# 通信系统收发端的奇偶校验设计

# 设

# 计

# 文

# 档

指导老师：陈亦欧

成员姓名：

目录

[通信系统收发端的奇偶校验设计 1](#_Toc3744)

[1 系统功能概述 3](#系统功能概述)

[1.1 功能 3](#_Toc17322)

[1.2 原理 3](#_Toc28967)

[1.2.1 奇偶校验位产生电路 3](#_Toc32689)

[1.2.2 奇偶校验检测电路 4](#_Toc5887)

[1.2.3 并串、串并转换电路 4](#_Toc12418)

[2 系统架构与模块分割 4](#系统架构与模块分割)

[3 各模块设计方案，端口定义及电路图 5](#各模块设计方案，端口定义及电路图)

[3.1 奇偶校验位产生电路/奇偶校验检测电路 5](#_Toc5440)

[3.1.1 电路图 5](#_Toc4920)

[3.1.2 端口定义 5](#_Toc2773)

[3.1.3 设计方案 6](#_Toc16417)

[3.2 并串，串并转换电路 6](#_Toc14788)

[3.2.1，并串转换电路 6](#_Toc30324)

[3.2.2串并转换电路 7](#_Toc8459)

[3.3 信道传输模块 8](#_Toc9987)

[3.4 通信系统收发端的奇偶校验顶层模块 8](#通信系统收发端的奇偶校验顶层模块)

[4 各种测试方案及测试报告 9](#各种测试方案及测试报告)

[4.1 直接传输数据的testbench代码以及时序图输出 10](#_Toc2534)

[4.2 改变传输数据的testbench代码以及时序图输出 11](#_Toc17468)

# 1 系统功能概述

## 功能

将数据进行串行传输，并通过奇偶校验的方法来检验一组数据在传输过程中是否发生传输错误

## 原理

### 奇偶校验位产生电路

若传输的二进制数据中有奇数个1，则奇偶校验电路输出1；若传输的二进制数据中有偶数个1，则奇偶校验电路输出0。输出的那一位称为数据的校验位。该原理和异或门的功能一致，所以可使用异或门组合来进行电路的设计。

若输入为3位，则：

奇校验位逻辑值的表达式为：*=*（）’

偶校验位逻辑值的表达式为：=

### 奇偶校验检测电路

通过校验电路产生的校验位与传输数据一同传输给奇偶校验检测电路，检测传输是否出错。若传输过程发生数据一位错或者奇数位错时，电路会输出1，代表error，否则输出0，代表传输无错误。

奇校验检测：

偶校验检测：

### 并串、串并转换电路（信道传输）

根据控制电路和相应的时钟信号，将并行的数据按顺序一位一位的传输到信道上，再将串行的数据从信道上传输到并行的奇偶校验检测电路中。该转换电路可节约传输数据所需要的的硬件资源。

