

GIẢI MÃ TÍN HIỆU 4-QAM PHỤC VỤ TRUYỀN NHẬN TÍN HIỆU QUANG SỬ DỤNG FPGA

Lương Công Duẩn, Vũ Anh Đào

* Học Viện Công Nghệ Bưu Chính Viễn Thông

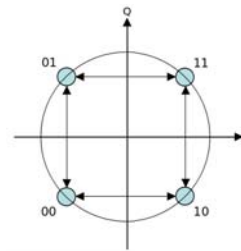
Tóm tắt: QAM được sử dụng nhiều trong cả giao tiếp dữ liệu tương tự và số. FPGA là một nền tảng đáng tin cậy trong thiết kế và triển khai đặc biệt trong lĩnh vực thông tin liên lạc. Bài báo trình bày mô hình thuật toán giải mã tín hiệu QAM-4 phục vụ truyền nhận tín hiệu quang sử dụng nền tảng FPGA(Field-Programmable Gate Array). Mô hình có vai trò đọc dữ liệu đầu vào ADC, xử lý tính toán dữ liệu theo chuẩn QAM-4 và đưa ra dữ liệu số được truyền tải tương ứng. Mô hình được xây dựng dựa trên 5 khối sau: Khối đọc dữ liệu ADC, Khối đệm lưu trữ dữ liệu tạm, Khối đồng bộ, Khối xác định tín hiệu số thông qua hàm tích chập và Khối dữ liệu đầu ra. Mô hình sau khi được xây dựng được tiến hành thử nghiệm trên Board FPGA Spartan3. Chương trình được thiết kế bằng ngôn ngữ VHDL dựa trên phần mềm Xilinx ISE và Modelsim Tool. Thiết kế của nhóm tác giả tổng hợp sử dụng khoảng 45K cổng, hoạt động ở tần số 50MHz và tiêu thụ khoảng 32mW.

Từ khóa: QAM, OFDM, DDS, FPGA.

I. ĐẶT VẤN ĐỀ

Quadrature Amplitude Modulation (QAM)[1] là hình thức điều chế dữ liệu được sử dụng rộng rãi để điều chế dữ liệu thành sóng mang được sử dụng rộng rãi trong thông tin vô tuyến và thông tin quang. QAM chứng tỏ được những ưu điểm nổi bật so với các phương pháp điều chế cũ như ASK, BPSK, QPSK. Điều chế QAM cho phép thay đổi số lượng dữ liệu thông tin trên mỗi symbol từ đó có các phiên bản QAM khác nhau như: 4-QAM, 8-QAM, 16-QAM, 32-QAM, 64-QAM[2]... Việc thay đổi thông số bit/symbol cho phép nâng cao tốc độ truyền dữ liệu, tiết kiệm Bandwidth tuy nhiên cũng đặt ra yêu cầu cao về việc tính toán và xử lý thông tin trên các nền tảng phần cứng. Về bản chất, truyền tải dữ liệu theo điều chế QAM có sự thay đổi dữ liệu truyền tải cả về pha và tần số. Tuy nhiên, với 4-QAM việc điều chế chỉ dừng lại ở điều chế biên độ với các góc lệch pha lần lượt là: 45° , 135° , 225° và 315° . Tín hiệu được truyền tải đi liên tục thay đổi pha tín hiệu. Bên phía thu tín hiệu cần tiến hành đồng bộ và xác định được pha của tín hiệu được truyền đến từ đó xác định dữ liệu số đang được truyền tải trên sóng

mang. Với các QAM bậc cao hơn, thuật toán giải mã cần tiến hành đồng bộ và xác định được biên độ của tín hiệu cần truyền đến.



Hình 1: Lược đồ phân bố của 4-QAM

Ở bài báo này, nhóm tác giả chỉ dừng lại ở khai thác một mô hình thuật toán để giải điều chế tín hiệu 4-QAM để thử nghiệm khả năng tính toán và đáp ứng của thuật toán trên nền tảng phần cứng FPGA để phục vụ các nghiên cứu tiếp theo của nhóm tác giả về các phương pháp giải mã tín hiệu theo chuẩn điều chế QAM tích hợp trên các thiết bị phần cứng có sử dụng các thuật toán liên quan đến trí tuệ nhân tạo như CNN, RNN[3].

