

# BÀI 6: GIAO TIẾP UART – RS232

## 1. MỤC ĐÍCH

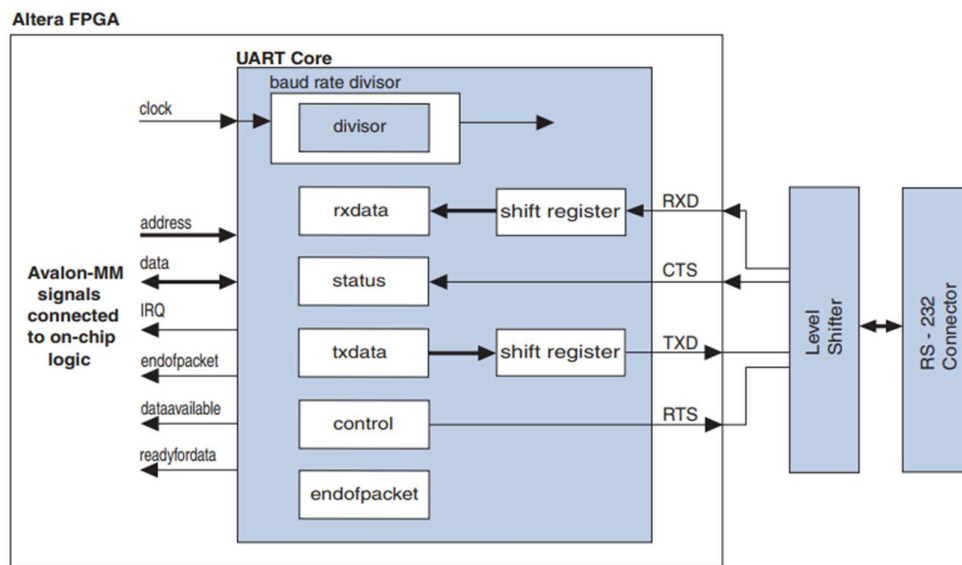
Thông qua bài thực hành này, sinh viên sẽ hiểu rõ:

- Thực hiện truyền nhận data thông qua SerialPort – RS232

## 2. TỔNG QUAN

Sơ đồ khối UART, với các Input/Output như sau:

**Figure 6–1.** Block Diagram of the UART Core in a Typical System



Trong đó, chúng ta sẽ truyền Data đi thông qua chân **Output TXD**, và nhận data tại chân **Input RXD**.

Viết code software sử dụng lõi IP Altera cung cấp sẵn sử dụng thư viện:

- altera\_avalon\_uart.h
- altera\_avalon\_uart\_regs.h
- altera\_avalon\_uart.c

Hàm đọc data: **IORD\_ALTERA\_AVALON\_UART\_RXDATA(UART\_BASE)**

Hàm ghi data: **IOWR\_ALTERA\_AVALON\_UART\_TXDATA(UART\_BASE, character)**

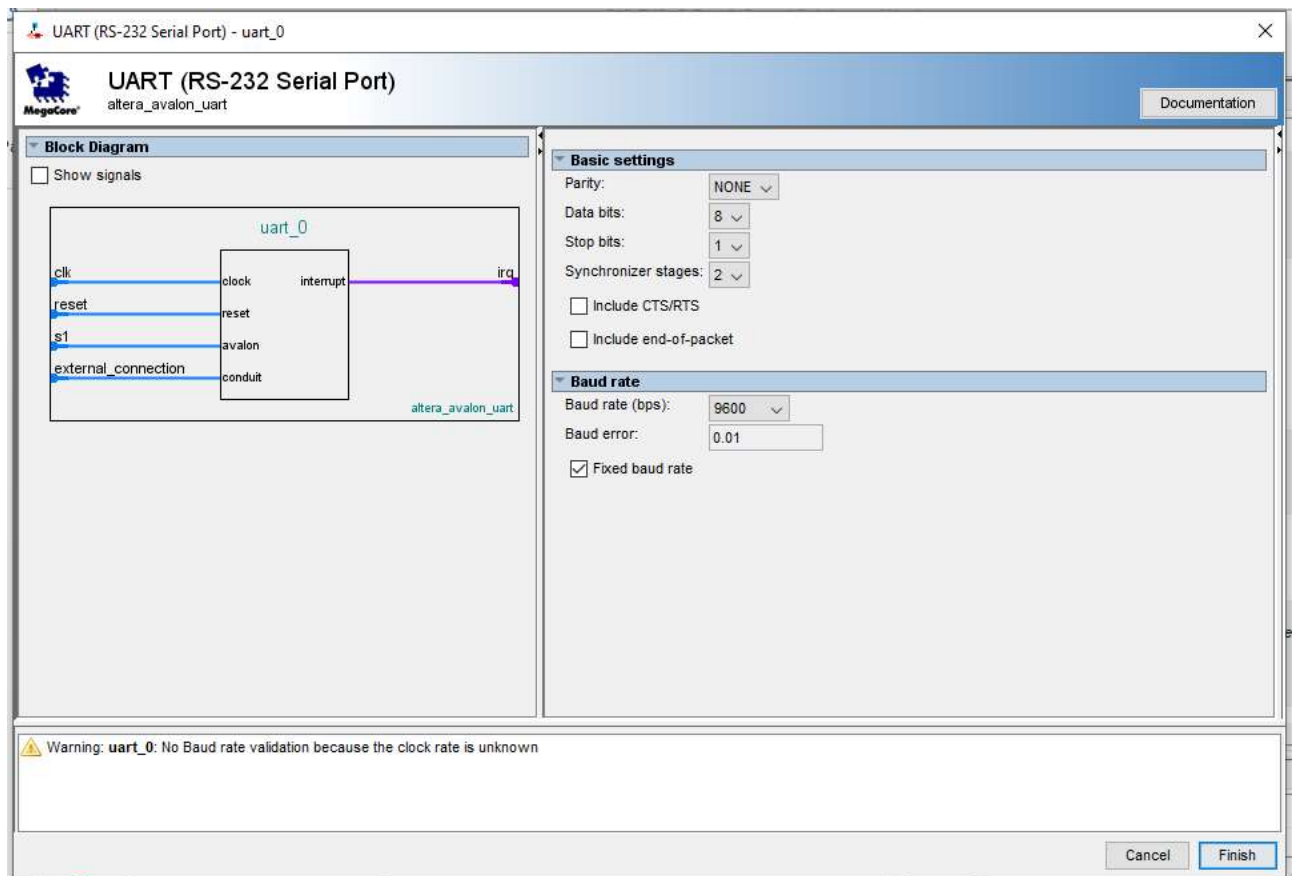
# BÀI TẬP

**Bài 1:** Thực hiện bộ đếm từ 00 – 99 thông qua UART. Đặt tên project là **lab6\_part1**.

Qsys:

- *On-chip memory – Ram*
- *Chip Nios II*
- *Clock 50*
- *JTAG UART*
- *UART*

Add thêm UART như sau, cài đặt Baud Rate=9600:



Qsys - qsys\_nios.qsys (D:\SoC\Example\com\_RS232\qsys\_nios.qsys)

