng cách sử dụng thêm hệ số offset. Phương pháp tiếp cận này giảm đáng kể tài nguyên phần cứng của khối xử lý nút kiểm tra nhưng lỗi nền rất cao trong kết quả giải mã vì giá trị sai khác giữa hai giá trị cực tiểu tùy thuộc vào mỗi vòng lặp. Để cải thiện hiệu suất giải mã của thuật toán này, các tác giả trong [37] đã đưa các hệ số offset để tính toán cho mỗi vòng lặp trong quá trình giải mã. Tuy nhiên, nhược điểm là việc tối ưu các giá trị hệ số hiệu chỉnh gặp khó khăn, mà các giá trị này phụ thuộc vào số các vòng lặp và giá trị SNR. Ngoài ra, việc tìm ra các hệ số offset luôn phức tạp hơn các hệ số chuẩn hóa, nhưng thuật toán giải mã sử dụng hệ số offset cho kết quả giải mã tốt hơn khi dùng hệ số chuẩn hóa [38]. Chính vì vậy, trong luận án này, với mục đích nâng cao hiệu suất giải mã và hạn chế tăng tài nguyên phần cứng, thay vì dùng đồng thời hai hệ số chuẩn hóa hay offset, thuật toán giải mã được cải tiến bằng cách sử dụng kết hợp hệ số chuẩn hóa và hệ số offset tác động lên quá trình xử lý các nút kiểm tra. Bên cạnh đó, để tiết kiệm tài nguyên phần cứng của khối xử lý này ở quá trình xử lý nút kiểm tra, luận án thực hiện tìm duy nhất giá trị cực tiểu thứ nhất và giá trị cực tiểu thứ hai được hiệu chỉnh từ giá trị cực tiểu thứ nhất bằng hệ số offset/chuẩn hóa. Để tránh sự xuất hiện lỗi nền khi sử dụng phương pháp tìm giá trị cực tiểu duy nhất, ở quá trình xử lý các nút biến cũng được tác động bằng hệ số hiệu chỉnh. Việc tác động này đã thực hiện ở khối xử lý các nút biến vì hạn chế tăng tính phức tạp của khối xử lý các nút kiểm tra nhằm khối kiểm tra là khối cần tính toán phức tạp nhất trong bộ giải mã [39]. Tác giả đã chứng minh trên cơ sở lý thuyết việc ảnh hưởng của các hệ số hiệu chỉnh (hệ số offset, hệ số chuẩn hóa) đến khả năng sửa lỗi và tiến hành mô phỏng Monte-Carlo bằng MATLAB để đánh giá chất lượng hệ thống thông qua tỷ lệ lỗi bit và tỷ số năng lượng bit trên nhiễu để kiểm chứng kết quả đã đề xuất.

Trong thuật toán giải mã lặp của bộ giải mã LDPC, các nút trao đổi thông tin qua các cạnh kết nối của giàn đồ Tanner [40]. Có hai kỹ thuật lịch trình trao đổi thông tin cơ bản: layer và flooding. Áp dụng cho cùng loại thuật toán Normalized Min-Sum (NMS), các tác giả trong [25] đã chứng minh việc áp dụng kỹ thuật layer đã tăng khả năng giải mã gần như gấp đôi so với dùng kỹ thuật flooding. Các tác giả trong [41] cũng sử dụng kỹ thuật layer và kết hợp việc xử lý song song giúp tăng đáng kể tốc độ xử lý. Việc sử dụng kỹ thuật flooding giúp tốc độ xử lý cao, độ trễ thấp nhưng hiệu suất sửa lỗi thấp và độ phức tạp của các khối kết nối cao [26]. Ngược lại, kỹ thuật layer giúp hiệu suất giải mã tăng, độ phức tạp định tuyến thấp nhưng tốc độ xử lý giảm hơn, độ trễ cao. Hướng nghiên cứu khác để đạt tối ưu về tài nguyên phần cứng trong thiết kế bộ giải mã là việc lựa chọn các kiến trúc phần cứng như song song hoàn toàn, bán song song hay nối tiếp. Trong cấu trúc song song, mạng định tuyến kết nối giữa các khối xử lý quá phức tạp, đặc biệt với trường hợp chiều dài mã lớn, gây hiện tượng tắc nghẽn định tuyến (routing bottleneck) [42] và kết cấu phần cứng công kênh (do sử dụng rất nhiều khối xử lý song song). Ngược lại, cấu trúc nối tiếp yêu cầu tài nguyên phần cứng ít nhưng tốc độ xử lý rất thấp [43]. Do vậy, trên thực tế phổ biến hơn cả là cấu trúc bán song song, các thiết kế này thường có sự đánh đổi giữa tốc độ xử lý và tài nguyên phần cứng. Có hai cách cơ bản có thể tăng tốc độ xử lý của cấu trúc bán song song là: tăng tần số làm việc và tăng số lượng các khối xử lý song song [44]. Tăng tần số làm việc bằng cách sử dụng kỹ thuật pipeline, nhưng nó gây trễ trong việc ghi dữ liệu vào bộ nhớ (khi có xảy ra xung đột pipeline). Giải pháp tránh xung đột pipeline là đưa thêm các chu kỳ dừng hoạt động (stall cycles). Các tác giả trong [45-47] đã sử dụng lịch trình đọc/ghi ngoại tuyến nhằm làm giảm số chu kỳ dừng hoạt động dựa trên kỹ thuật thay đổi trật tự ma trận kiểm tra chẵn lẻ (Parity Check Matrix-PCM). Thực chất việc thay đổi trật tự PCM không làm giảm số chu kỳ dừng hoạt động, mà nó làm giảm các khối xử lý song song. Khi đó, làm giảm tài nguyên phần cứng đồng nghĩa là số chu kỳ dừng hoạt động giảm. Hạn chế của phương pháp này là việc xử lý song song giảm và tốc độ xử

lý cũng giảm. Các tác giả trong [48] đã đề xuất kiến trúc bộ giải mã layer với bộ nhớ lưu trữ dạng đơn kênh. Khi có xung đột pipeline, các tác giả đã bỏ qua quá trình cập nhật các từ mã. Các thông tin các nút kiểm tra khi có xung đột sẽ được lưu vào bộ nhớ riêng và sẽ sử dụng chúng để cập nhật các thông tin từ mã mới cập nhật khi không còn xung đột pipeline. Tuy nhiên, việc bỏ qua quá trình cập nhật các từ mã sẽ làm giảm hiệu suất sửa lỗi. Các tác giả trong [49] hạn chế sự ảnh hưởng dòng dữ liệu bằng cách sử dụng giải mã cùng lúc nhiều khung dữ liệu nhưng độ trễ tăng theo cấp số nhân với số khung dữ liệu sử dụng và tài nguyên phần cứng cũng tăng. Các tác giả trong [50] đã thiết kế bộ giải mã áp dụng kỹ thuật pipeline mà không sử dụng chu kỳ dừng hoạt động khi xảy ra xung đột truy cập bộ nhớ hay khi các bậc nút kiểm tra thay đổi. Thiết kế này sử dụng lịch trình giải mã kết hợp, điều này có nghĩa lúc bình thường bộ giải mã sẽ hoạt động với lịch trình layer, nếu có xung đột pipeline xảy ra nó lập tức chuyển sang hoạt động với lịch trình flooding. Kết quả là tài nguyên phần cứng tiết kiệm đáng kể, tuy nhiên số lượng thanh ghi trong thiết kế tăng. Trong luận án này, tác giả thực hiện thiết kế bộ giải mã bằng cách sử dụng cấu trúc bán song song kết hợp với kỹ thuật layer nhằm tiết kiệm tài nguyên phần cứng, giảm sự phức tạp của mạng kết nối và tăng kết quả sửa lỗi. Giúp tăng tốc độ xử lý, thiết kế trong luận án sẽ sử dụng kỹ thuật pipeline.

Khi sử dụng kỹ thuật lịch trình layer, thông tin các nút kiểm tra và nút biến cần được lưu trữ để xử lý ở các vòng lặp tiếp theo, do đó, bộ nhớ thường chiếm phần khá lớn trong toàn bộ diện tích chip và mức tiêu thụ công suất trong bộ giải mã LDPC [27]. Với mục đích tiết kiệm không gian bộ nhớ, các nghiên cứu đã áp dụng kỹ thuật như “xóa” bit thông tin các nút kiểm tra có trọng số nhỏ nhất (Least Significant Bit-LSB) [51], chia mỗi phân lớp thành 2 module bằng nhau (phương pháp Split row) [52-55], sử dụng ít bit để biểu diễn thông tin [44], kỹ thuật tạo mới ma trận mở rộng: thay vì mở rộng bằng ma trận hoán vị  $Z \times Z$ , các tác giả trong [56] đã thay thế bằng ma trận mở rộng  $Z \times r$  với  $r$  là số các phần tử có giá trị bằng -1. Mặc dù, các nghiên cứu được đề cập đã cho thấy sự giảm không gian lưu trữ, nhưng hiệu suất sửa lỗi giảm đi đáng kể. Cấu trúc

tiết kiệm tài nguyên phần cứng của bộ giải mã khác được giới thiệu trong công bố [48]. Dựa trên tính chất trực giao của mã 5G LDPC (nghĩa là hai lớp của ma trận BG1 hay BG2 ở cùng một cột sẽ không có nhiều hơn 1 giá trị không âm), các tác giả đã kết hợp các lớp có tính chất trực giao với nhau và kỹ thuật phân chia bộ nhớ nhằm cải thiện tài nguyên phần cứng và giảm đáng kể không gian bộ nhớ. Hạn chế của thiết kế này là do kết hợp các phân lớp với nhau nên các từ mã mới cập nhật sẽ lớn hơn, bộ nhớ lưu trữ các từ mã này sẽ cần thêm không gian.