# 2 系统架构与模块分割

整体系统共可分割为六个模块：

1, 奇偶校验位产生模块：将一组数据进行输入，输出该组数据的校验位。

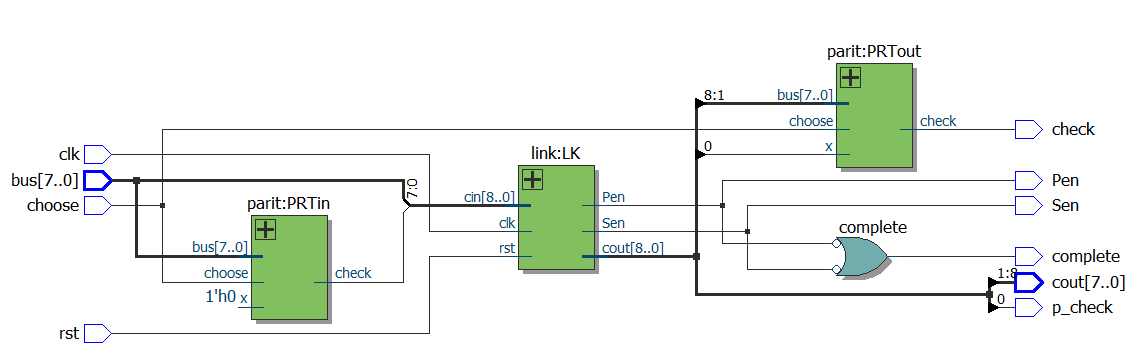
2, 并串转换模块：将数据进行并串转换，输入信道中。

1. 串并转换模块：将信道中串行数据，进行并行转换输出。
2. 信道传输模块：将并串，串并模块进行整合。

5, 奇偶校验检测模块：将数据与校验位并行输入，根据输出检测数据传输过程中是否出现错误。**该模块与奇偶校验位产生模块的电路是一样的。**

6, 奇偶校验信道传输的顶层模块

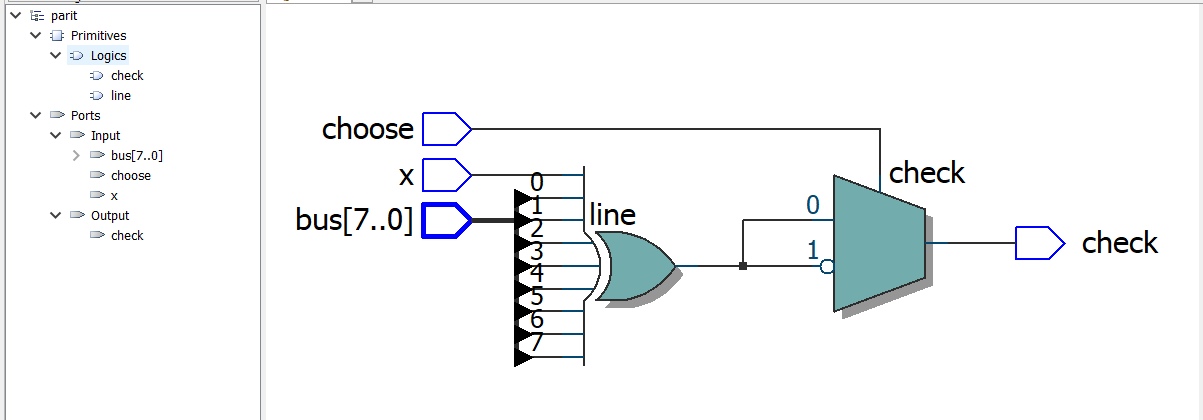
整体系统架构图如下:



