

清华大学电子工程系

数字逻辑与处理器基础 实验

DE2 实验版 V8

课程教学组

2016/3/21

目录

实验安排	2
注意事项	2
DE2 实验板外围资源的 FPGA 引脚分配	3
演示实验：数码管七段译码器	6
实验一 计数器	8
实验二 序列检测器设计	10
实验三 频率计设计	12
实验四 串口收发器设计	16

实验安排

分数根据现场完成情况和实验报告综合评定，思考题完成比较好可以酌情加分。实验具体安排见下表：

序号	实验名称	实验学时	备注
1	演示实验、实验 1：计数器	6	提交计数器实验报告，第 7、8 周完成
2	实验 2：序列检测器	6	单独提交实验报告，现场验收截至第 10 周
3	实验 3：频率计	6	单独提交实验报告，现场验收截至第 13 周
4	实验 4：异步串口收发器	6	单独提交实验报告，验收截至第 15 周

注意事项

1. 每人一套实验器材，领用和归还实验器材需登记。
2. 实验共四组，实验按顺序进行，每组实验完成后才能开始下一组实验。
3. 每个实验硬件调试通过后，务必请助教进行现场验收；验收必须在规定时间点以前现场进行，每人每次实验最多 2 次验收机会。
4. 现场验收与实验报告各占该实验成绩的 50%，实验报告请在每次实验现场验收截止时间后一周内提交。
5. 实验时间：根据二级选课严格按照选课时间进行，**每组实验没有完成现场验收前必须按时到实验室进行实验**，完成现场验收后，该实验后续时间可以根据个人情况自由安排。
6. FPGA 管脚分配一定要按照实验要求进行，否则**可能造成短路，烧毁芯片**。
7. 注意使用 U 盘或其他方法保存备份设计文件和工程。
8. 实验报告内容包括：实验目的；设计方案（原理说明及框图）；关键代码及文件清单；仿真结果及分析；综合情况（面积和时序性能）；硬件调试时遇到的问题和解决情况。
9. 提交方式：实验报告（word 或者 pdf）和设计代码打包后提交到网络学堂，提交打包文件名按照“学号_姓名_实验编号”的规则命名。

有抄袭嫌疑（实验报告或者设计代码出现雷同）的实验报告（不管是抄袭者还是被抄袭者）都一律按零分处理。

DE2 实验板外围资源的 FPGA 引脚分配

- 拨码开关

Signal Name	FPGA Pin No.	Description
SW[0]	PIN_N25	Toggle Switch[0]
SW[1]	PIN_N26	Toggle Switch[1]
SW[2]	PIN_P25	Toggle Switch[2]
SW[3]	PIN_AE14	Toggle Switch[3]
SW[4]	PIN_AF14	Toggle Switch[4]
SW[5]	PIN_AD13	Toggle Switch[5]
SW[6]	PIN_AC13	Toggle Switch[6]
SW[7]	PIN_C13	Toggle Switch[7]

- 按键开关

Signal Name	FPGA Pin No.	Description
KEY[0]	PIN_G26	Pushbutton[0]
KEY[1]	PIN_N23	Pushbutton[1]
KEY[2]	PIN_P23	Pushbutton[2]
KEY[3]	PIN_W26	Pushbutton[3]

- LED

Signal Name	FPGA Pin No.	Description
LEDR[0]	PIN_AE23	LED Red[0]
LEDR[1]	PIN_AF23	LED Red[1]
LEDR[2]	PIN_AB21	LED Red[2]
LEDR[3]	PIN_AC22	LED Red[3]
LEDR[4]	PIN_AD22	LED Red[4]
LEDR[5]	PIN_AD23	LED Red[5]
LEDR[6]	PIN_AD21	LED Red[6]
LEDR[7]	PIN_AC21	LED Red[7]

● 7 段数码管

Signal Name	FPGA Pin No.	Description
HEX0[0]	PIN_AF10	Seven Segment Digit 0[0]
HEX0[1]	PIN_AB12	Seven Segment Digit 0[1]
HEX0[2]	PIN_AC12	Seven Segment Digit 0[2]
HEX0[3]	PIN_AD11	Seven Segment Digit 0[3]
HEX0[4]	PIN_AE11	Seven Segment Digit 0[4]
HEX0[5]	PIN_V14	Seven Segment Digit 0[5]
HEX0[6]	PIN_V13	Seven Segment Digit 0[6]

HEX1[0]	PIN_V20	Seven Segment Digit 1[0]
HEX1[1]	PIN_V21	Seven Segment Digit 1[1]
HEX1[2]	PIN_W21	Seven Segment Digit 1[2]
HEX1[3]	PIN_Y22	Seven Segment Digit 1[3]
HEX1[4]	PIN_AA24	Seven Segment Digit 1[4]
HEX1[5]	PIN_AA23	Seven Segment Digit 1[5]
HEX1[6]	PIN_AB24	Seven Segment Digit 1[6]