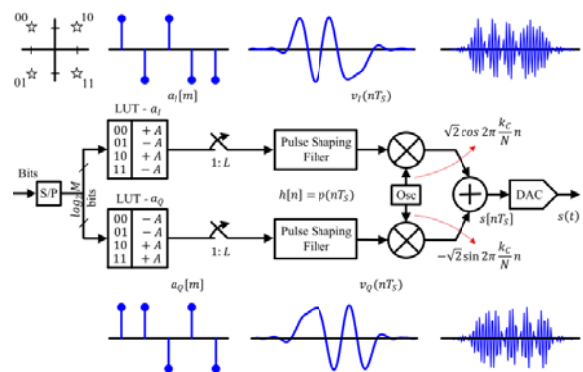
Bài báo được phân tích theo các phần chính sau:

- Phân tích lý thuyết
- Xây dựng mô hình
- Xây dựng chương trình cho các khối
- Thử nghiệm, đánh giá

II. PHÂN TÍCH LÝ THUYẾT, XÂY DỰNG MÔ HÌNH

2.1. Phân tích lý thuyết

a. Điều chế tín hiệu QAM[4][5]



Hình 2: Sơ đồ khối bộ điều chế tín hiệu 4-QAM trên FPGA

Dữ liệu số được đồng bộ với bộ phát tín hiệu theo các

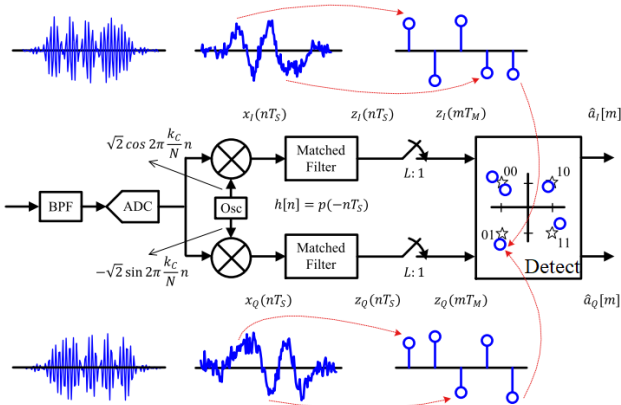
Tác giả liên hệ: Lương Công Duẩn, Vũ Anh Đào

Email: duanlc@ptit.edu.vn, daova@ptit.edu.vn

Đền tòa soạn: 11/2019, chỉnh sửa: 12/2019, chấp nhận đăng: 12/2019

khe thời gian tạo thành chuỗi dữ liệu số. Chuỗi dữ liệu số được đưa đến khối S/P có vai trò tách các tín hiệu ở vị trí lẻ và chẵn lần lượt vào 2 khối I và Q. Các dữ liệu này được lưu trữ trong bộ đệm và được đi qua bộ LUT (Look Up Table) của FPGA để xác định các pha tương ứng cần điều chế đầu ra. Ở sử dụng 2 bảng LUT có nội dung khác nhau cho khối I và khối Q. Sau khi đã xác định pha, dữ liệu của 2 khối được tạo thành dữ liệu xung theo tốc độ lấy mẫu được cài đặt tại L. Tại đây hệ thống sử dụng LUT để giảm thời gian tính toán. Ngoài ra, nhóm tác giả đã sử dụng công cụ DDS[6] của Xilinx để hỗ trợ tạo ra tín hiệu chuẩn sin trước khi đưa dữ liệu ra DAC. Sau đó, tín hiệu đầu ra được đưa qua bộ lọc dạng xung và đưa qua bộ trộn tần và được tổng hợp thành tín hiệu ra trước khi đưa ra DAC để xuất thành tín hiệu theo miền thời gian.

b. Giải điều chế tín hiệu QAM[7][8]



