

THIẾT KẾ VÀ ĐÁNH GIÁ CHẤT LƯỢNG MODULE GIAO TIẾP NỐI TIẾP TRONG CHUẨN USB

DESIGN AND VERIFICATION OF SERIAL INTERFACE ENGINE MODULE IN USB STANDARD

Trần Đức Sang, Nguyễn Văn Thành Lộc, Trần Quốc Tấn,
Trần Thị Quỳnh Như, Huỳnh Hoàng Hà, Đỗ Duy Tân
Trường Đại học Sư phạm Kỹ thuật TP.HCM, Việt Nam

Ngày toà soạn nhận bài 19/5/2021, ngày phản biện đánh giá 25/6/2021, ngày chấp nhận đăng 19/7/2021.

TÓM TẮT

Bài báo này trình bày thiết kế chi tiết và đánh giá chất lượng của module giao tiếp nối tiếp (Serial Interface Engine hay SIE) trong chuẩn USB 2.0 dựa trên công nghệ FPGA. Khối SIE có chức năng nhận dữ liệu được truyền từ bộ xử lý trung tâm (Central Processing Unit hay CPU) và chuyển dữ liệu đó đến Universal Transceiver Macrocell Interface (UTMI). Ở phía phát, dữ liệu được chuyển từ song song thành nối tiếp, sau đó đi qua khối mã hóa kiểm tra lỗi và bộ mã hóa kênh trước khi được phát qua kênh truyền. Ở phía thu, ngược lại, sau khi đi qua bộ giải mã hóa kênh và kiểm tra lỗi, dữ liệu nối tiếp sẽ được chuyển đổi thành dữ liệu song song. Thông qua các kết quả mô phỏng, chúng tôi chỉ ra rằng hoạt động của module SIE được thiết kế đáp ứng đầy đủ các yêu cầu kỹ thuật đề ra.

Từ khóa: khối giao tiếp nối tiếp; USB 2.0; FPGA; Xilinx ISE; đánh giá và xác thực.

ABSTRACT

This paper presents a design and verification of a Serial Interface Engine (SIE) module in the USB 2.0 standard based on FPGA technology. The SIE unit receives data sent from the Central Processing Unit (Central Processing Unit or CPU) and passes it to the Universal Transceiver Macrocell Interface (UTMI). At the sender, the data is converted from parallel to serial, then passed through the error control encoder and the channel encoder before being transmitted over the channel. At the receiver, on the contrary, after going through the channel decoder and the error control decoder, the serial data is converted back to parallel. By means of various simulations, we show that the operation of the designed SIE module fully meets the technical requirements.

Keywords: Serial Interface Engine; USB 2.0; FPGA; Xilinx ISE; verification.

1. GIỚI THIỆU

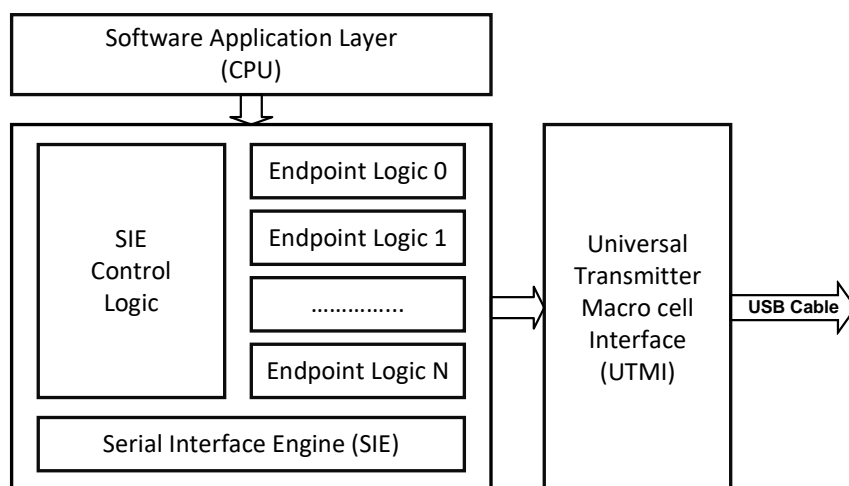
Universal Serial BUS (USB) là một giao diện truyền dữ liệu nối tiếp tiện lợi và nhanh chóng [1], thường được sử dụng trong các thiết bị được thiết kế bởi nhiều công ty khác nhau, như Intel, Compaq, NEC, Digital, Northern, IBM và Microsoft [2]. Trong đó thiết kế USB có thể được thực hiện bằng cách sử dụng nền tảng Field Programmable Gate Array (FPGA) [3], [4], dùng các ngôn ngữ mô tả phần cứng như Verilog, VHDL (VHSIC Hardware Description Language) để

thiết kế và xác thực. Thiết kế phức tạp đặt ra câu hỏi về tính khả thi của việc ứng dụng giao tiếp USB trong các thiết bị chuyên dụng và các phân khúc điện năng thấp của thiết bị điện tử, đặc biệt khi so sánh sang giao tiếp RS232 / UART truyền thống [5].

Bộ giao tiếp USB bao gồm hai đơn vị - Universal Transceiver Macrocell Interface (UTMI) và Parallel Interface Engine (PIE) [4]. UTMI có thể kết nối với cáp USB và có thể được sử dụng để đồng bộ hóa khung thời gian và truyền dữ liệu nối tiếp. Trong khi đó,

PIE chịu trách nhiệm về việc xây dựng hoặc trích xuất gói tin và giao tiếp với thiết bị ngoại vi. Các mô-đun nói trên được thiết kế bằng máy trạng thái hữu hạn và được triển khai bằng ngôn ngữ mô tả phần cứng Verilog [6], [7], [8]. Trong [9], mạch tích hợp dành

riêng cho ứng dụng (Application-Specific Integrated Circuit hay ASIC) của bộ truyền nhận USB được triển khai bằng công nghệ CMOS 0,18- μ m của TSMC ở [10] và được cải tiến hơn ở [9].



