JTE
HCMUTE
ISSN: 1859-1272

JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.hcmute.edu.vn
Email: jte@hcmute.edu.vn

FPGA-Based Design and Implementation of CRC-16 Encoder and Decoder

Le Hoang Trieu, Tran Quoc Duy, Nguyen Hoang Hieu, Nguyen Thi Hong Hao, Nguyen Van Thanh Loc, Do Duy Tan*

Faculty of Electrical and Electronics Engineering, Ho Chi Minh City University of Technology and Education, Vietnam

* Corresponding author. *Email: tandd@hcmute.edu.vn*

ARTICLE INFO

Received: 17/2/2022
Revised: 21/5/2022
Accepted: 23/8/2022
Published: 30/8/2022

KEYWORDS

CRC;
FPGA;
Verilog;
Encoder;
Decoder.

ABSTRACT

Error control coding is an important functional block to ensure the reliability of any communication system. Specifically, Cyclic Redundancy Check (CRC) codes are widely used in many fields such as civil communication and industrial communication. CRC codes efficiently provide the first layer of protection against data corruption during data transmission sent from a transmitter to a receiver over channels. With the advantage of being simple but effective in detecting and possibly correcting errors in data transmission and storage of digital data. In this paper, we present a detailed design and implementation of a CRC-16 encoder and decoder based on Field Programmable Gate Array (FPGA) using Verilog hardware description language. Then, we evaluate the design using both Xilinx ISE software and AX309 FPGA kit where error detection capability and resource usage are tested in detail. Extensive simulations and FPGA board based experimental results have been conducted to confirm the effectiveness of our proposed design.

Thiết Kế Và Thi Công Bộ Mã Hóa Và Giải Mã CRC-16 Dựa Trên Công Nghệ FPGA

Lê Hoàng Triệu, Trần Quốc Duy, Nguyễn Hoàng Hiếu, Nguyễn Thị Hồng Hảo, Nguyễn Văn Thành Lộc, Đỗ Duy Tân*

Khoa Điện-Điện Tử, Trường Đại Học Sư Phạm Kỹ Thuật TPHCM, Việt Nam

** Tác giả liên hệ. Email: tandd@hcmute.edu.vn*

THÔNG TIN BÀI BÁO

Ngày nhận bài: 17/2/2022
Ngày hoàn thiện: 21/5/2022
Ngày chấp nhận đăng: 23/8/2022
Ngày đăng: 30/8/2022

TỪ KHÓA

CRC;
FPGA;
Verilog;
Mã hóa;
Giải mã.

TÓM TẮT

Mã kiểm tra lỗi là một khối chức năng quan trọng giúp đảm bảo độ tin cậy trong các hệ thống thông tin liên lạc. Đặc biệt, mã hóa CRC (Cyclic Redundancy Check) được sử dụng rộng rãi trong nhiều lĩnh vực như truyền thông dân dụng và truyền thông công nghiệp. Mô hình mã CRC là một trong những mô hình mã kiểm tra lỗi hiệu quả giúp khắc phục dữ liệu bị ảnh hưởng của nhiễu trong quá trình truyền dữ liệu qua kênh truyền. Ưu điểm chính của mã CRC là đơn giản nhưng đạt được hiệu quả cao trong việc phát hiện lỗi và lưu trữ dữ liệu số. Trong bài báo này, chúng tôi trình bày chi tiết thiết kế và thi công bộ mã hóa và bộ giải mã mã CRC-16 dựa trên công nghệ FPGA bằng ngôn ngữ mô tả phần cứng Verilog. Các kết quả đánh giá được thực hiện trên cả phần mềm mô phỏng và kit FPGA để so sánh độ chính xác và tính hiệu quả của thiết kế so với lý thuyết.

Doi: https://doi.org/10.54644/jte.71B.2022.1140

1. Giới thiệu

Mã kiểm tra lỗi CRC (Cyclic Redundancy Check) thuộc loại mã FEC (Forward Error Correction), có ưu điểm là dễ thực hiện mã hóa và giải mã cũng như khả năng phát hiện lỗi mạnh mẽ [1]. Do đó, mã CRC được sử dụng rộng rãi trong nhiều lĩnh vực như truyền thông dân dụng và truyền thông công nghiệp,... Bộ mã CRC được thiết kế đặc biệt để giúp khắc phục các loại lỗi phổ biến gây ra bởi kênh

JTE
HCMUTE
ISSN: 1859-1272

JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.hcmute.edu.vn
Email: jte@hcmute.edu.vn

