



## ỨNG DỤNG CÔNG NGHỆ FPGA ĐỂ THIẾT KẾ BỘ TRUYỀN, NHẬN DỮ LIỆU GIAO TIẾP VỚI MÁY TÍNH TRÊN THIẾT BỊ DE1 QUA ĐƯỜNG TRUYỀN UART

**Đặng Lành<sup>1\*</sup>, Phạm Xuân Hải<sup>2</sup>, Trương Trường Sơn<sup>3</sup>**

<sup>1</sup> Trường Đại học Đà Lạt

<sup>2</sup> Viện Nghiên cứu Hạt nhân – Đà Lạt

<sup>3</sup> Trường Đại học Sư phạm Thành phố Hồ Chí Minh

Ngày nhận bài: 23-7-2018; ngày nhận bài sửa: 03-12-2018; ngày duyệt đăng: 21-12-2018

### TÓM TẮT

Ứng dụng công nghệ FPGA vào thiết kế, chế tạo các thiết bị điện tử khả lập trình cho phép khai thác thế mạnh của kỹ thuật lập trình, tạo sự mềm dẻo linh hoạt và hiệu quả cho thiết kế phần cứng. Bài báo này trình bày thiết kế mạch truyền, nhận dữ liệu 8 bit giữa FPGA (sử dụng bản mạch DE1 của hãng Altera) và máy tính theo chuẩn giao tiếp UART chuyển cổng USB. Sản phẩm thiết kế ứng dụng hiệu quả trong điều khiển tự động.

**Từ khóa:** DE1-Altera, FPGA, UART, USB.

### ABSTRACT

**Application of FPGA technology in design of data transfer-receive devices interfacing to PC in DE1 board via UART**

Application of FPGA technology in design and construction of programmable electronics devices enables the exploitation of the advantages of programming techniques, creating flexibility effectively for hardware design as well as control of peripheral devices. This work presents an 8-bit data transfer-receive circuit design between FPGAs (using Altera's DE1 board) and computer via UART standard to USB cable. Designed products can be used effectively for control of specialized automation.

**Keywords:** DE1\_Altera, FPGA, UART, USB.

### 1. Đặt vấn đề

Kỹ thuật điều khiển tự động hóa được coi là một trong những ngành mũi nhọn, then chốt của các trường đại học. Vì vậy, các trường cần đầu tư xây dựng nhiều mô hình, thí nghiệm thực hành về chuyên ngành tự động hóa. Với sự phát triển mạnh mẽ của công nghệ điện tử đương đại, những vi mạch bán dẫn có thể lập trình được đáp ứng kích thước nhỏ, công suất tiêu thụ thấp, tạo ra các hệ thống điều khiển tự động ngày càng thông minh hơn và giải quyết được nhiều bài toán phức tạp hơn, trong đó việc ứng dụng FPGA (*Field Programmable Gate Array*) là thiết bị lập trình được, và là công nghệ đột phá mới nhất hiện nay. FPGA có khả năng tái lập trình nên đạt độ linh hoạt cao, đáp ứng nhiều thuật

\* Email: lanhd@dlu.edu.vn

toán hay các yêu cầu khác nhau về phần cứng, do đó thích hợp với các mạch điều khiển tự động thông minh, giao tiếp PC, truyền nhận và xử lý dữ liệu ở tốc độ cao. Từ đó, sử dụng FPGA để thiết kế, xây dựng một số bài thí nghiệm thực hành dùng trong chuyên ngành tự động hóa là cần thiết. Để có bài thí nghiệm thực hành trên chip FPGA, quy trình thiết kế hoàn chỉnh từ xây dựng phần cứng đến phần mềm phải được chú trọng. Mục tiêu của bài báo là dùng bản mạch DE1, hãng Altera sử dụng ngôn ngữ mô tả phần cứng VHDL (Very high speed integrated circuit Hardware Description Language) để thiết kế mạch truyền, nhận dữ liệu 8 bit giữa ngoại vi với máy tính qua cổng giao tiếp RS 232 chuyển USB. Sản phẩm thiết kế ứng dụng hiệu quả trong điều khiển tự động.

