|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **信号名** | **位数** | **方向** | **来源/去向** | **备注** |
| **接收控制逻辑模块** | | | | |
| **CLK** | 1 | I | 时钟控制逻辑 | 接收时钟 |
| **CLRN** | 1 | I | 控制逻辑模块 | START控制计数器启动 |
| **ENABLE** | 1 | I | 控制逻辑模块 | START控制译码有效 |
| **Q8** | 1 | O | 锁存器 | 锁存时钟 |
| **Q9** | 1 | O | 控制逻辑模块 | 收到校验位 |
| **Q11** | 1 | O | 控制逻辑模块 | 接收数据结束 |
| **串并转换模块** | | | | |
| **CLK** | 1 | I | 时钟控制逻辑 | 接收时钟 |
| **CLRN** | 1 | I | 控制逻辑模块 | START控制移位启动 |
| **SERIAL** | 1 | I | 接收总线 | 串行数据 |
| **Q** | 8 | O | 锁存器、控制逻辑模块（低2位） | 最晚的8位数据 |
| **数据缓冲模块** | | | | |
| **CLK** | 1 | I | 译码器 | Q8上升沿锁存 |
| **CLRN** | 1 | I | 控制逻辑模块 | 系统复位 |
| **D** | 8 | I | 移位寄存器 | 8位并行数据 |
| **Q** | 8 | O | 控制逻辑模块的数据通道 | 8位有效数据 |
| **控制逻辑模块** | | | | |
| **D9** | 1 | I | 译码器 | 校验位 |
| **D11** | 1 | I | 译码器 | 接收数据结束 |
| **SQ0** | 1 | I | 移位寄存器 | 串行数据最低位 |
| **SQ1** | 1 | I | 移位寄存器 | 串行数据次低位 |
| **RQ** | 8 | I | 锁存器 | 8位有效数据 |
| **CLRN** | 1 | I/O | 控制逻辑模块 | 系统复位 |
| **START** | 1 | O | 计数器、译码器、移位寄存器 | 计数、译码和移位启动信号 |
| **SERIAL** | 1 | I | 控制逻辑模块 | 串行数据 |
| **CLK** | 1 | I | 时钟控制逻辑 | 接收时钟 |
| **CSN** | 1 | I | 处理器 | 内部寄存器片选 |
| **RDN** | 1 | I | 处理器 | 内部寄存器读 |
| **WRN** | 1 | I | 处理器 | 内部寄存器写 |
| **ADDR** | 2 | I | 处理器 | 内部寄存器地址 |
| **DATA** | 8 | O | 处理器 | 输出数据缓冲区或中断状态寄存器内容 |
| **INTN** | 1 | O | 处理器 | 中断请求信号 |

注：信号名末尾带字母N的信号，都是低电平有效，其余信号均为高电平有效。

**元件例化的代码：**

CtrlUnit: ctrl PORT MAP (

d9 => t9,

d11 => t11,

sq0 => tdata(0),

sq1 => tdata(1),

rq => treg,

clrn => clrn,

start => start,

serial => serialdata,

clk => clk,

csn => csn,

wrn => wrn,

rdn => rdn,

addr => addr,

data => data,

intn => intn

);

CountUnit: count4 PORT MAP (

clk => clk,

clrn => start,

q => c

);

DecodeUnit: decode4 PORT MAP (

d => c,

enable => start,

q8 => t8,

q9 => t9,

q11 => t11

);

ShiftUnit: sreg PORT MAP (

clk => clk,

clrn => start,

serial => serialdata,

q => tdata

);

RegUnit: reg8 PORT MAP (

clrn => clrn,

clk => t8,

d => tdata,

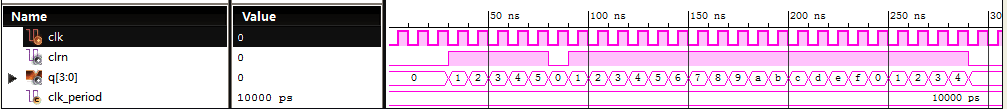
q => treg

);