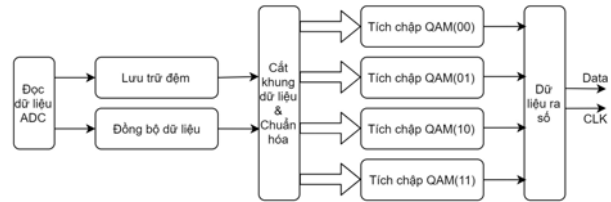
Hình 3: Sơ đồ khối bộ giải điều chế dữ liệu 4-QAM trên FPGA

Dữ liệu đầu vào sau khi đi qua môi trường truyền dữ liệu và đưa qua bộ lọc BPF sau đó đưa qua khối đọc dữ liệu ADC chuyển dữ liệu từ miền tương tự về miền số để có thể xử lý trên FPGA. Ngược lại với phía điều chế, tín hiệu được tách làm 2 luồng sau đó đưa qua bộ lọc và được lấy mẫu theo tần số chuẩn hóa. Dữ liệu được lưu trữ vào khối Detect để tính toán và xử lý để chuyển đổi thành dữ liệu số. Đây là khối chính của mô hình giải mã dữ liệu QAM. Ở bài báo này, nhóm tác giả tập trung vào phân tích, xây dựng và thiết kế chương trình cho khối Detect làm nền tảng để tiếp tục xây dựng các chương trình, thuật toán hỗ trợ giải điều chế tín hiệu QAM.

Sơ chi tiết khối Detect sẽ được trình bày ở phần tiếp theo của bài báo này với giả thuyết các khối khác đều đã hoàn thành. Trong đó đầu vào của khối là dữ liệu số của tín hiệu tương tự đã được lấy mẫu có độ lệch pha khác nhau và đầu ra của khối là dữ liệu số đã được tính toán đồng bộ theo dữ liệu đã được truyền từ phía phát. Ở bài báo này nhóm tác giả đã sử dụng bộ đọc ADC để đọc trực tiếp dữ liệu được điều chế pha và đưa vào khối xử lý Detect để rút ngắn thời gian xây dựng và kiểm thử hoạt động của khối.

2.2. Xây dựng mô hình

Từ những phân tích và điều kiện giả sử ở trên, nhóm tác giả đưa ra mô hình khối giải mã dữ liệu như sau:



Hình 4: Sơ đồ khối giải mã dữ liệu số 4-QAM

Trong đó, các khối có vai trò và được thiết kế cụ thể như sau:

Khối đọc dữ liệu ADC: Được xây dựng để giao tiếp với IC giao tiếp ADC MCP3204 thông qua chuẩn giao tiếp SPI. IC MCP3204 cung cấp khả năng đọc dữ liệu với tần số lấy mẫu tối đa là 100KHz và độ phân giải 12 bits. Khối đọc dữ liệu ADC thông qua MCP3204 để lấy mẫu tín hiệu với tần số lấy mẫu là: 80KHz. Ở bài báo này do hạn chế về IC đọc dữ liệu ADC nên nhóm tác giả thử nghiệm truyền tải dữ liệu ở tần số là 5Ksps tương ứng với dữ liệu là 10Kbps. Dữ liệu từ khối đọc dữ liệu ADC được chuyển tiếp đến 2 khối Lưu trữ dữ liệu đệm và Đồng bộ dữ liệu để tiếp tục xử lý.

Khối lưu trữ đệm: Nhóm tác giả xây dựng chương trình tính toán dựa trên 16 Mẫu/Symbol. Do đó, mỗi chu kỳ symbol được lưu trữ 16 mẫu dữ liệu ADC. Khối lưu trữ đệm có vai trò lưu trữ các dữ liệu tạm phục vụ cho quá trình tính toán. Khối lưu trữ dữ liệu lưu trữ dữ liệu cho tối đa 2 symbol (1 symbol đang tính toán và 1 symbol đang được truyền đến) tương ứng với tối đa 32 mẫu dữ liệu. Mỗi mẫu dữ liệu được lưu trữ bởi một ô nhớ tùy biến 12 bit và được đặt trên RAM của FPGA. Dữ liệu này được lưu trữ theo dạng vòng nối tiếp, dữ liệu mới được đẩy vào lần thứ 32+n sẽ tự động thay thế ô dữ liệu tại vị trí thứ n tạo thành chuỗi dữ liệu liên kết vòng.