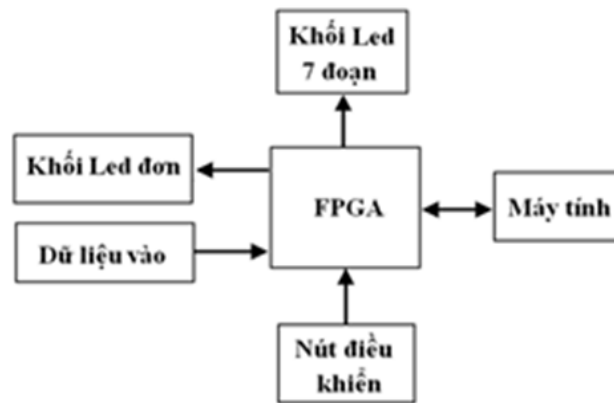
## 2. Khái quát về giao tiếp UART

UART (Universal Asynchronous Receiver Transmitter) gọi là bộ thu, phát dữ liệu không đồng bộ nối tiếp [2]. Truyền, nhận dữ liệu theo chuẩn UART có thể thực hiện theo phương thức song công, hay bán song công. Bộ truyền UART làm nhiệm vụ phát từng bit trong byte dữ liệu một cách tuần tự. Bộ thu UART chịu trách nhiệm lắp ghép các bit này lại thành các byte hoàn chỉnh. Mỗi UART gồm có hai thanh ghi dịch, được dùng làm thành phần cơ bản trong việc chuyển giữa nối tiếp sang song song và ngược lại, nghĩa là bộ phát có nhiệm vụ chuyển đổi 1 byte dữ liệu từ song song sang nối tiếp và thêm vào các bit start, stop để tạo thành khung truyền (frame). Dữ liệu truyền đi được điều khiển bằng một xung clock gọi là tốc độ baud, ở bộ thu sẽ lấy mẫu các tín hiệu, tập hợp lại và khôi phục tín hiệu gốc. Cơ chế lấy mẫu tín hiệu: một bit thông tin được chia thành 16 điểm lấy mẫu [2], điểm lấy mẫu tín hiệu là điểm giữa của bit thông tin.

## 3. Thiết kế bộ truyền, nhận trên FPGA

### 3.1. Sơ đồ khối tổng thể của thiết kế

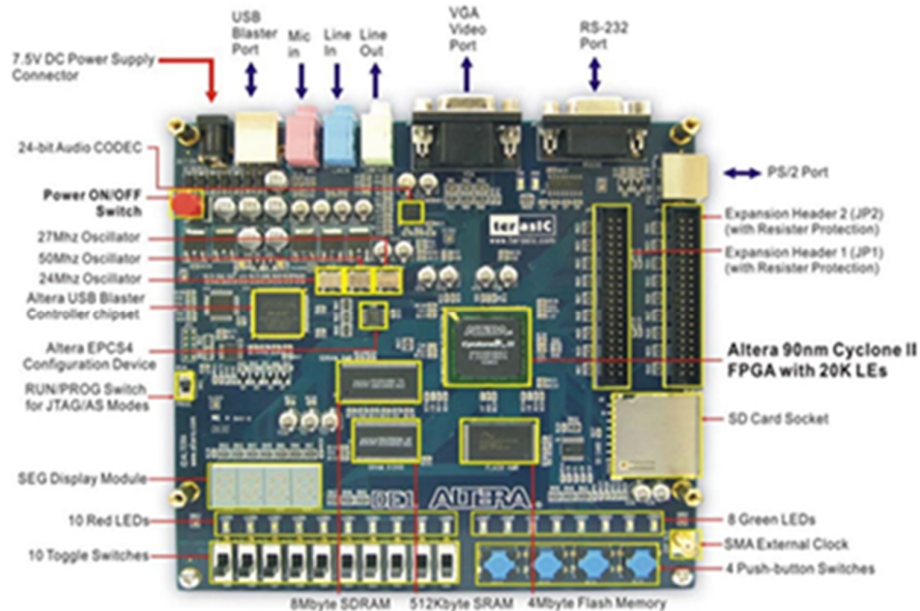
Sơ đồ khối cấu trúc tổng thể của mạch truyền, nhận dữ liệu giữa FPGA và máy tính được trình bày ở Hình 1. Mỗi khối có một chức năng hoạt động riêng, trong đó FPGA là khối trung tâm kết nối tất cả các khối chức năng còn lại. Các khối đó có nhiệm vụ cụ thể như sau: Khối led 7 đoạn – hiển thị dữ liệu từ FPGA gửi đi, dữ liệu FPGA nhận từ máy tính và hiển thị tham số cấu hình tốc độ baud; khối led đơn – hiển thị trạng thái các bit dữ liệu từ FPGA gửi lên máy tính, hiển thị trạng thái khi các nút điều khiển tích cực; khối dữ liệu vào – là các công tắc có thể ‘on’ hoặc ‘off’ thể hiện các bit dữ liệu vào là ‘1’ hoặc ‘0’; khối nút điều khiển – là các nút nhấn cho phép bộ truyền, nhận hoạt động, cho phép truyền dữ liệu, reset và nạp tham số cấu hình tốc độ baud; máy tính – thực hiện quá trình truyền nhận dữ liệu. Để kết nối với FPGA, tín hiệu từ cổng RS232 của máy tính phải qua một mạch chuyển đổi MAX232 [4] để trở thành tín hiệu theo chuẩn UART; FPGA – là mảng các phần tử logic lập trình được và được thiết kế bằng ngôn ngữ lập trình VHDL.



