

## 实验二 基于 FPGA 的 CMI 编译码

### 一. 实验目的

1. 熟悉指定序列 NRZ 码生成原理以及光纤线路 CMI 编译码原理。
2. 初步熟练 Altera 公司 QUARTUS II 仿真平台的使用。
3. 进一步熟悉数字电路设计技巧。
4. 基本掌握 Altera 公司 FPGA 设计与仿真方法。
5. 了解 FPGA 功能的物理验证。

### 二. 实验环境

1. QUARTUS 13.1
2. 示波器
3. FPGA 开发板
4. 芯片型号: Cyclone EP3C25E144C8

### 三. 实验基本原理

#### 1. NRZ 码

NRZ (Non-Return-to-Zero) 码是一种常用的数字编码方式。在 NRZ 编码中, 二进制数据 0 和 1 分别用两种不同的电平表示, 通常为高电平和低电平。在一个比特周期内, 电平保持不变。NRZ 编码简单易行, 但存在一定的缺陷, 如长时间的同一电平会导致接收端难以判断比特边界。

#### 2. CMI 编码

CMI (Coded Mark Inversion) 编码是一种常用的线路编码技术, 可以克服 NRZ 编码的缺陷。CMI 编码规则如下:

- 0 码: 用 01 表示, 即前半个比特周期为高电平, 后半个比特周期为低电平。
- 1 码: 用 00 或 11 交替表示, 即第一个 1 用 00 表示, 第二个 1 用 11 表示, 第三个 1 又用 00 表示, 以此类推。

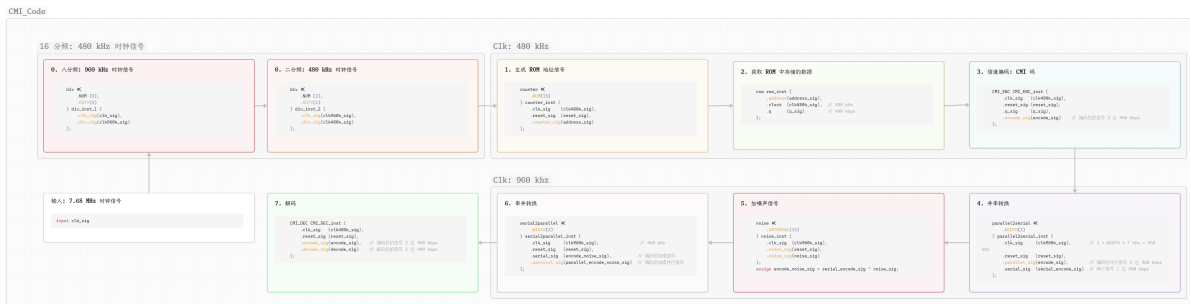
#### 3. CMI 译码 (帧同步)

CMI 译码过程需要进行帧同步, 以正确识别数据中的 0 和 1。帧同步的目的是确定比特边界, 以便正确解码数据。常见的帧同步方法有:

- 帧同步字: 在数据帧的开头添加一个特定的同步字, 接收端通过检测同步字来确定帧边界。
- 电平跳变: 利用 CMI 编码的特点, 通过检测电平跳变来确定比特边界。

## 四. 实验流程设计

### 流程图



### 业务代码

### 编码模块

```
module CMI_ENC (
    input clk_sig,
    input reset_sig, // 复位信号 低电平有效
    input q_sig,      // 信息信号

    output reg [1:0] encode_sig // 编码后信号
);
    reg [0:0] flag_sig;

    initial begin
        encode_sig <= 2'b01;
        flag_sig    <= 1'b0;
    end

    always @(posedge clk_sig) begin
        if (!reset_sig) begin
            flag_sig    <= 1'b0;
            encode_sig <= 2'b01;
        end else begin
            if (q_sig) begin
                flag_sig    <= ~flag_sig;
                encode_sig <= {~flag_sig, ~flag_sig};
            end else begin
                encode_sig <= 2'b01;
            end
        end
    end

endmodule
```