Như vậy, việc thiết kế bộ giải mã LDPC với các đặc tính tốc độ xử lý cao, tiết kiệm phần cứng và hiệu suất giải mã cao là điều cần thiết.

## **1.2 Mục tiêu của luận án**

Nhiệm vụ cụ thể của luận án là giải quyết các vấn đề sau:

- Nghiên cứu và đề xuất phương pháp cải thiện hiệu suất giải mã của thuật toán giải mã dựa trên cơ sở thuật toán MS bằng cách hiệu chỉnh lên các thông tin trong quá trình xử lý các nút biên và/hoặc quá trình xử lý các nút kiểm tra. Xác định thuật toán giải mã đề xuất ứng dụng cho việc thiết kế bộ giải mã với mục đích tiết kiệm phần cứng và cải thiện hiệu suất giải mã.
- Nghiên cứu tính chất đặc biệt của mã 5G LDPC với mục đích tìm phương pháp tiết kiệm tài nguyên phần cứng.
- Nghiên cứu các kỹ thuật nhằm thiết kế bộ giải mã LDPC dựa trên thuật toán giải mã đề xuất trên nền tảng FPGA với tiết kiệm tài nguyên phần cứng, đặc biệt là với mục đích tiết kiệm bộ nhớ và cải thiện hiệu suất giải mã.

Để đạt được kết quả đặt ra, tác giả đã thực hiện việc cải tiến thiết kế bằng các phương pháp sau:

- a. Phương pháp thứ nhất là cải tiến khối thực hiện quá trình xử lý thông tin các nút kiểm tra bằng cách tính toán duy nhất giá trị cực tiểu thứ nhất của các

thông tin ngõ vào và vị trí của nó và ghép hai khối thực hiện xử lý các thông tin nút biến và khối thực hiện quá trình cập nhật từ mã.

- b. Phương pháp thứ hai là không lưu trữ giá trị cực tiểu thứ hai và lưu trữ các thông tin trong bộ nhớ dưới định dạng nén. Giá trị cực tiểu thứ hai được tính toán bằng cách hiệu chỉnh từ giá trị cực tiểu thứ nhất trong khối chuyển đổi thông tin từ định dạng nén sang không nén sau khi thông tin đọc được từ bộ nhớ.
- c. Phương pháp cuối cùng là áp dụng tính chất “không đều” của mã 5G LDPC, tác giả thực hiện việc phân chia bộ nhớ phụ thuộc vào bậc các nút kiểm tra.

Trong thiết kế phần cứng cho bộ giải mã LDPC, luận án sẽ thực hiện trên nền tảng công nghệ FPGA (Field-Programmable Gate Array).

### **1.3 Đối tượng nghiên cứu**

Đối tượng nghiên cứu chính của luận án là:

- Lý thuyết cơ bản về mã LDPC (khái niệm, phân loại và đặc điểm của mã LDPC trong mạng 5G).
- Thuật toán giải mã của mã LDPC.
- Cấu trúc phần cứng bộ giải mã LDPC.
- Các khối xử lý và bộ giải mã hoàn chỉnh.

### **1.4 Phạm vi nghiên cứu của luận án**

- Thuật toán giải mã dựa trên thuật toán giải mã MS.
- Mã LDPC trong mạng 5G với ma trận cơ sở BG1, BG2 chiều dài mã 4080, 13056, 7424, 8832, 6720, 8448 và tỷ lệ mã 1/2, 2/3, 3/4, 3/5, 1/4.
- Thiết kế bộ giải mã LDPC trên nền tảng FPGA với ma trận cơ sở BG1 của mã 5G, chiều dài mã 8832 và tỷ lệ mã 1/2 với mục đích tiết kiệm tài nguyên phần cứng và cải thiện hiệu suất giải mã.



## 1.5 Những đóng góp của luận án

Luận án có các đóng góp cụ thể như sau:

- i. Khảo sát sự ảnh hưởng của các giải thuật giải mã đến khả năng sửa lỗi của mã LDPC, cụ thể là mã QC-LDPC dùng trong mạng 5G.
- ii. Đề xuất các thuật toán giải mã dựa trên cơ sở thuật toán MS theo hướng hiệu chỉnh các thông tin của quá trình xử lý các nút kiểm tra sử dụng hai hệ số hiệu chỉnh.
- iii. Đề xuất các thuật toán giải mã bằng cách sử dụng các hệ số hiệu chỉnh lên đồng thời cả hai quá trình xử lý nút biến và nút kiểm tra của thuật toán MS.
- iv. Thiết kế bộ giải mã LDPC với mục đích tiết kiệm tài nguyên phần cứng và hiệu suất giải mã được cải thiện.
- v. Thiết kế bộ giải mã với mục đích tiết kiệm không gian bộ nhớ.

## 1.6 Bố cục của luận án

**Bố cục của luận án** bao gồm 5 chương cùng với phần mở đầu, tài liệu tham khảo và danh mục các báo cáo khoa học đã được công bố của tác giả.

*Chương 1: Giới thiệu.*

*Chương 2: Cơ sở lý thuyết về mã LDPC*

*Chương 3: Thuật toán giải mã sử dụng mã LDPC.*

*Chương 4: Thiết kế phần cứng bộ giải mã LDPC với mục đích tiết kiệm bộ nhớ và cải thiện hiệu suất giải mã.*

*Chương 5: Kết luận và hướng phát triển.*

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VỀ MÃ LDPC

### 2.1 Đôi nét về lịch sử phát triển của mã LDPC

Năm 1962, Gallager đã giới thiệu mã LDPC trong luận án tốt nghiệp [59]. Tuy nhiên, vào thời điểm đó khả năng tính toán của máy tính còn rất hạn chế nên ông đã không chứng minh được kết quả giải mã của nó tiến gần giới hạn Shannon. Chính vì vậy, công trình của Gallager đã bị bỏ quên gần 30 năm. Vào những năm 1996, MacKay và Neal chỉ ra những đặc tính ưu việt của mã khối tuyến tính có ma trận kiểm tra chẵn lẻ mật độ thấp và đã đề xuất ra mã MacKay-Neal [68].

### 2.2 Phương pháp biểu diễn mã LDPC

Có hai cách dùng để biểu diễn mã LDPC: biểu diễn bằng ma trận và biểu diễn bằng giản đồ.

### 2.3 Mã LDPC đều và không đều

Trong ma trận  $\mathbf{H}$ , ký hiệu  $d_c$  là số các số hạng có giá trị bằng 1 trên mỗi hàng và  $d_v$  là số các số hạng có giá trị bằng 1 trên mỗi cột.  $d_c, d_v$  lần lượt được gọi là bậc nút kiểm tra và bậc nút biến. Mã LDPC được gọi là đều nếu như các giá trị  $d_c$  không thay đổi trên mỗi hàng, và các giá trị  $d_v$  không thay đổi trên mỗi cột.

### 2.4 Mã Quasi-Cyclic LDPC

Mã Quasi-Cyclic (QC-LDPC) [75] là một dạng của mã LDPC được ứng dụng rộng rãi trong thực tế. Các mã QC-LDPC được định nghĩa bởi ma trận cơ sở  $\mathbf{B}$  có kích thước  $R \times C$  với các số hạng  $b_{i,j} \geq -1$  với  $i \in (1, \dots, R); j \in (1, \dots, C)$ .

### 2.5 Mã LDPC ứng dụng trong 5G

Mã LDPC ứng dụng trong 5G (hay mã 5G QC-LDPC) được đặc trưng bởi hai ma trận cơ sở BG1 và BG2 cùng với 51 hệ số mở rộng  $Z$ .

Cần phải lưu ý rằng mã 5G LDPC là mã không đều đối với cả các bậc nút biến lẫn bậc nút kiểm tra.

Bảng 2-1. Thống kê bậc các nút kiểm tra của BG1

<b>Bậc nút kiểm tra (<math>d_c</math>)</b>	3	4	5	6	7	8	9	10	19
<b>Số các dòng của ma trận BG1</b>	1	5	18	8	5	2	2	1	4

## 2.6 Quan hệ tỷ lệ tín hiệu trên nhiễu (SNR) với tỷ lệ lỗi bit (BER).

Một trong những đặc tính dùng để đánh giá thuật toán giải mã là kết quả sửa lỗi. Đó chính là mối quan hệ của tỷ lệ tín hiệu trên nhiễu (SNR) với tỷ lệ lỗi bit (BER). Trong hệ thống thông tin số, SNR còn được thể hiện qua tỷ số năng lượng ký hiệu trên mật độ tạp âm ( $E_s/N_0$ ).

## 2.7 Các thuật toán giải mã.

Về cơ bản, các thuật toán giải mã cho mã LDPC có thể được phân thành hai loại chính: quyết định cứng (hard-decision) [80] và phương pháp quyết định mềm (soft-decision) [81]. Trong luận án này, chúng tôi sử dụng phương pháp quyết định mềm. Thuật toán giải mã LDPC dựa trên cơ sở thuật toán MP (Message Passing) và được chia làm hai loại chính: Belief Propagation (BP) (hay Sum-Product (SP)) và Min-Sum (MS).

