



ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  
KHOA KỸ THUẬT MÁY TÍNH

**ĐỒ ÁN THIẾT KẾ HỆ THỐNG SỐ VỚI HDL:  
GIAO TIẾP DỮ LIỆU SONG PHƯƠNG UART GIỮA  
FPGA VÀ MÁY TÍNH – THIẾT KẾ HƯỚNG ỨNG  
DỤNG THỰC TẾ TRÊN KIT DE2**

<b>Thành viên nhóm</b>	Trương Thiên Quý – MSSV: 23521321
	Ngô Đỗ Anh Quân – MSSV: 23521257
	Lê Huỳnh Thanh Phương – MSSV: 23521242
<b>Giáo viên hướng dẫn</b>	Hồ Ngọc Diễm
<b>Mã lớp</b>	CE213.P21

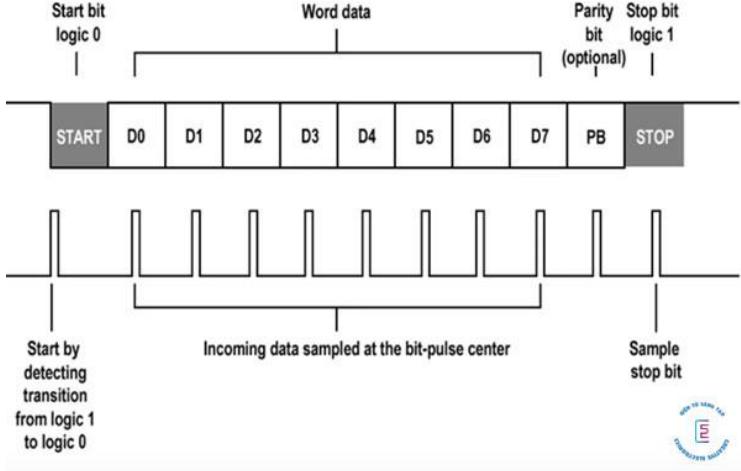
TP. HỒ CHÍ MINH, 2025

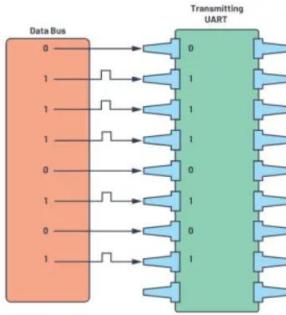
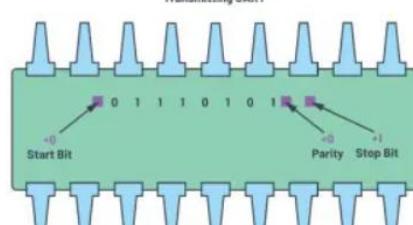
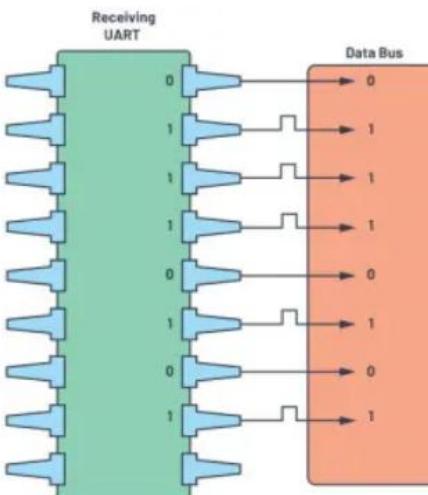
## Chương 1: Mô tả đề tài (Giới thiệu tổng quát)