Hình 1. Kiến trúc tổng quát của bộ truyền nhận USB [1].

Sơ đồ chung của khối SIE được thể hiện trong Hình 1. Cụ thể như sau:

- Khối Logic Control SIE chứa Logic nhận dạng gói tin USB (PID), Logic nhận dạng địa chỉ và Logic tuần tự khác để quản lý các gói và giao dịch USB.
- Khối Logic Endpoint chứa nhận dạng số Endpoint, FIFO và Logic Control FIFO. Khối SIE nhận dữ liệu từ đơn vị xử lý trung tâm của PC và gửi dữ liệu đã chuyển đến UTMI [11].
- Khối UTMI xử lý giao thức USB cấp thấp và tín hiệu dữ liệu. Khối UTMI thực hiện các tác vụ như tuần tự hóa dữ liệu, giải mã dữ liệu, nhồi bit, khôi phục xung đồng hồ và đồng bộ hóa xung đồng hồ. Nhiệm vụ chính của UTMI là truyền dữ liệu từ USB sang USB khác tương thích với logic vật lý, chẳng hạn như tín hiệu vi sai D+ và D-.

Đóng góp chính của bài báo này là trình bày thiết kế chi tiết module SIE, phần quan trọng trong chuẩn giao tiếp USB 2.0. Sau đó các mô phỏng được thực hiện để xác thực tính đúng đắn của module SIE được thiết kế.

Khác với những bài báo đã xuất bản, chúng tôi thực hiện kiểm tra khả năng phát hiện lỗi bit được tạo ngẫu nhiên để gần với thực tế hơn.

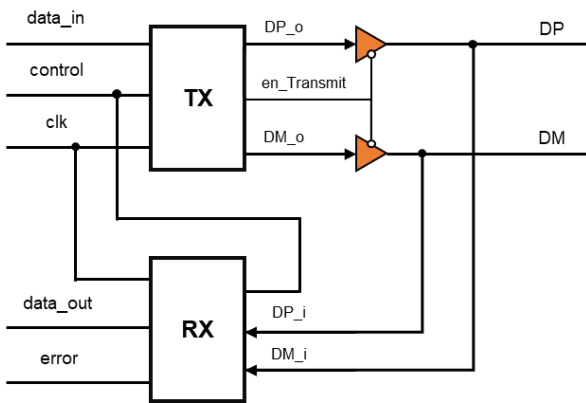
2. KIẾN TRÚC TỔNG QUÁT

Module SIE thực hiện mã hóa và giải mã dữ liệu nối tiếp, nhồi bit, phát hiện lỗi và các nhiệm vụ quản lý cấp tín hiệu khác. Gói tin định dạng phân tích và gói tin nhận dạng được thêm vào trong dữ liệu. Sau khi quá trình trên hoàn thành, dữ liệu ngõ vào được gửi đến thiết bị nhận. Bên nhận phản hồi xác nhận nếu dữ liệu nhận được chính xác. Nếu dữ liệu nhận được đã chính xác, dữ liệu sẽ được gửi đến khối UTMI (tầng vật lý). Nếu dữ liệu nhận được có lỗi thì sẽ hủy gói tin. Kiến trúc hệ thống được đề xuất của khối SIE (Hình 2) gồm 2 khối chính: Transmitter (TX) và Receiver (RX). Sơ đồ chi tiết của khối TX và RX được trình bày ở Hình 3 và 4 gồm Data Control, Parallel-Input Serial-Output (hay PISO), PID Check, Cyclic Redundancy Check 5 (hay CRC5), CRC16, Bit-Stuffing, NRZI Encoder, NRZI, Bit-Unstuffing và Serial-Input Parallel-Output (hay SIPO).

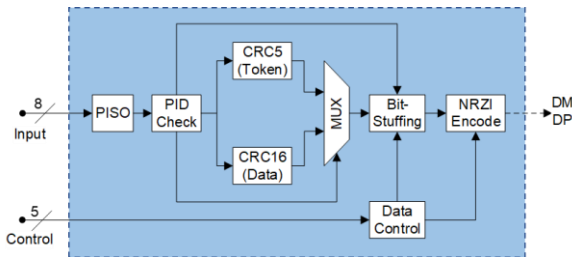
3. THIẾT KẾ CHI TIẾT HỆ THỐNG

3.1. Khối PISO

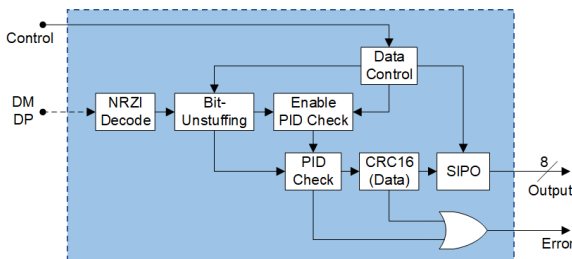
Khối PISO có chức năng chuyển đổi dữ liệu dạng song song thành nối tiếp thông qua cơ chế dịch từng bit ra ngõ ra. Bit có trọng số thấp nhất được truyền đi đầu tiên. Ban đầu khi có tín hiệu rst, khối PISO ở trạng thái IDLE cho đến khi nội dung thanh ghi dữ liệu nhận là gói tin SYNC thì chuyển sang trạng thái TRANSMIT.