File Edit System View Tools Help

Component Library

uart

Project

Library

Interface Protocols

Serial

Altera 16550 Comp

JTAG UART

UART (RS-232 Ser

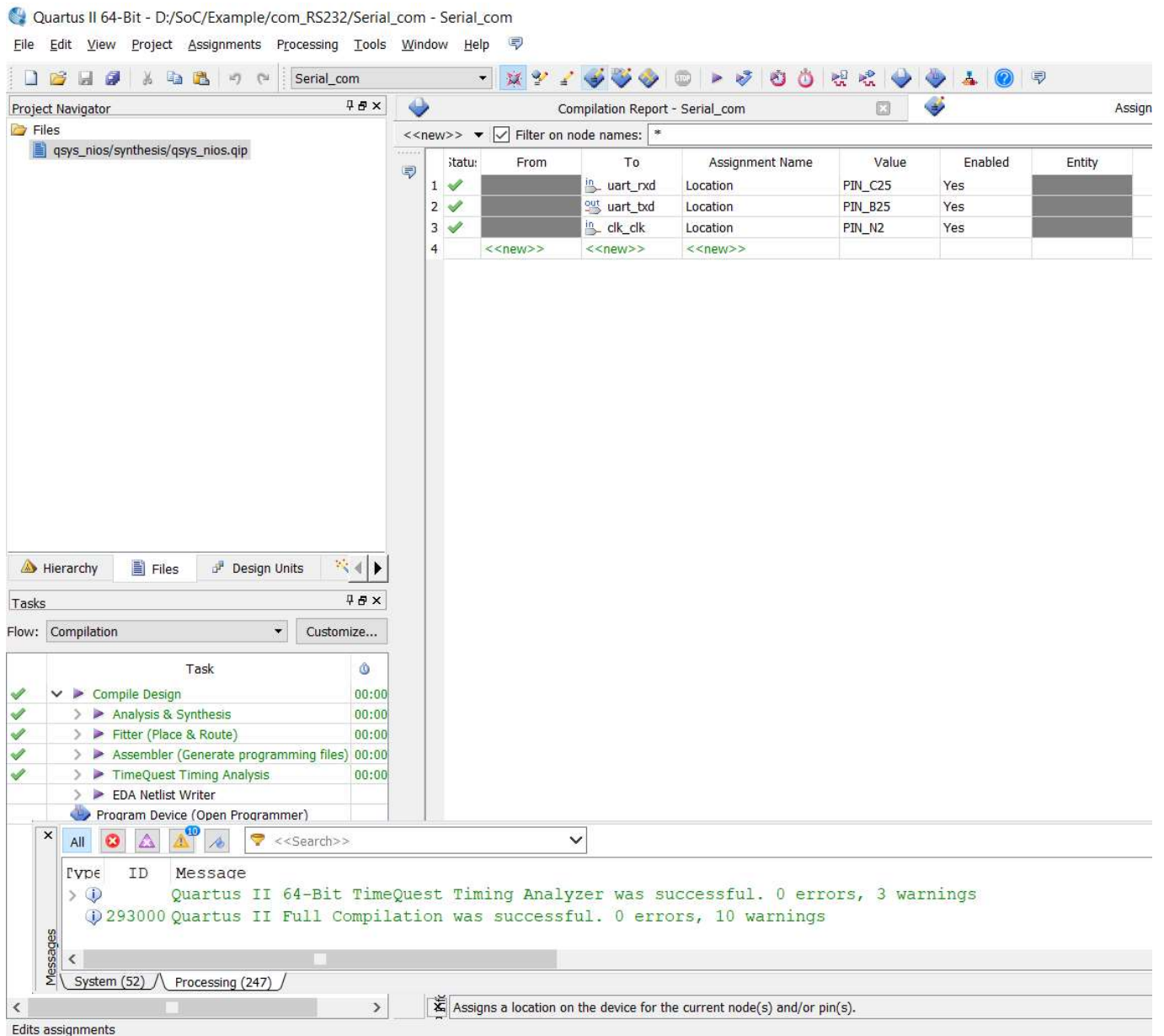
New... Edit... Add...

System Contents

Connections	Name	Description	Export	Clock	Base	End	IRQ	Opcode Name
	clk_0	Clock Source	clk	clk_0				
	clk_in	Clock Input	Double-click to export					
	clk_in_reset	Reset Input	Double-click to export					
	clk	Clock Output	Double-click to export					
	clk_reset	Reset Output	Double-click to export					
	onchip_memory2_0	On-Chip Memory (RAM or ROM)						
	clk1	Clock Input	Double-click to export	clk_0				
	s1	Avalon Memory Mapped Slave	Double-click to export	[clk1]	0x1000	0x1fff		
	reset1	Reset Input	Double-click to export	[clk1]				
	nios2_qsys_0	Nios II Processor						
	clk	Clock Input	Double-click to export	clk_0				
	reset_n	Reset Input	Double-click to export	[clk]				
	data_master	Avalon Memory Mapped Master	Double-click to export	[clk]			IRQ 0	
	instruction_master	Avalon Memory Mapped Master	Double-click to export	[clk]				
	jtag_debug_module_re...	Reset Output	Double-click to export	[clk]				
	jtag_debug_module	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x2800	0x2fff		
custom_instruction_m...	Custom Instruction Master	Double-click to export						
	jtag_uart_0	JTAG UART						
	clk	Clock Input	Double-click to export	clk_0				
	reset	Reset Input	Double-click to export	[clk]				
	avalon_fag_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x3030	0x3037		
	UART	UART (RS-232 Serial Port)						
	clk	Clock Input	Double-click to export	clk_0				
	reset	Reset Input	Double-click to export	[clk]				
	s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x3000	0x301f		
	external_connection	Conduit	uart					

Messages

Description	Path
0 Errors, 0 Warnings	



Hình . Gán chân cho UART.