<b><u>1. Giới thiệu chung</u></b>	<p>Trong bối cảnh công nghệ điều khiển và truyền thông nhúng ngày càng phát triển, nhu cầu giao tiếp giữa hệ thống điều khiển và thiết bị ngoại vi như máy tính là rất cần thiết. Giao tiếp UART (Universal Asynchronous Receiver/Transmitter) là một trong những phương pháp truyền dữ liệu nối tiếp phổ biến và đơn giản nhất hiện nay, đặc biệt trong các hệ thống nhúng và thiết kế phần cứng có thể lập trình như FPGA.</p> <p>Đề tài này tập trung vào việc thiết kế và hiện thực một hệ thống giao tiếp UART hoàn chỉnh trên nền tảng <b>FPGA DE2</b> của hãng Altera (Cyclone II). Thay vì sử dụng các module UART tích hợp sẵn hoặc IP core từ thư viện, toàn bộ giao tiếp UART sẽ được thiết kế từ đầu bằng <b>ngôn ngữ mô tả phần cứng Verilog HDL</b>, bao gồm các thành phần:</p> <ul style="list-style-type: none"><li>• <b>Bộ truyền UART (UART Transmitter)</b></li><li>• <b>Bộ nhận UART (UART Receiver)</b></li><li>• <b>Bộ chia tần số (Baud Rate Generator)</b></li><li>• <b>Hệ thống điều khiển và hiển thị dữ liệu truyền/nhận theo thời gian thực</b></li></ul>
<b><u>2. Mục tiêu đề tài</u></b>	<ul style="list-style-type: none"><li>✓ Hiểu và hiện thực chuẩn truyền thông UART một cách chi tiết từ tầng phần cứng.</li><li>✓ Thiết kế hệ thống giao tiếp nối tiếp hai chiều giữa FPGA và máy tính qua cổng RS232 hoặc USB.</li><li>✓ Hiển thị dữ liệu truyền và nhận thông qua <b>đèn LED 7 đoạn</b> và <b>LED đơn</b> trên kit DE2.</li><li>✓ Hỗ trợ cấu hình tốc độ baud tùy biến thông qua <b>công tắc SW</b> và <b>nút nhấn KEY</b>.</li><li>✓ Đánh giá hiệu năng truyền thông thông qua phần mềm mô phỏng giao tiếp Serial như PuTTY hoặc TeraTerm.</li></ul>
<b><u>3. Điểm nổi bật của đề tài</u></b>	<ul style="list-style-type: none"><li>► Thiết kế <b>thuần phần cứng</b> bằng Verilog, không phụ thuộc thư viện bên ngoài.</li><li>► Tái hiện chính xác cơ chế truyền UART: ghép bit start–data–stop và lấy mẫu bit nhận theo baud rate.</li><li>► Hỗ trợ thay đổi baud rate <b>động</b> trong quá trình chạy chương trình.</li><li>► Cho phép <b>tương tác hai chiều</b> giữa người dùng và FPGA thông qua <b>giao diện máy tính</b>.</li></ul>

	<p>► Có thể mở rộng thành mô hình giao tiếp ngoại vi hoặc ứng dụng điều khiển từ xa.</p>
<b><u>4. Phạm vi thực hiện</u></b>	<ul style="list-style-type: none"> <li>- Nền tảng phần cứng: FPGA DE2 (Cyclone II EP2C20F484C7), cổng RS232 hoặc USB-UART.</li> <li>- Ngôn ngữ: Verilog HDL</li> <li>- Công cụ: Quartus II, ModelSim (mô phỏng), phần mềm PuTTY/TeraTerm trên PC</li> <li>- Kết nối UART 2 dây: TX, RX</li> <li>- Hiển thị dữ liệu: 4 LED 7 đoạn (HEX0–HEX3), LED đơn (LEDR), switch và button để nhập liệu và điều khiển</li> </ul>
<b><u>5. Kết quả mong đợi</u></b>	<ul style="list-style-type: none"> <li>- Một hệ thống UART hai chiều có thể gửi và nhận dữ liệu 8-bit giữa FPGA và máy tính.</li> <li>- Dữ liệu nhập từ switch trên kit DE2 có thể gửi lên máy tính và ngược lại.</li> <li>- Giao diện truyền thông hoạt động ổn định ở nhiều tốc độ baud, có khả năng tùy chỉnh linh hoạt.</li> <li>- Tạo nền tảng cho các đề tài mở rộng như giao tiếp với cảm biến, module Bluetooth/GSM, hoặc truyền nhận dữ liệu trong mạng nhúng.</li> </ul>