Hình 1. Sơ đồ khối tổng thể của thiết kế

### 3.2. Thực hiện thiết kế trên bản mạch DE1 của Altera

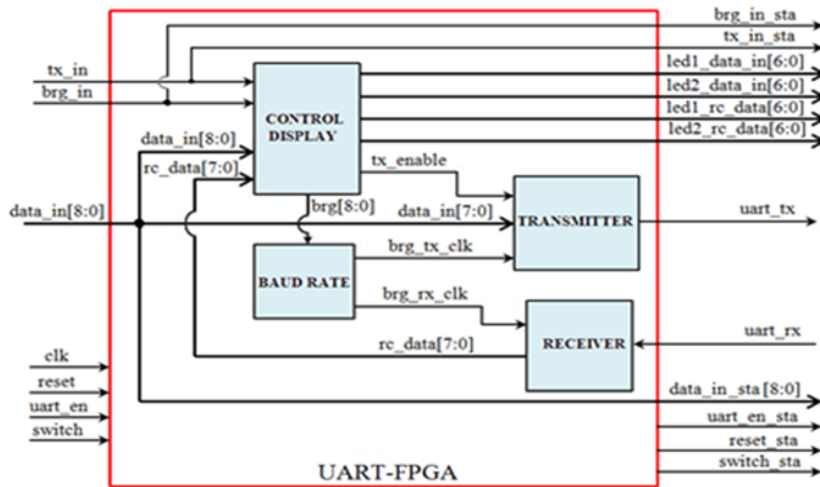
#### 3.2.1. Thiết kế các khối chức năng trên chip FPGA của bản mạch DE1



Hình 2. Bản mạch DE1

Bản mạch DE1 được trình bày ở Hình 2, sử dụng những công nghệ mới nhất cả về phần cứng và công cụ CAD (Computer Aid Design) [5], cung cấp một phương tiện tối ưu để nghiên cứu và phát triển về các lĩnh vực logic số [4]. Công trình nghiên cứu này sử dụng 10 nút chuyển mạch (switch), 4 nút nhấn (button), 10 led đơn đỏ (led red), 4 led đơn xanh (green led), 4 led 7 đoạn (7-seg display) và cổng RS232 để thiết kế. Sử dụng ngôn ngữ lập trình phần cứng VHDL [1], [2], [6] để thiết kế các khối chức năng của bộ truyền,