# 3 各模块设计方案，端口定义及电路图

## 3.1 奇偶校验位产生电路/奇偶校验检测电路--parit模块

### 3.1.1 电路图



### 3.1.2 端口定义

1, choose为1位输入端，设置校验方式是奇校验还是偶校验，若choose=1，则为奇校验，反之则为偶校验。

2, x为校验位，在奇偶校验电路中置0即可（。。。。。。。。。。。。。。。。。。。。。。）

3, bus为8位输入数据线

4, check为1位输出端，根据choose设置的奇偶校验方式，产生输入7位数据的校验位。

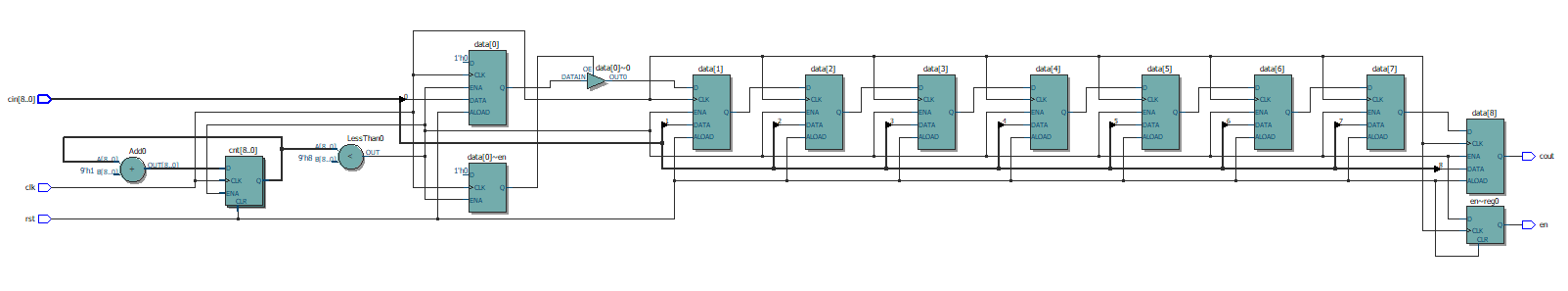