## Chương 2: Cơ sở lý thuyết

<b><u>1. Giao tiếp UART là gì?</u></b>	UART (Universal Asynchronous Receiver/Transmitter) là một giao thức truyền thông nối tiếp <b>không đồng bộ</b> , cho phép truyền và nhận dữ liệu giữa hai thiết bị kỹ thuật số, ví dụ như <b>máy tính ↔ vi điều khiển</b> , <b>máy tính ↔ FPGA</b> , hoặc <b>FPGA ↔ module ngoại vi</b> (Bluetooth, WiFi, GPS...).
<b><u>2. Cấu trúc khung dữ liệu UART (frame)</u></b>	<p>- Khi một byte dữ liệu được truyền đi qua UART, nó sẽ được đóng gói thành một khung dữ liệu:</p>  <p>Trong đề tài này, ta dùng chuẩn <b>8N1</b>, nghĩa là:</p> <ul style="list-style-type: none"><li><b>1 bit Start</b> (logic ‘0’) → báo hiệu bắt đầu truyền</li><li><b>8 bit Data</b> (từ LSB → MSB)</li><li><b>No parity</b> (không dùng bit chẵn lẻ)</li><li><b>1 bit Stop</b> (logic ‘1’) → kết thúc khung truyền</li></ul> <p>Ví dụ: Gửi ký tự ‘A’ (mã ASCII 65 = 01000001): Khung UART truyền đi sẽ là:</p> <p style="text-align: center;">Start   0 1 0 0 0 0 0 1   Stop 0   1st bit → 8th bit  1</p>
<b><u>3. Baud Rate và chia tần số</u></b>	Baud rate là số lượng bit truyền đi trong một giây. Vì UART là <b>giao tiếp không đồng bộ</b> , nên baud rate phải được thiết lập giống nhau ở cả hai bên.

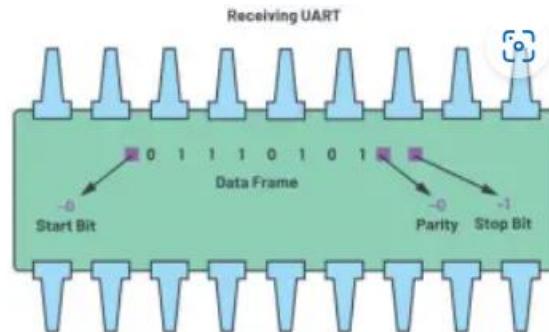
	<p>Kit DE2 dùng thạch anh <b>50 MHz</b> → cần chia tần để tạo ra tín hiệu UART phù hợp. Ví dụ, để tạo ra baud rate 9600bps:</p> $\text{Tần số chia (BRG)} = 50,000,000 / (9600 \times 16) \approx 325$ <p><b>Lưu ý:</b> Hệ thống UART thực hiện lấy mẫu dữ liệu nhiều lần trên mỗi bit (thường là 16x), để tăng độ tin cậy. Vì vậy hệ số chia còn tính thêm hệ số 16.</p>
<p><b>4. Cấu trúc bộ truyền UART (UART Transmitter)</b></p>	<p><b>Chức năng:</b> Nhận dữ liệu 8 bit song song → thêm bit start/stop → truyền tuần tự từng bit ra chân UART_TX.</p> <p><b>Nguyên lý hoạt động:</b></p> <ul style="list-style-type: none"> <li>UART truyền nhận dữ liệu song song từ bus dữ liệu.</li> </ul>  <ul style="list-style-type: none"> <li>Khi có tín hiệu tx_enable, bộ truyền ghép dữ liệu thành chuỗi 10 bit:</li> </ul> <p style="text-align: center;"><b>start_bit (0) + data[7:0] + stop_bit (1)</b></p>  <ul style="list-style-type: none"> <li>Mỗi bit được phát ra theo nhịp xung của baud_tick</li> </ul>  <ul style="list-style-type: none"> <li>Khi gửi xong, bộ truyền trở lại trạng thái chờ</li> </ul>

## 5. Cấu trúc bộ nhận UART (UART Receiver)

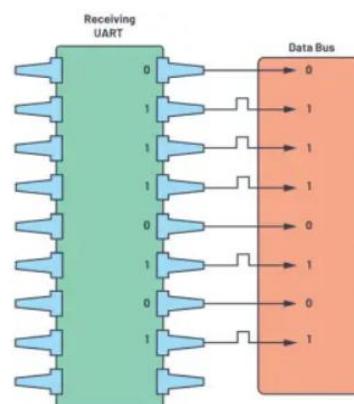
**Chức năng:** Nhận dữ liệu nối tiếp từ chân UART\_RX → phân tích khung → chuyển thành dữ liệu song song 8 bit.