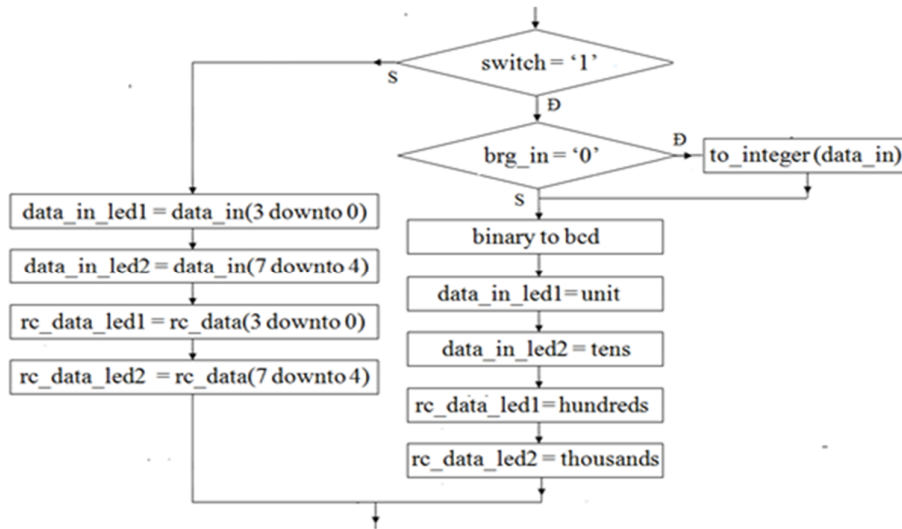
nhận UART trên chip FPGA – Cyclone II EP2C20F484C7. Cấu trúc chương trình viết theo mô hình gồm bốn phần tử con (Hình 3): Khối tạo tốc độ baud, khối điều khiển – hiển thị, khối truyền dữ liệu nối tiếp và khối nhận dữ liệu nối tiếp. Trong các khối con tác vụ xử lý đồng thời (process) được sử dụng để thực hiện các đoạn chương trình nhỏ xử lý song song như: Tạo ra các bộ đếm; bộ chia tần; bộ phát bit; bộ lấy mẫu từng bit.



Hình 3. Sơ đồ thiết kế các khối chức năng của bộ truyền nhận UART

3.2.2. Nguyên tắc hoạt động từng khối

Khối điều khiển – hiển thị (Control - Display)

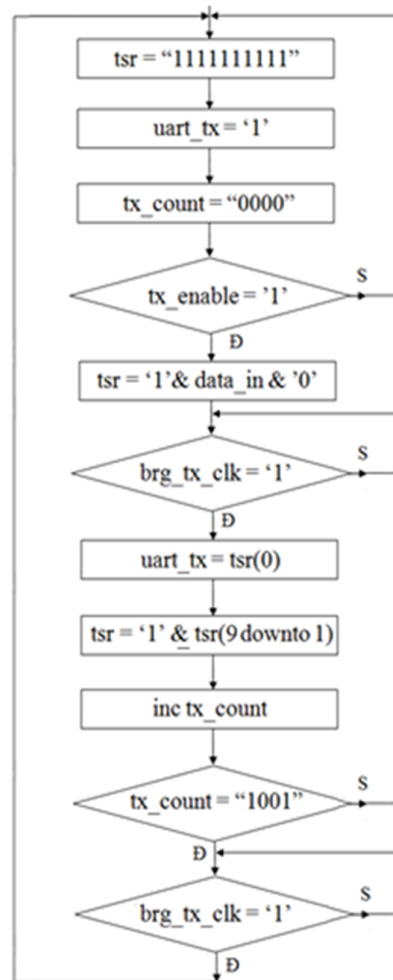


Hình 4. Lưu đồ thuật toán của khối hiển thị

Khối này (Hình 4) thực hiện hai chức năng: điều khiển và hiển thị. Tiếp nhận tín hiệu điều khiển tx\_in (mức thấp) để tạo ra tín hiệu tx\_enable (mức cao) cho phép khối truyền dữ liệu hoạt động, một lần nhấn nút tx\_in (có thể giữ nút kéo dài vài xung clock) chỉ có một xung tín hiệu tx\_enable cho phép bộ truyền hoạt động và truyền một khung dữ liệu. Chức năng thứ hai là hiển thị dữ liệu truyền, nhận và hiển thị giá trị tham số brg. Hình 7 trình bày lưu đồ thuật toán của khối hiển thị. Khi switch = '1', nếu nhấn nút brg\_in xuống mức thấp (brg\_in = '0'), dữ liệu vào data\_in sẽ chuyển sang số integer (to\_integer (dat\_in)) nạp vào tham số brg của khối baud rate, đồng thời dữ liệu data\_in cũng chuyển sang số BCD (Binary Coded Decimal). Sau đó chuyển đổi và tách thành 4 số thập phân tương ứng với hàng ngàn, trăm, chục, đơn vị, hiển thị 4 số thập đó lên 4 led 7 đoạn. Nếu switch = '0' dữ liệu vào data\_in và dữ liệu nhận rc\_data được hiển thị trên 4 led 7 đoạn.