## 解码模块

```

module CMI_DEC (
    input      clk_sig,
    input      reset_sig, // 复位信号 低电平有效
    input [1:0] encode_sig, // 编码后信号

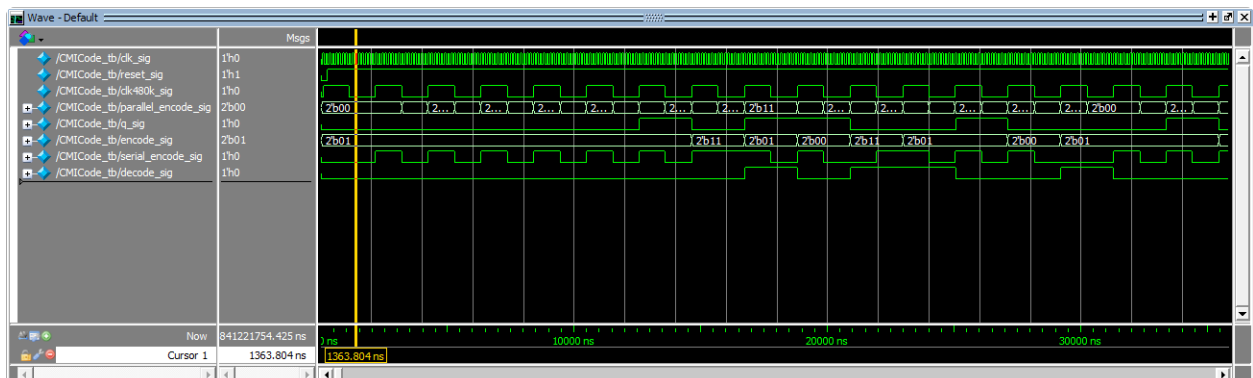
    output reg decode_sig
);
    initial begin
        decode_sig <= 1'b0;
    end

    always @(posedge clk_sig) begin
        if (!reset_sig) begin
            decode_sig <= 1'b0;
        end else begin
            case (encode_sig)
                2'b00: decode_sig <= 1'b1;
                2'b01: decode_sig <= 1'b0;
                2'b10: decode_sig <= 1'b0;
                2'b11: decode_sig <= 1'b1;
                default: decode_sig <= 1'b0;
            endcase
        end
    end
endmodule

```

## 五. 实验测试结果及分析计算

## 实验仿真结果



## 结果分析

## 1. 输入

十进制：1425

二进制：

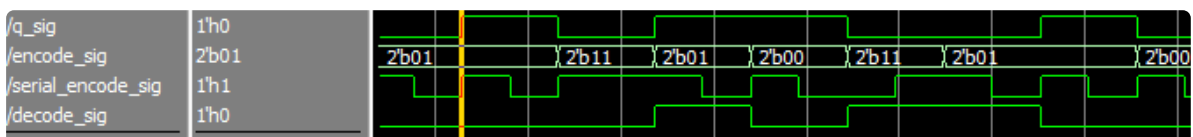
```
15'b0000101110010001
```

## 2. 信息速率



1 bit 持续时长大约为  $14639 - 12559 = 2080\text{ns}$ ，即  $2.08\text{ }\mu\text{s}$ ，对应信息速率为  $480.77\text{ kbps}$ ，考虑到估读误差的存在，可以认为我们产生了信息速率为  $480\text{ kbps}$  的信源。

## 3. 编解码分析



1. `q_sig` 为信源，图中产生的序列为 `01011001` 对比可知是 15 位学号的一部分；
2. `encode_sig` 为编码输出，`q_sig` 0 输出 `01`，1 交替输出 `00`，`11`，符合编码设计；
3. `serial_encode_sig` 为并串转换后串行编码信号；
4. `decode_sig` 为译码输出信号，对比 `q_sig` 可知，译码输出正确。

## 六. 实验中遇到的问题及解决方法

## 七. 心得体会

## 八. 参考文献

1. 光通操作指导 202403
2. 光通信实验 20240329