Khối đồng bộ dữ liệu: Thực hiện đồng bộ giữa khối lưu trữ dữ liệu và khối cắt khung dữ liệu. Khối này hoạt động trên 2 cơ chế chính là: Dựa vào tần số tín hiệu, tần số lấy mẫu và Sự thay đổi đột ngột dữ liệu ADC tại các thời điểm chuyển giao giữa các symbol khác nhau. Trong đó, khối hoạt động chính dựa trên các bộ đếm đồng bộ thời gian và tự hiệu chỉnh dựa vào sự thay đổi dữ liệu ADC đột biến. Khi có sự chuyển đột ngột giữa 2 dữ liệu 00 và 11 hoặc 01 và 10 sẽ xảy ra sự thay đổi đột ngột về mức điện áp mà khối ADC đọc được. Đây là cơ sở để khối đồng bộ có khả năng tự điều chỉnh điểm đồng bộ. Do đó, khối này có thể dựa chính vào dữ liệu nhận được để tăng chính xác của khối đồng bộ dữ liệu. Mỗi khi khối này xác định có một khung vừa được truyền tải đến hoàn thành, khối sẽ xuất 1 xung tín hiệu cho khối cắt khung dữ liệu & Chuẩn hóa để phục vụ đồng bộ tính toán.

Khối cắt khung dữ liệu & Chuẩn hóa: Khi nhận được tín hiệu đồng bộ từ khối đồng bộ dữ liệu, khối này tự động sao chép 16 mẫu tín hiệu gần nhất và đưa vào khối tính toán đồng thời gửi một tín hiệu đến khối lưu trữ dữ liệu để xóa bỏ 16 mẫu tín hiệu đó phục vụ cho việc lưu trữ tiếp theo. Do sự suy hao trong quá trình truyền tải nên biên độ dữ liệu có thể thay đổi không đồng nhất. Do đó,

để thuận lợi cho quá trình tính toán các dữ liệu cần được chuẩn hóa đồng mức theo tín hiệu có biên độ lớn nhất. Sau khi xác định được 16 mẫu tín hiệu được sử dụng để tính toán, khối này tự xác định giá trị lớn nhất của chuỗi và chuẩn hóa toàn bộ dữ liệu theo hệ số nhân từ số lớn nhất lên 0x0FFF.

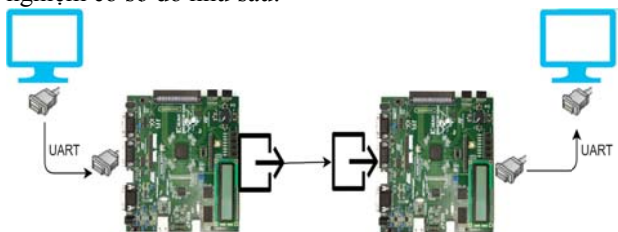
Khối tích chập: Khối tích chập tiến hành nhập dữ liệu từ khối cắt khung dữ liệu & chuẩn hóa để tính toán. Bài báo này xây dựng chương trình cho 4-QAM vì vậy nhóm tác giả xây dựng 4 khối tích chập độc lập cho 4 dạng tín hiệu khác nhau của 4-QAM. Với mỗi dữ liệu sẽ có giá trị tích chập khác nhau. Tín hiệu được truyền tải được xác định dựa trên đầu ra của khối tích chập có kết quả lớn nhất. Các khối này được thiết kế tương tự nhau với dữ liệu được lưu trữ tại bảng LUT khác nhau theo dữ liệu dạng sóng của các symbol 00,01,10 và 11 tương ứng. Các dữ liệu được lưu trữ trong các khối LUT này được tạo bởi IP Core DDS của Xilinx để đảm bảo tính chính xác về dữ liệu. Các khối tính toán tích chập này được đặt độc lập do đó được thực hiện song song giúp giảm thời gian tính toán.

