

数字逻辑实验三 | 波形分析

200110617 蔡嘉豪

一、概述

题目要求使用时钟 IP 和存储器 IP 实现 16 位 LED 灯从右往左依次点亮，直到全亮，按复位键全灭，再次按启动键，重复上述过程。

存储器 IP 名为 `led_mem`，负责存储深度 16、宽度 16 的数据（16 个地址对应灯的 16 个状态，每个状态有 16 个灯的电平高低描述，为 16 bits 长度的二进制串）

时钟 IP 分频为 10 MHz，也就是说导出的时钟 `clk_g` 是频率为 10 MHz 的时钟

二、总思路

先一次性写入 16 个 16 bits 的二进制串，再逐秒读出赋值给 `led`

三、状态定义和转换

定义 4 种状态，分别为复位、待机、写数据、读数据

按复位键进入复位状态，完成状态初始化后，进入待机状态

待机状态按启动键进入写状态

```
1  reg [2:0] status;  
2  parameter DEVICE_RESET = 3'b000;  
3  parameter DEVICE_NOT_WORK = 3'b001;  
4  parameter DEVICE_WRITE = 3'b010;  
5  parameter DEVICE_READ = 3'b011;
```

四、写数据

定义了一个计数器 `write_cnt`，用于每 2 个时钟周期给出写信号

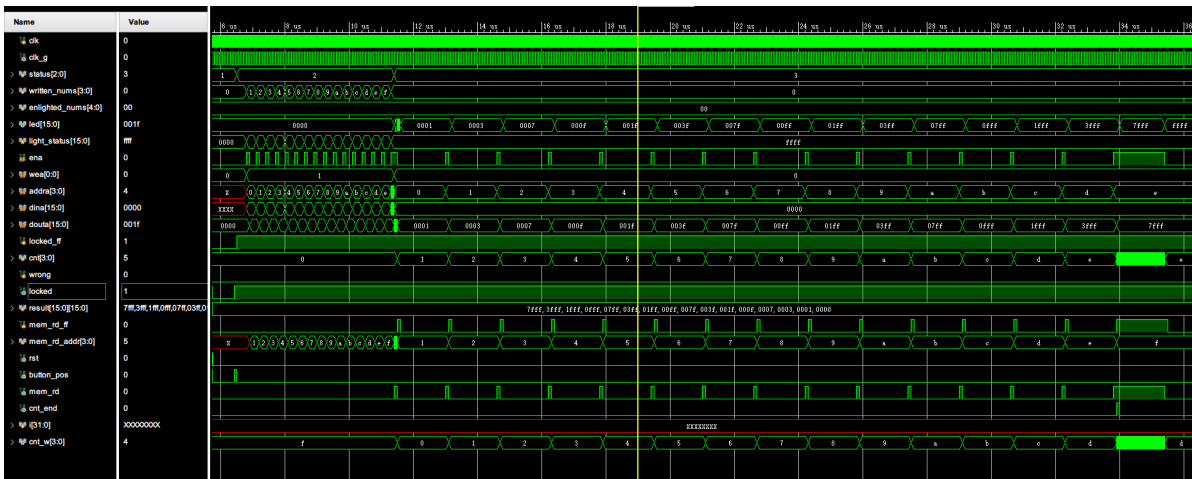
即每 2 个时钟周期写入一个灯的状态数据

五、读数据

定义计数器 `read_cnt`，每隔 1 秒给出读信号，改变 `addra` 并将 `douta` 中数据赋值给 `led`

最后一个灯稍微做一些特判即可

六、波形分析



此处仿真中给出读信号的周期为 15 个时钟周期

1.时钟

从图中可见，`clk_g` 与原时钟 `clk` 相比实现了无毛刺分频

2.读写分析

从图中可见，约 6.5 ns 到 11.4 ns，`status` 为 `DEVICE_WRITE` 为写数据状态，此阶段中存储器写信号 `wea` 一直拉高，但存储器使能信号 `ena` 只在需要写数据的时刻拉高（保证其不持续工作，降低能耗），经过该阶段，`addra` 从 0 到 15 都已经写入对应的数据

从 11.4 ns 之后 `status` 为 `DEVICE_READ` 为读数据状态，此阶段中，写信号 `wea` 一直为 0，存储器使能信号 `ena` 只在需要读数据的那一刻拉高，并且该阶段中，`led` 与 `douta` 保持同步，即亮灯的状态与读出的数据时刻保持一致