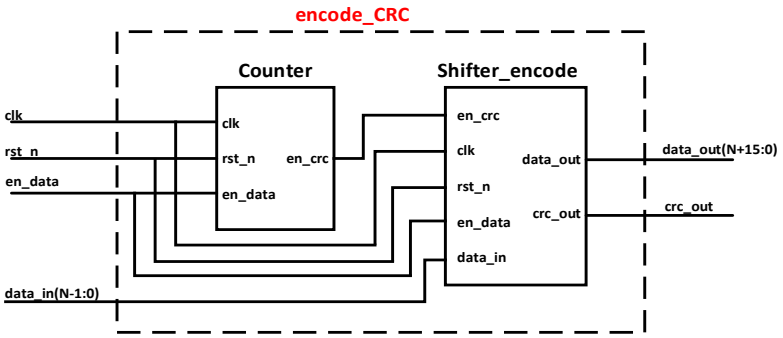
truyền hữu tuyến hoặc vô tuyến, đảm bảo tính toàn vẹn của thông tin (information message) được gửi từ thiết bị phát đến thiết bị thu [1]. Trong mã CRC, các bít dư thừa hay còn gọi là các mã CRC được tạo ra bằng phép chia nhị phân không nhớ hay phép chia modulo-2. Đối với bộ mã hóa, phần dư của phép chia nhị phân chính là mã CRC được gắn vào phía cuối của chuỗi dữ liệu truyền đi. Còn đối với bộ giải mã, kết quả của phép chia cho kết quả bằng không thì chuỗi dữ liệu nhận được tại sau khi qua kênh truyền là chính xác[2]. Các phép chia của bộ mã hóa và giải mã được thực hiện thông qua bộ chia. Độ dài của bộ chia sẽ bằng với số bậc cộng thêm 1 . Ví dụ, CRC-16 có bậc là 16 tương ứng độ dài của bộ chia là 17 bit. Một cách khác để biểu diễn bộ chia là dưới dạng đa thức, người ta gọi đó là đa thức sinh. Ưu điểm của đa thức sinh là ngắn ngọn và dễ dàng phân tích phát hiện lỗi. Một đa thức sinh có khả năng bao quát được hầu hết các trường hợp lỗi phải thỏa mãn các đặc điểm sau: (1) không chia hết cho *x*; (2) không chia hết cho x^t+1, với $2 < t < n-1$; (3) khi phân tích thành nhân tử có $(x+1)$. Đặc điểm thứ nhất nhằm bảo đảm phát hiện được tất cả các lỗi bệt (Burst) có độ dài bằng bậc của đa thức sinh. Đặc điểm thứ hai bảo đảm phát hiện được lỗi 2 bit ngẫu nhiên. Đặc điểm cuối cùng bảo đảm phát hiện được tất cả các lỗi với tổng số bit bị lỗi là số lẻ [3].

Mã CRC đã được trình bày trong nhiều bài báo [4-6]. Trong bài báo [4], tác giả thiết kế và thi công bộ mã CRC cho hệ thống ADS-B (Automatic Dependent Surveillance Broadcast) dựa trên công nghệ FPGA. Trong bài báo [5], tác giả đề xuất thuật toán CRC có khả năng giảm tỉ lệ lỗi của hệ thống bằng cách phát hiện và kiểm soát lỗi trong truyền thông mạng máy tính. Trong bài báo [6], tác giả xét tính năng sửa lỗi của mã CRC áp dụng cho hai tiêu chuẩn yêu cầu năng lượng thấp Bluetooth Low Energy (BLE) và IEEE 802.15.4. Trong bài báo này, chúng tôi trình bày chi tiết thiết kế và thi công bộ mã hóa và bộ giải mã CRC-16 ứng dụng trong chuẩn Modbus RTU dựa trên công nghệ FPGA [7-9] – thường được sử dụng trong hệ thống truyền thông công nghiệp, để xác minh độ tin cậy của dữ liệu. Thay vì chỉ đánh giá thực thi chức năng thiết kế qua mô phỏng như nhiều bài báo đang có, chúng tôi đánh giá kết quả thiết kế trên cả phần mềm Xilinx ISE và kit FPGA Xilinx AX309, trong đó khả năng phát hiện lỗi, tài nguyên sử dụng và thời gian hoạt động được xem xét và đánh giá chi tiết.

Phần còn lại của bài báo được trình bày như sau: phần 2 trình bày nguyên lý của mô hình mã hóa CRC, phần 3 trình bày thiết kế chi tiết hệ thống, phần 4 trình bày đánh giá chất lượng thiết kế và cuối cùng kết luận được trình bày trong phần 5.

2. Thiết kế chi tiết hệ thống

2.1. Module CRC_encoder



Hình 1. *Sơ đồ khối bộ mã hóa CRC-16.*

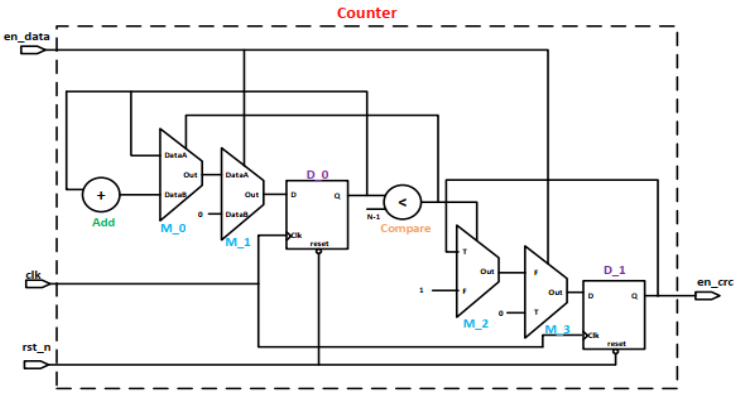
Hình 1 mô tả tổng quan các khối trong mạch mã hóa CRC-16 với đa thức sinh là $g(x) = x^{16} + x^{15} + x^2 + 1$ có N bit ngõ vào. Mạch gồm 2 khối chính là khối Counter và khối Shifter_encode [10]. Trong đó khối Counter có nhiệm vụ đếm số lần dịch để tạo tín hiệu cho phép en_crc để khối Shifter_encode đưa dữ liệu tới ngõ ra data_out và crc_out. Trạng thái hoạt động của khối Counter và khối Shifter_encode được trình bày lần lượt ở **Bảng 1** và **Bảng 2**. Khi N bit dữ liệu (ở ngõ vào data_in) được dịch hoàn toàn vào khối Shifter_encode thì mã CRC được tạo ra (CRC_encode). Khối Counter sẽ đếm thêm 16 lượt đếm để dịch 16 bit "0" vào khối Shifter_encode, đẩy 16 bit CRC ra ngõ ra crc_out đồng thời ra ngõ ra data_out kết hợp với dữ liệu (ở ngõ vào data_in) tạo thành từ mã (codeword)

JTE
HCMUTE
ISSN: 1859-1272

JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.hcmute.edu.vn
Email: jte@hcmute.edu.vn

truyền ra kênh truyền [11]. Sơ đồ chi tiết, nguyên tắc hoạt động của các khối được trình bày trong phần sau.

