基于PCM-Hamming码的FSK调制解调通信系统

1. 小组成员分工

|  |  |  |
| --- | --- | --- |
| 姓名 | 学号 | 分工 |
| 王童 | 2015011201 | Hamming码、并行串行转换、串口通信、整合、调试 |
| 李百双 |  |  |
| 张栩铭 |  |  |

1. 实验原理

运用大学期间所学过的知识来设计一个用于语音传输的通信系统，主要用到的知识有：PCM编解码、Hamming码、串口通信、FSK调制解调技术。

其中，信源编码部分用到了PCM编解码技术，信道编码采用Hamming提供1bit纠错能力，信道传输采用FSK调制解调技术。

【**PCM编解码**】

【**Hamming码**】

1. **编码**

信息码位有8位，经过计算可得校正子有4位，一共是12位码

**确定Q**

**生成矩阵**

用8bit数据左乘生成矩阵可得到12bit Hamming码

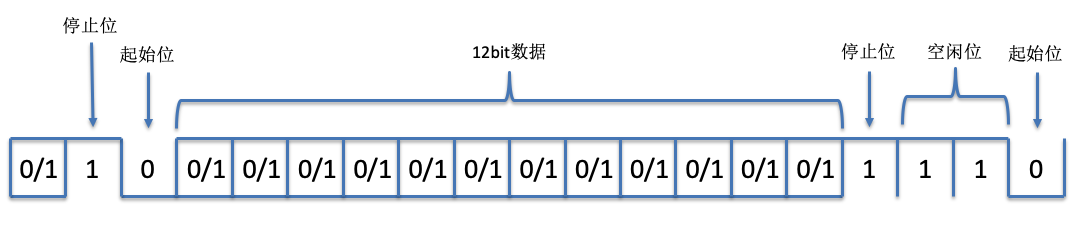
1. **解码**

**监督矩阵**

用接收到的12bit Hamming码左乘监督矩阵，可得到4bit 监督码，当监督码4位均为0时表示信道传输无误码，可将Hamming码的前8位提取出来作为解码出来的数据。若监督码等于的某一行，表示存在误码，由于我们设计的Hamming码只具有1bit纠错能力，所以我们假设只存在1bit误码，则监督码等于监督矩阵的哪一行，则接收到的12bit Hamming码的对应行存在误码，将其置换后取前8bit可得出解码出来的数据。

【**串口通信**】

由于信道存在延时，为了保证从bit流中正确读出传输的数据而不出现错位，我们采用了起止式异步协议

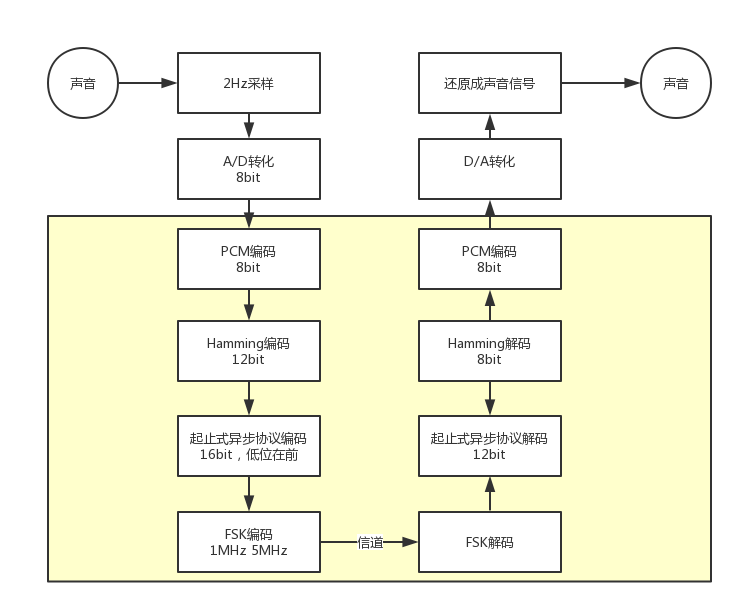


传送时，数据的低位在前，高位在后

起止式异步协议的特点是一个字符一个字符传输，并且传送一个字符总是以起始位开始，以停止位结束，字符之间没有固定的时间间隔要求。其格式如上图所示。每一个字符的前面都有一位起始位（低电平，逻辑值0），字符本身由12位数据位组成，接着是1位停止位，停止位后面是不定长度的空闲位。停止位和空闲位都规定为高电平（逻辑值），这样就保证起始位开始处一定有一个下跳沿。

【**FSK调制解调技术**】

1. 系统设计框图



其中黄色部分是我们具体实现的部分，为了方便展示，我们省去了采样及A/D和D/A转换部分，预设了一些8bit的数据作为输入。

【**时钟分析**】

以下时钟频率均为仿真的时候做的假设，实际烧录到板子上时进行了微调。除了CLK以外的所有时钟信号均基于CLK经过分频得来。

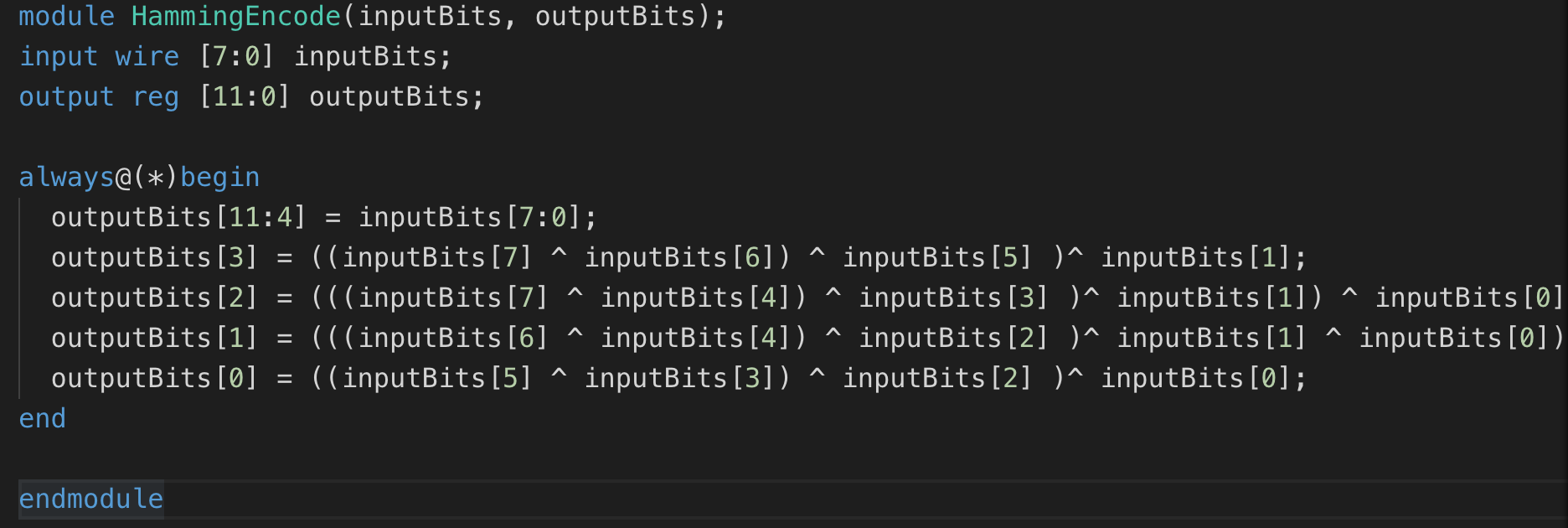
1. 系统时钟CLK：60MHz
2. 采样率CLK\_Sample：2Hz（为了保证输出肉眼可观测到）
3. 传输bit率：CLK\_Serial\_bits：2\*16=32Hz（一个字符对应16位的传输bit流）
4. FSK传输信号：（为了契合串口通信协议，必须保证1对应低频Fsk信号）
   1. Fsk\_0：5MHz（对应0）
   2. Fsk\_1：1MHz（对应1）
5. 各部分的具体实现方法