## 2.8 Phương pháp Density Evolution

Density Evolution là phương pháp hiệu quả để phân tích kết quả giải mã của các thuật toán giải mã lặp MP. Phương pháp này xác định các hệ số tác động tối ưu lên các quá trình xử lý thông tin.

## 2.9 Cấu trúc phần cứng bộ giải mã LDPC

Cấu trúc bộ giải mã LDPC cơ bản được chia thành 3 loại: cấu trúc song song, cấu trúc nối tiếp và cấu trúc bán song song.

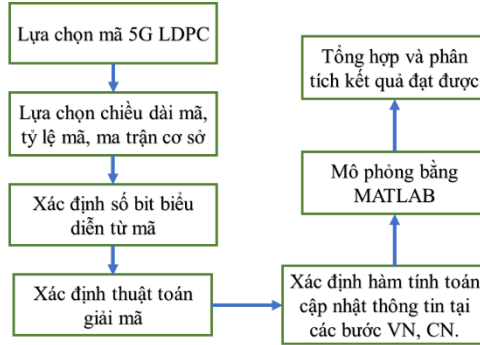
## 2.10 Lịch trình

Lịch trình của quá trình giải mã LDPC xác định trình tự thực hiện các nút VN và CN hay khả năng cùng lúc nhiều nút thực hiện một cách song song. Có nhiều loại lịch trình nhưng có hai loại phổ biến là lịch trình flooding [88, 89] và lịch trình layer [90, 91].

## 2.11 Qui trình thiết kế bộ giải mã LDPC trên FPGA.

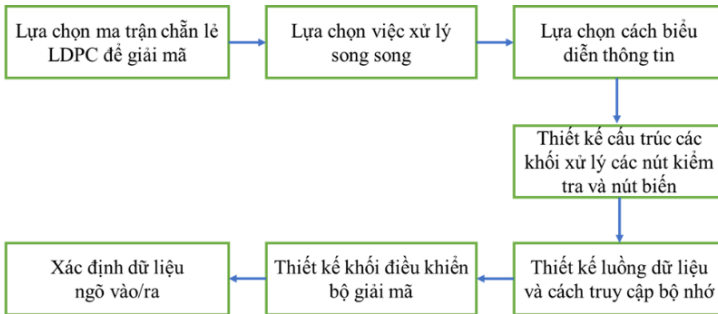
### a. Nghiên cứu xây dựng thuật toán cho bộ giải mã.

Qui trình thực hiện xác định thuật toán sẽ tiến hành cụ thể như Hình 2-14.



Hình 2-14. Quy trình xác định thuật toán giải mã LDPC

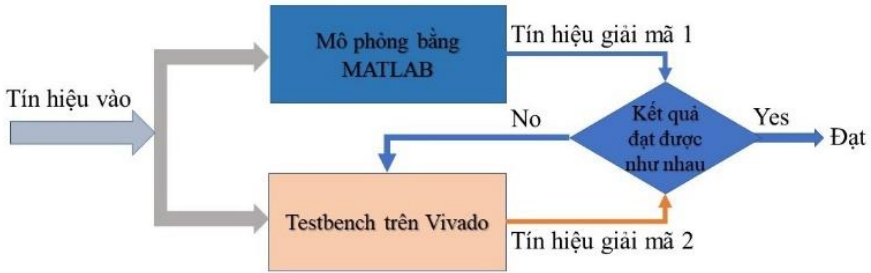
### b. Qui trình thiết kế bộ giải mã LDPC



Hình 2-15. Qui trình thiết kế bộ giải mã LDPC

### a. Kiểm tra kết quả

Để đảm bảo kết quả đạt ta cần kiểm tra kết quả phần cứng và phần mềm như sơ đồ ở Hình 2-16.



Hình 2-16. Sơ đồ kiểm tra kết quả giữa phần cứng và phần mềm

## 2.12 Ký hiệu

Mã LDPC có chiều dài mã là  $N$  và chiều dài khối tin  $K$  sẽ được biểu diễn bằng ma trận chẵn lẻ  $\mathbf{H}$  với kích thước  $M \times N$  trong đó mỗi dòng tương ứng với một nút kiểm tra (CN), mỗi cột tương ứng với một nút biến (VN).

Ký hiệu:  $\alpha_{m,n}$  (hay còn được gọi là thông tin VTC) là thông tin truyền từ VN  $n$  (nút biến thứ  $n$ ) đến CN  $m$ ;  $\beta_{m,n}$  (hay còn được gọi là thông tin CTV) là thông tin truyền từ CN  $m$  đến VN  $n$ ;  $\gamma_n$  là thông tin nhận được từ kênh truyền;  $\tilde{\gamma}_n$  là từ mã được cập nhật. Để thực hiện thiết kế phần cứng, giả sử số bit biểu diễn các thông tin  $\alpha_{m,n}, \gamma_n, \tilde{\gamma}_n$  là  $\tilde{q}$  bit; thông tin  $\beta_{m,n}$  là  $q$  ( $q < \tilde{q}$ ). Tập hợp các giá trị của thông tin  $\tilde{q}$  bit được ký hiệu là  $\tilde{\mathcal{M}} = (-\tilde{Q}, \dots, -1, 0, +1, \dots, +\tilde{Q})$  với  $\tilde{Q} = 2^{\tilde{q}} - 1$ . Tương tự, tập hợp các giá trị của thông tin  $q$  bit là  $\mathcal{M} = (-Q, \dots, -1, 0, +1, \dots, +Q)$  với  $Q = 2^q - 1$ .

## 2.13 Kết luận chương

Chương 2 đã trình bày một cách khái quát về mã LDPC như hai cách biểu diễn mã, hai phân loại và mã QC-LDPC ứng dụng nhiều trên thực tế. Ngoài ra, chương 2 cũng giới thiệu các thuật toán giải mã, phân loại cấu trúc phần cứng và lịch trình giải mã. Nhìn chung, những nội dung được trình bày trong chương này sẽ là cơ sở lý thuyết cho các nội dung nghiên cứu ở các chương tiếp theo của luận án.

## CHƯƠNG 3. THUẬT TOÁN GIẢI MÃ SỬ DỤNG MÃ LDPC

### 3.1 Khảo sát sự ảnh hưởng của các giải thuật giải mã đến khả năng sửa lỗi của mã 5G LDPC

Thuật toán MS được mô tả theo công thức sau:

$$\beta_{m,n} = \left( \prod_{n' \in H(m) \setminus n} \text{sgn}(\alpha_{m,n'}) \right) \left( \min_{n' \in H(m)/n} |\alpha_{m,n'}| \right) \quad (3.1)$$

Ở khía cạnh phân cứng, quá trình xử lý các nút kiểm tra có dạng:

$$\beta_{m,n} = \left( \prod_{n' \in H(m) \setminus n} \text{sgn}(\alpha_{m,n'}) \right) \times \begin{cases} \min 2 & \text{vị trí của } |\alpha_{m,n'}| \text{ là } index\_min1 \\ \min 1 & \text{các vị trí khác} \end{cases} \quad (3.2)$$

với  $\min 1 = \min_{n \in H(m)/n} |\alpha_{m,n}|$ ;  $index\_min1$  là vị trí của  $\min 1$ . Trong trường hợp có nhiều giá trị  $\min 1$ ,  $index\_min1$  là vị trí thấp nhất;  $\min 2$  là giá trị cực tiểu thứ hai của các thông tin ngõ vào VTC  $\alpha_{m,n}$ .

Thuật toán Normalized Min-Sum (NMS) [26, 83]. Khi đó (3.1) được viết dưới dạng:

$$\beta_{m,n} = \alpha \cdot \left( \prod_{n' \in H(m) \setminus n} \text{sgn}(\alpha_{m,n'}) \right) \left( \min_{n' \in H(m)/n} |\alpha_{m,n'}| \right) \quad (3.3)$$

Thuật toán Offset Min-Sum (OMS) là trong quá trình xử lý các nút kiểm tra bằng cách sử dụng hệ số offset  $\beta > 0$ :

$$\beta_{m,n} = \left( \prod_{n' \in H(m) \setminus n} \text{sgn}(\alpha_{m,n'}) \right) \cdot \max \left( \min_{n' \in H(m)/n} |\alpha_{m,n'}| - \beta, 0 \right) \quad (3.5)$$

### 3.2 Các thuật toán giải mã đề xuất cho bộ giải mã LDPC

Thuật toán giải mã LDPC phân lớp thường gồm ba giai đoạn chính: xử lý các nút biên, xử lý các nút kiểm tra và cập nhật từ mã. Luận án sẽ đề xuất thuật toán giải mã dựa trên thuật toán Min-Sum (MS) được phát triển trên cơ sở toán học sử dụng logarit Jacobian [92]. Phần tiếp theo sẽ trình bày các hướng cải tiến so với giải thuật cơ sở MS.

### 3.2.1 Các thuật toán giải mã đề xuất sử dụng hai hệ số hiệu chỉnh các thông tin của quá trình xử lý các nút kiểm tra

#### a. Thuật toán Improved Offset Min-Sum (IOMS) đề xuất

Thuật toán đề xuất được trình bày ở bài báo khoa học [2] ở phần Danh mục công trình đã công bố.

Quá trình xử lý nút kiểm tra (CN) được viết như sau:

$$\beta_{m,n} = \prod_{n' \in H(m) \setminus n} \text{sign}(\alpha_{m,n'}) \cdot \max \left\{ \left( \gamma \cdot \min_{n' \in H(m) \setminus n} |\alpha_{m,n'}| \right) - \eta, 0 \right\} \quad (3.22)$$

với  $0 < \gamma < 1$ ;  $\eta > 0$ .