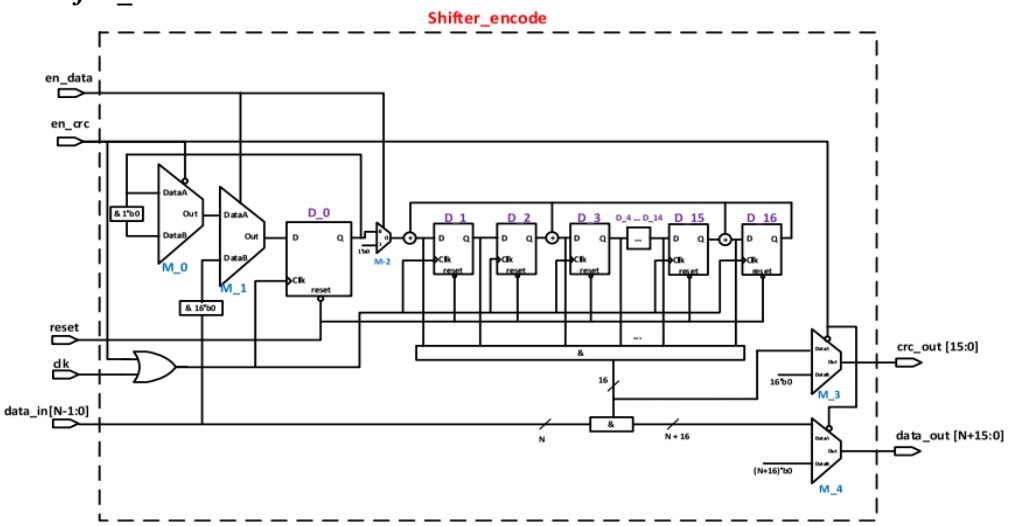
2.1.1. Khối Counter



***Hình 2**. Sơ đồ chi tiết bộ Counter.*

Hình 2 thể hiện sơ đồ khối bộ Counter với mux M_0 cho phép giá trị đếm bên trong mạch tăng lên 1 đơn vị khi giá trị đếm nhỏ hơn N-1, Mux M_1 điều khiển mạch bắt đầu đếm khi en_data = 0. Khi giá trị đếm bằng giá trị N-1, mạch sẽ giữ nguyên giá trị đếm và cho tín hiệu en_crc = 1. Các bộ mux M_2, M_3 và Flip Flop D_1 đóng vai trò như một mạch chốt cho ngõ ra en_crc.

2.1.2. Khối Shifter_encode



Hình 3. *Sơ đồ chi tiết bộ Shifter_encode.*

Hình 3 mô tả sơ đồ chi tiết khối Shiter_encode, mạch bắt đầu nhận dữ liệu data_in và nối 16 bit 0 vào vị trí LSB của dữ liệu khi en_data = 1. Thông qua mux M_1, mux M_0 có nhiệm vụ dịch trái logic 1 bit khi en_data = 0. Đồng thời khi en_data = 0, mux M_2 cho phép dữ liệu đi vào bộ tính toán CRC-16 để tính toán ra 16 bit CRC qua các Flip Flop từ D_1 đến D_16. Tín hiệu en_crc = 1 báo hiệu đã tính toán xong và cho phép xuất giá trị crc_out thông qua mux M_3 và nối data_in với crc_out đưa đến ngõ ra data_out thông qua mux M_4.

***Bảng 1.** Bảng trạng thái của khối Counter.*

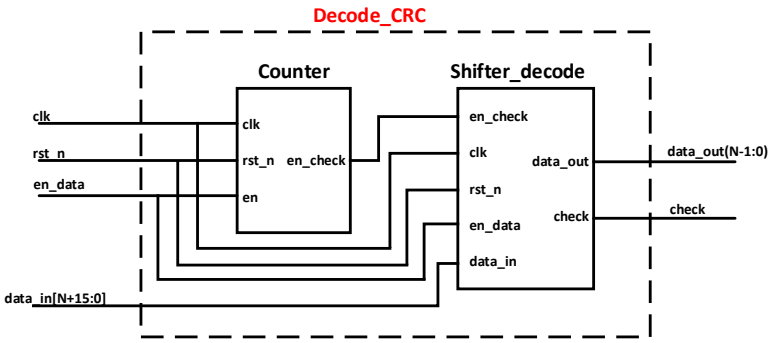
Input			i=0 ; i ≤ N+15 ; i++	Output
clk	rst_n	en_data		en_crc
1	1	0	0	0
1	1	1	≤ N-1	0
1	1	1	≤ N + 15	1

JTE HCMUTE
ISSN: 1859-1272
JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.hcmute.edu.vn
Email: jte@hcmute.edu.vn