【PCM编解码】

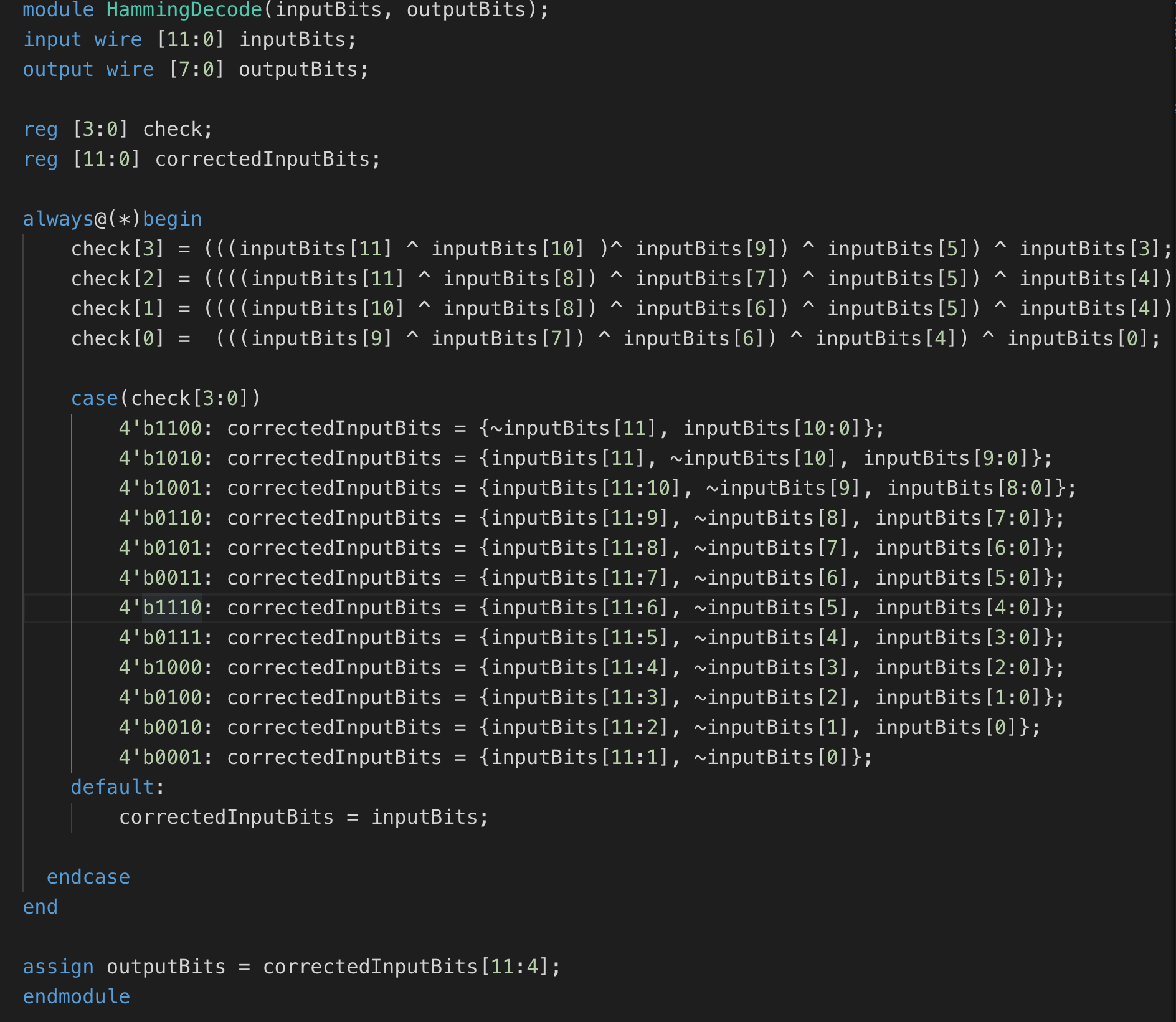
【Hamming码】

二进制乘法可以用异或操作来实现

1. 编码