#### b. Thuật toán Advanced Offset Min-Sum (AOMS) đề xuất

Thuật toán đề xuất được trình bày ở bài báo khoa học [7] ở phần Danh mục công trình đã công bố.

Phương trình xử lý CN của thuật toán AOMS được viết dưới dạng:

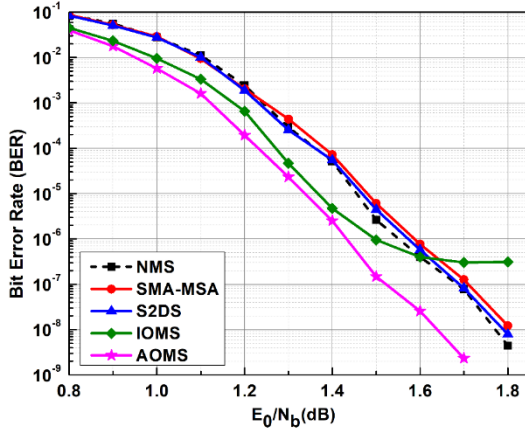
$$\beta_{m,n} = \prod_{n' \in H(m) \setminus n} \text{sign}(\alpha_{m,n'}) \cdot \begin{cases} \min 2' & \text{vị trí của } |\alpha_{m,n'}| \text{ là } \text{index\_min1} \\ \min 1' & \text{các vị trí khác} \end{cases} \quad (3.23)$$

trong đó:

$$\begin{aligned} \min 1' &= \max\{\min 1 - \delta, 0\} \\ \min 2' &= \begin{cases} \max\{\min 2 - \delta, 0\} & \text{nếu } \min 1 \geq \min 2 - \tau \\ \max\{\min 2 - (\tau + \delta), 0\} & \text{nếu } \min 1 \leq \min 2 - \tau \end{cases} \end{aligned} \quad (3.24)$$

với các hệ số  $\delta, \tau > 0$ .

Từ kết quả mô phỏng cho thấy thuật toán IOMS xuất hiện lỗi nền sớm hơn so với các thuật toán khác, lỗi nền bắt đầu ở giá trị BER 10<sup>-6</sup>. Cả hai thuật toán S2DS và SMA-MSA có kết quả gần như giống nhau và gần với kết quả hiệu suất sửa lỗi với NMS. Thuật toán AOMS có khả năng cải thiện kết quả giải mã và không xuất hiện lỗi nền khi BER 10<sup>-8</sup>. Ở giá trị BER này, thuật toán IOMS cải tiến có độ lợi giải mã tốt hơn 0.21 dB, 0.25 dB và 0.26 dB so với các thuật toán NMS, S2DS và SMA-MS.



Hình 3-8. Kết quả giải mã của các bộ giải mã khác nhau với chiều dài khối 8832 và tỷ lệ mã 1/2.

### 3.2.2 Các thuật toán giải mã đề xuất sử dụng hai hệ số hiệu chỉnh các thông tin của quá trình xử lý các nút kiểm tra và nút biến.

#### a. Thuật toán Hybrid Offset Min-Sum (HOMS) đề xuất.

Thuật toán đề xuất được trình bày ở bài báo khoa học [3] ở phần Danh mục công trình đã công bố.

Phương trình mô tả quá trình CN viết dưới dạng:

$$\beta_{m,n} = A \cdot \begin{cases} \left( \min_{n \in H(m)} |\alpha_{m,n}| \right) & \text{vị trí của } |\alpha_{m,n'}| \text{ là } index\_min1 \\ \max \left\{ \left( \left( \min_{n \in H(m)} |\alpha_{m,n}| \right) \right) - \beta, 0 \right\} & \text{các vị trí khác} \end{cases} \quad (3.26)$$

Trong đó  $A = \prod_{n' \in H(m) \setminus n} \text{sgn}(\alpha_{m,n'})$ ;  $\beta > 0$ ;  $index\_min1$  là vị trí của giá trị cực tiểu thứ nhất ( $min1$ ). Trong trường hợp nếu có nhiều hơn một giá trị  $min1$  thì  $index\_min1$  sẽ nhận vị trí thấp nhất.

Phương trình mô tả quá trình xử lý VN có dạng:

$$\alpha_{m,n} = \gamma_n + \text{sgn} \left( \sum_{m' \in H(n) \setminus m} \beta_{m',n} \right) \cdot \max \left( \sum_{m' \in H(n) \setminus m} \beta_{m',n} - \delta, 0 \right) \quad (3.27)$$

với hệ số  $\delta > 0$ .



## b. Thuật toán Variable Offset Min-Sum (VOMS) đề xuất

Thuật toán đề xuất được trình bày ở bài báo khoa học [3] ở phần Danh mục công trình đã công bố.

Trong quá trình xử lý CN, thuật toán VOMS hoạt động giống như với thuật toán OMS. Các thông tin VN được cập nhật theo phương trình sau với  $\tau$  ( $0 < \tau < 1$ )

$$\alpha_{m,n} = \gamma_n + \tau \cdot \sum_{m' \in H(n) \setminus m} \beta_{m',n} \quad (3.28)$$

## c. Thuật toán giải mã Enhanced single minimum Min-Sum (EsmMS) đề xuất

Thuật toán đề xuất được trình bày ở bài báo khoa học [6] ở phần Danh mục công trình đã công bố. Phương trình mô tả quá trình CN viết dưới dạng:

$$\beta_{m,n} = A \cdot \begin{cases} \left( \min_{n \in H(m)} |\alpha_{m,n}| + \omega \right) & |\alpha_{m,n}| = \left( \min_{n \in H(m)} |\alpha_{m,n}| \right) \\ \left( \min_{n \in H(m)} |\alpha_{m,n}| \right) & |\alpha_{m,n}| \neq \left( \min_{n \in H(m)} |\alpha_{m,n}| \right) \end{cases} \quad (3.29)$$

Phương trình mô tả quá trình xử lý VN có dạng:

$$\alpha_{m,n} = \gamma_n + \text{sgn} \left( \sum_{m' \in H(n) \setminus m} \beta_{m',n} \right) \cdot \max \left( \sum_{m' \in H(n) \setminus m} \beta_{m',n} - \delta, 0 \right) \quad (3.30)$$

với các hệ số  $A = \prod_{n' \in H(m) \setminus n} \text{sgn}(\alpha_{m,n'})$ ;  $\omega, \delta > 0$ .

## d. Kết luận

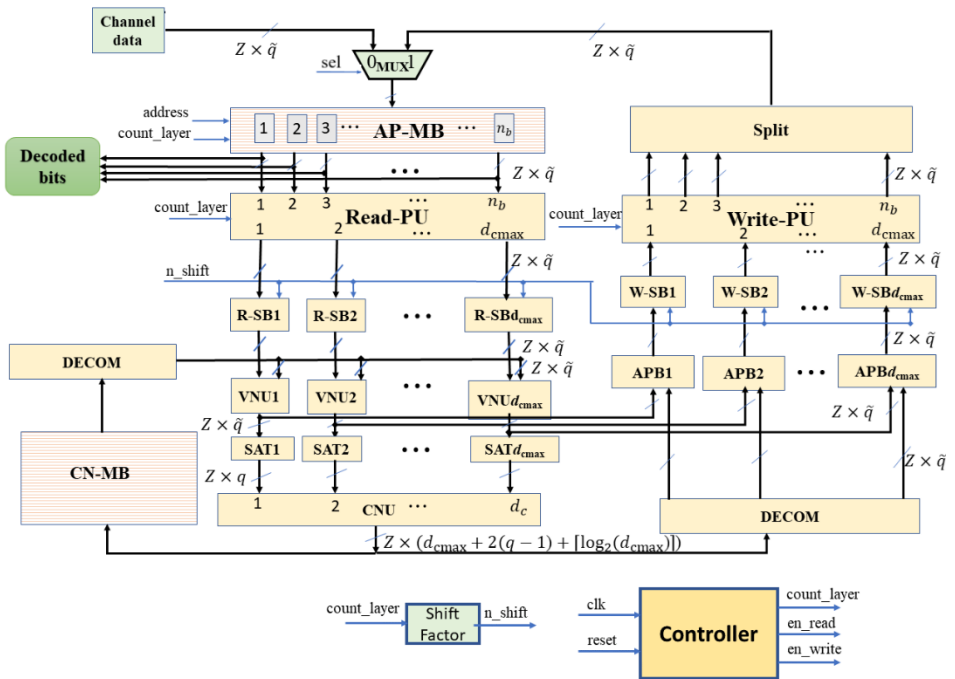
Trong chương này, năm thuật toán giải mã được đề xuất nhằm nâng cao chất lượng giải mã của thuật toán MS đã được trình bày. Các thuật toán IOMS và AOMS đề xuất đã sử dụng hai hệ số hiệu chỉnh vào quá trình xử lý CN. Hai thuật toán VOMS và HOMS đã tác động vào cả hai quá trình xử lý CN và VN. Điều chú ý nhất là thuật toán HOMS, EsmMS còn giảm đáng kể tải nguyên phần cứng của quá trình xử lý CN bằng phương pháp chỉ tính duy nhất giá trị

cực tiểu thứ nhất của các thông tin VTC ngõ vào, giá trị cực tiểu thứ hai được tính nhờ giá trị cực tiểu thứ nhất và hệ số hiệu chỉnh. Các hệ số hiệu chỉnh được tối ưu bằng phương pháp DE kết hợp với phương pháp mô phỏng. Kết quả mô phỏng cho thấy các thuật toán đề xuất đã cải thiện khả năng giải mã, đặc biệt thuật toán AOMS cho kết quả sửa lỗi tốt hơn khoảng 0.26 dB so với các thuật toán NMS, S2DS và SMA-MS khi áp dụng cho mã 5G LDPC. Trên cơ sở kết quả giải mã và tài nguyên phần cứng sử dụng của các thuật toán giải mã đã đề xuất, luận án sẽ sử dụng thuật toán HOMS và EsmMS làm cơ sở để thực hiện thiết kế bộ giải mã. Chi tiết cách thiết kế các bộ giải mã sẽ được trình bày chi tiết ở Chương 4.