Bảng 2. *Bảng trạng thái của khối Shifter_encode.*

Input					Output	
clk	rst_n	en_data	en_crc	data_in	crc_out	data_out
1	1	0	0	0	0	0
1	1	1	0	Data input [0:N-1] >>1	0	0
1	1	1	1	16 bit "0" >>1	CRC_encode	{Data input, CRC_encode}

2.2. Module CRC_decoder



Hình 4. *Sơ đồ khối bộ giải mã CRC-16.*

Hình 4 mô tả sơ đồ khối tổng quát của mạch giải mã CRC-16 với đa thức sinh $g(x) = x^{16} + x^{15} + x^2 + 1$ có N+16 bit ngõ vào, hoạt động của mạch tương tự với mạch mã hóa CRC-16. Mạch gồm 2 khối chính là khối Counter và khối Shifter_decode. Hoạt động của khối Counter trong bộ giải mã CRC-16 tương tự với bộ mã hóa CRC-16, ký hiệu en_crc được thay bằng ký hiệu en_check. Trạng thái hoạt động của khối Shifter_decode được trình bày trong **Bảng 3**. Khi N bit dữ liệu (ở ngõ vào data_in) được dịch hoàn toàn vào khối Shifter_decode thì mã CRC_decode được tạo ra. Khối Counter sẽ đếm thêm 16 lượt đếm để dịch 16 bit CRC_encode vào khối Shifter_decode. Khối Shifter_decode sẽ so sánh CRC_encode và CRC_decode, nếu giống nhau (chứng tỏ không xảy ra lỗi) thì ngõ ra check=1 đồng thời ngõ ra data_out xuất N bit dữ liệu, ngược lại ngõ ra check=0 đồng thời ngõ ra data_out=0.

Bảng 3. *Bảng trạng thái của khối Shifter_decode.*

Input					Output	
clk	rst_n	en_data	en_check	data_in	check	data_out
1	1	0	0	0	0	0
1	1	1	0	Data input [16:N+15] >>1	0	0
1	1	1	1	Data input [0:15]>>1	if (CRC code_decode = CRC code_encode), check= 1	Data input [16:N+15]

3. Đánh giá chất lượng của thiết kế

3.1. Tài nguyên sử dụng

Thiết kế đầy đủ các chức năng được tổng hợp trên FPGA Spartan-6 XC6SLX9-TQG144, khi dữ liệu vào có chiều dài 1600 bit thì tài nguyên logic được sử dụng và tần số hoạt động tối đa như tóm tắt trong các **Bảng 4-7.**

Bảng 4. *Tóm tắt tài nguyên sử dụng cho bộ mã hóa.*

Logic Utilization	Used	Available	Utilization (%)
Number of Slice Registers	1634	11440	14
Number of Slice LUTs	3302	5720	57
Number of fully used LUT-FF pairs	1632	3304	49

Bảng 5. *Tóm tắt các thông số liên quan tới thời gian xử lý của bộ mã hóa.*

Speed Grade	-3
Maximum Frequency	273.338 MHz
Minimum input arrival time before clock	5.991 ns
Maximum output required time after clock	6.297 ns
Maximum combinational path delay	5.296 ns

Bảng 6. *Tóm tắt tài nguyên sử dụng cho bộ giải mã.*

Logic Utilization	**Used**	**Available**	**Utilization(%)**
Number of Slice Registers	1650	11440	14
Number of Slice LUTs	3306	5720	57
Number of fully used LUT-FF pairs	1648	3304	49

Bảng 7*. Tóm tắt các thông số liên quan tới thời gian xử lý của bộ giải mã.*

Speed Grade	-3
Maximum Frequency	273.338 MHz
Minimum input arrival time before clock	5.995 ns
Maximum output required time after clock	8.485 ns
Maximum combinational path delay	5.296 ns

3.2. Kết quả đánh giá qua mô phỏng

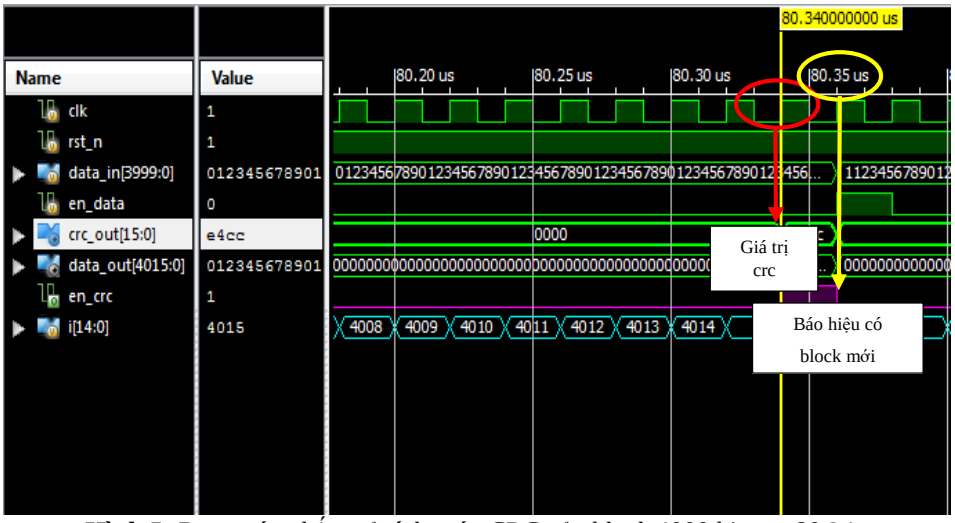
Thiết kế của bộ mã hóa CRC-16 được đánh giá qua các Testcase thể hiện trong **Bảng 8.** Testcase 1 thể hiện thời gian cần thiết để mã hóa 1 block có độ dài 4000 bit (ví dụ cho việc tạo mã CRC bảo vệ gói dữ liệu của chuẩn USB 2.0). Kết quả được thể hiện trên **Hình 5**. Bộ mã hóa cần khoảng thời gian 80,34 us để thực hiện mã hóa 1 block độ dài 4000 bit. Testcase 2 cho thấy bộ mã hóa CRC-16 tính toán CRC khi dữ liệu được xor với đa thức sinh và đa thức sinh đã dịch sang trái 1 bit. Kết quả được thể hiện trên **Hình 6**. Một số trường hợp xảy ra việc tính toán mã CRC bị trùng nhau. Với 3 dữ liệu block khác nhau nhưng lại cho ra cùng một kết quả CRC, đây là một trường hợp khi bộ giải mã cho ra kết quả sai.