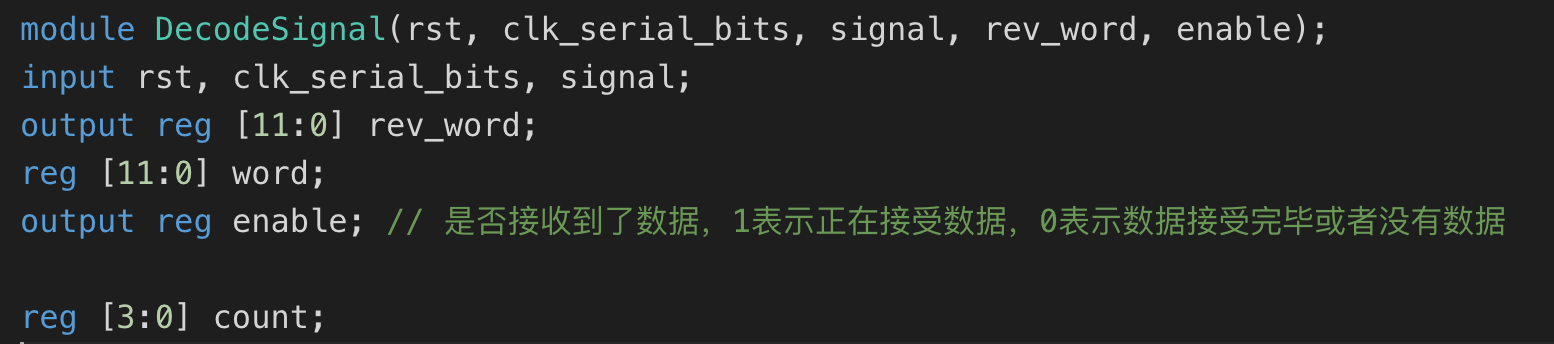


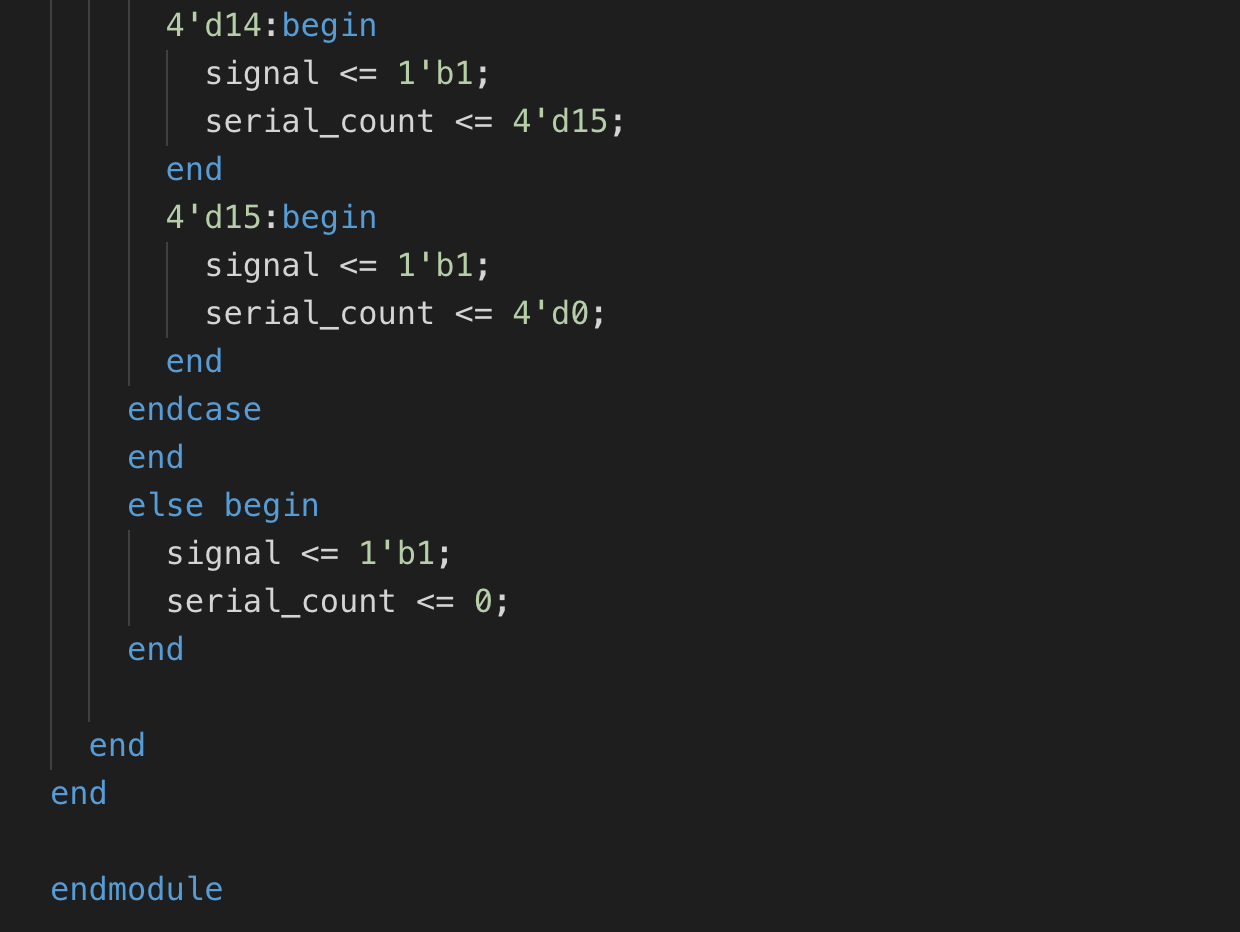