Khối đầu ra dữ liệu số: Dựa vào kết quả tính toán của 4 khối tích chập, khối này có nhiệm vụ xác định tín hiệu (symbol) được truyền đến có giá trị tương ứng nào. Sau khi xác định dữ liệu, khối này gửi dữ liệu ra theo chuẩn nối tiếp và sử dụng tín hiệu CLK (Clock) để đồng bộ. Với chuẩn xuất dữ liệu này các ngoại vi khác có thể truy cập dữ liệu để sử dụng cho các ứng dụng về truyền dẫn dữ liệu khác.

Các khối được thiết kế tích hợp và chạy trên Board FPGA Spartan3 XC3S1000. Chương trình được cấu hình hoạt động ở tần số 50MHz (*Tần số mặc định của Board*). Chương trình được xây dựng sử dụng khoảng 45K cổng (Chiếm khoảng 11% tài nguyên của XC3S1000 – 442.368K) và tiêu tốn khoảng 32mW điện năng khi hoạt động.

III. THỬ NGHIỆM VÀ ĐÁNH GIÁ

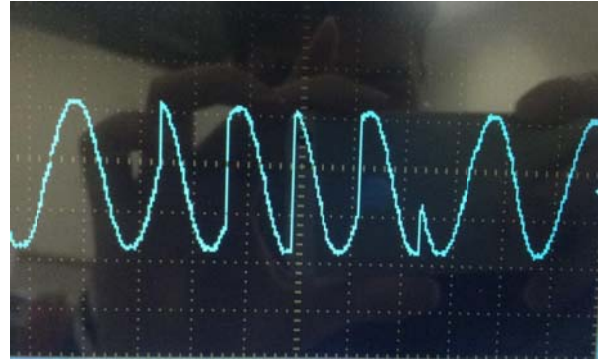
Sau khi tiến hành xây dựng chương trình nhóm tác giả tiến hành thử nghiệm đánh giá kết quả của mô hình. Nhóm tác giả sử dụng phương pháp đầu vòng để so sánh giữa dữ liệu truyền đi và dữ liệu giải mã. Hệ thống thử nghiệm có sơ đồ như sau:



Hình 5: Mô hình thử nghiệm truyền và nhận dữ liệu

Do điều kiện thực nghiệm, nhóm tác giả đã tiến hành cả phần phát và phần thu trên một Board Spartan-3 và tiến hành thử nghiệm thông qua đầu vòng tín hiệu phát và tín hiệu thu. Dữ liệu gửi đi là dữ liệu do người dùng gửi

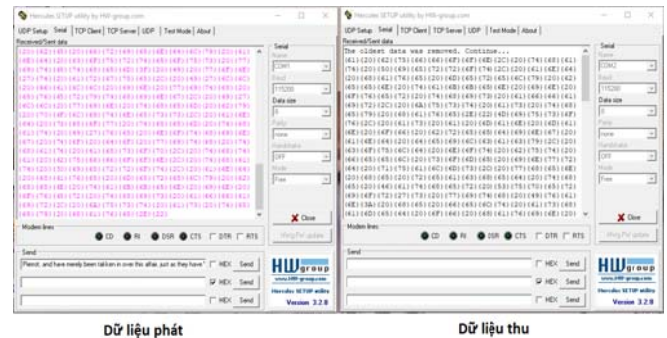
xuống board thông qua giao tiếp UART ở tần số 115200. Sau khi nhận hết một khung dữ liệu Board tiến hành đưa dữ liệu số vào khối phát và chuyển dữ liệu ra đường phát thông qua IC DAC MCP4822. Dữ liệu liên tục được nhận và truyền xuống Board. Mỗi khung truyền được phân biệt với nhau bởi nhóm ký tự đánh dấu kết thúc: “\0\0\r\n”.



Hình 6: Dữ liệu 4-QAM được phát trên chân DAC

Sau khi dữ liệu được truyền ra chân của DAC, tín hiệu được dẫn qua một cable điện dài 10m để giả lập suy hao và nhiễu sau đó được nối vào chân ADC của khối giải mã dữ liệu 4-QAM.

