

实验题目：计数器和序列检测器的设计

班级：无 06

学号：2020010796

姓名：闫珺博

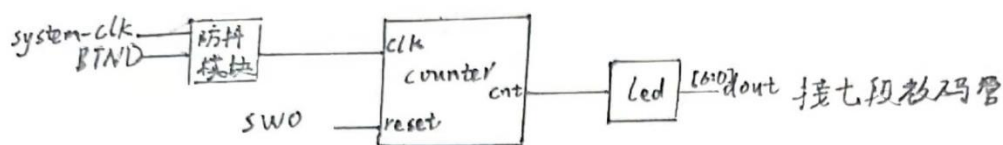
一、实验目的

1. 掌握简单时序逻辑电路的设计方法
2. 了解任意进制计数器的设计方法
3. 掌握有限状态机的实现原理和方法
4. 掌握序列检测的方法

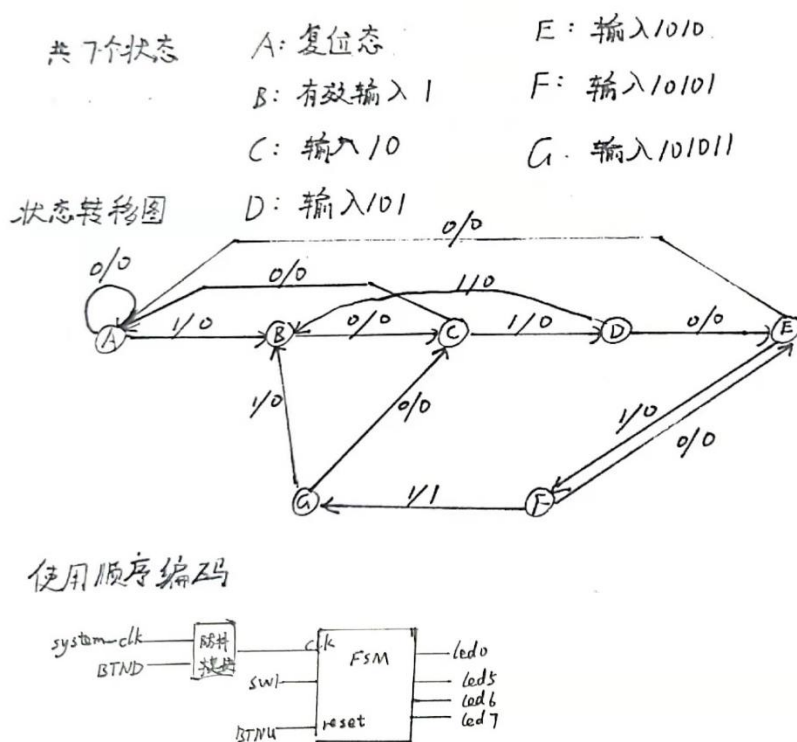
二、设计方案

1. 计数器

以按键 BTND 输入作为时钟，4bits 十进制计数器只需要从 4'b0000 计数到 4'b1001，超过置零，计数器从 0 到 9 唯一对应七段数码管显示 7bit，在计数的时序逻辑部分后面加上组合逻辑部分得到对应七段数码管信号输入



2. 有限状态机实现序列检测器



3. 移位寄存器和组合逻辑实现序列检测器

实际上是用六位存储输入的序列，每次改变储存序列都和要检测的序列进行比较，如果相等 led0 就输出高电平，点亮 LED0



三、 关键代码及文件清单

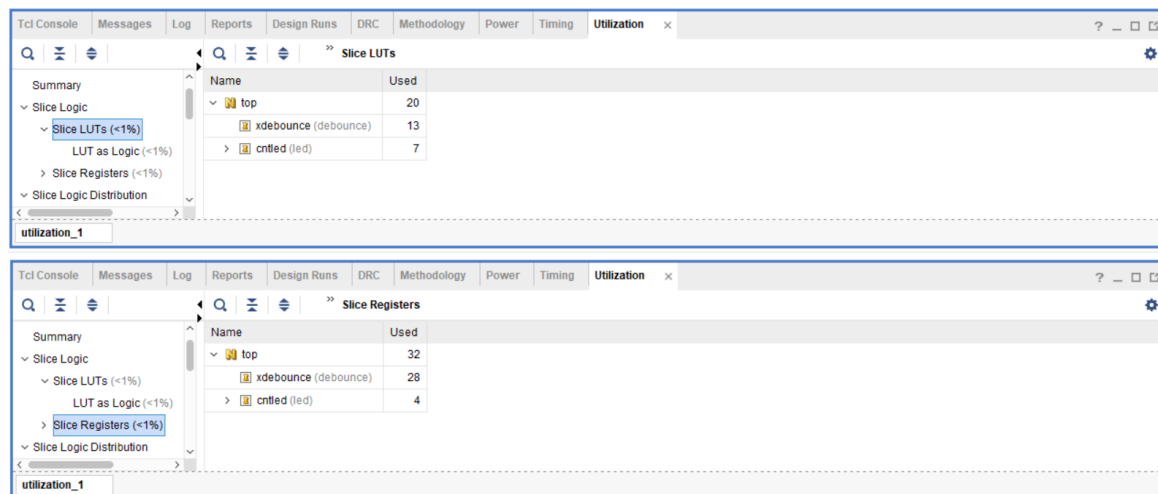
计数器(counter 文件夹)中, counter.v 中写了两个 module, 一个是负责计数的 counter, 另一个是负责根据内部例化的 counter 点亮对应灯的 led; debounce.v 是防抖模块; top.v 是顶层模块, 例化 led 和 debounce