**Nguyên lý hoạt động:**

- Luôn theo dõi đường UART\_RX để phát hiện **bit start** (**mức 0**)



- Sau khi phát hiện bit start, hệ thống:
  - Chờ **nửa chu kỳ baud** để vào giữa bit
  - Sau đó lấy mẫu liên tiếp 8 bit tiếp theo (1 mẫu/chu kỳ baud)
  - Kiểm tra bit stop (mức 1), nếu đúng → cho ra dữ liệu hợp lệ



## 6. Mô hình UART trên FPGA

Để hiện thực giao tiếp UART bằng Verilog, ta cần thiết kế các khối logic mô phỏng phần cứng sau:

Tên khối	Chức năng
Baud Generator	Chia tần số clock 50MHz thành xung phù hợp với baud rate UART
UART_TX	Truyền dữ liệu: từ song song sang nối tiếp
UART_RX	Nhận dữ liệu: từ nối tiếp sang song song
Control & Display	Điều khiển việc truyền, cấu hình baud, hiển thị dữ liệu

## 7. Tương tác thực tế

- Dữ liệu đầu vào nhập bằng công tắc (SW0–SW7)
- Nhấn nút (KEY0) để bắt đầu truyền dữ liệu
- Dữ liệu gửi đi được hiển thị trên HEX0–1
- Dữ liệu nhận được từ máy tính hiển thị trên HEX2–3
- Giao tiếp thực hiện qua **cáp chuyển USB–UART** (ví dụ CP2102 hoặc PL2303)

## 8. Ứng dụng thực tế

- Giao tiếp UART là nền tảng để kết nối với các module như:
  - HC-05 Bluetooth
  - ESP8266 WiFi
  - GSM SIM800L
- ➔ Cũng là cơ sở cho việc truyền dữ liệu từ/đến các thiết bị ngoại vi, cảm biến, điều khiển qua máy tính...