Hình 2. Kiến trúc tổng quát của module SIE.



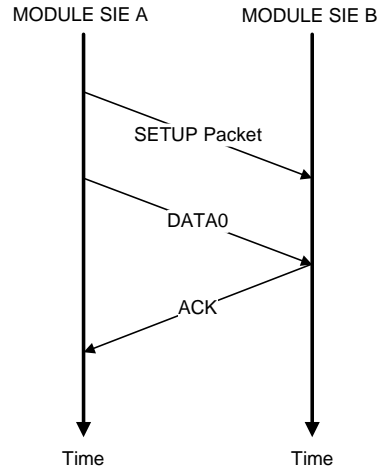
Hình 3. Kiến trúc khối TX của module SIE



Hình 4. Kiến trúc khối RX của module SIE.

3.2. Khối PID Check và Enable PID Check

Khối PID Check có chức năng kiểm tra định danh của các gói tin như Token, Data, Handshake được sử dụng trong giao thức USB (ví dụ như Hình 5). Khối PID Check được kích hoạt khi tín hiệu En_PID ở mức cao.

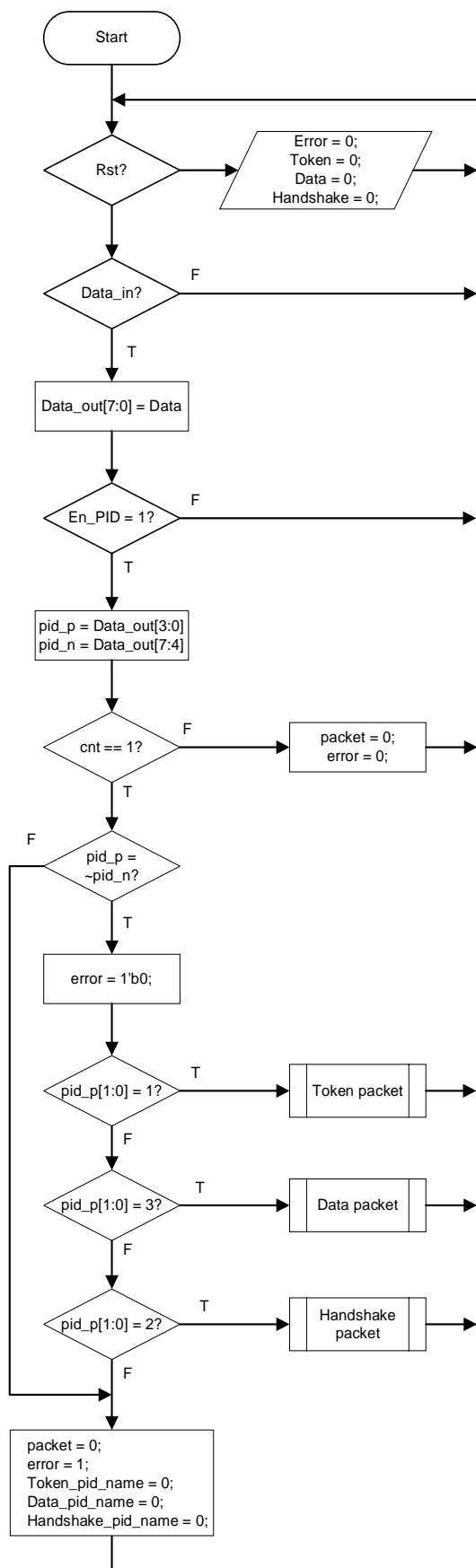


Hình 5. Bước Setup trong Control Transfer của giao thức USB.

Hình 6 minh họa lưu đồ giải thuật của khối PID check. Đầu tiên, dữ liệu đầu vào 8bit (Data_in) được lưu trữ trong một thanh ghi có ký hiệu Data_out [7: 0]. Nếu tín hiệu cho phép của khối kiểm tra PID ở mức cao, nửa byte cao (Data_out [7: 4]) được so sánh với phần bù của 1 nửa byte thấp (~Data_out [3: 0]). Nếu các nửa byte cao và thấp giống nhau thì việc phân tích và phân loại gói được tiến hành bằng cách sử dụng bảng tra cứu và bảng này đã được thiết lập trước ở bảng 1. Nếu hai nửa byte không giống nhau, tín hiệu error được đặt thành mức cao và dữ liệu đầu vào bị loại bỏ.

Bảng 1. Bảng phân loại gói tin trong giao thức USB.

Loại gói tin	Gói tin
Token	IN
	OUT
	SETUP
	SOF
Data	DATA0
	DATA1
	DATA2
	MDATA
Handshake	ACK
	NAK
	NYET
	STALL



Hình 6. Lưu đồ giải thuật của khối PID Check.