Bảng 8. *Tóm tắt testcase của bộ mã hóa CRC-16*

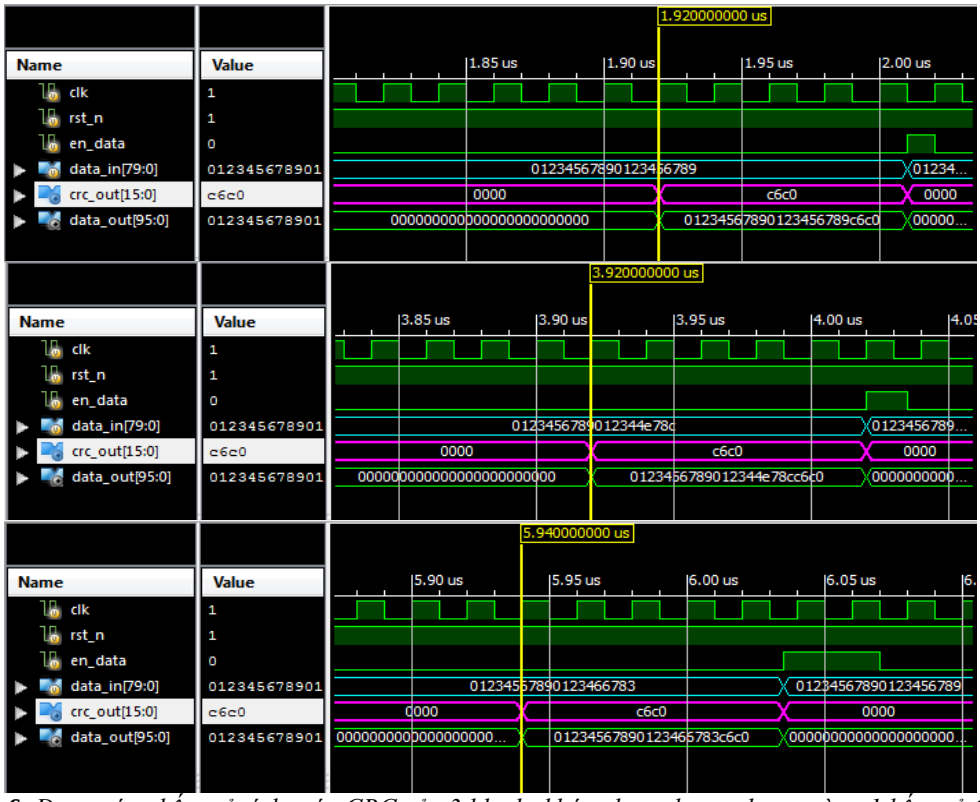
Testcase	**Nội dung**	
Testcase 1	Block-length: 4000 bit. Truyền 2 block cách nhau: 80,34 us.	Tần số xung clock là: 50 MHz
Testcase 2	Block-length: 80 bit. Truyền 3 block, 2 block kế nhau truyền cách nhau: 2,005 us. Dữ liệu block 2 là dữ liệu block 1 XOR với đa thức sinh. Dữ liệu block 3 là dữ liệu block 1 XOR với đa thức sinh đã dịch sang trái 1 bit.	

JTE
HCMUTE
ISSN: 1859-1272

JOURNAL OF TECHNOLOGY EDUCATION SCIENCE
Ho Chi Minh City of Technology and Education
Website: https://jte.hcmute.edu.vn/index.php/jte/index
Email: jte@hcmute.edu.vn



Hình 5. *Dạng sóng kết quả tính toán CRC của block 4000 bit sau 80,34 us.*



Hình 6. *Dạng sóng kết quả tính toán CRC của 3 blocks khác nhau nhưng cho ra cùng 1 kết quả CRC.*