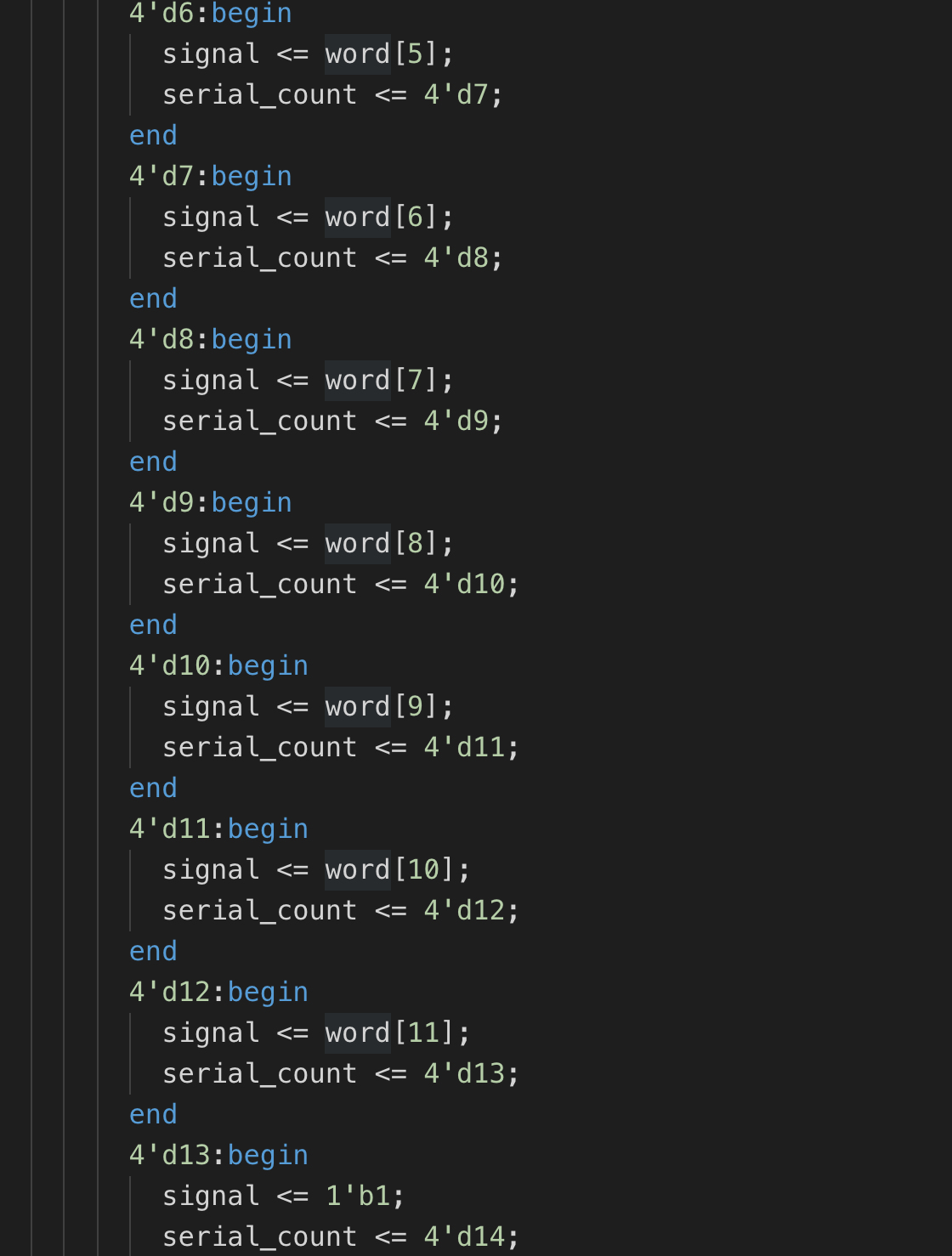
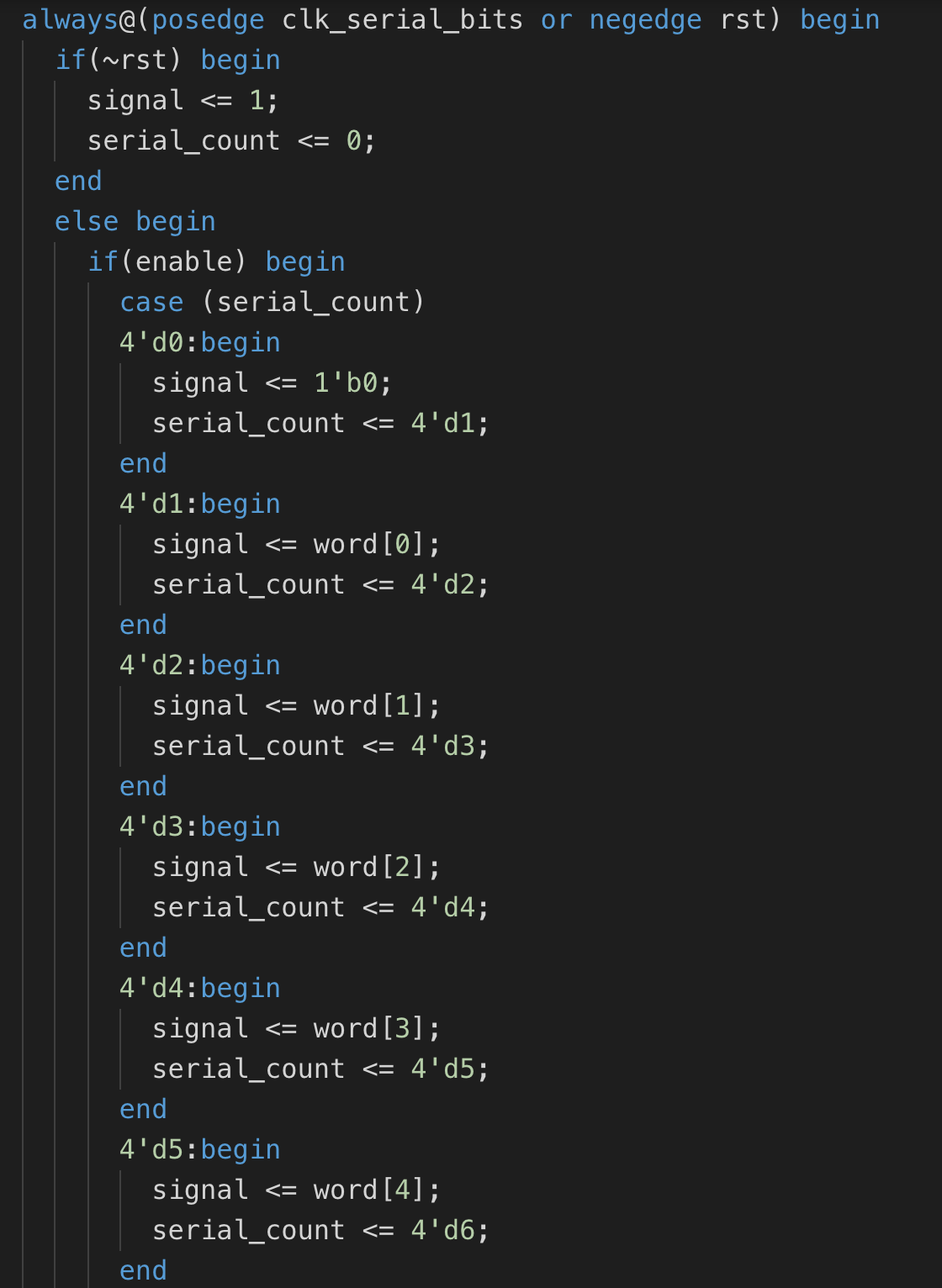
1. 解码



