

串口收发器实验报告

5 Jun 2014

目录

实验十：串口收发器设计.....	2
实验目的：	2
实验原理：	2
关键代码：	2
文件清单.....	7
仿真结果.....	7
综合情况.....	8
硬件调试情况.....	8

实验十：串口收发器设计

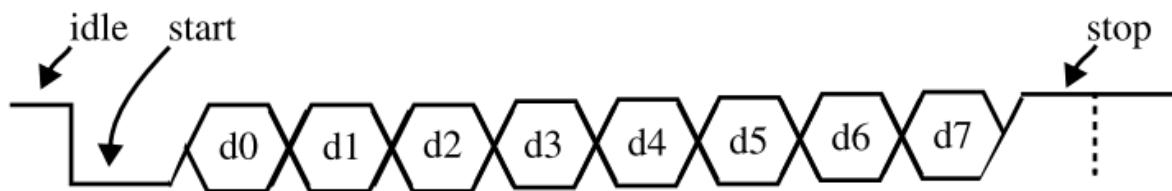
实验目的：

了解和掌握 UART 的工作原理

实验原理：

1. 串口基本原理：

UART（Universal Asynchronous Receiver/Transmitter）是一种通用串行数据总线，用于异步通信。该总线双向通信，可以实现全双工传输和接收。在嵌入式设计中，UART 用来与 PC 进行通信，包括与监控调试器和其它器件。与 UART 相关的一个概念是 RS232-C 标准，该标准由美国电子工业协会 EIA(Electronic Industry Association)制定的一种串行物理接口标准，其规定了若干标准的数据速率，并且采用较高电平来保证 20 米以内的有线传输。UART 是计算机与嵌入式系统中串行通信端口的关键部分，速率有规定的 9600 等波特率。在实际应用中，通用串口的电气特性兼容 RS232 规范信号，即逻辑“1”信号相对于地为-3 到-15 伏，而逻辑“0”相对于地为 3 到 15 伏。因此，当一个微控制器的 UART 与外界电路相连时，需要采用一个符合 RS232 标准的驱动器来将控制器管脚的 CMOS 电平或 TTL 电平转换为 RS232 标准电平。TTL 电平是 3.3V 的，而 RS232 是负逻辑电平，如果没有类似 MAX232 的驱动芯片进行电平转换，这么高的电压很可能会把芯片烧坏。



关键代码：

```
module Rx(DataIn,Data,reset,clk,DataEn);  
    input wire DataIn,reset,clk;  
    output reg [7:0]Data;  
    output reg DataEn;  
    reg [4:0]sample;
```

```
reg [2:0]count;
reg status;

always@(posedge clk,negedge reset)
begin
    if(reset==0)
    begin
        sample<=0;
        Data<=0;
        status<=0;
        DataEn<=0;
        count<=0;
    end
    else if(status==0)
    begin
        if(sample!=0)
            sample <= sample - 1;
        else
            begin
                DataEn<=0;
                if(DataIn==0)
                begin
                    status<= 1;
                    sample<= 23;
                end
            end
        end
    end
    else
```

```
begin
    if(sample==0)
        begin
            sample<= 15;
            Data[7:0]<={DataIn,Data[7:1]};
            count<=count+1;
            if(count==7)
                begin
                    status<= 0;
                    DataEn<= 1;
                end
            end
        end
    else
        sample<=sample-1;
    end
end
endmodule
```

```
module Tx(DataIn,DataInEn,DataOut,data,reset,clk);
```

```
    input wire [7:0]DataIn;
```

```
    input wire DataInEn,reset,clk;
```

```
    output reg DataOut;
```

```
    output reg [7:0]data;
```

```
    reg [3:0]count1,count2;
```

```
    reg outready;
```

```
    always@(negedge reset or posedge clk)
```

```
    begin
```

```
        if(reset==0)
```

```
begin
    data<=0;
    count1<=0;
    count2<=0;
    DataOut<=1;
    outready<=0;
end
else if(DataInEn)
begin
    data<=DataIn;
    outready<=1;
    count1<=0;
    count2<=0;
end
else if(outready)
begin
    count2<=count2-1;
    if(count2==0)
begin
    if(count1==0)
        DataOut<=0;
    else if(count1==9)
begin
        DataOut<=1;
        outready<=0;
    end
    else
        DataOut<=data[count1-1];
```

```
        count1<=count1+1;
    end
end
end
endmodule

module control(DataIn,clk,reset,Data,DataInEn,DataEn);
    input [7:0]DataIn;
    input clk,reset,DataInEn;
    output reg DataEn;
    output reg [7:0]Data;
    reg [7:0]data;
    reg [3:0]count;
    reg outready;

    always@(posedge clk,negedge reset)
    begin
        if(reset==0)
        begin
            count<=15;
            Data<=0;
            DataEn<=0;
            outready<=0;
        end

        else if(DataInEn)
        begin
            outready<=1;
            data<=DataIn;
        end
    end
end
```

```
        end
        else if(outready)
        begin
            count<=count-1;
            if(count==0)
            begin
                DataEn<=0;
                outready<=0;
            end
        else
        begin
            DataEn<=1;
            if(data[7]==1)
                Data<=data^8'b11111111;
            else
                Data<=data;
            end
        end
    end
end
endmodule
```

文件清单

Control.v	控制模块
Rx.v	发送模块
Tx.v	接收模块

仿真结果

由于串口发送和接收对时钟要求较高，本实验的调试使用逻辑分析仪查看波形，未进行仿真。

综合情况

频率计:	Total logic elements	91
	Total registers	65
	Worst-case	13.765 ns

硬件调试情况

串口线会导致 win8.1 蓝屏，故使用 pl2303USB-ttl 串口模块连接 GPIO 调试。