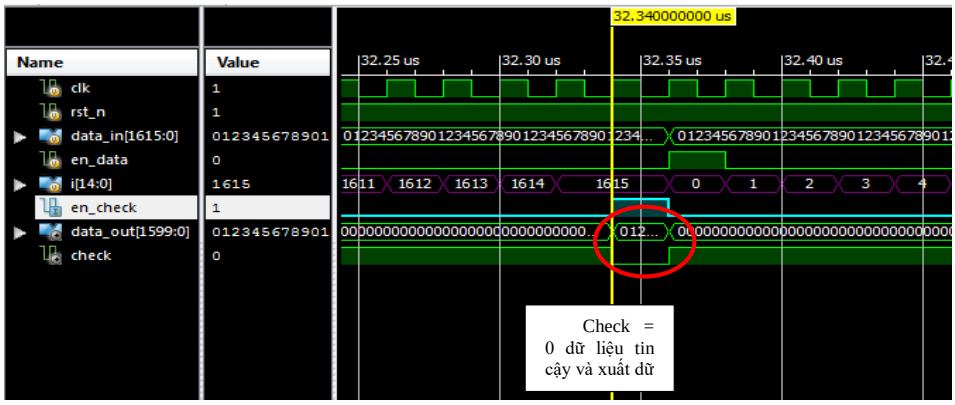
Thiết kế của bộ giải mã CRC-16 được đánh giá qua các Testcase được thể hiện trong **Bảng 9**. Testcase 3 mô tả giải mã CRC của 1 block có chiều dài 1616 bit (ví dụ giao thức Modbus). Kết quả được thể hiện trên **Hình 7**. Bộ giải mã cần 32,34 us để giải mã 1 block có chiều dài 1616 bit. Các Testcase từ 4.1 đến Testcase 4.5 đánh giá xác suất có thể giải mã của bộ giải mã CRC-16. Các Testcase 4.1, 4.2, 4.4 cho ra cùng kết quả đều phát hiện được 100/100 block sai truyền đi, được thể hiện trên **Hình 8**. Testcase 4.3 và Testcase 4.5 cho ra cùng kết quả, phát hiện được 99/100 block sai truyền đi, được thể hiện trên **Hình 9**.

JTE
HCMUTE
ISSN: 1859-1272

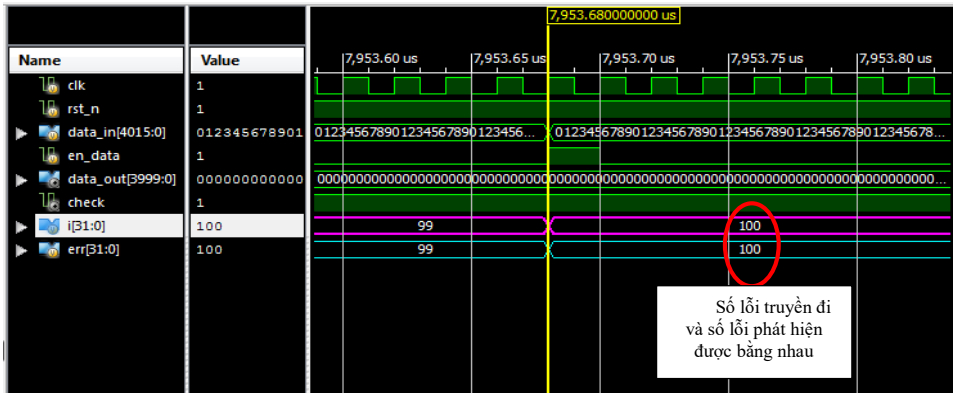
JOURNAL OF TECHNOLOGY EDUCATION SCIENCE
Ho Chi Minh City of Technology and Education
Website: https://jte.hcmute.edu.vn/index.php/jte/index
Email: jte@hcmute.edu.vn

Bảng 9. *Tóm tắt testcase của bộ giải mã CRC-16.*

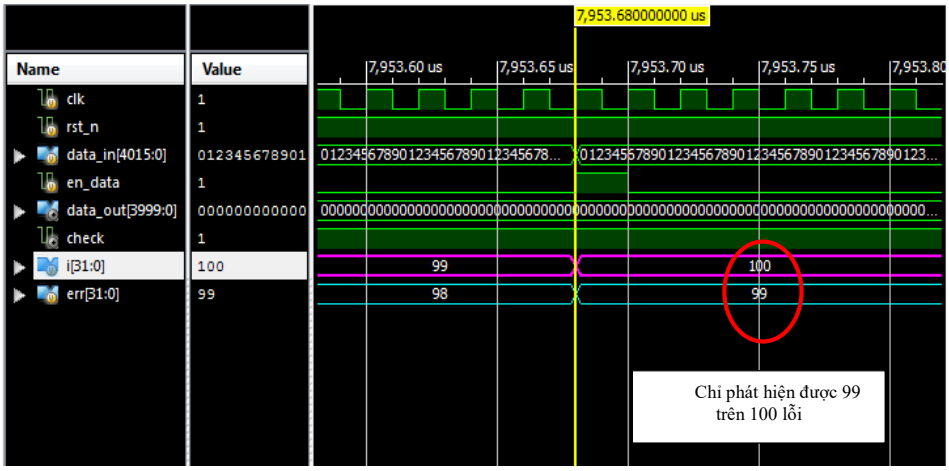
Testcase	Nội dung	
Testcase 3	Block-length: 1616 bit. 2 block truyền cách nhau: 32,34 us.	Tần số xung clock là: 50MHz
Testcase 4.1	Block-length: 4016 bit. Truyền 100 block. Mỗi block sai 2 bit.	
Testcase 4.2	Block-length: 4016 bit. Truyền 100 block. Mỗi block sai 3 bit.	
Testcase 4.3	Block-length: 4016 bit. Truyền 100 block. Mỗi block sai 4 bit.	
Testcase 4.4	Block-length: 4016 bit. Truyền 100 block. Mỗi block sai 5 bit.	
Testcase 4.5	Block-length: 4016 bit. Truyền 100 block. Mỗi block sai 6 bit.	