- Khối truyền dữ liệu nối tiếp (Transmitter)

Khối truyền dữ liệu (Hình 5) nhận được tín hiệu cho phép truyền tx\_enable, khối truyền sẽ nhận 8 bit dữ liệu từ ngõ vào data\_in [7:0], sau đó chuyển dữ liệu từ dạng song song sang dạng nối tiếp và phát đi từng bit dữ liệu nối tiếp qua chân tín hiệu uart\_tx dựa trên tốc độ baud đã được tính trong khối baud rate. Quá trình truyền dữ liệu được thể hiện cụ thể ở lưu đồ thuật toán (Hình 8). Bắt đầu, thanh ghi truyền tsr được gán 10 bit trạng thái '1' (tsr = "1111111111"), ngõ ra bộ truyền uart\_tx gán bit '1' (uart\_tx = '1'); bộ đếm bit dữ liệu phát đi tx\_count gán bằng 0 (tx\_count = "0000"). Khi chưa có tín hiệu cho phép truyền (tx\_enable), bộ truyền đang ở trạng thái chờ; khi có tín hiệu cho phép truyền, 8 bit dữ liệu vào sẽ ghép với bit start (trạng thái '0') và bit stop (trạng thái '1') tạo thành khung truyền 10 bit. Bộ truyền chờ cạnh lên cạnh lên của xung truyền brg\_tx\_clk (brg\_tx\_clk = '1'), nếu có cạnh lên của xung truyền brg\_tx\_clk thì bit đầu tiên, bit start (có trọng số thấp nhất) của thanh ghi tsr được truyền đi, nghĩa là dữ liệu một bit này sẽ gán cho ngõ ra uart\_tx của bộ truyền; đồng thời bộ đếm bit dữ liệu phát



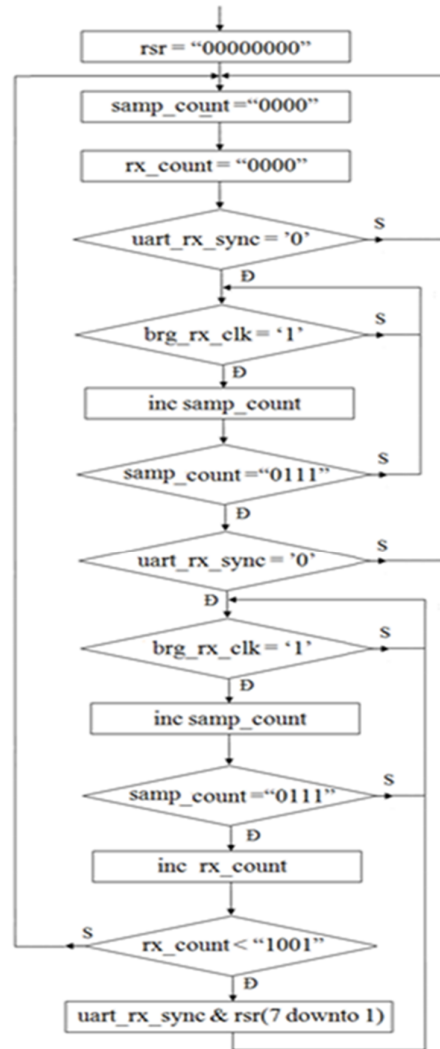
Hình 5: Lưu đồ thuật toán của khối truyền dữ liệu

tx\_count tăng lên một đơn vị. Nếu số lượng bit phát đi chưa đủ 10 bit thì bộ phát cứ tiếp tục phát bit dữ liệu lên đường truyền theo xung nhịp brg\_tx\_clk; nếu đủ 10 bit (tx\_count = "1001") thì đợi của xung nhịp brg\_tx\_clk (brg\_tx\_clk = '1') để kết thúc quá trình truyền 1 byte dữ liệu và bộ truyền trở về trạng thái chờ, chuẩn bị cho quá trình truyền dữ liệu tiếp theo.