Dữ liệu sau khi đưa vào khối giải mã được tiến hành truyền qua giao tiếp UART ngược lại máy tính để tiến hành so sánh dữ liệu và đánh giá. Nhóm tác giả sử dụng phần mềm Hercules[9] để truyền và nhận dữ liệu UART. Phần mềm Hercules hỗ trợ tính năng gửi dữ liệu từ file và nhận dữ liệu từ file giúp quá trình thử nghiệm được nhanh chóng. Giao diện thử nghiệm có dạng như sau:



Hình 7: Giao diện thử nghiệm truyền và nhận dữ liệu 4-QAM

Kết quả thử nghiệm với các đoạn dữ liệu ngắn cho thấy dữ liệu nhận được tương đồng với dữ liệu đã thực hiện truyền đi và gần như không có các sai lệch.

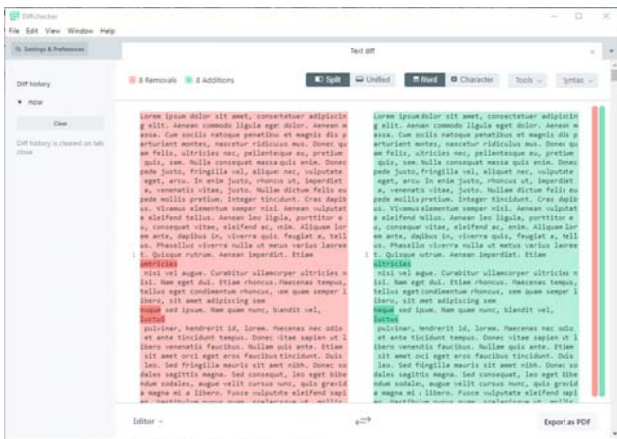


Hình 8: Kết quả thử nghiệm với chuỗi văn bản ngắn

Để kiểm thử với độ chính xác cao hơn, sau khi thử nghiệm với các chuỗi văn bản ngắn, nhóm tác giả đã sử dụng website Text-Generator[10] để tạo một file văn bản ngẫu nhiên có độ dài là 9000 từ (61525 ký tự) sau đó thử nghiệm truyền qua phần mềm Hercules truyền dữ liệu xuống Board và thu về để tiến hành đánh giá kết quả thử nghiệm. Chuỗi văn bản nhận được lưu vào file LogReceive.log để tiến hành so sánh với file LogSend.log đã được sử dụng để gửi đi.

Sau khi tiến hành truyền và nhận file, nhóm tác giả sử dụng phần mềm Diffchecker[11] để so sánh giữa văn bản truyền và nhận để tiến hành đánh giá. Kết quả so sánh cho thấy trong 61525 ký tự truyền đi chuỗi nhận và chuỗi truyền có sai khác 8 ký tự (0.013%).

Kết quả cho thấy mô hình của nhóm tác giả đã đáp ứng được nhu cầu thiết kế cơ bản về giải mã tín hiệu 4-QAM. Hiện tại sai số của mô hình còn khá lớn do các nhiễu đến từ môi trường ngoài can thiệp vào đường dẫn dữ liệu của mô hình thử nghiệm. Nhóm tác giả dự kiến tiếp tục nâng cấp, hoàn thiện mô hình để có thể áp dụng thử nghiệm trên miền truyền dẫn quang nhằm đưa ra đánh giá toàn diện và chính xác hơn về khả năng hoạt động của mô hình.



Hình 9: Kết quả so sánh giữa file truyền và file nhận

IV. KẾT LUẬN

Dựa trên kết quả thiết kế và thử nghiệm cho thấy mô hình nhóm tác giả đã đề xuất đã đáp ứng được các yêu cầu về giải mã dữ liệu 4-QAM trên nền tảng FPGA. Mô hình có kiến trúc tương đơn giản, tốc độ đáp ứng về tính toán nhanh. Độ chính xác thử nghiệm hiện tại còn tương đối thấp so với một số mô hình khác[7][12]. Tuy nhiên, nhóm tác giả không sử dụng hoàn toàn các kết quả đi trước do nhóm muốn chủ động xây dựng mô hình nền tảng để phát triển theo hướng ứng dụng kiến trúc CNN và RNN để xây dựng các mô hình về giải mã dữ liệu QAM sử dụng công cụ học máy ở giai đoạn nghiên cứu kế tiếp[3].