HEX2[0]	PIN_AB23	Seven Segment Digit 2[0]
HEX2[1]	PIN_V22	Seven Segment Digit 2[1]
HEX2[2]	PIN_AC25	Seven Segment Digit 2[2]
HEX2[3]	PIN_AC26	Seven Segment Digit 2[3]
HEX2[4]	PIN_AB26	Seven Segment Digit 2[4]
HEX2[5]	PIN_AB25	Seven Segment Digit 2[5]
HEX2[6]	PIN_Y24	Seven Segment Digit 2[6]

HEX3[0]	PIN_Y23	Seven Segment Digit 3[0]
HEX3[1]	PIN_AA25	Seven Segment Digit 3[1]
HEX3[2]	PIN_AA26	Seven Segment Digit 3[2]
HEX3[3]	PIN_Y26	Seven Segment Digit 3[3]
HEX3[4]	PIN_Y25	Seven Segment Digit 3[4]
HEX3[5]	PIN_U22	Seven Segment Digit 3[5]
HEX3[6]	PIN_W24	Seven Segment Digit 3[6]

HEX4[0]	PIN_U9	Seven Segment Digit 4[0]
HEX4[1]	PIN_U1	Seven Segment Digit 4[1]
HEX4[2]	PIN_U2	Seven Segment Digit 4[2]
HEX4[3]	PIN_T4	Seven Segment Digit 4[3]
HEX4[4]	PIN_R7	Seven Segment Digit 4[4]
HEX4[5]	PIN_R6	Seven Segment Digit 4[5]
HEX4[6]	PIN_T3	Seven Segment Digit 4[6]

HEX5[0]	PIN_T2	Seven Segment Digit 5[0]
HEX5[1]	PIN_P6	Seven Segment Digit 5[1]
HEX5[2]	PIN_P7	Seven Segment Digit 5[2]
HEX5[3]	PIN_T9	Seven Segment Digit 5[3]
HEX5[4]	PIN_R5	Seven Segment Digit 5[4]
HEX5[5]	PIN_R4	Seven Segment Digit 5[5]
HEX5[6]	PIN_R3	Seven Segment Digit 5[6]

- 时钟输入

Signal Name	FPGA Pin No.	Description
CLOCK_50	PIN_N2	50 MHz clock input

- RS232 串口

Signal Name	FPGA Pin No.	Description
UART_RXD	PIN_C25	UART Receiver
UART_TXD	PIN_B25	UART Transmitter

演示实验：数码管七段译码器

7 个 LED 管排列成图 1 所示的模式，这样就形成了不同的位。DE2 开发板上有 8 个共阳极的 7 段数码管（实验使用其中的 4 个）。在共阳极的情况下，**FPGA 引脚输出 0 点亮 LED，输出 1 关掉 LED**。表 1 所示的真值表给出了显示 0~F 的十六进制数所需的 a~g 的输出阴极值。

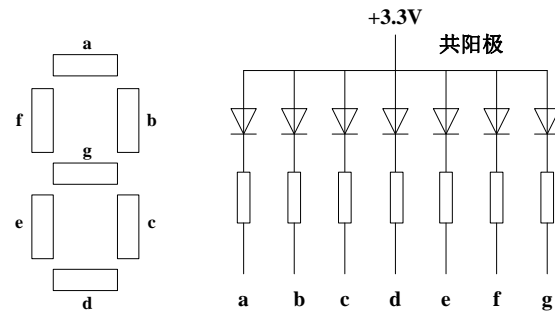


图 1 包含 7 个发光二极管（LED）的 7 段显示管

表 1 显示十六进制数 0~F 的 7 段代码

x	a	b	c	d	e	f	g
0	0	0	0	0	0	0	1
1	1	0	0	1	1	1	1
2	0	0	1	0	0	1	0
3	0	0	0	0	1	1	0
4	1	0	0	1	1	0	0
5	0	1	0	0	1	0	0
6	0	1	0	0	0	0	0
7	0	0	0	1	1	1	1
8	0	0	0	0	0	0	0
9	0	0	0	0	1	0	0
A	0	0	0	1	0	0	0
B	1	1	0	0	0	0	0
C	0	1	1	0	0	0	1
D	1	0	0	0	0	1	0
E	0	1	1	0	0	0	0
F	0	1	1	1	0	0	0