- Khởi nhận dữ liệu nối tiếp (Receiver)

Khởi nhận dữ liệu nối tiếp (Hình 6), phát hiện bit start (mức '0'), sẽ lấy mẫu dữ liệu từ chân nhận tín hiệu nối tiếp uart\_rx; sau đó chuyển dữ liệu từ dạng nối tiếp sang dạng song song. Quá trình nhận dữ liệu được trình bày cụ thể ở lưu đồ thuật toán Hình 9. Bắt đầu, thanh ghi nhận dữ liệu rsr được gán 8 bit '0' (rsr = "00000000"); bộ đếm bit lấy mẫu samp\_count được gán bằng 0 (samp\_count = "0000"); bộ đếm bit dữ liệu nhận rx\_count cũng được gán bằng 0 (rx\_count = "0000"). Khi bit lấy mẫu chưa bằng '0' (uart\_rx\_sync = '1'), tức là bộ nhận chưa phát hiện được bit start nên bộ nhận dữ liệu đang ở trạng thái chờ; khi phát hiện được bit start, bộ nhận chờ cạnh lên của xung nhận brg\_rx\_clk (brg\_rx\_clk = '1'), nếu có cạnh lên của xung brg\_rx\_clk thì bộ đếm mẫu samp\_count tăng lên 1 đơn vị; tiếp theo, nếu bộ đếm mẫu đếm chưa đủ 8 bit thì vẫn tiếp tục đếm lên, khi bộ đếm đếm đủ 8 bit mà tín hiệu lấy mẫu vẫn là mức thấp thì bit start này hợp lí và chấp nhận chu trình tiếp theo là thu nhận các bit dữ liệu, còn nếu bit start không hợp lí thì bộ

truyền quay lại trạng thái chờ ban đầu. Chu trình tiếp theo thu các bit dữ liệu, trong khi bit lấy mẫu start vẫn ở mức thấp, bộ nhận chờ cạnh lên của xung nhận brg\_rx\_clk, nếu có cạnh lên của xung brg\_rx\_clk thì bộ đếm mẫu samp\_count tăng lên 1 đơn vị, nếu bộ đếm mẫu đếm chưa đủ 8 bit thì vẫn tiếp tục đếm lên đến khi bộ đếm đếm đủ 8 bit thì bộ đếm bit dữ liệu nhận rx\_count tăng lên 1 đơn vị, nghĩa là bộ nhận đã lấy mẫu (thu) được 1 bit dữ liệu và dịch bit dữ liệu đó vào thanh ghi rsr (uart\_rx\_sync & rsr (7 downto 1)). Tương tự



Hình 6: Lưu đồ thuật toán của khối nhận dữ liệu

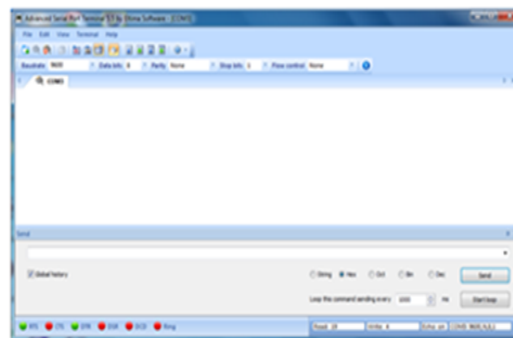
như vậy quá trình đếm mẫu (samp\_count) lấy mẫu (rx\_count) và dịch bit dữ liệu vào thanh ghi rsr cứ tiếp tục cho đến khi bộ nhận, nhận đủ 9 bit còn lại. Kết thúc quá trình nhận 1 byte dữ liệu và bộ nhận trở về trạng thái chờ, chuẩn bị cho quá trình nhận dữ liệu tiếp theo.