Hình 7. *Dạng sóng kết quả giải mã CRC của block 1616 bit sau 32.34 us.*



Hình 8. *Dạng sóng kết quả truyền 100 block có độ dài 4016 bit với mỗi block sai 2 bit, 3 bit hoặc 5 bit.*



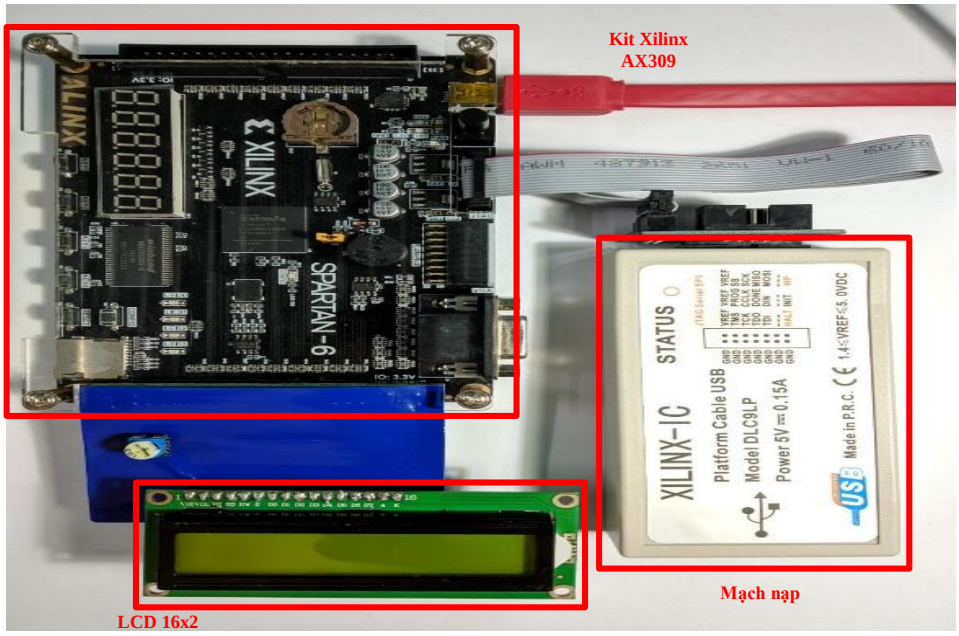
Hình 9. *Dạng sóng kết quả truyền 100 block có độ dài 4016 bit với mỗi block sai 4 bit hoặc 6 bit.*

JTE
HCMUTE
ISSN: 1859-1272

JOURNAL OF TECHNOLOGY EDUCATION SCIENCE
Ho Chi Minh City of Technology and Education
Website: https://jte.hcmute.edu.vn/index.php/jte/index
Email: jte@hcmute.edu.vn

3.3. Kết quả đánh giá thực tế trên kit FPGA

Mạch kiểm tra của bộ mã hóa CRC-16 được điều khiển bằng một nút nhấn, khi nhấn sẽ gửi dữ liệu cố định sẵn. Kết quả tính toán của bộ mã hóa CRC-16 hiển thị lên màn hình LCD 16x2. Bên cạnh đó, mạch kiểm tra của bộ giải mã CRC-16 được điều khiển bằng hai nút nhấn, một nút nhấn có nhiệm vụ thay đổi dữ liệu đầu vào để tạo các lỗi khác nhau. Kết quả kiểm tra của bộ giải mã CRC-16 hiển thị lên màn hình LCD 16x2. Các linh kiện có trong mạch kiểm tra thực tế như **Hình 10** bao gồm kit FPGA Xilinx AX309, mạch nạp cho kit, màn hình hiển thị LCD 16x2 và nút nhấn tích hợp sẵn trên kit**.**



Hình 10. *Phần cứng thực tế.*

Kết quả tính toán của bộ mã hóa CRC với block dữ liệu 4000 bit (Testcase 1.1) và 1600 bit (Testcase 2.1) khi nạp vào kit thực tế được thể hiển trên **Hình 11** và **Hình 12.**

Hình 11. *Kết quả tính toán CRC của bộ mã hóa với block dữ liệu dài 4000 bit.*

Hình 12. *Kết quả tính toán CRC của bộ mã hóa với block dữ liệu dài 1600 bit.*

JTE HCMUTE
ISSN: 1859-1272
JOURNAL OF TECHNOLOGY EDUCATION SCIENCE
Ho Chi Minh City of Technology and Education
Website: https://jte.hcmute.edu.vn/index.php/jte/index
Email: jte@hcmute.edu.vn

Từ mô phỏng và thực nghiệm ta rút ra được những đánh giá sau. Bộ giải mã và mã hóa cần thời gian tối thiểu để có thể thực hiện mã hóa và giải mã CRC. Thời gian phụ thuộc vào block-length, block-length càng dài thời gian chờ càng lâu. Cụ thể, thời gian mã hóa tối thiểu khi block-length là 4000 bit là 80,32 ns, thời gian giải mã tối thiểu khi block-length là 1616 bit là 32,34 ns. Nếu có block tiếp theo đến trước khi quá trình mã hóa hoặc giải mã hoàn tất thì ưu tiên tính toán cho block thứ 2. Từ **Bảng 10**, chúng ta có thể đưa ra kết luận về bộ giải mã CRC-16 0x8005:

i. Luôn phát hiện được block sai nếu số bit sai của block nhỏ hơn 4.
ii. Luôn phát hiện được block sai nếu block có số bit sai của block đó là số lẻ. Chứng minh lại lý thuyết đã được trình bày trong [3].
iii. Nếu đa thức của lỗi mà chia hết cho đa thức sinh bộ giải mã sẽ không phát hiện được.