LỜI CẢM ƠN

Nghiên cứu này được tài trợ bởi Học Viện Công nghệ Bưu chính Viễn thông (PTIT), cơ sở Hà Nội trong đề tài mã số 01-2019-HV-KTĐT1.

TÀI LIỆU THAM KHẢO

- [1] X. Li, Simulink-based simulation of quadrature amplitude modulation (QAM) system, Proceedings of the IAIC-IJME International Conference ISBN 978-1-60643-379-9, 2008.
- [2] C.H. Dick and H.M. Pedersen, Design and implementation of high-performance FPGA signal processing data paths for software defined radio, Xilinx, INC.
- [3] J. A. Maya, N. A. Casco, P. A. Roncagliolo and J. G. Garcia, "A high data rate BPSK receiver implementation in FPGA for high dynamics applications," 2011 VII Southern Conference on Programmable Logic (SPL), Cordoba, 2011, pp. 233-238.
- [4] D. M. Klymyshyn and D. T. Haluzan, "FPGA implementation of multiplierless M-QAM modulator," in Electronics Letters, vol. 38, no. 10, pp. 461-462, 9 May 2002.
- [5] J. Östth, M. Karlsson, A. Serban and S. Gong, "M-QAM Six-Port Modulator Using Only Binary Baseband Data, Electrical or Optical," in IEEE Transactions on Microwave Theory and Techniques, vol. 61, no. 6, pp. 2506-2513, June 2013.
- [6] X. Du and J. Zhang, "DDS phase-locked swept source and study design," Proceedings of 2011 International Conference on Computer Science and Network Technology, Harbin, 2011, pp. 146-149.
- [7] H. E. Zorlu, M. Çeven, S. Taşdöken and H. Özer, "Implementation of PSK and QAM demodulators on FPGA," 2012 20th Signal Processing and Communications Applications Conference (SIU), Mugla, 2012, pp. 1-4.
- [8] B. Bornoosh, A. Nabavi, M. E. Nick and A. Haghbin, "A New Architecture for Reducing Phase Noise of Digital Carrier Recovery Algorithms in High-Order QAM Demodulators," 2007 IEEE International Conference on Signal Processing and Communications, Dubai, 2007, pp. 668-671.
- [9] <https://www.hw-group.com/software/hercules-setup-utility>
- [10] <https://www.blindtextgenerator.com/lorem-ipsum>
- [11] <https://www.diffchecker.com/>
- [12] Changxing Lin, Beibei Shao and Jian Zhang, "A high data rate parallel demodulator suited to FPGA implementation," 2010 International Symposium on Intelligent Signal Processing and Communication Systems, Chengdu, 2010, pp. 1-4.

4-QAM DEMODULATION FOR OPTICAL SIGNAL TRANSMISSION APPLICATION

Abstract -Quadrature Amplitude Modulation (QAM) is used in both analog and digital communication. FPGA chips are Field-upgradable, reliable in designing and implementation. FPGA has been playing a significant role in mobile communication. This paper proposes a 4-QAM demodulation model based on FPGA technology. The model reads ADC signal, calculator based on 4-QAM standard, and export to digital signals.

Keywords — QAM, OFDM, DDS, FPGA.



Lương Công Dẫn tốt nghiệp Thạc Sĩ ngành Kỹ thuật Viễn thông tại Học viện Công nghệ Bưu chính Viễn thông năm 2018. Hiện đang là giảng viên tại khoa Kỹ thuật Điện tử I, Học viện Công nghệ Bưu chính Viễn thông.



Vũ Anh Đào tốt nghiệp Thạc Sĩ ngành Kỹ thuật đo lường và điều khiển tự động tại Đại học Bách khoa Hà Nội năm 2002. Hiện đang là nghiên cứu sinh tại đại học Bách khoa từ năm 2017 và là giảng viên tại khoa Kỹ thuật Điện tử I, Học viện Công nghệ Bưu chính Viễn thông.