#### 4. Thục nghiệm và bình luận kết quả

Sử dụng phần mềm Quartus II 9.0 của hãng Altera [4] để soạn thảo và biên dịch thiết kế. Gán các tín hiệu vào, ra của thiết kế với các chân I/O của chip FPGA (Cyclone II EP2C20F484C7) trên bản mạch DE1, nạp thiết kế lên bản mạch DE1. Sơ đồ kết nối giữa bản mạch DE1 và máy tính được trình bày ở Hình 7. Dùng các switch (từ SW0 đến SW9) để thiết lập tốc độ baud và đưa dữ liệu vào cho bộ truyền; các KEY làm tín hiệu điều khiển, KEY0: tín hiệu cho phép truyền, KEY1: cho phép bộ truyền hoạt động, KEY2: reset và KEY3: nạp tham số brg; bốn Led 7 đoạn để hiển thị dữ liệu truyền và nhận giữa FPGA máy tính, và hiển thị tham số brg, thạch anh dao động cấp cho hệ thống 50MHz. Sử dụng phần mềm Advanced Serial Port Terminal 5.5 để thực hiện và quan sát quá trình truyền, nhận dữ liệu (Hình 8).



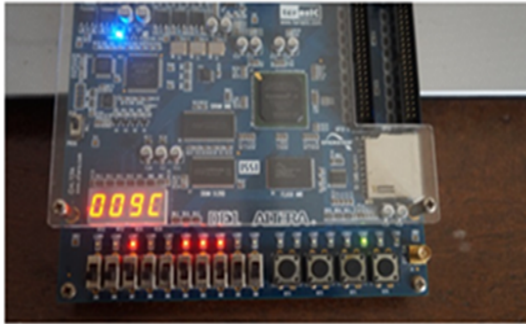
**Hình 7.** Kết nối giữa Bản mạch DE1 và máy tính



**Hình 8.** Giao diện phần mềm Advanced Serial Port Terminal 5.5

#### 4.1. FPGA truyền dữ liệu lên máy tính

Để thực hiện truyền, nhận dữ liệu thì phải chuyển SW9 xuống mức '0'. Đưa dữ liệu vào đường truyền bằng cách thay đổi trạng thái 8 bit của các switch (từ SW0 đến SW7), dữ liệu được hiển thị mã nhị phân trên 8 led đơn (từ LEDR0 đến LEDR7) và mã thập lục phân trên 2 led 7 đoạn (HEX0 và HEX1). Nhấn nút cho phép truyền dữ liệu (KEY0), một byte dữ liệu (ví dụ: 9C) hiện trên 2 led 7 đoạn HEX0, HEX1 sẽ truyền sang máy tính (Hình 9). Phần mềm Advanced Serial Port Terminal sẽ nhận dữ liệu này và hiển thị trên cửa sổ phần mềm (Hình 10). Tương tự có thể tiếp tục truyền nhiều byte dữ liệu khác nhau với nhiều tốc độ baud khác nhau để kiểm tra độ chính xác của bộ truyền.



**Hình 9.** Bảng mạch DEI truyền byte dữ liệu (9C) lên máy tính



**Hình 10.** Máy tính nhận byte dữ liệu (9C) từ bảng mạch DEI truyền đến

#### 4.2. FPGA nhận dữ liệu từ máy tính truyền đến

Vẫn giữ SW9 mức '0', từ cửa sổ phần mềm Advanced Serial Port Terminal (Hình 11), thực hiện gửi "send" một byte dữ liệu (ví dụ: A4) thì bảng mạch DEI nhận giá trị này và hiển thị trên hai led 7 đoạn HEX2 và HEX3 (Hình 12). Tương tự có thể tiếp tục gửi nhiều byte dữ liệu khác nhau với nhiều tốc độ baud khác nhau để kiểm tra độ chính xác của bộ nhận dữ liệu. Thực nghiệm với nhiều tốc độ baud khác nhau ở Bảng 1, thực hiện nhiều lần quá trình truyền, nhận dữ liệu, nhằm kiểm tra độ hoạt động ổn định của bộ truyền và bộ nhận, kết quả quá trình truyền, nhận hoạt động tốt, tuy nhiên ở tốc độ baud quá lớn thỉnh thoảng xảy ra lỗi.