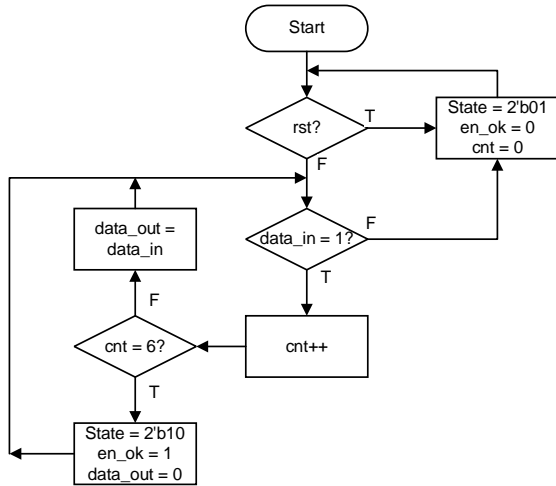
3.3. Khối CRC5, CRC16

Tính toán CRC5 được áp dụng cho tất cả các trường địa chỉ và dữ liệu đầu cuối USB chỉ ngoại trừ trường dữ liệu SYNC, lệnh USB và CRC. Trong đó, khối CRC5 dùng để kiểm tra lỗi bit trong gói tin Token. Khối CRC16 dùng để kiểm tra lỗi bit trong gói tin Data [1]. Đa thức biểu diễn của CRC5 được sử dụng là: $x^5 + x^2 + 1$. Tương tự, khối CRC16 dùng đa thức biểu diễn là: $x^{16} + x^{15} + x^2 + 1$. Khối CRC16 được thiết kế theo dạng thanh ghi dịch, Linear Feedback Shift Registers (hay LFSRs) [12].

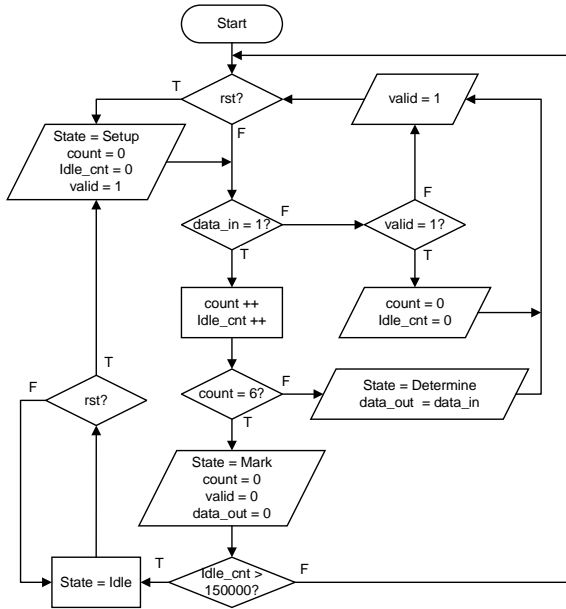
3.4. Khối Bit-Stuffing và Bit-Unstuffing

Bit-Stuffing là phương pháp chèn thêm một bit '0' sau mỗi 6 bit '1' liên tiếp. Hình 7 trình bày giải thuật của khối Bit-Stuffing. Ban đầu khi có tín hiệu rst, các tín hiệu state, en_ok, cnt được gán bằng 0 và tiếp tục giữ nguyên giá trị ban đầu cho đến khi data_in bằng 1 thì cnt được cộng 1. Khi cnt chưa bằng 6 thì data_out được gán bằng data_in. Ngược lại khi cnt bằng 6 (tức là có 6 bit '1' liên tiếp) thì state, en_ok và data_out được gán lần lượt cho 2'b10, '1' và '0'.

Hình 8 trình bày giải thuật cho khối Bit-Unstuffing. Ban đầu khi có tín hiệu rst, count và idle_cnt được gán bằng 0, valid bằng 1 và khối Bit-Unstuffing ở trạng thái Setup. Khi tín hiệu ngõ vào data_in = 1, count và idle_cnt tăng 1. Ngược lại, khi data_in = 0, nếu valid bằng 1 thì gán count và idle_cnt bằng 0. Nếu valid bằng 0 thì gán valid bằng 1. Khi count khác 6, khối ở trạng thái Determine. Tiếp tục cho đến khi count = 6 thì khối chuyển sang trạng thái Mark và gán valid, count, data_out bằng 0. Giả sử trạng thái được xây dựng để tránh các mức cao trong thời gian dài, ví dụ 3ms. Nếu tần số clock là 50MHz thì sẽ có 150.000 xung trong 3ms. Sau khoảng thời gian 3ms (tức idle_cnt > 150.000) thì quay về trạng thái Setup. Ngược lại sẽ ở trạng thái Idle và chờ tín hiệu rst.



Hình 7. Lưu đồ giải thuật của khối Bit-Stuffing.



Hình 8. Lưu đồ giải thuật của khối Bit-Unstuffing.

3.5. Khối NRZI Encoder và Decoder (Mã hóa và Giải mã NRZI)

Khối NRZI Encoder ở phía phát thực hiện ánh xạ tín hiệu nhị phân thành tín hiệu vật lý để truyền trên đường truyền. Ngược lại, khối NRZI Decoder thực hiện ánh xạ ngược tín hiệu vật lý nhận được từ đường truyền thành tín hiệu nhị phân ở phía thu.

3.6. Khối Data Control

Khối điều khiển dữ liệu điều khiển chức năng xử lý dữ liệu bằng cách cung cấp tín hiệu kích hoạt cho các khối như khối Bit-Stuffing, NRZI Encoder, Bit-Unstuffing, Enable PID Check và SIPO. Nếu các khối

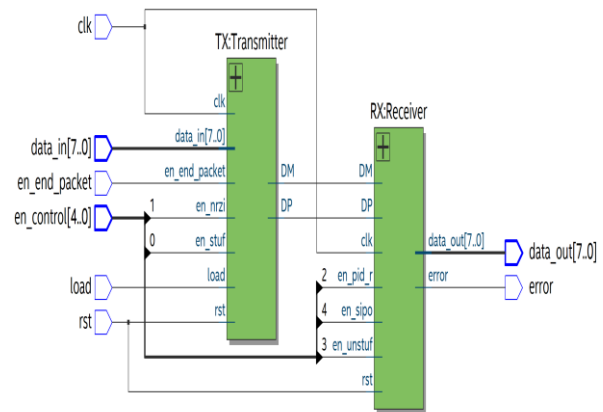
này không nhận được tín hiệu kích hoạt, chúng sẽ ở trạng thái không hoạt động.