# CHƯƠNG 4. THIẾT KẾ PHẦN CỨNG BỘ GIẢI MÃ LDPC VỚI MỤC ĐÍCH TIẾT KIỆM BỘ NHỚ VÀ CẢI THIỆN HIỆU SUẤT GIẢI MÃ

## 4.1 Cấu trúc bộ giải mã dựa trên thuật toán Min-Sum (MS) thông thường

Sơ đồ khối tổng quát của bộ giải mã MS tham khảo [105] như minh họa ở Hình 4-1.



Hình 4-1. Sơ đồ khối của bộ giải mã LDPC sử dụng thuật toán layer MS [105]

Nội dung thiết kế bộ giải mã MS tham khảo như thuật toán giải mã MS theo cấu trúc phân lớp, sơ đồ khối tổng quát, tóm tắt chức năng của các khối và nguyên tắc hoạt động của bộ giải mã MS được trình bày ở phần 4.1.1, 4.1.2, 4.1.3, 4.1.4.

#### 4.1.5 Khối xử lý các nút kiểm tra (CNU)

Nhiệm vụ chính của CNU là tính toán các thông tin CTV ( $\beta_{m,n}$ ).

#### 4.1.6 Khối xử lý các nút biến (VNU) và khối cập nhật từ mã (APB)

Mỗi thông tin VTC được tính theo công thức:  $\alpha_{m,n}^{new} = \tilde{\gamma}_n^{old} - \beta_{m,n}^{old}$  với  $\tilde{\gamma}_n^{old}$  là từ mã đọc từ bộ nhớ AP-MB;  $\beta_{m,n}^{old}$  là thông tin CTV đọc từ bộ nhớ CN-MB.

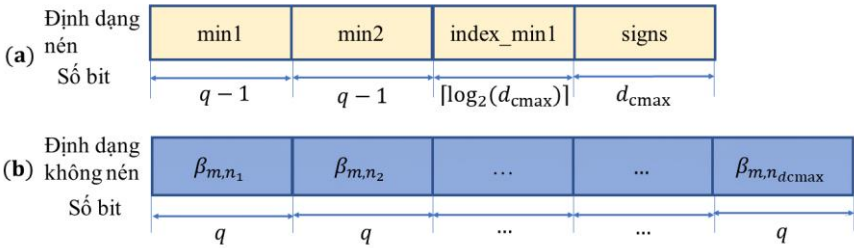
Các thông tin  $\alpha_{m,n}^{new}$  vừa nhận được sẽ đưa tới khối CNU để tính các thông tin  $\beta_{m,n}^{new}$ . Các từ mã được cập nhật theo công thức:  $\tilde{\gamma}_n^{new} = \alpha_{m,n}^{new} + \beta_{m,n}^{new}$ .

#### 4.1.7 Các bộ nhớ

Kiến trúc bộ giải mã LDPC theo lịch trình phân lớp cần có hai bộ lưu trữ: một bộ nhớ dùng để lưu các từ mã  $\tilde{\gamma}_n$  (AP-MB), một bộ nhớ dùng để lưu các thông tin CTV  $\beta_{m,n}$  ở ngõ ra khối CNU (CN-MB).

#### 4.1.8 Bộ giải nén (DECOM) của bộ giải mã MS

Vì các thông tin CTV lưu trữ trong bộ nhớ CN-MB có định dạng nén nên cần có bộ giải nén (DECOM) để chuyển thông tin CTV từ định dạng nén dạng không nén.



Hình 4-2. Các thông tin CTV ở ngõ ra khối CNU định dạng nén (a) và không nén (b).

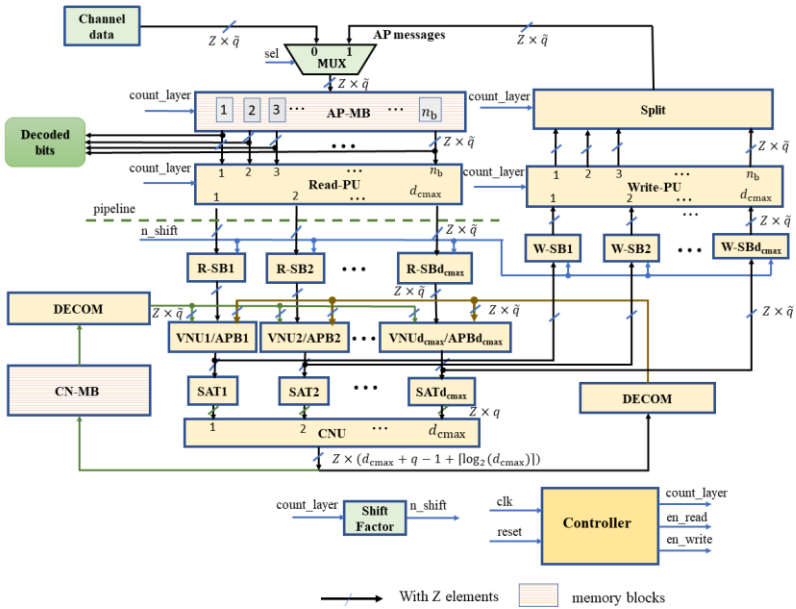
## 4.2 Thiết kế phần cứng của bộ giải mã HOMS đề xuất

### 4.2.1 Thuật toán Hybrid Offset Min-Sum (HOMS) đề xuất

Luận án sử dụng thuật toán HOMS phân lớp để thiết kế bộ giải mã.

### 4.2.2 Kiến trúc của bộ giải mã HOMS đề xuất

Sơ đồ khối tổng quát của bộ giải mã HOMS đề xuất minh học ở Hình 4-11.



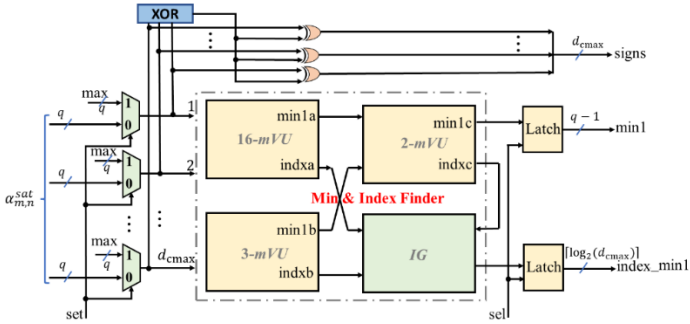
Hình 4-11. Sơ đồ khối tổng quát của bộ giải mã HOMS đề xuất

### 4.2.3 Chọn số bit biểu diễn thông tin

L luận án sẽ chọn số bit biểu diễn thông tin  $(q, \tilde{q}) = (4, 6)$  với mục đích giảm tài nguyên phần cứng trong đó số bit biểu diễn các thông tin  $\alpha_{m,n}, \gamma_n, \tilde{\gamma}_n$  là  $\tilde{q}$  bit; thông tin  $\beta_{m,n}$  là  $q$  ( $q < \tilde{q}$ ) và giảm không gian lưu trữ nhưng vẫn đảm bảo hiệu suất giải mã.

### 4.2.4 Khối xử lý các nút kiểm tra (CNU)

Khác với khối CNU của bộ giải mã MS, khối CNU trong thiết kế này bao gồm  $d_{cmax}$  khối ghép kênh (MUX), khối tìm giá trị cực tiểu và vị trí của nó (Min & Index Finder), một khối thực hiện phép nhân các dấu của thông tin ngõ vào và một số tín hiệu điều khiển. Trong thiết kế đề xuất không cần tính giá trị cực tiểu thứ 2 nên không có khối Compare & select.

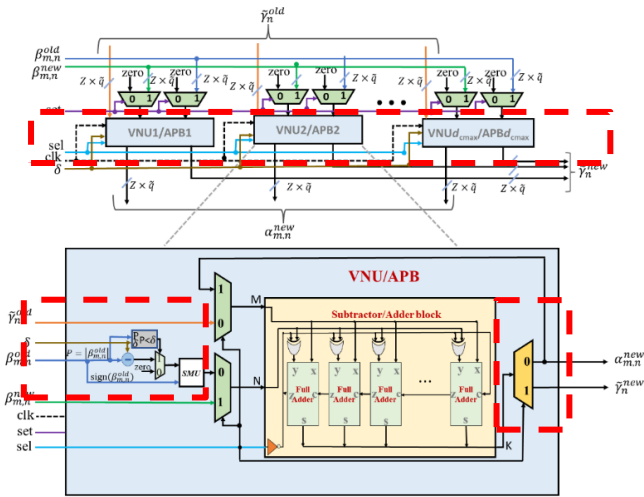


Hình 4-16. Cấu trúc phần cứng khối CNU của bộ giải mã HOMS đề xuất với  $d_{cmax} = 19$

#### 4.2.5 Khối xử lý các nút biến (VNU) và khối cập nhật từ mã (APB)

Các khối VNU thực hiện quá trình xử lý VN nghĩa là tìm các thông tin  $\alpha_{m,n}$ , còn các khối APB xử lý quá trình cập nhật các từ mã  $\tilde{\gamma}_n$ . Ở bộ giải mã MS thông thường, hai khối này được thiết kế riêng biệt. Sự khác nhau cơ bản của hai quá trình xử lý này là khối VNU thực hiện phép trừ còn khối APB dùng phép cộng. Vì vậy, luận án đã kết hợp hai khối này thành một khối VNU/APB và sử dụng thêm tín hiệu điều khiển “sel” để chọn thực hiện chức năng khối VNU hay chức năng khối APB.