### Chương 3: Mô hình thiết kế đề xuất

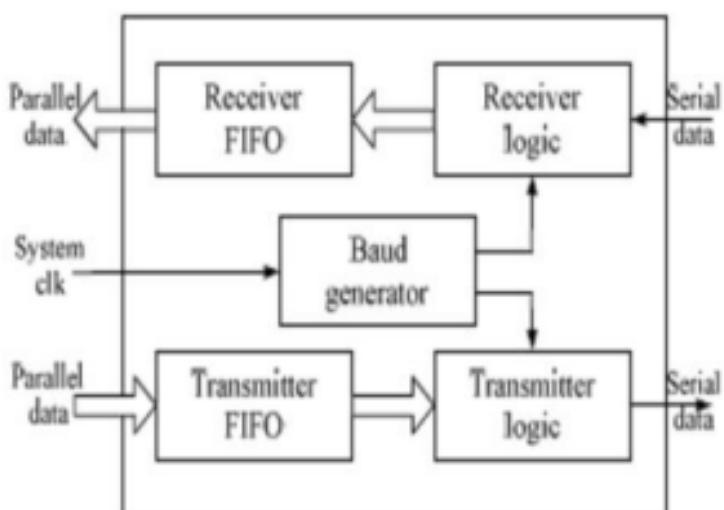
<p><b><u>Mục tiêu tổng thể của mô hình</u></b></p>	<p>Thiết kế một hệ thống giao tiếp <b>UART hai chiều</b> giữa FPGA (kit DE2) và máy tính, hiện thực bằng Verilog, với các tính năng:</p> <ul style="list-style-type: none"> <li>Gửi dữ liệu từ <b>SW0–SW7</b> trên kit → PC</li> <li>Nhận dữ liệu từ PC → hiển thị lên <b>HEX2–HEX3</b></li> <li>Cấu hình tốc độ truyền động bằng công tắc (SW9) và nút nhấn (KEY)</li> <li>Hiển thị dữ liệu truyền/nhận trên <b>LED 7 đoạn</b></li> </ul>
<p><b><u>1. Sơ đồ khái tổng thể</u></b></p>	<p>Khi truyền dữ liệu từ Máy tính (PC) sang DE2:</p> <pre> graph LR     PC["Máy tính (PC) [Virtual Terminal]"] -- TX --&gt; DE2["FPGA DE2 [UART_RX Block]"]     DE2 -- RX --&gt; HEX["Hiển thị trên HEX[2-3]"]     DE2 -- "Song song" --&gt; HEX   </pre> <p>Khi truyền dữ liệu từ DE2 sang Máy tính (PC):</p> <pre> graph LR     PC["Máy tính (PC) [Virtual Terminal]"] &lt;-- RX --&gt; DE2["FPGA DE2 [UART_TX Block] Hiển thị trên HEX[0-1]"]     DE2 &lt;-- "Song song" --&gt; SW["Điều khiển Công tắc SW[7-0]"]   </pre>
<p><b><u>2. Chi tiết các khối chức năng</u></b></p>	<p><b><u>2.1. Baud Rate Generator (BRG):</u></b></p> <ul style="list-style-type: none"> <li><b>Chức năng:</b> Chia tần số 50 MHz thành xung baud phù hợp với tốc độ truyền UART (ví dụ 9600 bps).</li> <li><b>Cấu hình tốc độ:</b> Sử dụng SW[8:9] để chọn hệ số chia tương ứng với các mức baud khác nhau.</li> <li><b>Tín hiệu ra:</b> baud_tick (xung đều đặn ứng với từng bit UART)</li> <li>•</li> </ul> <p><b><u>2.2. UART Transmitter (UART_TX):</u></b></p> <ul style="list-style-type: none"> <li><b>Chức năng:</b> Nhận dữ liệu 8 bit từ switch, đóng khung dữ liệu UART (start–data–stop), phát nối tiếp từng bit ra chân UART_TX</li> <li><b>Tín hiệu điều khiển:</b> <ul style="list-style-type: none"> <li>tx_enable: bật truyền khi nhấn nút KEY0</li> </ul> </li> <li><b>Tín hiệu ra:</b> <ul style="list-style-type: none"> <li>UART_TX: đầu ra nối tiếp, truyền sang máy tính</li> </ul> </li> </ul>