设计一个 BCD 码到 7 段码的译码器，输入是一个 4 位的待译码信号 x[3:0]，输出是一个 7 段的数值。

图 2 是 DE2 实验板 7 段数码管显示电路图。DE2 开发板提供了 8 个独立的 7 段数码管，每一段都与 FPGA 有单独的引脚连接。

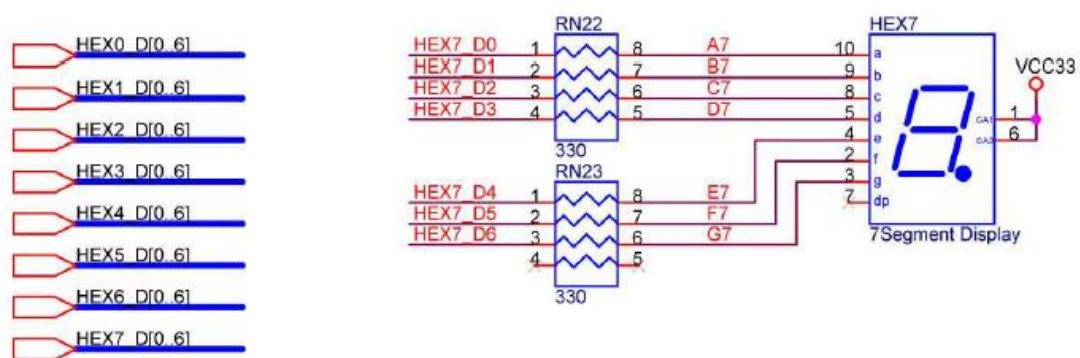


图 2 DE2 实验板 7 段数码管显示电路图

本实验中，设计一个数码管七段译码器电路，通过 4 个拨码开关（SW0~SW3）作为待显示的 BCD 码输入，输出为对应 0~F 的数码驱动信号。

外部电路连接方法：

表 2 数码管七段译码器 FPGA 引脚连接

信号名称	用途	对应 FPGA 引脚
SW[0]	Input	N25
SW[1]	Input	N26
SW[2]	Input	P25
SW[3]	Input	AE14
HEX0[0] (a)	Output	AF10
HEX0[1] (b)	Output	AB12
HEX0[2] (c)	Output	AC12
HEX0[3] (d)	Output	AD11
HEX0[4] (e)	Output	AE11
HEX0[5] (f)	Output	V14
HEX0[6] (g)	Output	V13

实验一 计数器

实验目的:

- 掌握简单时序逻辑电路的设计方法;
- 了解任意进制计数器的设计方法;

实验原理:

计数器的原理

计数器是一种常用的时序电路，它按照规定的方式改变内部各触发器的状态，以记录输入的时钟脉冲的个数。按照规定的计数顺序的不同，计数器可以分为加法计数器、减法计数器、可逆计数器和不同进制的计数器；按照工作方式的不同，又可以分为异步计数器和同步计数器。

以二进制计数器为例，加法计数器在计数脉冲依次输入时，相应的二进制数据是依次增加的。表 3 给出了 4 位加法计数器的功能表。可以看出，每来一个计数脉冲，最低位 QA 的状态变化一次，其后各位则在低一位触发器的状态由 1 变为 0 时发生状态变化。这样，利用低一位的反相输出作为高一位的时钟输入，便可以构成加法计数器。

表 3 4 位加法计数器的功能表

计数器脉冲数目	计数器输出			
	QD	QC	QB	QA
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0

15	1	1	1	1
----	---	---	---	---

减法计数器的计数规律与加法计数器相反，每来一个计数脉冲，计数器的值是减一。利用 D 触发器可以方便地构成减法计数器，与加法计数器不同的是，减法计数器利用低一位的输出作为高一位的时钟输入，而加法计数器则是利用低一位的反相输出作为高一位的时钟输入。

上面介绍的加法计数器和减法计数器属于异步计数器，由于计数控制信号是在各级间逐级传递的，这种计数器从时钟脉冲上升沿达到最后一个触发器翻转到规定的状态，需要较长的延时；计数器位数越多，翻转到稳定状态的时间就越长。为了提高计数器的工作速度，可以采用同步计数器。在同步计数器中，各个触发器使用同一个计数控制时钟，每一位在时钟上升沿到来时是否翻转取决于比其低的位是否都是“1”。其中，触发器的翻转是在时钟上升沿同步进行的，其翻转稳定时间仅仅取决于单级触发器的翻转时间，而与计数器的位数无关。

其他进制的计数器的设计类似于二进制计数器，可以根据其功能表进行设计。

实验内容：

采用行为级设计方法设计计数器：

- (1) 设计一个具有异步复位控制的 4bits 二进制同步加法计数器。
- (2) 设计一个具有异步复位控制的 4bits 二进制异步加法计数器。
- (3) 在上述计数器的基础上设计一个减法计数器（同步、异步均可）