有限状态机实现的序列检测器(fsm 文件夹)中, fsm.v 中只有一个 module fsm, 检测上升沿复位信号和上升沿时钟信号并进行状态改变; debounce.v 是防抖模块; top.v 是顶层模块, 例化 fsm 和 debounce

移位寄存器实现的序列检测器(shiftreg 文件夹)中, shiftreg.v 中只有一个 module shiftreg, 检测上升沿复位信号和上升沿时钟信号, 检测当前寄存器中存的数字是不是 101011; debounce.v 是防抖模块; top.v 是顶层模块, 例化 shiftreg 和 debounce

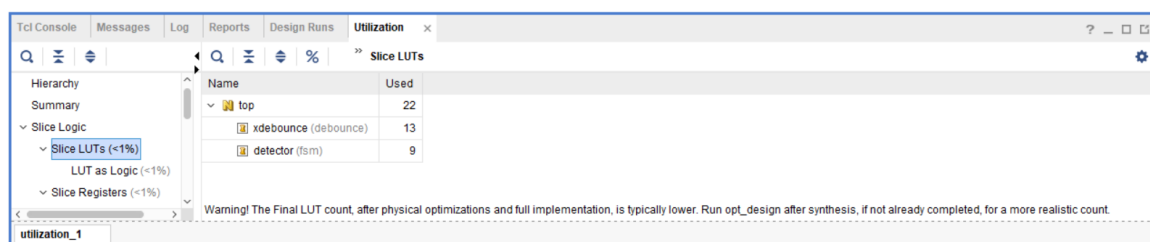
四、 综合情况(面积和时序性能)

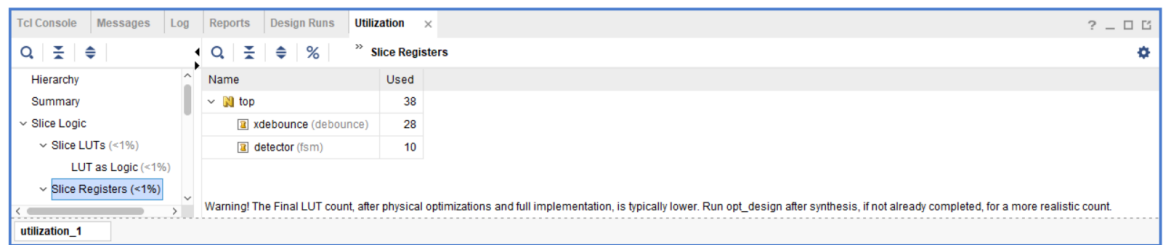
1. 计数器



cntled 占用了 7 个查找表, 4 个寄存器

2. 有限状态机实现序列检测器





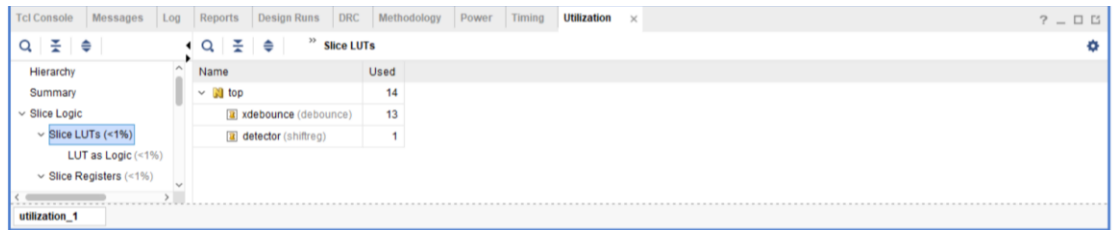
Name	Used
top	38
xdebounce (debounce)	28
detector (fsm)	10

Warning! The Final LUT count, after physical optimizations and full implementation, is typically lower. Run opt_design after synthesis, if not already completed, for a more realistic count.

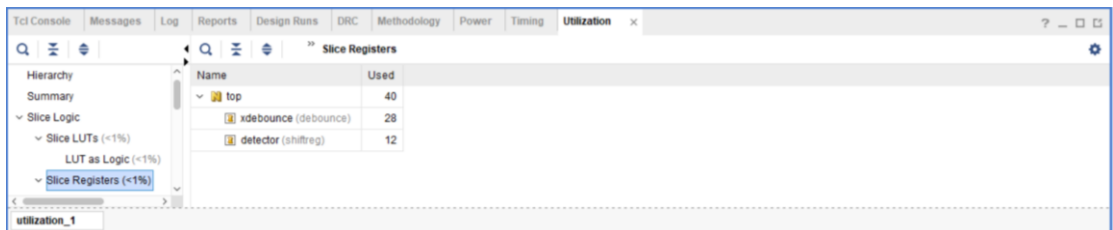
Detector

占用了 9 个查找表，10 个寄存器

3. 移位寄存器和组合逻辑实现序列检测器



Name	Used
top	14
xdebounce (debounce)	13
detector (shiftreg)	1



Name	Used
top	40
xdebounce (debounce)	28
detector (shiftreg)	12

detector 占用了 1 个查找表，12 个寄存器

4. 硬件调试情况

烧入以上代码能实现基本的功能，但是我在测试过程中发现如果按键按的太快板子会没有反应，我分析是因为加入了防抖模块，防抖模块把过快的按键按动和机械振动一起消除了，因此太快按会没有反应。