Bảng 10. *Bảng so sánh xác suất phát hiện được block sai khi truyền 100 block.*

Block-length	Số block truyền	Số bit sai mỗi block	Số block phát hiện được bị sai
4016 bit	100	2	100
		3	100
		4	99
		5	100
		6	99

4. Kết luận

Trong bài báo này, chúng tôi đã thực hiện thiết kế, thi công và đánh giá chi tiết bộ mã hóa và giải mã mã CRC-16 dựa trên công nghệ FPGA. Thiết kế đáp ứng được trường kiểm tra lỗi (error-checking field) của Modbus RTU chứa 16 bit CRC và được tính toán dựa trên dữ liệu thông tin. Các bit CRC được tạo ra luôn đảm bảo nằm ở vị trí trường cuối cùng trong thông tin. Các đánh giá về mã CRC được xem xét trên các block dữ liệu tiệm cận với ứng dụng trong các chuẩn truyền thông như USB 2.0 hay Modbus thông qua các Testcase mô phỏng. Bài báo này có giá trị như một tài liệu tham khảo cho việc học tập các môn liên quan tới thiết kế hệ thống số và thiết kế vi mạch số, đặc biệt là thiết kế các khối mã hóa kiểm soát lỗi ứng dụng trong thông tin số.

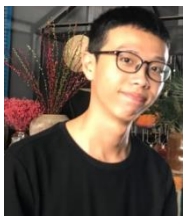
Lời cám ơn

Bài báo này được tài trợ kinh phí nghiên cứu năm 2022 bởi trường Đại Học Sư phạm Kỹ thuật Thành Phố Hồ Chí Minh (mã số đề tài SV2022-23).

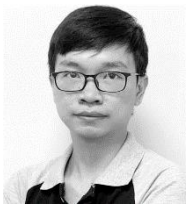
TÀI LIỆU THAM KHẢO

[1] Martin P.Clark, *Wireless Access Networks: Fixed Wireless Access and WLL Networks-Design and Operation*, John Wiley & Sons Ltd, 2000.
[2] Todd K.Moon, *Error Control Coding—Mathematical Methods and Algorithms*, New York: John Wiley & Sons, 2005.
[3] Yuan Jiang, *A Practical Guide to Error-Control Coding Using MATLAB*, Artech House Publishers, 2010.
[4] Ma Yuping & Zhang Jun, "Design and Implementation of CRC Error Correction for ADS-B System Responding Based on FPGA". *Applied Mechanics and Materials*, 2014.
[5] Zonglin Zhong, Wengui Hu, "Error detection and control of IIoT network based on CRC algorithm", *Computer Communications*, volume 153, 2020, pages 390-396.
[6] E. Tsimbalo, X. Fafoutis and R. J. Piechocki, "CRC Error Correction in IoT Applications", *IEEE Transactions on Industrial Informatics*, vol. 13, no. 1, Feb. 2017, pp. 361-369.
[7] J. N. Chhatrawala, N. Jasani and V. Tilva, "FPGA based data acquisition with Modbus protocol", *International Conference on Communication and Signal Processing (ICCSP)*, 2016, pp. 1251-1254.
[8] Sarah L.Harris, David Money Harris, *Digital Design and Computer Architecture -Arm Edition*, Elsevier USA, 04/2015.
[9] Stephen M. Trimberger, *Field-Programmable Gate Array Technology*, Springer Science & Business Media, 1994.
[10] Lin, S., and D. J. Castello, *Error Control Coding—Fundamentals and Application*, Upper Saddle River, NJ: Prentice-Hall, 2004.
[11] Behrouz A. Forouzan, *Data Communications and Networking (Fourth Edition)*, McGraw-Hill, USA, 2007.

Le Hoang Trieu is currently a student at the Ho Chi Minh City University of Technology and Education (HCMUTE), Vietnam. His main research interests include wireless communication networks and FPGA-based designs for DSP applications.



Tran Quoc Duy received his B.S degree from Ho Chi Minh City University of Technology and Education (HCMUTE), Vietnam, in 2021. His main research interests include wireless communication networks and FPGA-based designs for DSP applications.



Nguyen Hoang Hieu received his B.S degree from Ho Chi Minh City University of Technology and Education (HCMUTE), Vietnam, in 2021. His main research interests include wireless communication networks and FPGA-based designs for DSP applications.

Nguyen Thi Hong Hao is currently a student at the Ho Chi Minh City University of Technology and Education (HCMUTE), Vietnam. Her main research interests include wireless communication networks and FPGA-based designs for DSP applications.

Nguyen Van Thanh Loc received his B.S degree from Ho Chi Minh City University of Technology and Education (HCMUTE), Vietnam, in 2020. His main research interests include communication networks and applications of error-control coding for wireless communications.

Do Duy Tan received his B.S. degree from Ho Chi Minh City University of Technology (HCMUT), Vietnam, and M.S. degree from Kumoh National Institute of Technology, Korea, in 2010 and 2013, respectively. He received his Ph.D. degree from Autonomous University of Barcelona, Spain, in 2019. He is currently with the Department of Computer and Communication Engineering, Ho Chi Minh City University of Technology and Education (HCMUTE) in Vietnam as an Assistant Professor. His main research interests include real-time optimisation for resource allocation in wireless networks and coding applications for wireless communications.