为了便于观察计数结果，计数器的时钟输入信号采用按键代替，计数结果通过七段数码管 HEX0 输出。

外部电路连接方法：

信号名称	用途	对应 FPGA 引脚
KEY[0]	Input / 时钟输入	G26
KEY[1]	Input / 异步复位（低有效）	N23
HEX0[0]	Output / 计数输出	AF10
HEX0[1]	Output / 计数输出	AB12
HEX0[2]	Output / 计数输出	AC12
HEX0[3]	Output / 计数输出	AD11
HEX0[4]	Output / 计数输出	AE11
HEX0[5]	Output / 计数输出	V14
HEX0[6]	Output / 计数输出	V13

实验二 序列检测器设计

实验名称：序列检测器设计/有限状态机实现

实验目的：掌握有限状态机的实现原理和方法；掌握序列检测的方法

实验原理：

有限状态机（Finite State Machine, FSM）是逻辑电路设计中经常要遇到的，在数字电路中，通过建立有限状态机来进行时序数字逻辑的设计。在复杂数字系统设计中，有限状态机主要通过硬件描述语言实现，硬件描述语言能够清晰的描述状态转移过程和输入输出变量关系，使得时序逻辑设计大大简化，进而极大降低系统设计复杂度，提高系统模块化程度。

有限状态机从本质上讲是由寄存器和组合逻辑构成的时序电路，各个状态之间的转移总是在时钟的触发下进行的。可以通过建立原始状态表和状态化简来设计电路。

实验内容：

在连续输入的串行数据流中检测特定序列“101011”，一旦检测到一个“101011”就输出一个宽度为1个时钟周期的高电平脉冲。例如，当输入为“0010101101011100010101100”时，输出为“0000000010000010000000010”。两个101011序列可以重叠，如例子中的第二个1。

- （1）用有限状态机设计序列检测器，检测序列“101011”。
- （2）用移位寄存器和组合逻辑实现序列检测器，检测序列“101011”。
- （3）要求：按键去抖，状态机要求最简

外部电路连接方法

有限状态机实现：

信号名称	用途	对应 FPGA 引脚
KEY[0]	Input / 时钟输入	G26
KEY[1]	Input / 异步复位（低有效）	N23
SW[0]	Input / 串行数据输入	N25
LEDR[0]	Output / 检测结果输出	AE23
LEDR[1]	Output / 状态机显示[0]	AF23
LEDR[2]	Output / 状态机显示[1]	AB21
LEDR[3]	Output / 状态机显示[2]	AC22

移位寄存器实现：

信号名称	用途	对应 FPGA 引脚
KEY[0]	Input / 时钟输入	G26
KEY[1]	Input / 异步复位（低有效）	N23

SW[0]	Input / 串行数据输入	N25
LEDR[0]	Output / 检测结果输出	AE23
LEDR[1]	Output / 移位寄存输出[0]	AF23
LEDR[2]	Output / 移位寄存输出[1]	AB21
LEDR[3]	Output / 移位寄存输出[2]	AC22
LEDR[4]	Output / 移位寄存输出[3]	AD22
LEDR[5]	Output / 移位寄存输出[4]	AD23
LEDR[6]	Output / 移位寄存输出[5]	AD21