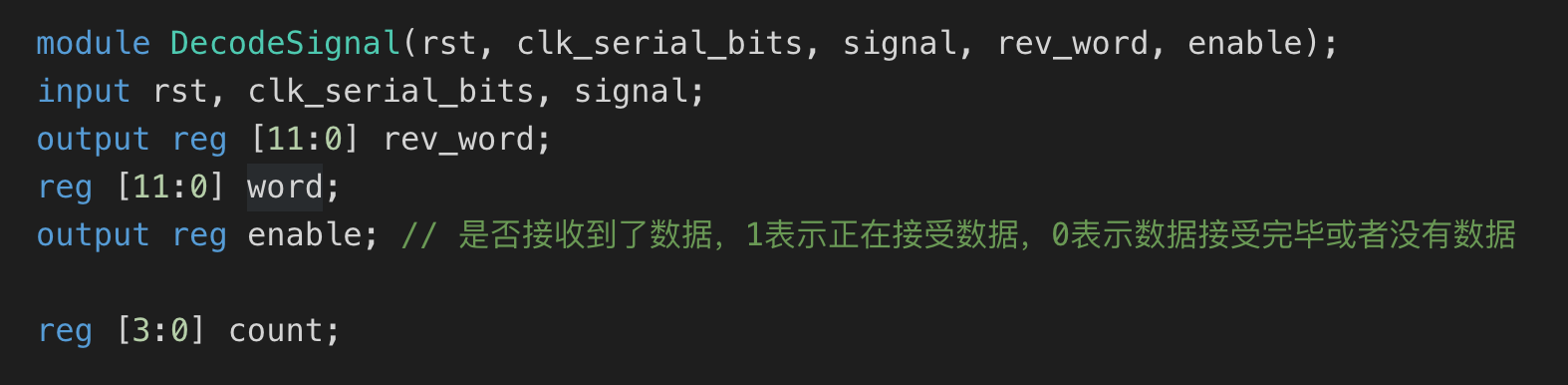
【串口通信】

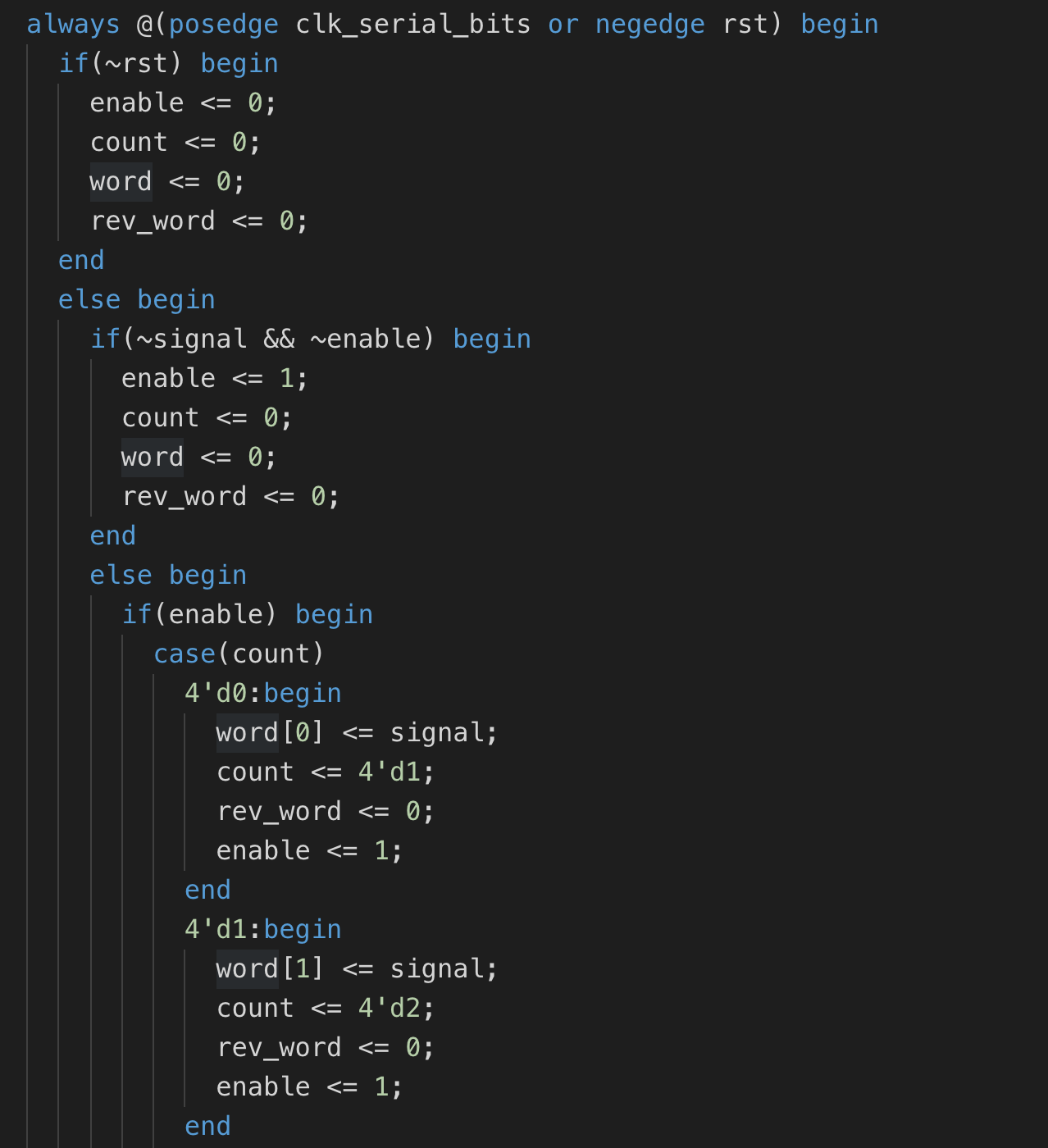
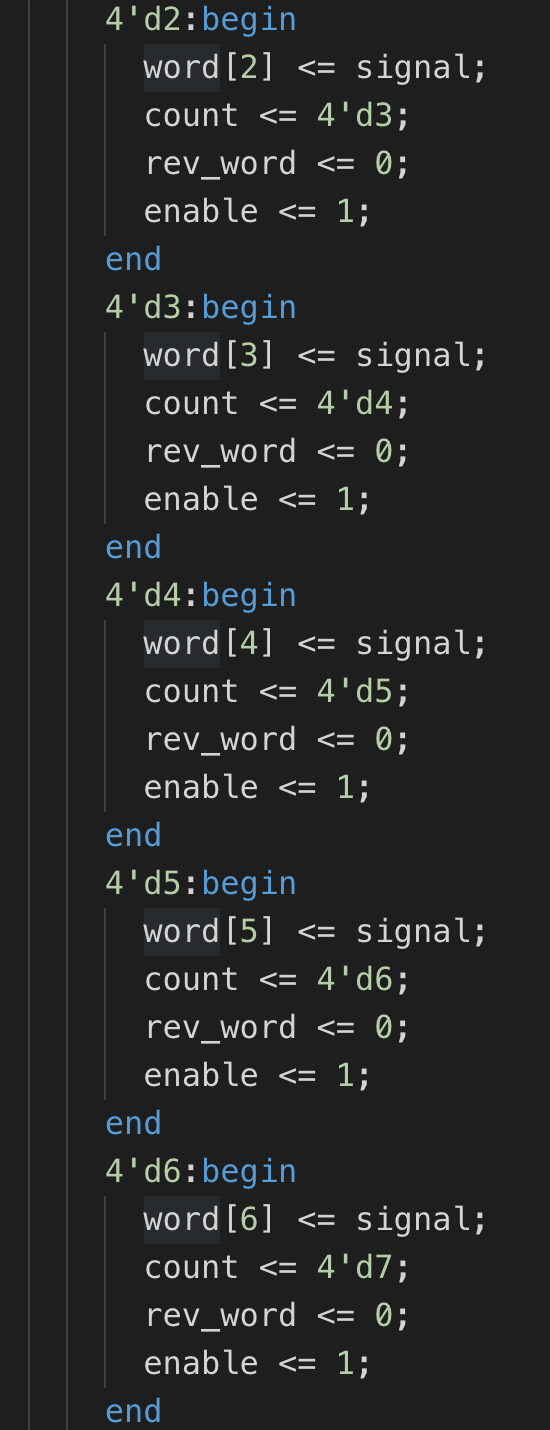
