

数字逻辑与处理器基础 第四次作业

串口收发器和存储器的使用

实验目的：了解和掌握 UART 的工作原理，进一步熟悉仿真验证方法，为后续设计做准备

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。UART是计算机与嵌入式系统中最常用的串行通信协议。

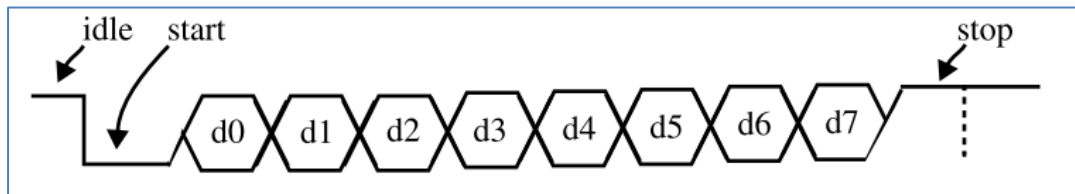


图 1 串口时序示意图

图1表明在异步传送中串行发送一个数据字节的位定时关系(图中没有包括奇偶校验位)。发送一个完整的字节信息，首先是一个作为起始位的逻辑“0”位，接着是8个数据位，然后是1个、1+1/2个或2个停止位逻辑“1”位，数据线空闲时呈现为高或“1”状态。在字符的8位数据部分，先发送数据的最低位(LSB)，最后发送最高位(MSB)。每位持续的时间是固定的，由发送器本地时钟控制，每秒发送的数据位个数，即为“波特率”。对于9600的波特率，每一位持续的时间为1/9600秒。

起始位和停止位起着很重要的作用。显然，他们标志每个字符的开始和结束，但更重要的是他们使接收器能把局部时钟与每个新开始接收的字符再同步。异步通信没有可参照的时钟信号，发送器随时都可能发送数据，需要从任何边沿的出现时刻开始正确地采样紧接着的10~11位(包括开始位、数据位和停止位)。接收器的时钟与发送器的时钟不是同一个，因此，接收器采样点的间隔跟由发送器时钟所确定的位间隔时间不同，接收器设计不好可能会导致采样错误。

目前的个人计算机在硬件上很少有专用的UART(RS232)接口了，其UART功能已经融合在USB接口中。开发板采用了FT2232芯片，将UART转换加载到USB传输接口上，其管脚约束如表1所示。

电路端口	FPGA 管脚
UART_RX	B18
UART_TX	A18

表 1 Basys3 UART 管脚约束

2. 实验设计原理:

串口收发器包括发送器和接收器两个模块。首先，通过串口接收器模块从外部接收数据，并将接收到的数据送给控制器模块，同时控制器模块根据接收的串口数据产生发送数据，并通过串口发送器模块将数据发送到外部。

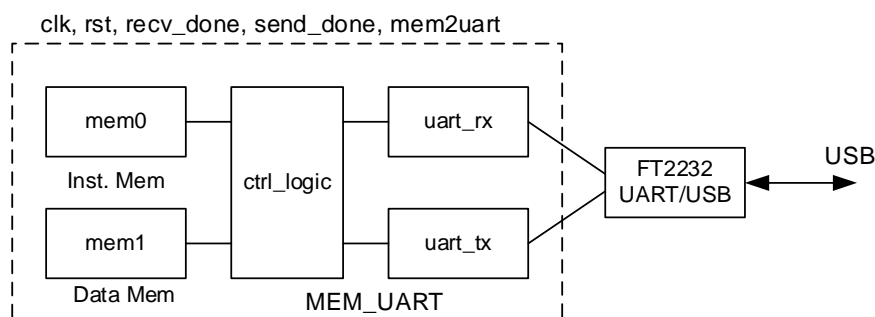
(1) 实现的功能

串口→存储器

从串口读取512*32*2bit的数据到FPGA中的指令存储器和数据存储器，其中指令存储空间为512*32bit，数据存储空间为512*32bit。数据接收完成，拉高recv_done信号。

存储器→串口:

当mem2uart为高电平时，FPGA从数据存储器中读取数据，并通过串口发送到上位机。数据发送完成，拉高send_done信号。



硬件接口及配置性参数

```
module UART_MEM(  
    input clk,          // 100MHz  
    input rst,          // BTNU  
    input mem2uart,     // SW0  
    /*-----MEM-----*/  
    output reg recv_done, // led 0  
    output reg send_done, // led 1  
    /*-----UART-----*/  
    input Rx_Serial,  
    output Tx_Serial  
);  
  
parameter CLKS_PER_BIT = 16'd10417; // 100M/9600  
parameter MEM_SIZE = 512;
```

上图所示，系统时钟clk为100MHz(W5)，复位按键rst为BTNU(T18)，接收完成信号绑定到LED0(U16)。将发送存储器中数据的控制信号mem2uart为SW0(V17)，发送完成信号绑定到LED1(E19)。

参数CLKS_PER_BIT可以配置串口的波特率：CLKS_PER_BIT = 系统时钟/波特率。

参数MEM_SIZE可以配置指令存储器和数据存储器的大小：MEM_SIZE*32bit。

本程序打通了串口到存储器以及存储器到串口的数据通路，更加具体的功能还需要在此基础上做进一步修改。

文件列表

- 发送：保存的串口发送的测试数据，大小为 $512 \times 32 \times 2 \text{bit}$ ，前半部分是指令，后半部分是数据
- 理论接收：保存的是串口接收的理论数据，大小为 $512 \times 32 \text{bit}$ ，内容与文件“发送”的后半部分相同
- src：源文件

实验内容：

- (1) 阅读 Verilog 代码，理解电路实现原理。
- (2) 在 Vivado 中，对代码通过行为级仿真和门级仿真进行验证。
- (3) 选做：添加一个控制信号，当该信号有效时，将数据存储器中的内容取出，逐字取反码，并存回数据存储器中。通过门级仿真对此设计进行验证。