	<p><b>2.3. UART Receiver (UART_RX)</b></p> <ul style="list-style-type: none"> <li>• <b>Chức năng:</b> Nhận dữ liệu từ máy tính, phát hiện khung truyền hợp lệ, giải mã và đưa ra dạng song song.</li> <li>• <b>Tín hiệu vào:</b> <ul style="list-style-type: none"> <li>◦ UART_RX: tín hiệu nối tiếp đầu vào từ máy tính</li> </ul> </li> <li>• <b>Tín hiệu ra:</b> <ul style="list-style-type: none"> <li>◦ rx_data[7:0]: dữ liệu 8 bit nhận được</li> <li>◦ rx_ready: báo hiệu có dữ liệu mới</li> </ul> </li> </ul>
	<p><b>2.4. Hiển thị dữ liệu</b></p> <ul style="list-style-type: none"> <li>• <b>HEX[0]–HEX[1]:</b> hiển thị giá trị truyền (TX)</li> <li>• <b>HEX[2]–HEX[3]:</b> hiển thị giá trị nhận (RX)</li> <li>• <b>LEDR[7:0]:</b> hiển thị nhị phân dữ liệu đang nhập từ SW[7:0]</li> </ul>
	<p><b>2.5. Điều khiển giao tiếp</b></p> <ul style="list-style-type: none"> <li>• <b>SW[7:0]:</b> nhập dữ liệu cần gửi</li> <li>• <b>SW[8:9]:</b> chọn tốc độ truyền</li> <li>• <b>KEY[0]:</b> nút nhấn để truyền dữ liệu (tx_enable)</li> <li>• <b>KEY[1]:</b> reset hệ thống</li> <li>• <b>KEY[2]:</b> nạp lại giá trị baud rate (nếu cần thiết)</li> </ul>
<b>3. Kết nối với máy tính</b>	<ul style="list-style-type: none"> <li>- UART_TX và UART_RX của FPGA được kết nối đến module chuyển USB–UART (VD: CP2102)</li> <li>- Dùng phần mềm terminal như <b>PuTTY</b>, <b>TeraTerm</b>, <b>RealTerm</b> để quan sát dữ liệu</li> <li>- Cài đặt trên máy tính: <ul style="list-style-type: none"> <li>• Baud rate: phù hợp với thiết lập của FPGA</li> <li>• Data bits: 8</li> <li>• Stop bit: 1</li> <li>• No parity</li> </ul> </li> </ul>
<b>4. Ví dụ dòng truyền dữ liệu</b>	<p><b>1/ Truyền:</b></p> <ul style="list-style-type: none"> <li>• Người dùng bật SW[7:0] = 01000001 (ASCII ‘A’)</li> <li>• Nhấn KEY0 → UART_TX đóng gói 0_01000001_1</li> <li>• Gửi đi 10 bit theo nhịp baud_tick</li> <li>• Máy tính nhận và hiển thị ký tự ‘A’</li> </ul> <p><b>2/ Nhận:</b></p> <ul style="list-style-type: none"> <li>• Người dùng gửi ‘B’ từ PuTTY</li> <li>• UART_RX phát hiện start bit → lấy mẫu → ghép lại byte 01000010</li> <li>• Dữ liệu hiện lên HEX2–3 là mã ASCII của ‘B’</li> </ul>

## Tóm tắt mô hình:

Khối	Chức năng chính	Thiết bị liên quan
Baud Generator	Tạo xung baud chính xác theo tần số	Clock 50 MHz
UART TX	Truyền dữ liệu từ SW[7:0] đến máy tính	SW, LEDR, HEX[0–1], KEY[0]
UART RX	Nhận dữ liệu từ máy tính và giải mã thành byte	UART_RX, HEX[2–3]
Điều khiển	Reset, enable, chọn baud	KEY[1], KEY[2], SW[8:9]
Giao tiếp với PC	Kết nối 2 dây TX, RX với module USB-UART	CP2102, PL2303, PuTTY

## Mô hình thiết kế cơ bản với FIFO:



## Chương 4: Kế hoạch kiểm tra thiết kế UART trên FPGA (DE1)

<b>I. MỤC ĐÍCH</b>	Đảm bảo thiết kế bộ truyền–nhận dữ liệu 8-bit hoạt động <b>chính xác, ổn định</b> giữa FPGA và máy tính thông qua chuẩn UART, với nhiều tốc độ baud khác nhau, ở cả hai chiều: <b>FPGA → PC</b> và <b>PC → FPGA</b> .	
<b>II. THIẾT BỊ – CÔNG CỤ</b>	<b>Thiết bị/Dụng cụ</b>	<b>Mục đích sử dụng</b>
	Bo mạch FPGA DE1 (Altera)	Thiết kế phần cứng truyền nhận dữ liệu
	Cáp USB–RS232 + MAX232	Chuyển đổi tín hiệu UART → USB
	Máy tính + phần mềm Terminal	Quan sát dữ liệu truyền nhận
	Switch (SW0–SW9)	Nhập dữ liệu và cấu hình tốc độ truyền
	Nút nhấn (KEY0–KEY3)	Điều khiển chức năng truyền, reset, nạp tốc độ
	LED đơn + LED 7 đoạn	Hiển thị dữ liệu truyền, nhận, và thông số baud
	Quartus II	Thiết kế, tổng hợp và nạp chương trình vào FPGA
<b>III. CẤU HÌNH CỔNG UART</b>	<b>Thông số</b>	<b>Giá trị</b>
	Tốc độ baud	9600 – 921600 (kiểm tra nhiều mức)
	Dữ liệu mỗi frame	8 bit
	Stop bit	1
	Parity	None
	Giao tiếp	Full Duplex (gửi + nhận)