实验三 频率计设计

实验名称：频率计设计

实验目的：掌握频率计的原理和设计方法。

实验原理：

频率计用于对一个未知频率的周期信号进行频率测量，在 1s 内对信号周期进行计数，即为此周期信号的频率。

频率计内部实现框图如下所示，其内部包括频率量程处理模块（10 分频）、时钟频率产生模块、控制信号产生模块、十进制计数器模块、锁存器模块、译码显示模块等。

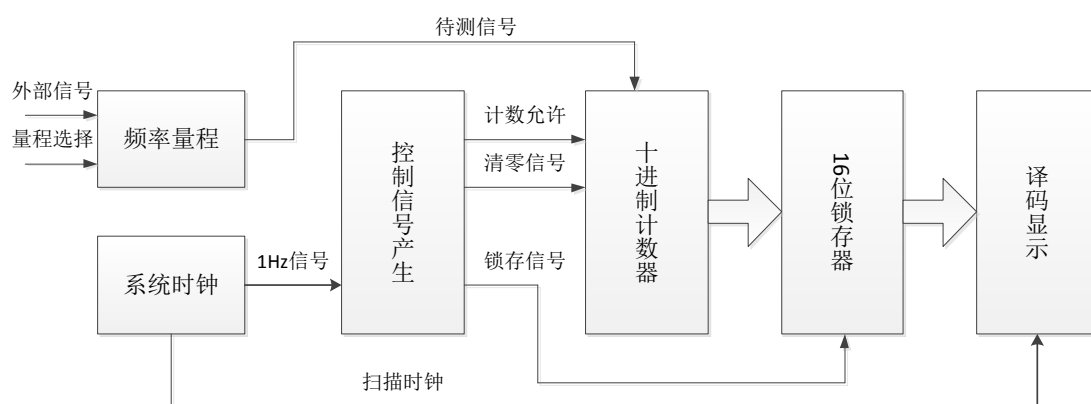


图 3 频率计实现原理图

利用系统时钟产生 1Hz 的控制信号，在 1s 的时长内利用计数器对待测信号进行计数，将计数结果锁存并输出到数码管中显示。其中，频率量程模块负责根据设定的量程控制信号决定是否对输入信号进行 10 分频；系统时钟模块根据外部输入的参考时钟（50MHz）产生标准 1Hz 的控制信号；控制信号产生模块产生计数所需的使能、清零信号以及保存测量结果所需的锁存信号；十进制计数模块在计数使能、清零信号控制下对外部输入信号（或其 10 分频信号）在 1s 周期内对其进行计数操作；锁存器模块在计数完成之后对计数结果进行锁存，保存上一测量周期的测量结果；译码显示模块将测量结果输出到 LED 数码管显示。

实验内容：

- (1) 设计一个频率计，满足对不同频率范围信号的测量，设置两个档位，对于高频信号需要首先进行 10 分频；低频档直接进行测量。
 - a. 待测信号产生模块产生 4 个不同频率的待测时钟信号，通过 SW[1:0]选择不同频率的时钟输出。
 - b. 频率量程处理。使用开关（SW2）来控制量程。SW2=0 表示低量程，直接测量周期；SW2=1 表示高量程，先对待测信号进行 10 分频。
 - c. 系统时钟模块。利用开发板上提供的 50MHz 时钟产生所需要的时钟和控制信号。

- d. 控制信号产生模块。利用 1Hz 时钟产生计数器使能信号，清零信号以及锁存器的锁存信号。
- e. 4 位十进制计数器。当计数使能有效时对计数器的 clock 信号进行十进制计数，reset 有效时对十进制计数器进行清零。4 位十进制计数器的每一位均从 0~9 计数（不可以出现 A~F 几个十六进制数）。
- f. 锁存器模块。结果锁存信号有效时输出锁定，否则，输出透明显示计数器值。
- g. 译码显示模块。4 个 7 段数码管分别显示千百十个位。另外使用 LED 灯（LEDR7）区别高低量程。

测试方法：为了测试方便，在实验指导书的最后提供一个待测信号产生模块

```
signalinput(sysclk, resetb, testmode[1:0], signin);
```

用于产生 4 个不同频率的待测试时钟信号。其中，testmode[1:0]表示 SW[1:0]的开关输入，00、01、10、11 分别选择 4 个不同频率的信号。sysclk 是开发板提供的 50MHz 时钟，signin 是输出的待测信号。

要求同学按照下图所示对测试信号输入模块以及自行设计的频率计模块进行连接：

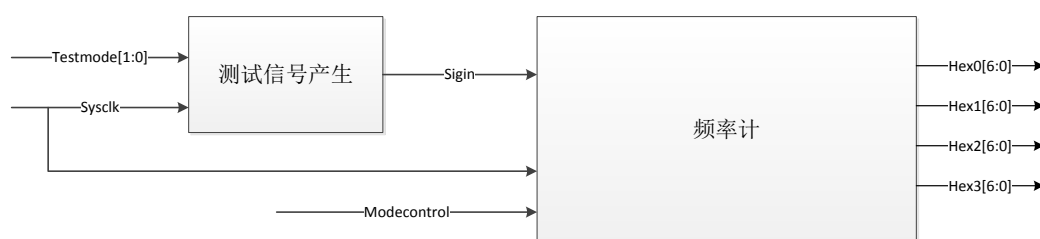


图 4 频率计顶层模块连接

外部电路连接方法

信号名称	用途	对应 FPGA 引脚
CLOCK_50	Input / 50MHz 时钟输入	N2
KEY[1]	Input / 异步复位（低有效）	N23
SW[0]	Input / 频率选择[0]	N25
SW[1]	Input / 频率选择[1]	N26
SW[2]	Input / 量程选择	P25
LEDR[0]	Output / 量程指示	AE23
HEX3~HEX0	Output / 频率计输出	