### 3.1.3 设计方案

根据choose来选择奇或偶校验方式，若choose=1，则为奇校验，反之则为偶校验。将8位输入与一位校验位同时作为输入，在奇偶校验位产生电路中，x置0，是为了不影响对输入的8位数据中1的个数的判断。在奇偶校验检测电路中，奇偶校验位产生电路的输出作为x的输入。

## 3.2 并串，串并转换电路

### 3.2.1，并串转换电路pal\_serial模块

#### 3.2.1.1 电路图



#### 3.2.1.2 端口定义

输入有三个，分别为9位数据输入（8位数据+1位校验位），1位时钟输入，1位复位输入。

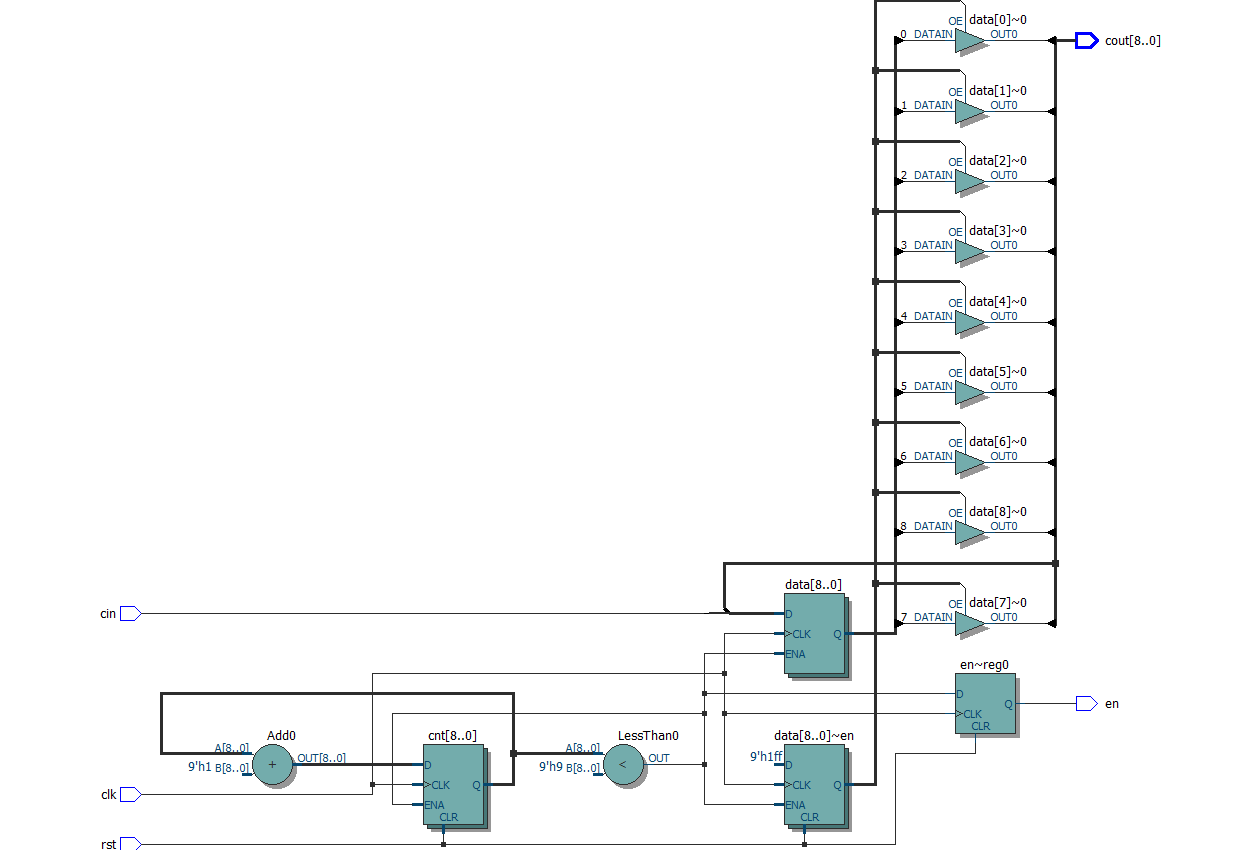
输出有两个，分别为1位数据输出和1位传输进行标志位（输出为1，则表示传输正在进行）。