#### IV. CÁC BÀI KIỂM TRA CỤ THỂ

<b>TEST 1 – Kiểm tra hiển thị và thiết bị đầu vào</b>	<b>Bước</b>	<b>Mô tả hành động</b>	<b>Kết quả mong đợi</b>	
	1	Gạt các SW0–SW7 với tổ hợp bất kỳ (ví dụ: 10110010)	8 LED đơn (LEDR0–7) sáng đúng theo dữ liệu	
	2	Xem HEX0, HEX1 hiển thị đúng mã hex (ví dụ: 0xB2)	Đúng mã hex hiển thị	
	3	Gạt SW8–SW9 và nhấn KEY3	Tốc độ baud hiển thị đúng trên 4 LED 7 đoạn	
<b>TEST 2 – FPGA truyền dữ liệu lên máy tính</b>	<b>Bước</b>	<b>Hành động</b>	<b>Kiểm tra</b>	
	1	Nhập dữ liệu bằng SW0–7 (ví dụ 10011100 → 0x9C)	HEX0, HEX1 hiển thị 9C	
	2	Nhấn <b>KEY0</b> để truyền	Gửi 1 byte lên máy tính	
	3	Trên phần mềm Terminal, nhận đúng 0x9C	Thành công nếu dữ liệu đúng	
	4	Thử lại với nhiều giá trị khác (00, FF, A5...)	Máy tính nhận chính xác	
	5	Thay đổi tốc độ baud (bằng SW + KEY3) → test lại	Máy tính vẫn nhận đúng nếu tốc độ phù hợp	
<b>TEST 3 – Máy tính gửi dữ liệu xuống FPGA</b>	<b>Bước</b>	<b>Hành động</b>	<b>Kiểm tra</b>	
	1	Gửi byte A4 từ phần mềm Terminal	HEX2, HEX3 trên DE1 hiển thị A4	
	2	Gửi thêm nhiều giá trị khác (00, FF, C3, ...)	Tất cả đều hiển thị chính xác	
	3	Kiểm tra khi gửi liên tục nhiều byte (dòng dữ liệu)	Hiển thị l่าน lượt, không mất dữ liệu	
	4	Thay đổi tốc độ baud gửi từ máy tính (đúng cấu hình FPGA)	Dữ liệu vẫn nhận ổn định	
<b>TEST 4 – Kiểm tra độ ổn định với nhiều tốc độ truyền</b>	<b>Tốc độ baud</b>	<b>Dữ liệu gửi/nhận</b>	<b>Sai số chấp nhận</b>	<b>Kết quả</b>
	9600	OK	~0.1%	Đúng, ổn định
	14400	OK	0%	Hoạt động chính xác nhất
	19200	OK	~0.1%	Ôn định
	115200	OK	~0.5%	Một vài lỗi nhẹ có thể
	230400+	Không ổn định	>3%	Đã lỗi dữ liệu

<b>TEST 5 – Kiểm tra lỗi và xử lý ngoại lệ</b>	<b>Bước kiểm tra</b>	<b>Mô tả</b>	<b>Kết quả mong đợi</b>
	Truyền dữ liệu với baud lệnh giữa FPGA và máy tính	Có tình trạng sai	Dữ liệu bị lỗi hoặc không hiện
	Truyền dữ liệu liên tục tốc độ cao	Gửi liên tiếp 100 byte từ PC	Không mất dữ liệu nếu baud đúng
	Gửi sai định dạng UART (parity, stop bit)	(Nếu hệ thống hỗ trợ parity)	Phát hiện lỗi hoặc bỏ qua dữ liệu

## V. TIÊU CHÍ ĐÁNH GIÁ THIẾT KẾ

- ✓ Tốc độ truyền ổn định ở mức baud chuẩn ( $\leq 115200$ )
- ✓ Dữ liệu chính xác cả chiều truyền và nhận
- ✓ Hiển thị đúng trạng thái và dữ liệu qua LED
- ✓ Thiết kế hoạt động ổn định trong thời gian dài
- ⚠ Nhận diện và phản hồi lỗi nếu cấu hình sai tốc độ hoặc định dạng

## Chương 5: Tổng Kết

### 1. HIỆU QUẢ HOẠT ĐỘNG MONG ĐỢI:

#### 1.1/ Đánh giá theo tốc độ truyền (baud rate):

- Giao tiếp UART giữa FPGA và máy tính hoạt động tốt ở các tốc độ phổ biến, đảm bảo dữ liệu không bị mất hoặc lỗi.