附：测试信号程序：signalinput.v

```
/*
sysclk    : 50MHz 系统时钟
resetb    : 低电平有效异步复位信号
testmode  : 00,01,10,11 分别代表 3125,6250,50,12500Hz, 使用 SW1~SW0 来控制
signin    : 待测时钟输出, 频率由 testmode 选择
*/

module signalinput(
    sysclk,
    resetb,
    testmode,
    signin
);

input          sysclk;
input          resetb;
input  [1:0]   testmode;
output         signin;

reg  [20:0]   divide;

always @(*)
begin
    case(testmode)
        2'b00 : divide=21'd16000;           //3125Hz
        2'b01 : divide=21'd8000;            //6250Hz
        2'b10 : divide=21'd1000000;         //50Hz
        2'b11 : divide=21'd4000;           //12500Hz
    endcase
end

reg          signin;
reg  [20:0]   state;

always @(posedge sysclk or negedge resetb)
begin
    if(~resetb) begin
        signin <= 1'b0;
        state <= 21'd0;
    end
    else begin
```

```
        if(state==divide)
            state <= 21'd0;
        else
            state <= state + 21'd2;

        sign <= (state==21'd0) ? ~sign : sign;
    end
end

endmodule
```


实验四 串口收发器设计

实验名称：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。

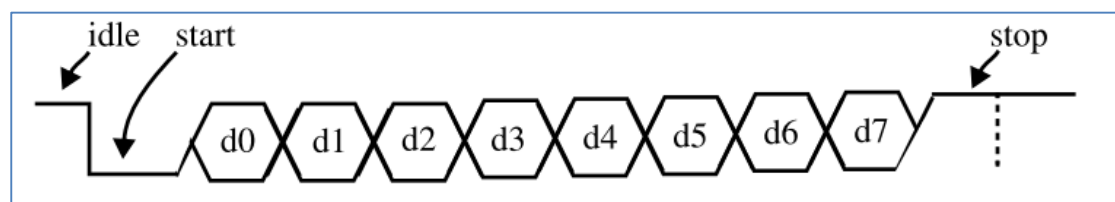


图 5 串口时序示意图

图 5 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，最后发送最高位。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”。

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，接收器设计不好会导致采样错误。

2. DE2 开发板相关电路介绍：

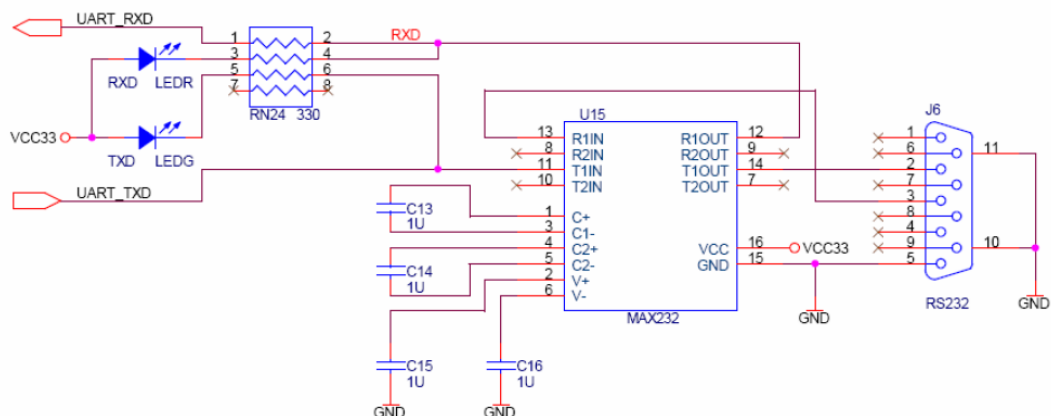


图 6 MAX232(RS-232)结构图

图 6 为实验中采用的 DE2 开发板上的串口连接示意图。MAX232 是 MAXIM 公司专门为 RS-232 标准串口设计的单电源电平转换芯片，FPGA 通过其与 PC 机上的串口通用程序通信。在 PC 机一侧通过串口调试助手选择对应的 USB COM 端口，设置波特率为 9600，1 位停止位，无硬件数据流控，无奇偶校验，如图 7 所示。