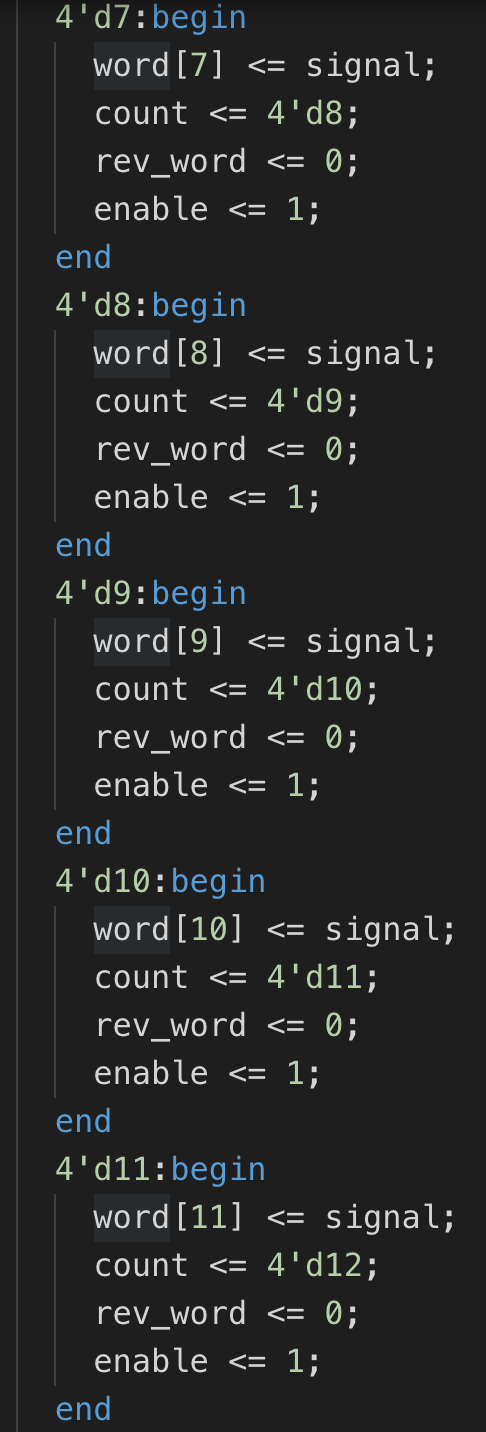
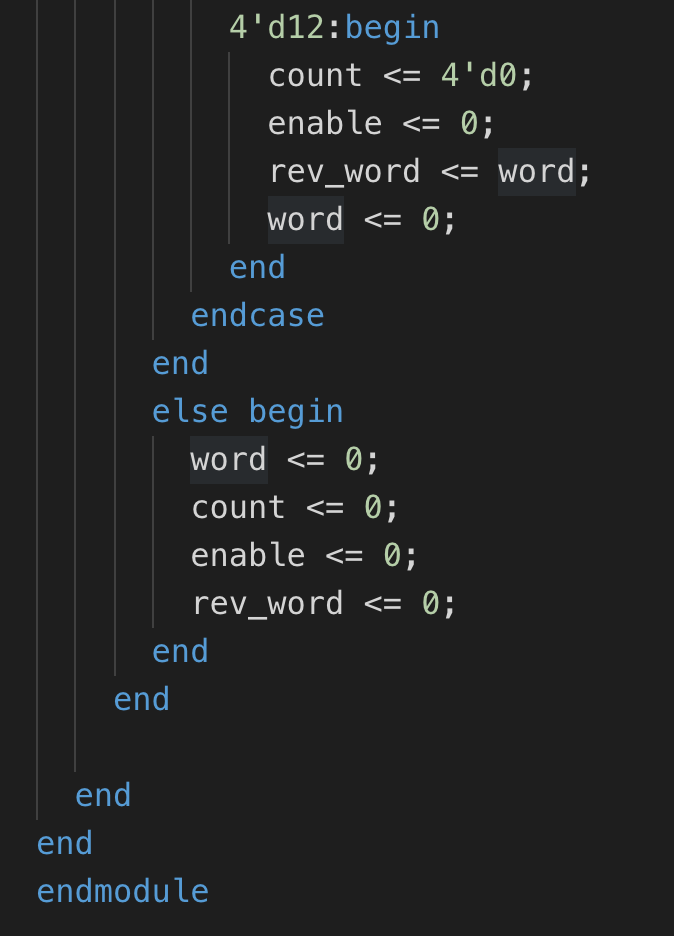
1. 发送