Code C tham khảo:

```
#include "sys/alt_stdio.h"
#include <stdio.h>
#include "system.h"
#include "altera_avalon_uart_regs.h"
#include <stdint.h>

void delay(int a)
{
    volatile int i=0;
    while(i<a*10000)
    {
        i++;
    }
}

int main()
{
```

```

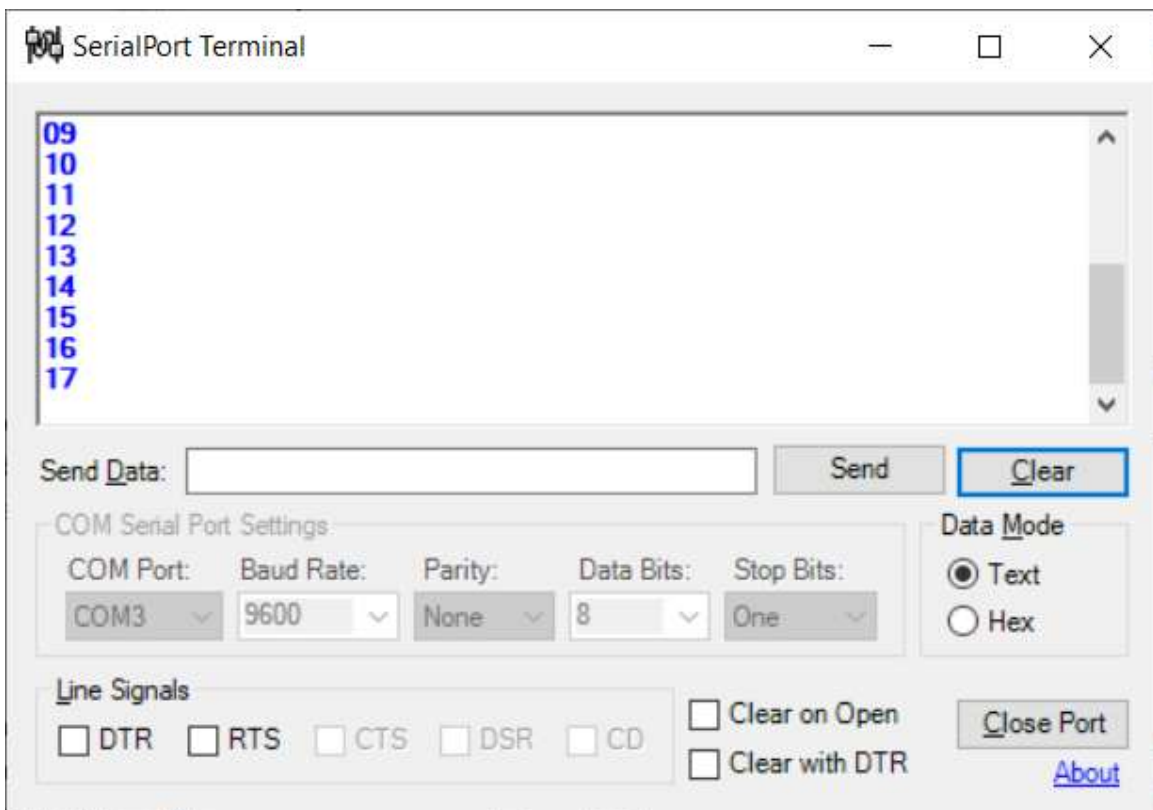
    int i,j;

    //char num[10]={0x30,0x31,0x32,0x33,0x34,0x35,0x36,0x37,0x38,0x39};
    char num[10]='0','1','2','3','4','5','6','7','8','9';
    while (1)
    {
        for(i=0;i<100;i++)
        {
            IOWR_ALTERA_AVALON_UART_TXDATA(UART_BASE, num[i/10]);
            IOWR_ALTERA_AVALON_UART_TXDATA(UART_BASE, num[i%10]);
            delay(100);
            IOWR_ALTERA_AVALON_UART_TXDATA(UART_BASE, '\n');
            delay(100);
        }
    }

    return 0;
}

```

Sau đó, Build Code và Run trên Nios. Đồng thời kết nối cáp **USBtoRS232** và sử dụng phần mềm **SerialPort Terminal** để Test. Kết quả như sau:



**Bài 2:** 1 Button và UART và JTAG UART, khi nhấn Button thì đếm số lần nhấn nút từ 00 – 99 và truyền số này qua cả 2: UART và JTAG UART. Đặt tên project là **lab6\_part2**.

**Bài 3:** Thực hiện các lệnh điều khiển như sau. Đặt tên project là **lab6\_part3**.

- Khi nhận ký tự ‘S’ thì Nios truyền về String “Down”;
- Khi nhận ký tự ‘W’ thì Nios truyền về String “Up”;

- Khi nhận ký tự ‘**A**’ thì Nios truyền về String “**Left**”;
- Khi nhận ký tự ‘**D**’ thì Nios truyền về String “**Right**”;

Gợi ý: Ta cần khai báo biến kiểu Char để nhận data, ví dụ như sau:

```
Char rxData;  
rxData = IORD_ALTERA_AVALON_UART_RXDATA(UART_0_BASE);
```