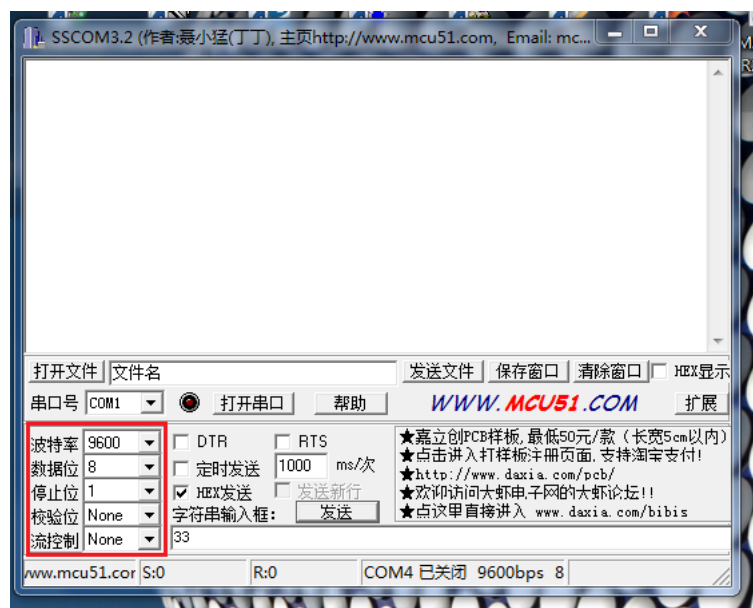


图 7 串口调试助手设置示意图

3. 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。

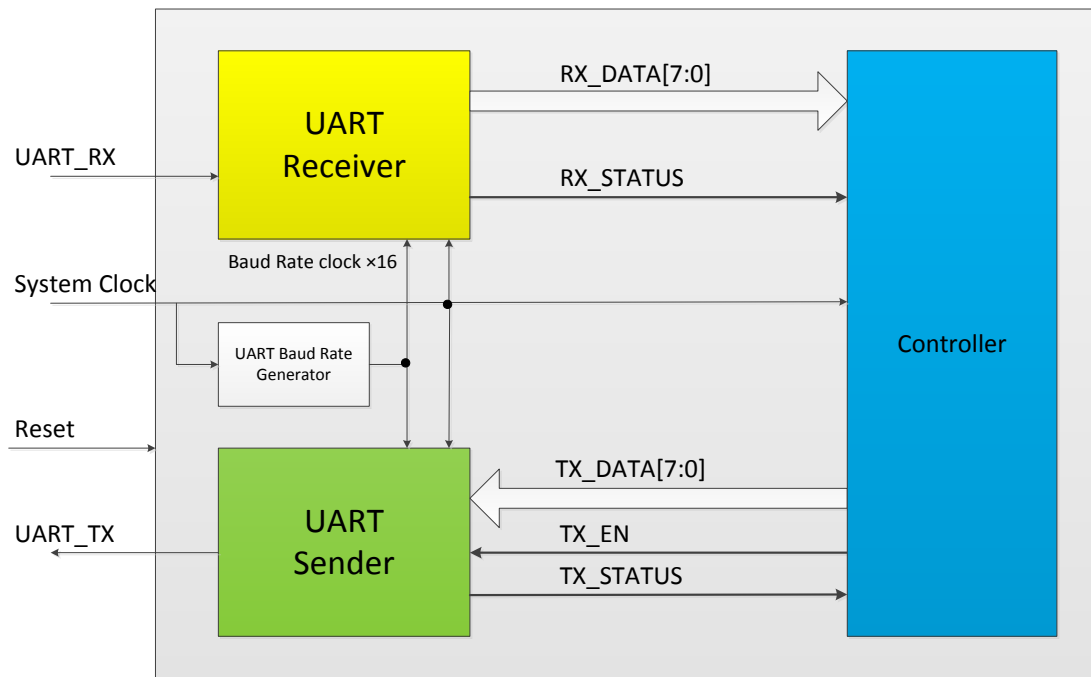


图 8 串口收发器功能实现框图

串口接收器（UART Receiver）模块负责从串口中接收串行数据流，并根据 UART 通讯协议提取接收到的数据并发送给控制器。每当串口接收器收到一个完整的数据，在 **RX_STATUS** 上输出一个高电平指示脉冲，并同时**在 RX_DATA 上输出接收到的有效数据**，RX_DATA 上的接收数据一致有效到下一个 RX_STATUS 脉冲位置。串口接收器与控制器的接口波形如下图所示