1. 计数器模块count4

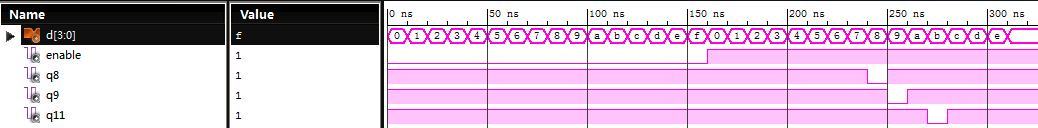
分别测试clrn=’1’和’0’的情况下观察几个时钟周期内计数器的输出情况。

先清零三个周期，然后工作五个周期，再清零一个周期，再工作20个周期，观察计数器的清零情况和循环情况。



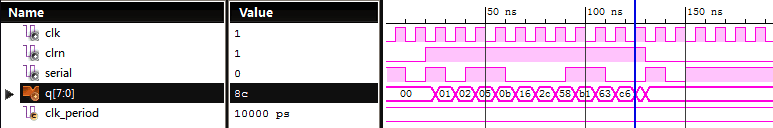
1. 译码器模块decode4

取遍所有输入，观察译码输出。



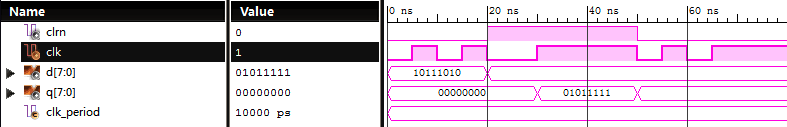
1. 移位寄存器模块sreg

给定时钟和串行输入，测试在清零和不清零时的输出。



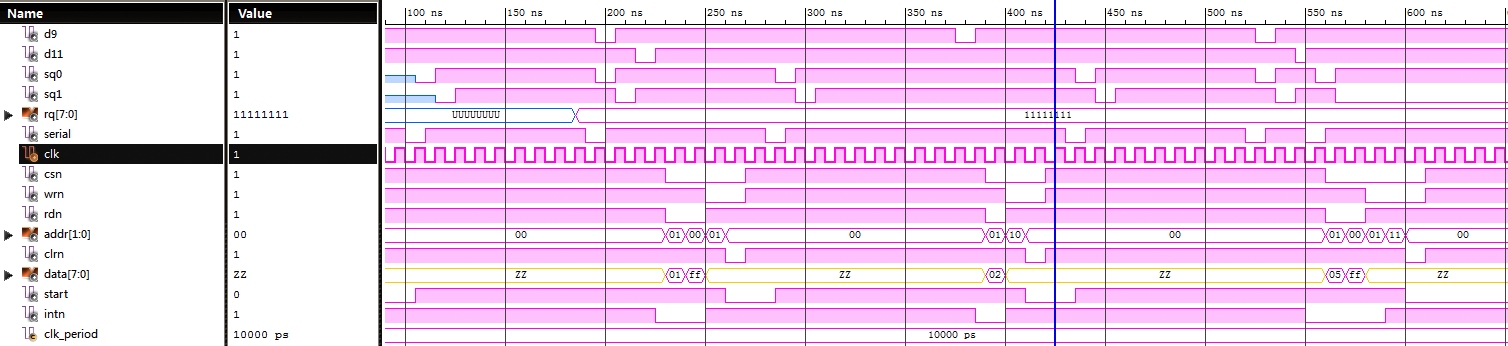
1. 锁存器模块reg8

给定时钟和8位输入，测试清零和不清零时的锁存结果。



1. 控制逻辑模块ctrl

分别测试成功传送、奇偶校验出错、数据溢出这三种情况下的控制输出。送的数据是8个’1’，奇偶校验位应为’0’，第一种情况奇偶校验位送’0’，第二种情况送’1’，第三种情况送’0’并在12位数据结束后送一个低电平代表开始新的数据帧。另外，还要模拟CPU处理中断请求，读数据和清除中断。



系统的功能测试——Serial模块

给定时钟、串行数据和CPU相应操作，测试成功传送、奇偶校验出错、数据溢出三种情况下的数据输出和中断请求输出。测试数据同ctrl模块的测试数据。最开始先发三个清除中断的命令，清除intn。