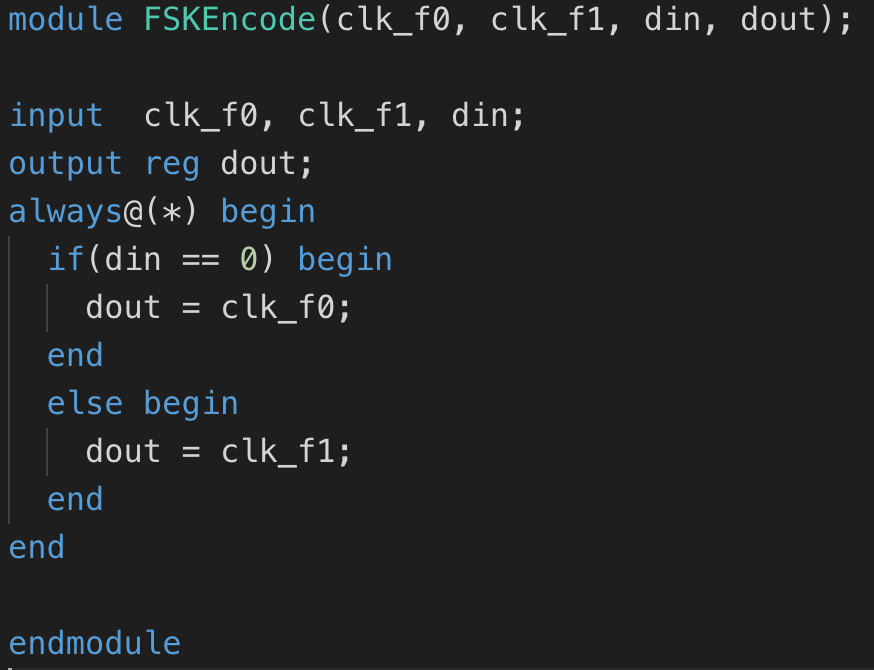
1. 接收



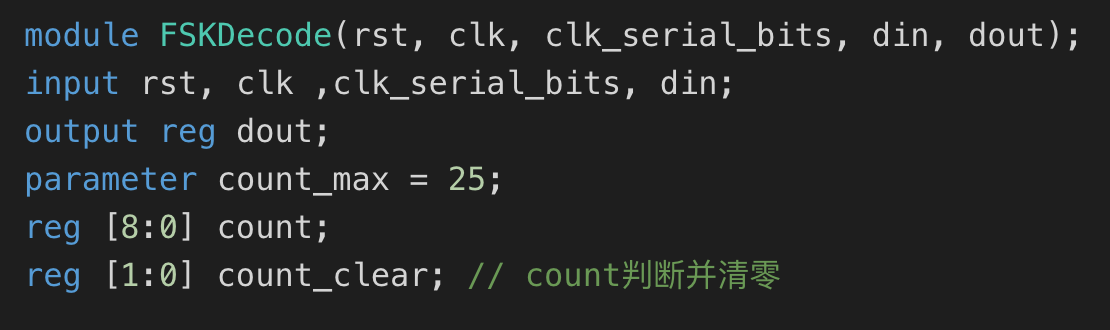
   

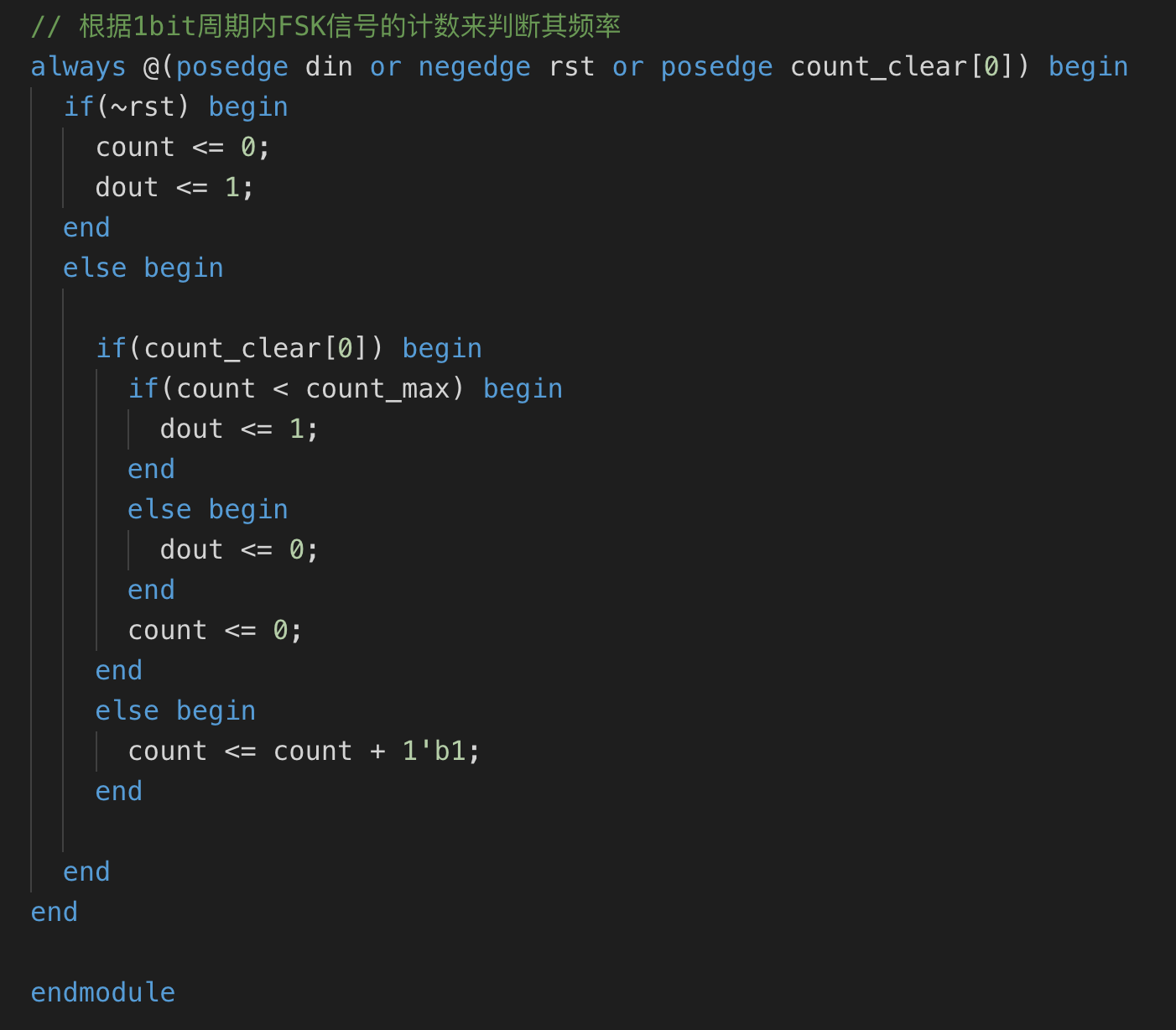
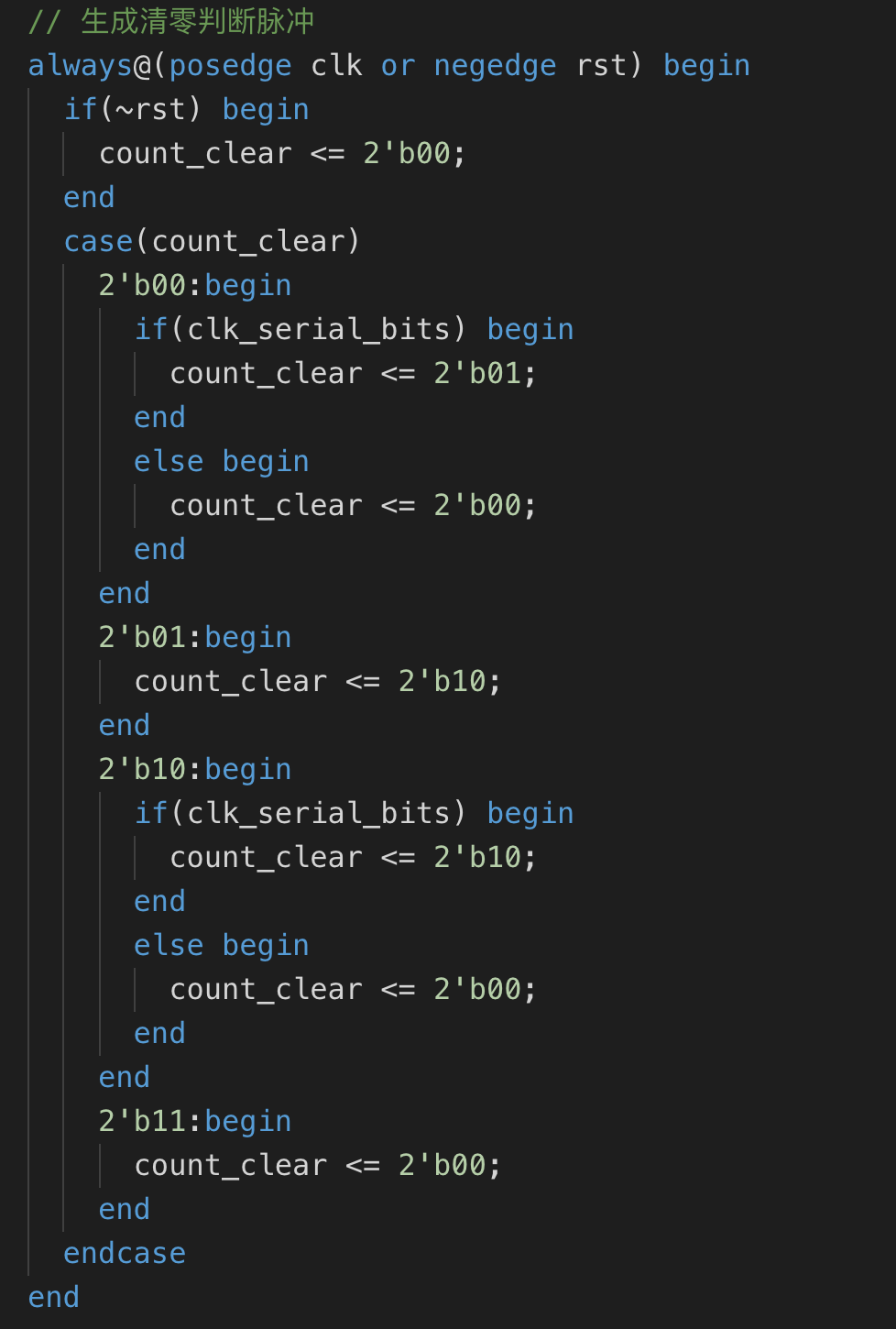
【FSK编解码】

1. FSK编码



1. FSK解码





1. 实验结果及分析

为了方便展示，将上排灯设为输入，下排灯设为输出，最右侧为低位，输入到输出之间经过了我们所设计的一套信道传输系统，以下为烧入板子后的显示结果及分析。由于FSK调制解调存在1bit延时，因此输出总是滞后于输入。

* 1. 输入：8’b00000010 输出：8’b00000001



* 1. 输入：8’b00000100 输出：8’b00000010



* 1. 输入：8’b00001000 输出：8’b00000100



* 1. 输入：8’b00010000 输出：8’b00001000



* 1. 输入：8’b00100000 输出：8’b00010000



* 1. 输入：8’b01000000 输出：8’b00100001



* 1. 输入：8’b10000000 输出：8’b01000010



* 1. 输入：8’b00000001 输出：8’b10000000



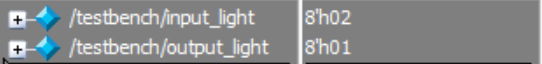