4. KẾT QUẢ ĐÁNH GIÁ HIỆU NĂNG CỦA MODULE SIE

4.1. Kết quả tổng hợp

Hình 9 trình bày sơ đồ khối tổng quát của module SIE với 2 khối TX, RX tổng hợp được từ phần mềm. Ngõ vào clk và rst là các tín hiệu xung dao động và reset tại ngõ vào. Tín hiệu data_out và data_in là dữ liệu ngõ ra và vào 8 bit. Ngoài ra, tín hiệu en_control, load, en_end_packet và error lần lượt là tín hiệu điều khiển cho phép, tải dữ liệu, báo kết thúc và lỗi gói tin.

Tài nguyên cần sử dụng trong thiết kế được tóm tắt ở Hình 10 và 11. Nhìn chung, module yêu cầu rất ít tài nguyên sử dụng với chỉ 296 LE (logic element). Module SIE được thiết kế có thể hoạt động ở tần số Fmax = 285.55MHz.



Hình 9. Sơ đồ khối module SIE tổng hợp được.

Flow Summary	
Flow Status	Successful - Fri Feb 05 14:03:19 2021
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Lite Edition
Revision Name	top
Top-level Entity Name	top
Family	Cyclone IV E
Total logic elements	296
Total registers	188
Total pins	26
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0

Hình 10. Tóm tắt tài nguyên sử dụng của thiết kế.

Slow 1200mV 85C Model Fmax Summary			
<<Filter>>			
	Fmax	Restricted Fmax	Clock Name
1	285.55 MHz	250.0 MHz	clk
limit due to minimum p... (max I/O toggle rate)			

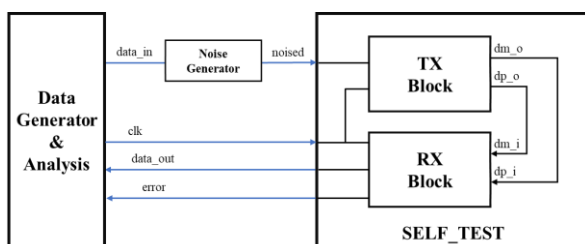
Hình 11. Tần số hoạt động tối đa của thiết kế.

4.2. Kết quả mô phỏng

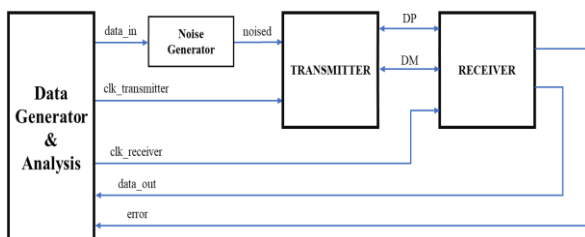
4.2.1. Mô hình testcase

Các mô hình dùng để mô phỏng self-test và truyền bất đồng bộ giữa khối SIE lần lượt được trình bày ở Hình 12 và 13. Trong cả 2 mô hình đều gồm có các khối như: Data Generator & Analysis (bộ tạo và phân tích dữ liệu), Transmitter (bộ phát), Receiver (bộ thu), Noise Generator (bộ tạo nhiễu). Trong mô hình truyền bất đồng bộ, khối SIE ở Transmitter và Receiver dùng 2 xung clock khác nhau và lệch pha nhau.

Khối Data Generator & Analysis chứa các task tạo chuỗi dữ liệu kiểm tra, tạo xung đồng hồ hoạt động và task hiển thị. Việc truyền dữ liệu đi và đọc các dữ liệu sẽ được hiển thị dưới dạng ký tự bằng các task hiển thị để dễ dàng kiểm tra. Khối Noise Generator chứa task tạo chuỗi dữ liệu lỗi ngẫu nhiên bằng cách thực hiện xor dữ liệu ngõ vào data_in với tín hiệu nhiễu có tỉ lệ xuất hiện lỗi gần với thực tế (3×10^{-6}) dùng để kiểm tra khả năng phát hiện lỗi của module SIE.



Hình 2. Mô hình testcase self-test.



Hình 3. Mô hình testcase truyền bất đồng bộ.

4.2.2. Tóm tắt các testcase

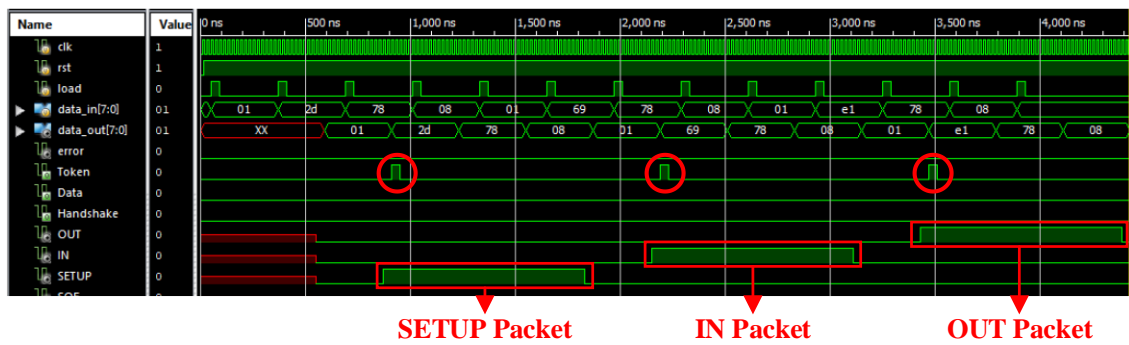
Thiết kế sẽ được mô phỏng qua 4 Testcase được liệt kê trong Bảng 2.