#### 3.2.1.3 设计方案

在复位端置1时，代表新一组数据进行输入，将输入的9位数据存储在内部存储器中，在每个时钟上升沿到来时，将内部存储器的数据一位一位的输出。

### 3.2.2串并转换电路serial\_pal模块

#### 3.2.2.1 电路图



#### 3.2.2.2 端口定义

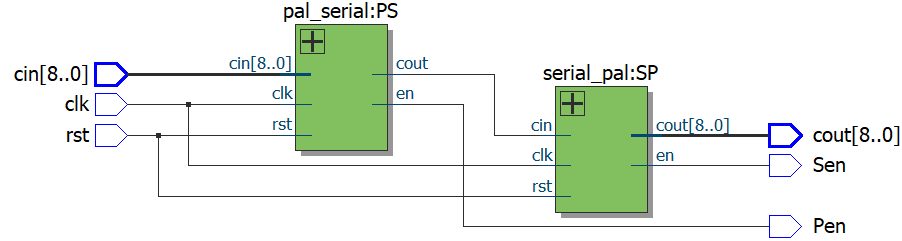
输入有三个，分别为1位数据输入，1位时钟输入，1位复位输入。

输出有两个，分别为9位数据输出（8位数据+1位校验位）和1位传输进行标志位（输出为1，则表示传输正在进行）。

#### 3.2.2.3 设计方案

在复位端置1时，将输出和内部寄存器置为高阻态，等待新一组数据输入，在每个时钟上升沿到来时，将输入的1位数据存入内部存储器当中，并传输给输出。

## 3.3 信道传输link模块

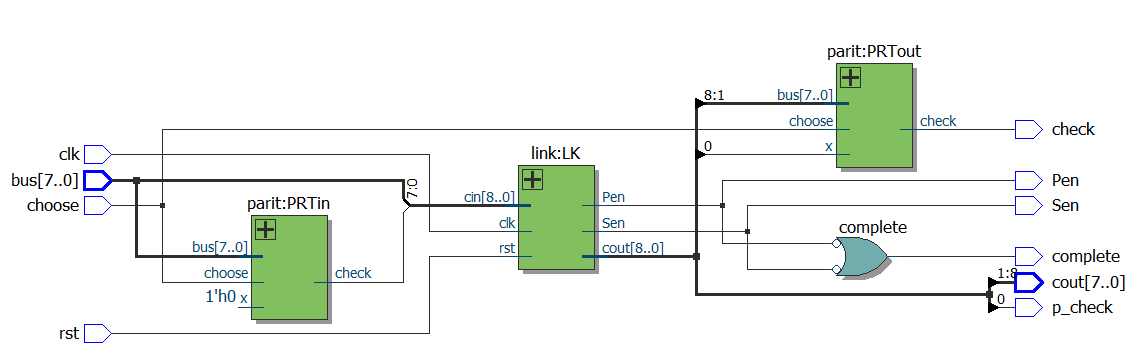


输入有三个，分别为9位数据输入（8位数据+1位校验位），1位时钟输入，1位复位输入。

输出有三个，分别为9位数据输出（8位数据+1位校验位），1位并串电路传输进行标志位，1位串并电路传输进行标志为。

## 3.4 通信系统收发端的奇偶校验顶层allLink模块

#### 3.4.1电路图



#### 3.4.2 端口定义

输入有4个，分别为8位数据输入，1位校验方式选择位输入，1位时钟输入，1位复位输入。

输出有6个，分别为：

1位并串电路传输进行标志位Pen（高电平表示正在传输，即此时并串电路正在传输数据）

1位串并电路传输进行标志位Sen（高电平表示正在传输，即此时串并电路正在传输数据）

1位传输结束标志位complete（高电平表示传输结束，也表示此时信道总线上没有数据在进行传输，即总线闲置）

8位数据输出位cout

1位校验检测位check

1位**校验位产生电路**的输出位p\_check（该位主要为了测试电路运行过程中是否正确，增强测试方案的准确性与详尽性）。

#### 3.4.3 设计方案

将所有模块进行连接，整合，封装，使系统组成结构更加清晰明了。

# 4 各种测试方案及测试报告

### 4.1 直接传输数据的testbench代码以及时序图输出

Testbench代码（只给出initial块中）

initial

begin

clk=0;

bus=8'b1010\_0100;choose=1;rst=1;

#5 rst=0;

#150 bus=8'b0000\_1111;choose=0;rst=1;

#5 rst=0;

end

initial

begin

forever

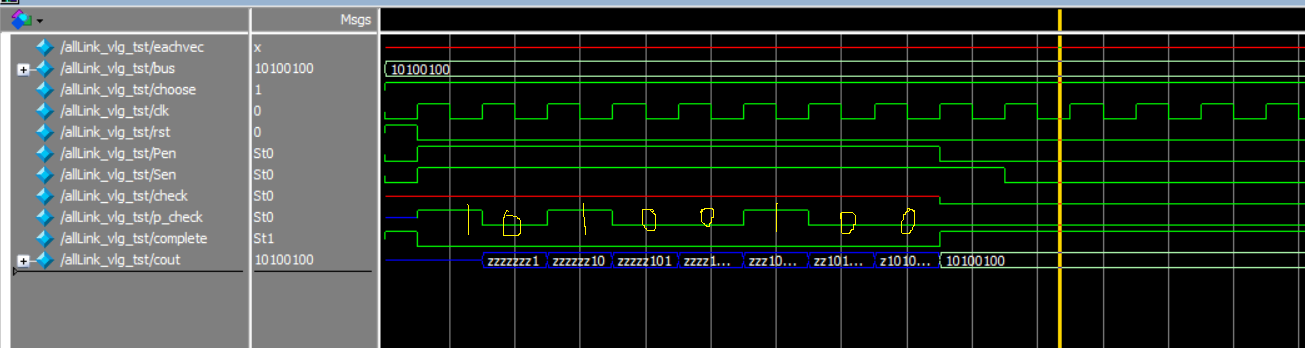
begin

#5 clk=~clk;

end

#500 $stop;