Nhằm tăng hiệu suất giải mã và giảm bớt tính phức tạp cho quá trình xử lý CN, quá trình xử lý các nút biến của thuật toán HOMS đề xuất được hiệu chỉnh bằng hệ số  $\delta > 0$  lên thông tin VTC. Trước khi đưa vào Bộ trừ/cộng (Subtractor/Adder), giá trị  $\beta_{m,n}^{old}$  thực hiện  $\beta_{m,n}^{old} - \delta$  hay gán bằng 0 nếu hiệu  $\beta_{m,n}^{old} - \delta < 0$ .



Hình 4-17. Cấu trúc phần cứng khối VNU/APB của bộ giải mã HOMS.

#### 4.2.6 Bộ nhớ CN-MB của bộ giải mã HOMS đề xuất

Độ rộng bộ nhớ CN-MB của bộ giải mã HOMS được xác định bởi:

$$W = Z \times (q - 1 + \lceil \log_2(d_{\text{cmax}}) \rceil + d_{\text{cmax}})$$

trong trường hợp  $d_{\text{cmax}} = 19 ; q = 4$  ta có:  $W = 27Z$ .

Vì các thông tin CTV ở ngõ ra khỏi CNU của tất cả các phân lớp đều phải lưu trữ vào bộ nhớ nên bộ nhớ CN-MB cần chứa  $m_b \times Z \times (q - 1 + \lceil \log_2(d_{\text{cmax}}) \rceil + d_{\text{cmax}}) = 124416$  bit. Dung lượng bộ nhớ CN-MB của bộ giải mã MS và bộ giải mã HOMS đề xuất được so sánh ở Bảng 4-1.

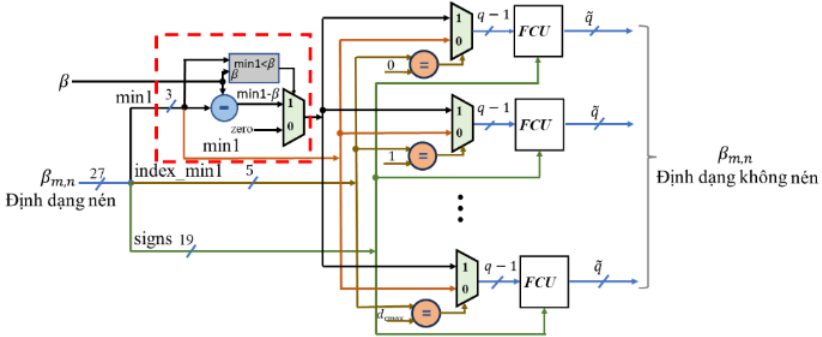
Bảng 4-1. So sánh bộ nhớ CN-MB của bộ giải mã MS thông thường và bộ giải mã HOMS đề xuất

Bộ nhớ CN-MB của bộ giải mã MS thông thường			Bộ nhớ CN-MB của bộ giải mã HOMS đề xuất		
Độ rộng	Độ sâu	Tổng	Độ rộng	Độ sâu	Tổng
30Z	24	720Z	27Z	24	648Z

Trong bộ giải mã HOMS, bộ nhớ AP-MB cần lưu 52992 bit và bộ nhớ CN-MB trữ 124416 bit. Kết quả dung lượng bộ nhớ cần có là 173.25 kb.

### 4.2.7 Bộ giải nén (DECOM) của bộ giải mã HOMS

Kiến trúc bộ DECOM của bộ giải mã HOMS đề xuất được minh họa ở Hình 4-18. Phần trong khung nét đứt là phần được cải tiến từ bộ giải mã MS.



Hình 4-18. Kiến trúc bộ DECOM của bộ giải mã HOMS

## 4.3 Thiết kế bộ giải mã Enhanced single minimum Min-Sum (EsmMS) đề xuất

### 4.3.1 Thuật toán giải mã EsmMS đề xuất

L luận án sử dụng thuật toán giải mã EsmMS phân lớp để thiết kế bộ giải mã.

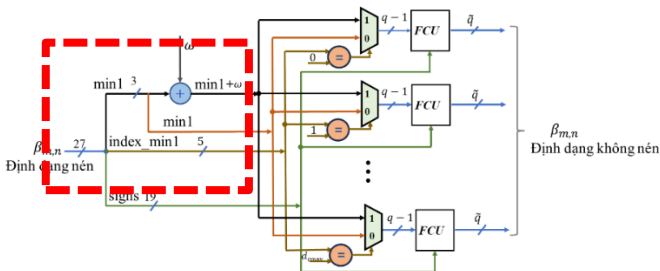
### 4.3.2 Kiến trúc và nguyên tắc hoạt động của bộ giải mã EsmMS đề xuất

Sơ đồ khối và nguyên tắc hoạt động của bộ giải mã EsmMS đề xuất trên cơ bản giống với bộ giải mã HOMS như đã giới thiệu ở phần 4.2. Tuy nhiên, khi thiết kế bộ giải mã EsmMS đề xuất, chúng tôi sẽ cải tiến khối DECOM và khối CN-MB, cụ thể sẽ được trình bày sau đây.

### Khối DECOM.

Phần trong khung nét đứt là phần cải tiến so với khối DECOM của bộ giải mã MS.





Hình 4-19. Kiến trúc của khối DECOM của bộ giải mã EsmMS.

**Bộ nhớ CN-MB của bộ giải mã EsmMS.**

Nếu chúng ta thực hiện với tất cả các lớp bộ nhớ CN-MB có độ rộng như nhau tương ứng  $d_{\text{cmax}} = 19$  sẽ làm tiêu tốn khá nhiều bộ nhớ không cần thiết. Chính vì vậy, chúng tôi đề xuất giải pháp chia bộ nhớ phù hợp với sự thay đổi của bậc nút kiểm tra. Cụ thể là, bộ nhớ CN-MB sẽ được chia thành hai bộ nhớ thành phần: CN-MB1 và CN-MB2.

Bảng 4-2. Dung lượng bộ nhớ CN-MB của các bộ giải mã MS, HOMS và EsmMS

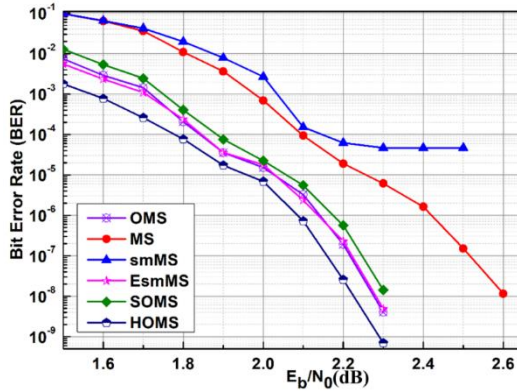
CN-MB của bộ giải mã MS			CN-MB của bộ giải mã HOMS			CN-MB của bộ giải mã EsmMS					
						Trước khi chia			Sau khi chia		
<i>W</i>	<i>D</i>	<i>Tổng</i>	<i>W</i>	<i>D</i>	<i>Tổng</i>	<i>W</i>	<i>D</i>	<i>Tổng</i>	<i>W</i>	<i>D</i>	<i>Tổng</i>
30Z	24	720Z	27Z	24	648Z	27Z	24	648Z	17Z	20	448Z
									27Z	4	

Dựa vào tính chất đặc biệt của mã 5G LDPC, sau khi sử dụng kỹ thuật chia bộ nhớ thì bộ nhớ CN-MB của bộ giải mã EsmMS tiết kiệm 37% bộ nhớ so với bộ giải mã MS và khoảng 29% so với bộ giải mã HOMS.

**4.4 Các kết quả thiết kế các bộ giải mã HOMS và EsmMS đề xuất**

**4.4.1 Kết quả giải mã**

Chúng tôi đã thực hiện mô phỏng Monte Carlo cho các thuật toán giải mã khác nhau sử dụng ma trận BG1 cho các mã 5G LDPC với tỷ lệ mã 1/2 và chiều dài mã 8832. Hệ số mở rộng  $Z = 192$ .



Hình 4-21. Kết quả BER của các bộ giải mã khác nhau với kích thước (8832, 4608) và tỷ lệ mã 1/2

Từ Hình 4-21, ở giá trị BER  $10^{-4}$  và  $10^{-8}$ , kết quả sửa lỗi của hai thuật toán HOMS và EsmMS đề xuất tốt hơn kết quả của thuật toán smMS và MS lên tới 0.32 dB và 0.38 dB.