Bảng 2. Bảng mô tả tóm tắt testcase.

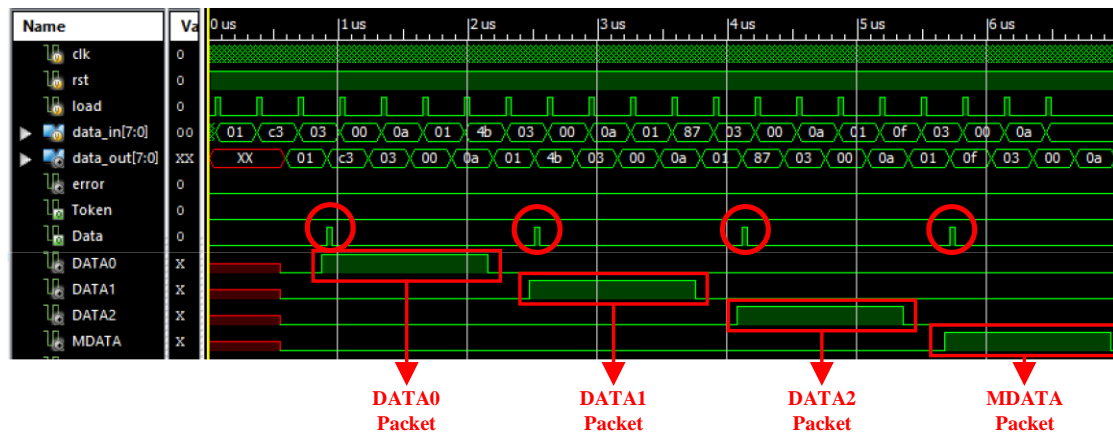
Testcase	Nội dung
1	Kiểm tra truyền dữ liệu trong mô hình self-test. Các gói tin gồm: Token, Data và Handshake.
2	Kiểm tra phát hiện lỗi ngẫu nhiên ở gói tin Data trong mô hình self-test. Tỉ lệ xuất hiện lỗi là 0.0003%.
3	Kiểm tra phát hiện lỗi ngẫu nhiên trong truyền dữ liệu bất đồng bộ giữa 2 khối SIE. Các gói tin lần lượt được kiểm tra tương tự như ở testcase 2. 2 module SIE có tần số hoạt động lệch pha nhau 180°.
4	Mô phỏng bước Setup trong chế độ truyền có điều khiển (Control Transfer) giữa module SIE A và B.

4.2.3. Testcase 1: Self-test nhận dạng gói

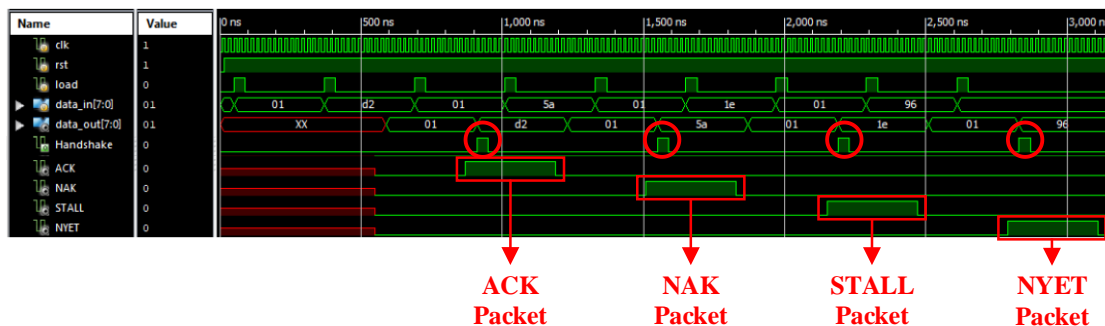
Mở đầu mỗi transaction là một gói tin Token nhằm báo hiệu mục đích của giao tiếp như IN, OUT và SETUP. Phần nội dung quan trọng nhất của mỗi transaction là dữ liệu được gửi đi (Data). Cuối của mỗi transaction là gói tin Handshake dùng để phản hồi gói tin Data. Các gói tin Handshake chỉ chứa 2 byte là SYNC và PID. Testcase này được thực hiện đầu tiên nhằm đảm bảo khối SIE hoạt động tốt trước khi thực hiện kiểm tra lỗi và truyền bất đồng bộ. Từ kết quả được trình bày ở Hình 14-16 cho thấy module SIE thực hiện đúng chức năng trên.



Hình 14. Dạng sóng mô phỏng phát hiện gói tin Token (Testcase 1).



Hình 15. Dạng sóng mô phỏng phát hiện gói tin Data (Testcase 1).



Hình 16. Dạng sóng mô phỏng phát hiện gói tin Handshake (Testcase 1).