end



由时序图可以看出，输入为1010\_0100，进行奇校验（choose置1），传输过程中不进行任何改变时：

1. 传输完毕时，输出p\_check为0，与输入合起来，一共3个1，表示校验位产生电路产生的校验位p\_check是正确的。

2.传输完毕时，输出check为0，表示数据传输无错误。

3.传输完毕时，输出cout确实为1010\_0100，传输无错误。

4.传输完毕时，输出complete置为1，表示**传输完毕**。

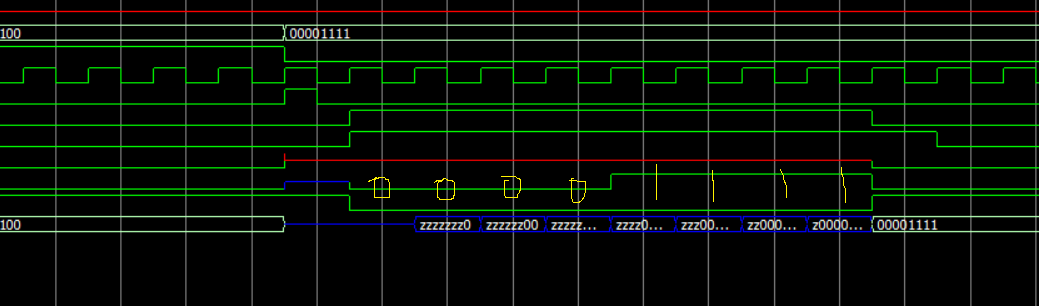
5，**需要特别说明的是：**

传输过程中，complete为0（代表总线忙）。**信道一位一位的数据先传到outlink[0]，再依次左移到outlink的高位（outlink为内部接口，图中无显示）。**

因为p\_check连接9位输出的最低位outlink[0]（在传输完毕时刚好对应输入的上级校验位），而cout[7:0]连接outlink[8:1]。

所以**传输过程中**，p\_check每一个clk上升沿代表的都是输入的一位数据，cout也会晚一个时钟周期显示输入数据，**但传输完毕时p\_check与cout显示均是对应正确的**。观察知，图中**实际输出与理论值一致。**

同理，输入为0000\_1111，进行偶校验（choose置0），过程中不进行任何改变时，时序图如下：



说明：

在重新向电路传输一组新数据时，rst需置1一次，将**新数据送入内部存储器**，然后便可开始传输数据，complete变为0，代表信道总线忙，正在传输一组新数据。由此可实现**可控的循环传输数据**。

### 4.2 改变传输数据的testbench代码以及时序图输出

**Testbench代码（只给出initial块中）**

initial

begin

bus=8'b1100\_0101; choose=0;x=0;

#10 choose=1;x=0;

#10 choose=0;x=1;

#10 choose=1;x=1;

#20 bus=8'b0011\_1110;choose=0;x=0;

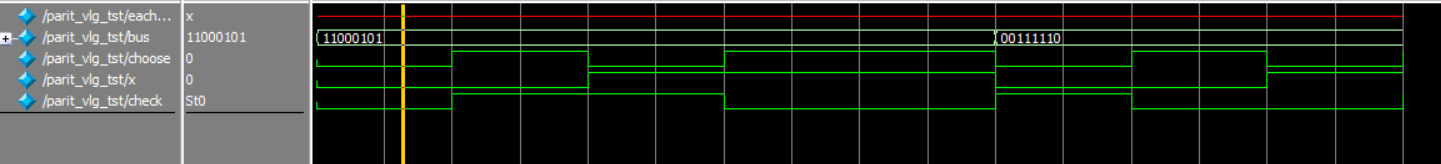
#10 choose=1;x=0;

#10 choose=0;x=1;

#10 choose=1;x=1;

end

由于模块设计分割方式的原因，没有办法把将输入数据在中间过程输出到io口进行篡改，再进行输出，而通过分析思考，可以发现，本系统总体来讲只有两个大模块（信道传输，奇偶校验），“在假定数据传输时出现错误”的前提下（电路连接部分可由整体结果的正确性得出，连接不会有问题），测试时只需测试奇偶校验位产生模块和奇偶校验检测模块，而该两个模块又是由同一电路组成，则只需检测该电路是否能正确产生校验位即可对系统正确性进行评估。所以下面给出相应testbench的时序图：



由图知：

1,bus=8'b1100\_0101;choose=0;x=0时，为偶校验，输出为0，9位输入和1位输出合并一共4个1，为偶数。

2,bus=8'b1100\_0101;choose=0;x=1时，为偶校验，输出为1, 9位输入和1位输出合并一共6个1，为偶数。

3,bus=8'b0011\_1110;choose=1;x=0时，为奇校验，输出为0，9位输入与1位输出合并一共5个1，为奇数。

同理，其他情况也正确。