Thuật toán EsmMS và OMS có hiệu suất giải mã gần như nhau ở giá trị BER  $10^{-6}$  và  $10^{-4}$ . Tuy nhiên, kết quả sửa lỗi của thuật toán EsmMS kém hơn của thuật toán HOMS khoảng 0.07 dB.

#### 4.4.2 Kết quả thực hiện phần cứng

Bộ giải mã LDPC được viết bằng ngôn ngữ phần cứng Verilog HDL (Verilog Hardware Description Language). Kết quả phần cứng được thực hiện trên Kintex UltraScale+ FPGA sử dụng phần mềm Xilinx Vivado 2021.2. Chúng tôi thực hiện mã 5G LDPC với chiều dài từ mã 8832 và tỷ lệ mã 1/2. Số vòng lặp giải mã cực đại là 10. Số bit biểu diễn các thông tin là  $(q, \tilde{q}) = (4, 6)$ . Tốc độ xử lý được tính bằng công thức [56]:

$$T = \frac{N \times F_{\max}}{L \times n_{iter} \times n_{cyc}} \text{ [Mbps]} \quad (4.8)$$

trong đó  $L$ : số lớp giải mã,  $n_{cyc}$ : số chu kỳ xung clock cần để thực hiện một lớp giải mã,  $n_{iter}$ : số vòng lặp cực đại,  $N$ : chiều dài từ mã và  $F_{\max}$  [MHz]: tần số làm việc cực đại.

Các bộ giải mã được thực hiện với những đặc tính khác nhau như chiều dài từ mã, số bit biểu diễn thông tin, thuật toán và số vòng lặp, để so sánh các thiết kế chúng ta cần dựa trên một hệ số chuẩn. Trong luận án này, chúng tôi sử dụng hệ số sử dụng hiệu quả phần cứng HUE (Hardware Usage Efficiency). Công thức tính hệ số HUE được xác định [56]:

$$HUE = \frac{H_u}{L \times T} \tag{4.9}$$

trong đó  $L$ : số lớp giải mã,  $T$ [Mbps]: tốc độ xử lý,  $H_u$  phản ánh bộ giải mã LDPC sử dụng bao nhiêu tài nguyên FPGA.

Kết quả phần cứng được so sánh với một số bộ giải mã LDPC tham khảo như liệt kê ở Bảng 4-3.

Bảng 4-3. Kết quả thực hiện trên FPGA và so sánh với các bộ giải mã tham khảo

Đặc tính	EsmMS [đề xuất]	HOMS [đề xuất]	2022 [117]	2021 [85]	2020 [50]
FPGA	Xilinx Kintex Ultrascale+	Xilinx Kintex Ultrascale+	Xilinx Virtex-7	Xilinx Kintex-7	Xilinx Kintex Ultrascale+
Số bit biểu diễn thông tin ( $q, \tilde{q}$ )	(4,6) bit	(4,6) bit	(8,8) bit	(8,8) bit	(8,8) bit
Tốc độ (Gbps)	2.63	2.82	2.85	0.391-1.1	3.2-4.9
Chiều dài mã	8832	8832	10368	3456	10368-26112
Hệ số mở rộng $Z$	192	192	384	384	384
Tần số làm việc cực đại (MHz)	142.8	153.5	255	160	404.8
Tỷ lệ mã	1/2	1/2	22/27	1/2-5/6	22/68-22/27
Số lớp	24	24	7	5-46	7-46

Số vòng lặp	10	10	11	8	10
Tầng pipeline	2	2	13	-	13
Dung lượng bộ nhớ (kb)	137.76	173.25	3888	7146	4914
Thuật toán giải mã	EsmMS	HOMS	OMS	OMS	OMS
Tài nguyên phần cứng H <sub>u</sub>	283094	314968	4204122	7438394	5233977
Loại bộ nhớ	BRAM	BRAM	BRAM	BRAM	BRAM
HUE	4.49	4.65	20.85	147.54	23.12
Số lần tăng HUE	1	1.04	4.64	32.9	5.15

Từ Bảng 4-3, ta thấy tốc độ xử lý của bộ giải mã HOMS đề xuất bằng 2.82 Gbps, tương đương với tốc độ xử lý của bộ giải mã trong thiết kế [117] và cao hơn gấp 2.5 lần so với của bộ giải mã [85]. Tốc độ xử lý của bộ giải mã EsmMS đề xuất đạt 2.63 Gbps chỉ chậm hơn 7% so với tốc độ xử lý của bộ giải mã HOMS.

Các bộ giải mã HOMS và EsmMS đề xuất có hệ số HUE bằng 4.65 và 4.49 tài nguyên phần cứng/lớp.Mbps, chứng tỏ bộ giải mã HOMS và EsmMS sử dụng tài nguyên phần cứng tiết kiệm hơn 4.5-5 lần so với các bộ giải mã ở [117] hay [50].

#### **4.5 Kiểm tra tính ứng dụng trong mạng 5G của bộ giải mã LDPC đề xuất**

Theo 3GPP TS 38.212, kênh truyền AWGN là một kênh truyền cơ bản được dùng thực hiện cho thuật toán giải mã LDPC cho mạng 5G [118].

Để các hệ thống ứng dụng hoạt động trong mạng 5G, tốc độ xử lý phải đạt 1-10Gbps. Từ kết quả phân cứng tốc độ xử lý 2.43 Gbps (bộ giải mã EsmMS) và 2.82 Gbps (bộ giải mã HOMS) thỏa tốc độ xử lý ứng dụng cho mạng 5G NR [58, 79, 120]. Như vậy, các bộ giải mã đề xuất có thể áp dụng cho mạng 5G.

#### **4.6 Kết luận**

Trong chương này, chúng tôi đã nghiên cứu và thiết kế hai bộ giải mã dựa trên các thuật toán giải mã đề xuất HOMS và EsmMS. Trong quá trình xử lý của các nút kiểm tra của hai thuật toán này đều chỉ cần xác định duy nhất giá trị cực tiểu thứ nhất của các thông tin ngõ vào và vị trí của nó thay vì phải xác định hai giá trị cực tiểu đầu tiên như ở thuật toán MS thông thường. Chính vì vậy, nó đã giúp tiết kiệm tài nguyên phần cứng một cách đáng kể. Thông tin ngõ ra của khối xử lý CNU được ghi ở định dạng nén giúp giảm không gian lưu trữ. Mặt khác, các kỹ thuật khác cũng được áp dụng như lịch trình layer, cấu trúc bán song song, chọn số bit biểu diễn thông tin phù hợp đã góp phần thu được bộ giải mã HOMS đề xuất với mục đích tiết kiệm phần cứng (so với một số bộ giải mã, bộ giải mã HOMS tiết kiệm lên đến 4.5-5 lần). Trên cơ bản, thiết kế phần cứng của bộ giải mã EsmMS gần giống với bộ giải mã HOMS. Tuy nhiên, dựa vào tính chất không đều của mã 5G LDPC, bộ giải mã EsmMS đã sử dụng kỹ thuật phân chia bộ nhớ giúp tiết kiệm lên đến 4.5 lần không gian lưu trữ. Hơn thế nữa, kết quả mô phỏng cho thấy hiệu suất giải mã cũng cải thiện đến 0.1 dB so với bộ giải mã SOMS.

## CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 5.1 Kết luận

Bộ giải mã LDPC có các ưu điểm như giải mã tốt, tốc độ xử lý cao và hiệu quả về tài nguyên phần cứng. Bộ giải mã đã được sử dụng cho nhiều chuẩn viễn thông, đặc biệt là hệ thống thông tin mới 5G. Chính vì vậy, bộ giải mã LDPC luôn được các nhà nghiên cứu chú ý và cải tiến. Trong luận án này, tác giả đã hoàn thành được các mục tiêu nghiên cứu đặt ra.

Nghiên cứu thuật toán giải mã Min-Sum (MS) và đề xuất các giải thuật nhằm cải thiện kết quả sửa lỗi của thuật toán MS bằng cách áp dụng hệ số hiệu chỉnh vào quá trình xử lý các nút kiểm tra (CN), quá trình xử lý các nút biên (VN). Các đề xuất đã được chứng minh trên cơ sở lý thuyết toán học và được kiểm chứng bằng kết quả mô phỏng Monte Carlo cho mã 5G LDPC. Thực chất, các hệ số hiệu chỉnh nhằm làm giảm việc ước lượng quá mức của phương pháp lấy xấp xỉ của thuật toán MS. Trong công trình nghiên cứu này đã đưa ra hai hướng tiếp cận: hướng thứ nhất là cải thiện hiệu suất giải mã bằng cách sử dụng hai hệ số hiệu chỉnh lên quá trình xử lý các nút kiểm tra (CN) (các thuật toán đề xuất IOMS và AOMS). Kết quả sửa lỗi của các thuật toán đề xuất này có độ lợi giải mã lên đến 0.26 dB so với các thuật toán NMS, S2DS và SMA-MS. Ngoài ra, kết quả mô phỏng còn cho thấy số vòng lặp ảnh hưởng rõ rệt tới độ lợi giải mã: độ lợi tăng 0.64 dB ở BER  $10^{-5}$  với số vòng lặp tăng từ 10 lên 20. Khối xử lý các nút kiểm tra là khối phức tạp nhất trong bộ giải mã, hầu hết các nghiên cứu khác đều tập trung sử dụng hiệu chỉnh vào khối này. Điều này dẫn tới việc làm tăng thêm tính phức tạp, luận án này đã đưa hướng giải pháp thứ 2 là cải thiện chất lượng giải mã của bộ giải mã LDPC bằng cách áp dụng các hệ số hiệu chỉnh ở cả hai quá trình xử lý nút kiểm tra (CN) và nút biên (VN). Với các chiều dài từ mã, tỷ lệ mã khác nhau cho mã 5G LDPC cho thấy: các thuật toán đề xuất (HOMS, VOMS và EmsMS) cho hiệu suất giải mã tốt hơn

lên tới 0.38 dB so với thuật toán tham khảo. Không những cho kết quả sửa lỗi cải thiện, hai thuật toán (HOMS và EmsMS) trong quá trình xử lý CN chỉ cần tìm duy nhất giá trị cực tiểu và vị trí của nó giúp giảm tài nguyên phần cứng sử dụng. Ở đây, hai cách tiếp cận của hai thuật toán khác nhau trong việc hiệu chỉnh các thông tin CTV. Tiêu chí thiết kế bộ giải mã đặt ra là tiết kiệm tài nguyên phần cứng và cải thiện hiệu suất giải mã, luận án đã chọn ra hai thuật toán HOMS và EmsMS làm cơ sở để thiết kế. Kết quả này đã được công bố ở công trình [3-5,7] trong phần Danh mục các công trình công bố của luận án.

