

[文章编号] 1671-802X(2008)02-0089-02

基于FPGA/CPLD的UART功能设计

贺良飞¹, 刘从跃²

(1.中国电子科技集团公司第四十一研究所, 安徽 蚌埠 233000;

2.蚌埠市邮政局设备维护科, 安徽 蚌埠 233000)

[摘要]采用通用集成电路实现UART的功能无法完全满足各种场合下的需要,将所需的UART功能集成到FPGA/CPLD内部,可以使整个设计更加紧凑、稳定且可靠。本文应用EDA技术,基于FPGA/CPLD器件设计与实现UART。

[关键词] FPGA/CPLD; UART; VHDL; Modelsim

[中图分类号] TN919

[文献标识码] B

1 引言

UART(即Universal Asynchronous Receiver Transmitter通用异步收发器)是广泛使用的串行数据传输协议。UART允许在串行链路上进行全双工通信。串行外设都会用到RS232-C异步串行接口协议,传统上采用通用的UART集成电路来实现,如TI、EXAR、EPIC公司的550、452等系列UART集成电路,但是我们一般在特定的电路设计中不需要使用完整的UART功能,而且对于多串口设备或需要加密通讯的场合,使用通用的UART集成电路也不是最合适的。如果设计上用到了FPGA/CPLD器件,那么我们就可以将所需要的UART功能集成到FPGA/CPLD内部,从而使整个设计更加紧凑、稳定且可靠。因此本文应用EDA技术,基于FPGA/CPLD器件设计与实现UART。

2 UART简介

2.1 UART基本结构

UART主要由数据总线接口、控制逻辑、波特率发生器、发送部分和接收部分组成。功能包括微处理器接口,发送缓冲器(THR)、发送移位寄存器(TSR)、帧产生、奇偶校验、并转串电路、数据接收缓冲器(RBR)、接收移位寄存器(RSR)、帧产生、奇偶校验、串转并电路等。UART电路的典型应用结构框图如图1所示。

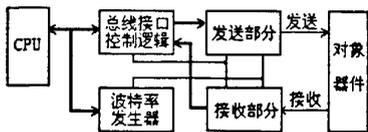


图1 UART电路典型应用框图

2.2 UART的帧格式

UART帧格式框图如图2所示。

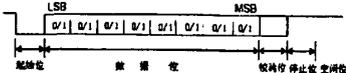


图2 UART帧格式框图

UART帧格式中包括收发空闲状态(idle),起始位(start-bit),数据位(5~8bit,可选),奇偶校验位(parity),停止位(stop

bit,位数可为1、1.5、2位)。UART内部的配置寄存器可以配置数据位数(5~8 bit)、是否有校验位和校验的类型、停止位的位数(1,1.5,2)等设置。

3 UART功能的设计实现

3.1 波特率发生器

UART的接收和发送是按照相同的波特率进行收发(当然也可以实现成对的不同波特率进行收发),波特率是可以由CPU的总线接口设置的。UART收发的每一个数据宽度都是波特率发生器输出时钟周期的16倍,即假定当前按照7200bps进行收发,那么波特率发生器输出的时钟频率应为7200×16Hz,当然这也是可以改变的,我们只是按照UART的方法进行设计。

假定我们提供的时钟为1.8432MHz,那么可以很简单地用CPU写入不同的数值到波特率保持寄存器,然后用计数器的方式生成所需要的各种波特率,这个值的计算原则就是1843200/(16×所期望的波特率),如果希望输出7200Hz的波特率,那么这个值就是1843200/(16×7200)=16。也就是说只需要将1.8432MHz时钟信号16分频便可得到期望的7200bps波特率时钟,在电路实现上将非常的方便。

3.2 UART发送器

发送器端口信号如图3所示。

数据的发送是由CPU控制,CPU给出写信号,发送器根据此信号将并行数据TxD_data[7..0]锁存进发送缓冲器tbl[7..0],并通过发送移位寄存器tsr[7..0]发送串行数据至串行数据输出端TxD。在数据发送过程中用输出信号busy作为标志信号,当一帧数据发送完毕时,busy信号为1,通知CPU在下一个时钟装入新数据。

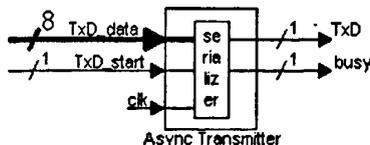


图3 UART发送器端口信号框图

根据上面的分析,实现UART发送功能的部分VHDL程序如下:

*[收稿日期] 2008-02-29

[作者简介] 贺良飞(1982.02-),河南淮滨人,助理工程师,中国电子科技集团公司第41研究所,研究方向:测量仪器。

刘从跃(1984.04-),安徽凤阳人,助理工程师,蚌埠市邮政局设备维护科,研究方向:自动控制。

```

if rst = '1' then
    txd <= '1';
    busy <= '1';
    tsr <= "00000000";
    paritybit <= '0';
elsif clk1x'event and clk1x = '1' then
    if clk1x_enable='1' then
        if std_logic_vector(cnt_t) = "0001" then
            tsr <= tbr;
            busy <= '0';
        elsif std_logic_vector(cnt_t) = "0010" then
            txd <= '0';
            elsif std_logic_vector(cnt_t) >= "0011" and
std_logic_vector(cnt_t) <= "1010" then
                tsr <= '0'&tsr(7 downto 1);
                txd <= tsr(0);
                paritybit <= paritybit xor tsr(0);
            elsif std_logic_vector(cnt_t) = "1011" then
                txd <= not paritybit;
            elsif std_logic_vector(cnt_t) = "1100" then
                paritybit <= '0';
                txd <= '1';
                busy <= '1';
            end if;
        end if;
    end if;
end if;

```

使用Modelsim仿真波形如图4所示:

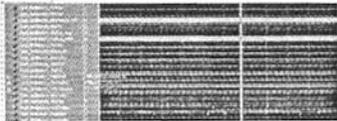


图4 UART发送器仿真波形框图

3.3 UART接收器

接收器端口信号如图5所示。

我们的设计目标是:1)当RxD线上有数据时,接收模块负责识别RxD线上的数据。2)当收到一个字节的数时,锁存接收到的数据到"data"总线,并使"data_ready"有效一个周期。3)串行数据帧和接收时钟是异步的,发送来的数据由逻辑1变为逻辑0可以视为一个数据帧的开始。接收器先要捕捉起始位,确定RxD输入由1到0,逻辑0要8个CLK16时钟周期,才是正常的起始位,然后在每隔16个CLK16时钟周期采样接收数据,移位输入接收移位寄存器rsr,最后输出数据data。还要输出一个数据接收标志信号data_ready标志数据接收完毕。

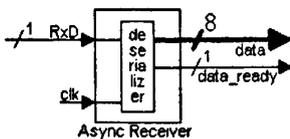


图5 UART接收器端口信号框图

根据上面的分析,实现UART接收功能的部分VHDL程序如下:

```

if rst = '1' then
    rsr <= "00000000";
    rbr <= "00000000";
    paritybit <= '1';
    framing_error1 <= '0';
    parity_error1 <= '0';
elsif clk1x'event and clk1x = '0' then
    if cnt_r >= 1 and cnt_r <= 8 then
        rsr(7) <= rxd2;
        rsr(6 downto 0) <= rsr(7 downto 1);
        paritybit <= paritybit xor rxd2;
        framing_error1 <= '0';
        parity_error1 <= '0';
    elsif (cnt_r) = 9 then
        rbr <= rsr;
        paritybit <= paritybit xor rxd2;
        elsif (cnt_r) = 10 then
            if (paritybit = '1') then
                parity_error1 <= '1';
            else
                parity_error1 <= '0';
            end if;
            if rxd2='0' then
                framing_error1 <= '1';
            else
                framing_error1 <= '0';
            end if;
            paritybit <= '1';
        end if;
    end if;
end if;

```

使用Modelsim仿真波形如图6所示:

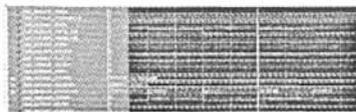


图6 UART接收器仿真波形框图

4 结论

通过波特率发生器、发送器和接收器模块的设计与仿真,能够较容易地实现通用异步收发器设计,对于收发数据帧和发生的波特率时钟频率能较灵活地改变,而且硬件实现不需要很多资源,尤其能较灵活地嵌入到FPGA/CPLD的开发中。在EDA技术平台上进行设计、仿真与实现具有较好的优越性。

[参考文献]

[1]潘名莲,马争等.微型计算机原理[M].电子工业出版社,1993.
 [2]侯波亨,顾新.VHDL硬件描述语言与数字逻辑电路设计[M].西安电子科技大学出版社,1997.
 [3]王诚等.FPGA/CPLD设计工具-Xilinx ise5.x 使用详解[M].人民邮电出版社,2003.