**Hình 11.** Máy tính truyền byte dữ liệu (A4) lên Bảng mạch DEI



**Hình 12.** Bảng mạch DEI nhận byte dữ liệu (A4) từ máy tính truyền xuống

#### 4.3. Sai số của bộ truyền, nhận dữ liệu

Sai số của quá trình truyền, nhận được tính [3] theo công thức (1). Dựa vào công thức này, tính được sai số của một số tốc độ baud thực nghiệm trình bày trong Bảng 1.

$$Error(\%) = \left( \frac{BaudRate_{Closest\ Match}}{BaudRate} - 1 \right) \times 100 \quad (1)$$

**Bảng 1.** Sai số của một số tốc độ baud thực nghiệm

Số thứ tự	Baud Rate	BRG		Baud Rate (Closest Match)	ERROR (%)
		Float	Integer		
1	9600	324,5208	325	9586	-0,1
2	14400	216,0139	216	14401	0
3	19200	161,7604	162	19172	-0,1
4	38400	80,3802	80	38580	0,5
5	56000	54,8036	55	55804	-0,4
6	57600	53,2535	53	57870	0,5
7	115200	26,1267	26	115741	0,5
8	128000	23,4141	23	130208	1,7
9	230400	12,5634	13	223214	-3,1
10	256000	11,2070	11	260417	1,7
11	460800	5,7817	6	446429	-3,1
12	921600	2,3908	2	1041667	13,0

Dựa vào bảng số liệu cho thấy, với tần số thạch anh 50MHz (trên bản mạch DE1) hầu hết đều có sai số khi thiết kế bộ truyền, nhận UART. Sai số chấp nhận được từ tốc độ truyền 9600bps đến 115200bps. Ở tốc độ truyền từ 128000bps đến 921600bps thì sai số lớn, quá trình truyền, nhận dữ liệu ở các tốc độ này sẽ xảy ra lỗi. Tốc độ truyền 14400bps có sai số thấp nhất nên quá trình truyền, nhận dữ liệu ở tốc độ này là chính xác nhất và không bị lỗi. Ở tốc độ 921600bps có sai số lớn nhất nên quá trình truyền, nhận xảy ra lỗi nhiều nhất.

## 5. Kết luận

Bài báo đã trình bày được mạch thiết kế trên FPGA ghép nối với máy tính, phần cứng sử dụng là bản mạch DE1 của Altera và phần mềm là chương trình viết bằng ngôn ngữ VHDL. Kết quả thiết kế thực hiện được sự truyền, nhận dữ liệu giữa FPGA và máy tính qua cổng RS 232 với sai số nhỏ hơn 1% thì quá trình truyền, nhận tốt nhất và không bị lỗi. Sản phẩm thiết kế làm tài liệu tham khảo, giảng dạy và học tập chuyên ngành điều khiển tự động. Qua sản phẩm thiết kế giúp người thực hiện nắm rõ bản chất của lập trình phần cứng, người học có thể thiết kế được các hệ thống logic số, hiểu được các môn học liên quan kỹ thuật số, vi điều khiển, kỹ thuật vi xử lý. Đây là phương pháp dạy học tích cực mới giúp sinh viên tiếp cận công nghệ chế tạo phần cứng trên FPGA và FPGA có thể thay thế mọi hệ thống số từ các IC (Intergrated Circuit) số đơn giản đến các bộ vi điều khiển phức tạp với kì vọng phát triển thiết kế phong phú và đa dạng về chuyên ngành tự động hóa.

• **Tuyên bố về quyền lợi:** Các tác giả xác nhận hoàn toàn không có xung đột về quyền lợi.

#### TÀI LIỆU THAM KHẢO

- [1] Bài giảng thiết kế số dùng ngôn ngữ mô tả phần cứng VHDL, Trường Đại học Kinh tế – Kỹ thuật Công nghiệp, 2012.
- [2] Thiết kế logic số, *Học viện Kỹ thuật Quân sự Hà Nội*, 2011.
- [3] Datasheet ATmega16 – Atmel.
- [4] DE1 Development and Education Board User Manual – Altera.
- [5] Quartus II Introduction Using VHDL Design (2011) – Altera.
- [6] Volnei A. Pedroni, *Circuit Design with VHDL*. MIT Press Cambridge, Massachusetts London, England, 2004.