Thiết kế bộ giải mã thường được tập trung vào ba hướng cơ bản: cải thiện hiệu suất giải mã, tăng tốc độ xử lý và tiết kiệm tài nguyên phần cứng. Luận án đã tập trung vào hai hướng là sử dụng hiệu quả tài nguyên phần cứng và tăng độ lợi giải mã. Việc thiết kế các khối xử lý riêng rẽ khá đơn giản, thiết kế bộ giải mã LDPC hoàn chỉnh phức tạp hơn vì các đặc trưng và thông số có sự ảnh hưởng và đánh đổi. Các đặc tính như lịch trình giải mã, cấu trúc thiết kế và số bit biểu diễn thông tin ảnh hưởng nhiều tới thiết kế, tác giả đã nghiên cứu để lựa chọn được các giải pháp thích hợp. Cụ thể, luận án đã chọn ra lịch trình phân lớp, cấu trúc bán song song, số bit biểu diễn thông tin và kỹ thuật nén thông tin lưu trữ trong bộ nhớ. Đặc biệt, quá trình xử lý CN của bộ giải mã HOMS đề xuất chỉ cần tính duy nhất giá trị cực tiểu thứ nhất của các thông tin VTC ngõ vào và vị trí của nó. Điều này giúp tiết kiệm bộ nhớ lưu trữ các thông tin CN khoảng 10% so với các bộ giải mã tham khảo khác. Bộ giải mã HOMS có tốc độ xử lý đạt 2.82 Gbps thỏa tiêu chuẩn tốc độ xử lý cho mạng 5G, cải thiện hiệu suất sửa lỗi lên 0.38 dB so với bộ giải mã MS. Đặc biệt, bộ giải mã HOMS đề xuất có hệ số HUE bằng 4.65 tài nguyên phần cứng/lớp.Mbps, chứng tỏ bộ giải mã HOMS sử dụng tài nguyên phần cứng tiết kiệm ít nhất 4.5-5 lần so với các bộ giải mã tham khảo. Kết quả này đã được công bố ở công trình [1, 2] trong phần Danh mục các công trình công bố của luận án.

Trong thiết kế bộ giải mã dạng phân lớp cần có hai bộ nhớ: bộ nhớ AP-MB lưu trữ thông tin được kết nối các khối xử lý giữa các phân lớp và bộ nhớ CN-MB

lưu trữ các thông tin các nút kiểm tra. Do vậy, các bộ nhớ chiếm một phần trong toàn bộ diện tích chip và mức tiêu thụ công suất trong bộ giải mã LDPC. Dung lượng bộ nhớ phụ thuộc vào các bậc nút kiểm tra, trong khi đó bậc nút kiểm tra trong mã 5G có đặc điểm phân bố "không đều". Chính vì vậy, nhằm tránh việc tiêu tốn bộ nhớ không cần thiết, bộ nhớ các nút kiểm tra CN-MB được chia thành hai bộ nhớ con dựa vào giá trị bậc nút kiểm tra. Ngoài ra, thay vì phải tìm hai giá trị cực tiểu thứ nhất và thứ hai như các bộ giải mã khác, thuật toán giải mã EsmMS yêu cầu chỉ cần tìm giá trị cực tiểu thứ nhất của các thông tin ngõ vào VTC và vị trí của nó. Thiết kế bộ giải mã EsmMS đề xuất trên cơ bản giống như bộ giải mã HOMS, tuy nhiên các khối DECOM cần cải tiến cho phù hợp. Kết quả thực hiện trên FPGA với mã 5G cho thấy bộ giải mã EsmMS đề xuất có tốc độ xử lý đạt 2.63 Gbps, hệ số HUE bằng 4.49 tài nguyên phần cứng/lớp.Mbps và tiết kiệm 4.5-5 lần so với bộ giải mã tham khảo. Kết quả cụ thể được công bố ở công trình [6] trong phần Danh mục các công trình công bố của luận án.

Tác giả đã thực hiện việc kiểm tra tính đúng đắn của các bộ giải mã thiết kế ứng dụng cho hệ thống thông tin mới, cụ thể là mạng 5G. Chính vì vậy, các kết quả nghiên cứu của luận án có thể bổ sung thêm mô hình về bộ giải mã LDPC. Các giải pháp thiết kế bộ giải mã LDPC trong luận án có thể làm cơ sở và gợi ý cho các nhà sản xuất ứng dụng trong các hệ thống thông tin mới.

## **5.2 Hướng phát triển.**

Tiết kiệm phần cứng, tốc độ xử lý và công suất là những thách thức của hệ thống thông tin. Tốc độ xử lý cao giúp hỗ trợ lưu lượng truy cập ngày càng tăng, trong khi công suất tiêu thụ liên quan đến tính bền vững của thiết bị và là mối lo ngại lớn đối với các thiết bị di động. Luận án này đã chú trọng vào thiết kế các bộ giải mã với mục đích tiết kiệm tài nguyên phần cứng đặc biệt là tiết kiệm bộ nhớ. Ngoài ra, thiết kế cũng cải thiện được tốc độ xử lý và cải thiện hiệu suất giải mã.



Vì vậy, một trong những hướng phát triển đầu tiên của luận án này là nghiên cứu tác động của các giải pháp đề xuất đến mức công suất tiêu thụ. Vấn đề tối ưu công suất tiêu thụ đặt ra cũng có thể tương ứng với yêu cầu tìm ra giải pháp cân đối giữa công suất tiêu thụ, diện tích sử dụng và tần số làm việc hay tốc độ xử lý.

## DANH MỤC CÔNG TRÌNH ĐÃ CÔNG BỐ

### Tạp chí quốc tế

1. **Bich Ngoc Tran-Thi**, Nguyen-Ly, T. T., & Hoang, T.. An FPGA Design with High Memory Efficiency and Decoding Performance for 5G LDPC Decoder. *Electronics*, 12(17), p. 3667, 2023. DOI: 10.3390/electronics12173667. (**SCIE, Scopus, Q2, IF=2.9**).
2. **Bich Ngoc Tran-Thi**, T. T. Nguyen-Ly, T. Hoang, "Further Improvements in Decoding Performance for 5G LDPC Codes Based on Modified Check-Node Unit," *Radioengineering Journal*, vol. 32(2), pp. 226-235, 2023. DOI: 10.13164/re.2023.0226 (**SCIE, Scopus, Q3, IF=0.947**).
3. **Bich Ngoc Tran-Thi**, Thien Truong Nguyen-Ly, and Trang Hoang, "High-Performance and Low-Complexity Decoding Algorithms for 5G Low-Density Parity-Check Codes," *Journal of Communications*, vol. 17, no. 5, pp. 358-364, May 2022. DOI: 10.12720/jcm.17.5.358-364. (**Scopus, Q3**).

### Tạp chí trong nước

4. **Trần Thị Bích Ngọc**, Nguyễn Hồng Hòa, Hoàng Trang, Nguyễn Lý Thiên Trường. "Khảo sát các giải thuật cho bộ giải mã LDPC ứng dụng trong mạng 5G". *Journal of Transportation Science and Technology*, Vol. 10- 4, pp. 9 – 15, 2022.
5. Nguyen Hong Hoa, **Tran Thi Bich Ngoc**, Hoang Trang, H. "Evaluate error correction performance of binary repeat accumulate code and Quasi Cyclic Low-density Parity-check code in 5G new-radio". *Journal of Science Technology and Food*, 20(4), pp. 23-32, 2020.

### Kỷ yếu hội nghị quốc tế

6. **Bich Ngoc Tran-Thi**, T. T. Nguyen-Ly and T. Hoang, "A Memory-Saving Approach for 5G New Radio Low-Density Parity-Check Decoders," 2023

*International Symposium on Electrical and Electronics Engineering (ISEE)*, Ho Chi Minh, Vietnam, pp. 52-57, 2023. DOI: 10.1109/ISEE59483.2023.10299826

7. **Bich Ngoc Tran-Thi**, Nguyen-Ly, T. T., Hong, H. N., & Hoang, T. An improved offset min-sum LDPC decoding algorithm for 5G new radio. In *2021 International Symposium on Electrical and Electronics Engineering (ISEE)*, IEEE, Ho Chi Minh, Vietnam, pp. 106-109, 2021. DOI: 10.1109/ISEE51682.2021.9418782