Baud Rate	Hoạt động	Sai số	Nhận xét
9600 bps	Tốt	~0.1%	Ôn định, phổ biến
14400 bps	Rất tốt	0%	<b>Chính xác nhất</b>
19200 bps	Tốt	~0.1%	Ôn định
115200 bps	Tốt	~0.5%	Hoạt động được, có sai số nhẹ
≥ 128000 bps	Không ổn định	>1% → 13%	Dễ lỗi, mất dữ liệu

#### 1.2/ Hiển thị dữ liệu:

- Dữ liệu chính xác cả chiều truyền và nhận.  
- Hiển thị dữ liệu chính xác và rõ ràng qua LED trên board DE2.

- Dữ liệu truyền đi: hiển thị trên **HEX[0], HEX[1]**.
- Dữ liệu nhận về: hiển thị trên **HEX[2], HEX[3]**.
- Trạng thái bit: thể hiện bằng **LEDR[0] – LEDR[7]**.

#### 1.3/ Phản hồi điều khiển và quản lý trạng thái truyền/nhận:

- Đảm bảo không bị mất đồng bộ hoặc lỗi khi truyền/nhận dữ liệu, mọi trạng thái đều phải hoạt động chính xác.

- Nút nhấn **KEY[0]** → truyền 1 byte dữ liệu.
- Nút **KEY[3]** → nạp tốc độ truyền mới.
- Phản ứng gần như tức thời.

#### 1.4/ Kiểm tra tính ổn định:

- Kiểm tra liên tục các quá trình truyền nhận trên board DE2 trong thời gian dài mà không xảy ra lỗi (ví dụ: mất bit, trễ bit).
- Kiểm tra các tín hiệu như Start bit, Stop bit, và dữ liệu song song từ FPGA trong quá trình giao tiếp UART.

## **2. HƯỚNG PHÁT TRIỂN (MỞ RỘNG DỰ ÁN):**

Mục tiêu nâng cấp	Nội dung triển khai cụ thể
Kiểm tra lỗi dữ liệu	Thêm parity bit hoặc tích hợp CRC để phát hiện lỗi khi truyền
Bộ đệm truyền/nhận	Thiết kế FIFO buffer giúp lưu trữ tạm thời dữ liệu
Giao tiếp đa byte	Cho phép truyền và nhận cả chuỗi dữ liệu dài
Thiết kế giao diện người dùng (PC)	Viết ứng dụng GUI hiển thị dữ liệu dạng bảng hoặc đồ thị
Tích hợp cảm biến	Kết nối cảm biến nhiệt độ, độ ẩm,... và truyền kết quả về PC
Tăng tốc độ truyền	Dùng thạch anh cao hơn, cải thiện khối tạo baud rate
Hỗ trợ nhiều giao thức	Mở rộng sang SPI, I2C, Bluetooth, USB,...

## **3. KẾT LUẬN**

- Thiết kế bộ truyền – nhận dữ liệu UART trên FPGA là một mô hình hiệu quả, trực quan và thực tế giúp sinh viên học tốt kỹ thuật số và lập trình phần cứng.
- Giao tiếp UART được thực hiện đúng chuẩn, dữ liệu truyền nhận chính xác tại các tốc độ phổ biến như 9600–115200 bps.
- Với cách thiết kế có tổ chức, dễ mở rộng, mô hình này không chỉ phù hợp cho mục đích học tập mà còn có thể áp dụng vào các hệ thống điều khiển thực tế trong tương lai.
- Nếu được phát triển thêm các chức năng kiểm lỗi, giao tiếp đa byte và cảm biến, thiết kế này hoàn toàn có thể trở thành nền tảng cho các dự án nhúng quy mô nhỏ và vừa.