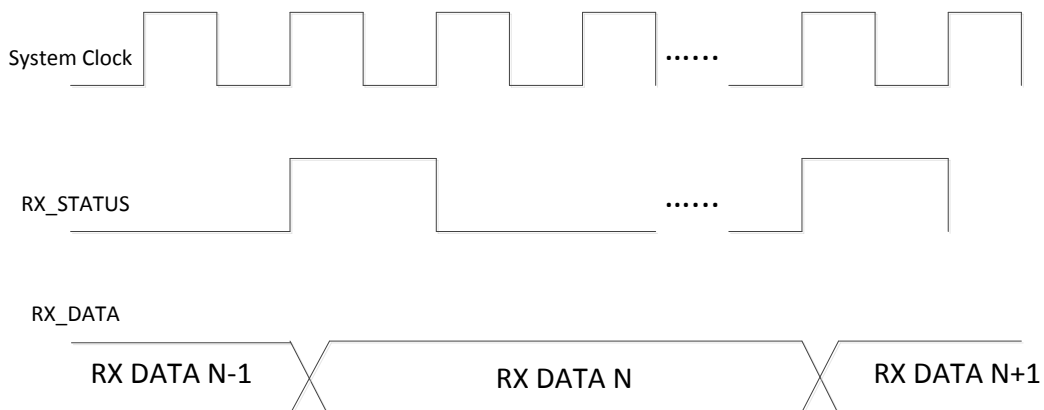


图 9 串口接收器与控制器接口信号

由于串行数据帧与接收时钟是异步的，所以接收器功能实现中的关键是**接收器时钟与每个接收字符的同步**。一个有效的方法是**接收器采用高速率时钟对串行数据进行采样**，通常采样频率是**位时钟频率的整数倍**。理论上倍数越高接收数据各位的分辨率越高，实际中，一般**最大选择 16 倍**。波特率发生器(Baud Rate Generator)模块负责根据 System clock 时钟产生所需的 16 倍（或者其他倍数）波特率的接收时钟。

接收器应该尽可能地靠近位周期的中心处对每位采样。如果接收器能很好地预测起始位的开始，那么他可在起始位的下降沿到来之后，等待半个位周期再采样数据位。此后，接收器每等待一个位周期采样一个数据位，直至收到最后一位为止。倘若接收时钟的频率足够接近发送时钟，使得最后位能在离该位的精确中心位置半个周期内对他采样，以上方案就能正确地工作。这意味着接收时钟相对于发送时钟在 10~11 个时钟周期内，其增加和减少应小于半个位的时间间隔。因此，要求收发双方 2 个时钟的误差容限在 5% 以内。

图 10 为采用高倍时钟进行接收信号的采样示意图。图 11 为 UART 仿真时序示意图。



图 10 接收端采样示意图



图 11 UART 信号仿真示意图

串口发送器（UART Sender）从控制器接收待发送数据，然后根据 UART 通讯协议串行发送出去。当控制器检测到 TX_STATUS 上出现高电平时，意味着此时串口发送器处于空闲状态可以接收一个新的发送数据，控制器在 TX_DATA 上输出待发送的数据，并同时在 TX_EN 上输出一个高电平脉冲，指示串口发送器启动一个新的数据发送。串口发送器与控制器之间的接口波形如下所示

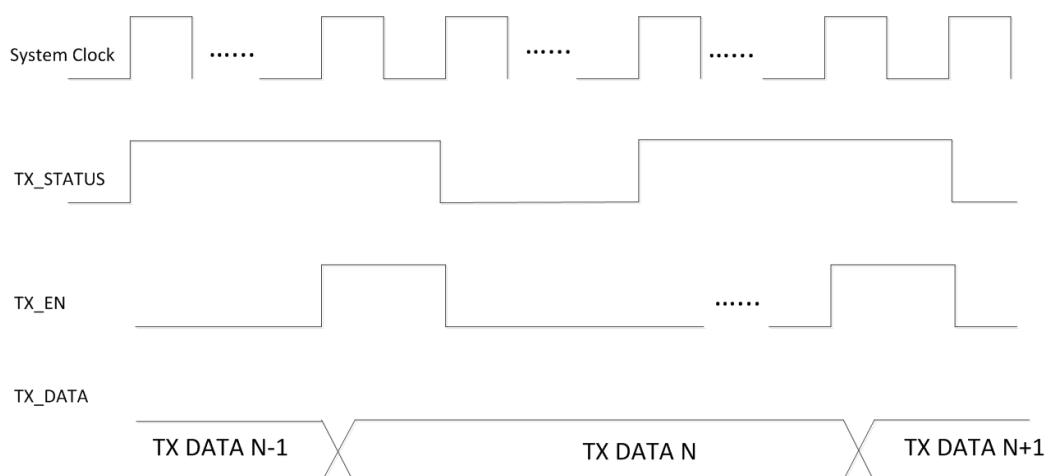


图 12 串口发送器与控制器接口信号

实验内容：

设计一个具有固定波特率的 UART 串口收发器，可以实现 9600 波特率的串口通信，能够与 PC 机串口进行通信，支持 8 比特数据位、1 比特停止位、无校验、无硬件流控模式。

- (1) 实现串口接收器的行为级设计，满足上述串口通信要求。
- (2) 实现串口发送器的行为级设计，满足上述串口通信要求。
- (3) 实现控制器，根据接收数据产生发送数据，规则如下：如果接收数据的 MSB 为‘1’，则产生接收数据的反码作为发送数据；如果接收数据的 MSB 为‘0’，则直接将接收数据的作为发送数据；
- (4) 将硬件平台的串口与计算机相连（USB 接口），通过串口调试助手进行验证。

外部电路连接方法

信号名称	用途	对应 FPGA 引脚
CLOCK_50	Input / 50MHz 时钟输入	N2
KEY[1]	Input / 异步复位（低有效）	N23
UART_RXD	Input / 串口接收	C25
UART_TXD	Output / 串口发送	B25