1. 信道噪声仿真分析

将Hamming码编码输出中的某一位置反，模拟信道噪声带来的1bit误差

wire [11:0] hammingEncodeWord\_Error;

assign hammingEncodeWord\_Error = {hammingEncodeWord[11:8], ~hammingEncodeWord[7], hammingEncodeWord[6:0]};

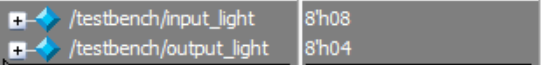
* 1. 输入：8’b00000010 输出：8’b00000001



* 1. 输入：8’b00000100 输出：8’b00000010



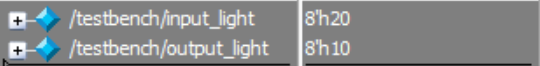
* 1. 输入：8’b00001000 输出：8’b00000100



* 1. 输入：8’b00010000 输出：8’b00001000



* 1. 输入：8’b00100000 输出：8’b00010000



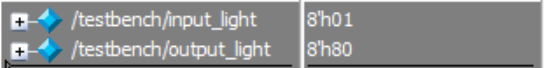
* 1. 输入：8’b01000000 输出：8’b00100001



* 1. 输入：8’b10000000 输出：8’b01000010



* 1. 输入：8’b00000001 输出：8’b10000000



由以上结果可以看出，Hamming码具有1bit纠错能力。

1. 小结与心得
2. 改进