4.2.4. Testcase 2: Self-test phát hiện lỗi ngẫu nhiên

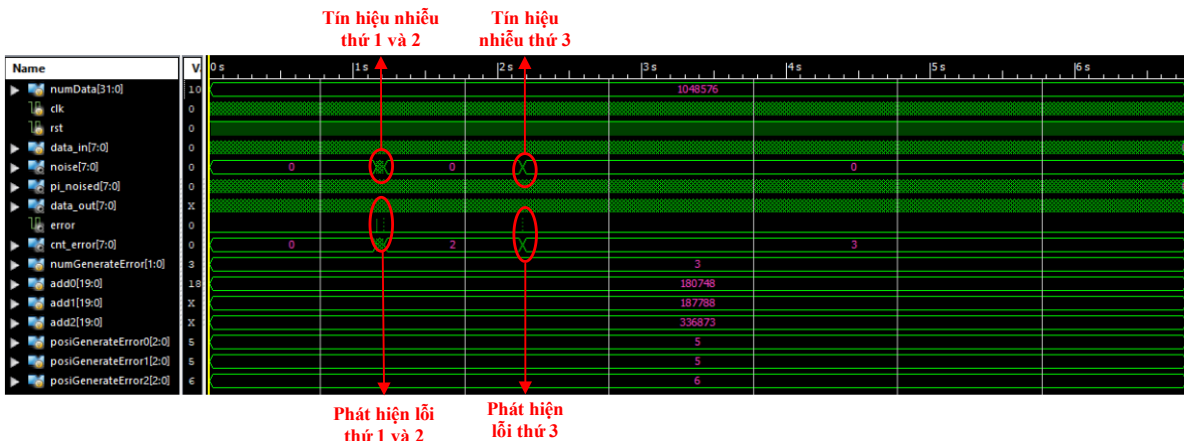
Testcase này được thực hiện nhằm đảm bảo module SIE phát hiện lỗi chính xác. Để tiết kiệm thời gian thì các gói tin Data được kiểm tra mang 16 byte chuỗi dữ liệu có nội dung giống nhau từ 0 đến 15. Dữ liệu mô phỏng trả về của Testbench sẽ được hiển thị dưới dạng chữ qua các task hiển thị như trình bày ở Hình 17. Kết quả mô phỏng được cho ở Hình 18 cho thấy module SIE hoạt động tốt. Với thông số:

- Num of Packets: số gói tin được kiểm tra
- Num of Errors Generated: số gói tin xuất hiện lỗi
- Packet Error: Vị trí của gói tin lỗi
- Position Error: Vị trí của byte lỗi trong gói tin
- Num of Errors Detected: Số gói tin lỗi phát hiện được

```

Console
ISim> run 1s
Num of Packets: 1048576
Num of Errors Generated: 3
Packet Error 0: 5
Position Error 0: 180748
Packet Error 1: 5
Position Error 1: 187788
Packet Error 2: 6
Position Error 2: 336873
Num of Errors Detected: 3
Stopped at time : 6773800990 ns : in File "D:\USB\USB_TESTCASE 2_1\tb.v" Line 235
    
```

Hình 17. Thông tin số lượng, vị trí gói tin và byte xuất hiện lỗi (Testcase 2)



Hình 18. Dạng sóng phát hiện lỗi ngẫu nhiên (Testcase 2).

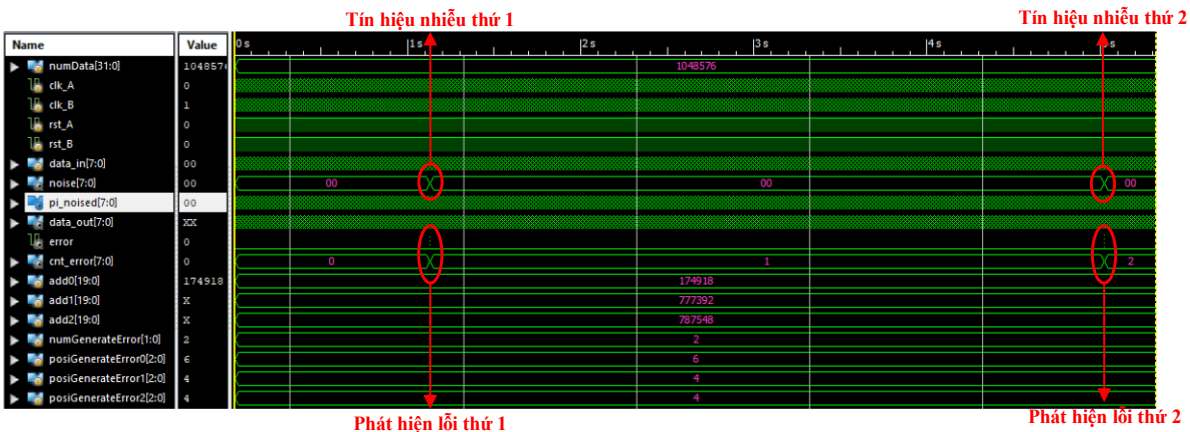
4.2.5. Testcase 3: Truyền bất đồng bộ phát hiện lỗi ngẫu nhiên

Testcase này được thực hiện nhằm đảm bảo 2 module SIE hoạt động tốt khi thực hiện truyền bất đồng bộ với nhau. Các gói tin Data được kiểm tra có nội dung tương tự như testcase 2 và được trình bày ở Hình 19. Kết quả mô phỏng được cho ở Hình 20 cho thấy module SIE hoạt động tốt ở mô hình truyền bất đồng bộ.

```

ISim> run 7s
A limit on number of transitions that can be stored in waveform database (.wdb) file is
tracing starting from time 0. You can trace only for the time widow of your interest b
with keyword "tracing limit" for more information
Num of Packets: 1048576
Num of Errors Generated: 2
Packet Error 0: 6
Position Error 0: 174918
Packet Error 1: 4
Position Error 1: 777392
Packet Error 2: 4
Position Error 2: 787548
Num of Errors Detected: 2
    
```

Hình 19. Thông tin số lượng, vị trí gói tin và byte xuất hiện lỗi (Testcase 3).

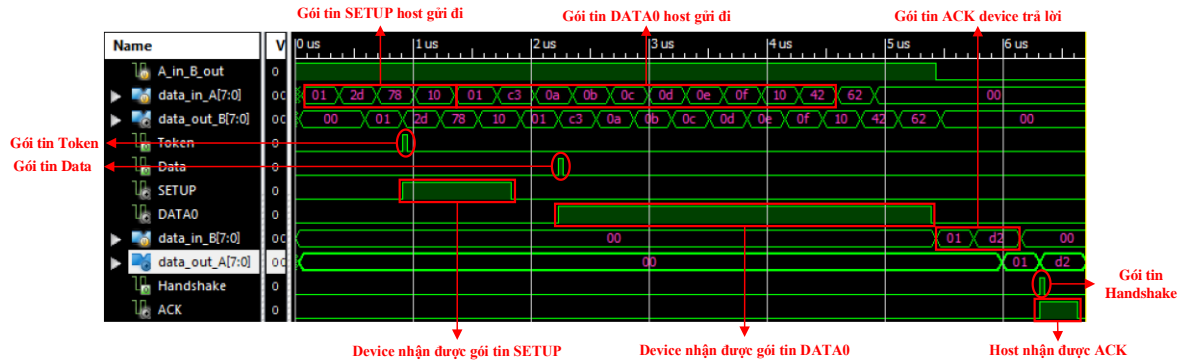


Hình 20. Dạng sóng phát hiện lỗi ngẫu nhiên (Testcase 3).

4.2.6. Testcase 4

Ở testcase này, chúng tôi thực hiện mô phỏng bước Setup trong chế độ truyền có điều khiển (Control Transfer) giữa 2

module SIE như được trình bày ở phần thiết kế khối PID Check. Kết quả mô phỏng được trình bày ở Hình 21 cho thấy rằng module SIE hoạt động tốt theo giao thức.



Hình 21. Dạng sóng mô phỏng của bước Setup (Testcase 4).

4.3. Đánh giá kết quả mô phỏng

Tất cả testcase đều được kiểm tra, so sánh và đều có kết quả đúng với yêu cầu thiết kế. Dựa vào đây có thể kết luận thiết kế có hoạt động tốt trên mô phỏng, đảm bảo chức năng, nhiệm vụ của module SIE ở cả mô hình self-test và truyền bất đồng bộ.

5. KẾT LUẬN

Trong bài báo này, chúng tôi đã thực hiện thiết kế module SIE trong chuẩn truyền thông USB 2.0 dựa trên công nghệ FPGA.

Sau đó, bài báo trình bày kết quả mô phỏng, đánh giá một cách tổng quát chức năng và hoạt động của module SIE được thiết kế. Trong khi hầu hết các thiết kế vi mạch ít được chia sẻ công khai và chi tiết, thiết kế module SIE trong bài báo này có giá trị cho việc học tập các môn liên quan tới thiết kế hệ thống số và thiết kế vi mạch số. Chúng tôi hướng đến thiết kế bộ truyền nhận USB 2.0 và USB 3.0 đầy đủ để thực thi thực tế trên nền tảng FPGA.

TÀI LIỆU THAM KHẢO

- [1] Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC, Philips, "Universal Serial Bus Specification", Revision 2.0, April 27, 2000.
- [2] L. Nardozza, "USB-IF Offers Guidance to Industry for Design Compliance of USB Devices," USB Implementers Forum (USB-IF), Barcelona, Spain, Feb. 21, 2016, pp. 1-3, <http://www.usb.org/>.
- [3] A. Bouhraoua and M. Al-Shammari, "A fundamentally secure payment device interfaced to regular PCs," in Proc. IEEE Region Conf., Kansas, MO, USA, Apr. 2008, pp. 1-5.
- [4] P. M. Szcwoka and K. J. Pyrzynski, "USB receiver/transmitter for FPGA implementation," 2012 International Conference on Signals and Electronic Systems (ICSES), Poland, Sep. 2012, pp. 1-6.
- [5] S. Kaul, P. Maheta, R. J. Rajesh and R. R. Das, "Novel multi-interface USB prototype device for merging commonly used peripheral devices," 2015 International Conference on Industrial Instrumentation and Control (ICIC), Pune, 2015, pp. 17-22.
- [6] Zainalabedin Navabi, Verilog Digital System Design: Register Transfer Level Synthesis, Testbench, and Verification, 2nd Edition, McGraw-Hill, 2005.
- [7] Xilinx ISE Web Pack, Ver 12.3. [Online]. Available: www.xilinx.com.

- [8] Xilinx Spartan -3 Family Complete Duta Sheet, Xilinx, San Jose, CA, USA, 2007.
- [9] G. Sung, L. Tung, H. Wang and J. Lin, "USB Transceiver with a Serial Interface Engine and FIFO Queue for Efficient FPGA-to-FPGA Communication," in IEEE Access, vol. 8, 2020, pp. 69788-69799.
- [10] G. Sung, H. Wang and J. Lin, "Serial interface engine asic with usb physical transceiver based on fpga development board," 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, 2017, pp. 410-413.
- [11] USB 2.0 Transceiver Macrocell Interface (UTMI) Specification, Version 1.05, Intel Corporation, Chennai, India, Mar.29, 2001.
- [12] Muthiah, D., and A. Arockia Bazil Raj. "Implementation of high-speed LFSR design with parallel architectures," 2012 IEEE International Conference on Computing, Communication and Applications, 2012, pp. 1-6.

Tác giả chịu trách nhiệm bài viết:

TS. Đỗ Duy Tân

Trường Đại học Sư phạm Kỹ thuật TP.HCM

Email: tandd@hcmute.edu.vn