点亮数字人生

计83 李天勤 2018080106

一，实验目的

1. 熟悉VHDL语言，掌握引荐程序得白那些规范。掌握EDA软件使用方法和工作流程。
2. 用硬件描述语言设计出通过输入端控制数码管输出显示数字。
3. 设计一个不带译码的七段数码管和一个带译码的七段数码管。
4. 第一个任务：设计由四个独立按键控制的七段数码管。第二个任务：设计一个由clk控制的七端数码管。

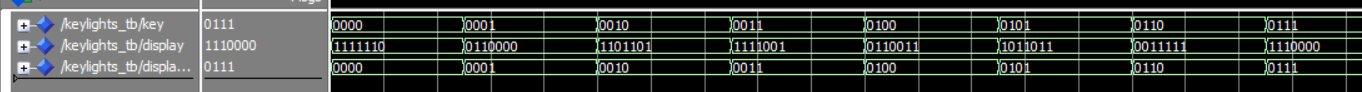
二，实验代码与原理分析

1. 第一个任务：由4个独立key 输入控制的七段数码管：

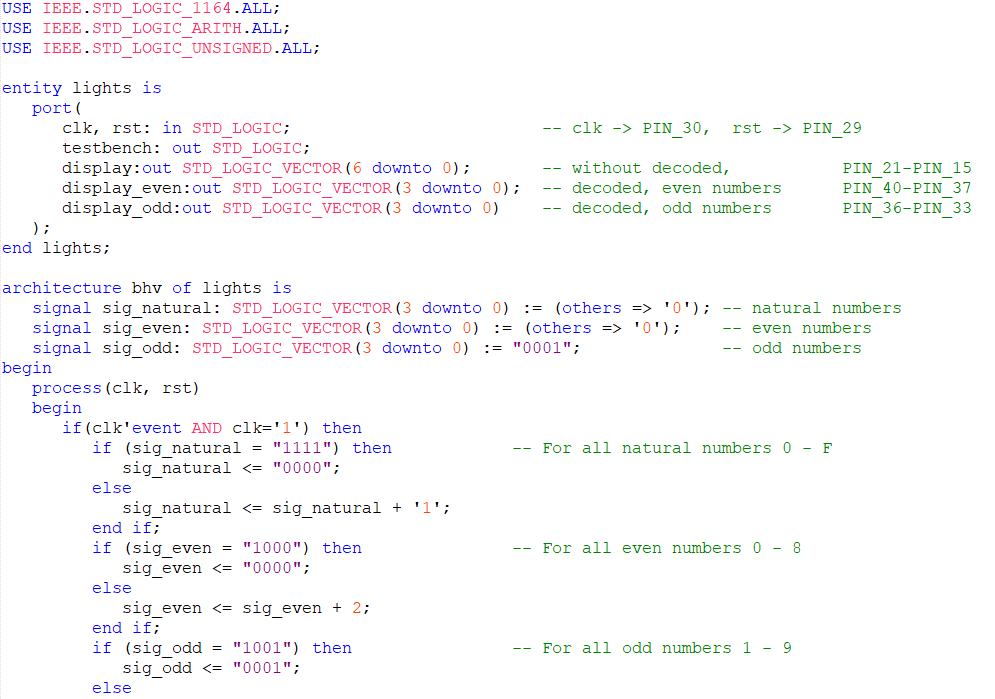
实验的这一部分非常容易。 描述该实验的文档末尾已经介绍了大多数代码。我唯一需要改变的部分就是b和d的4位二进制到7位二进制转换。它们是仅有的两个小写字母，因为大写字母B和D在七段数码管上太相似了8和0。

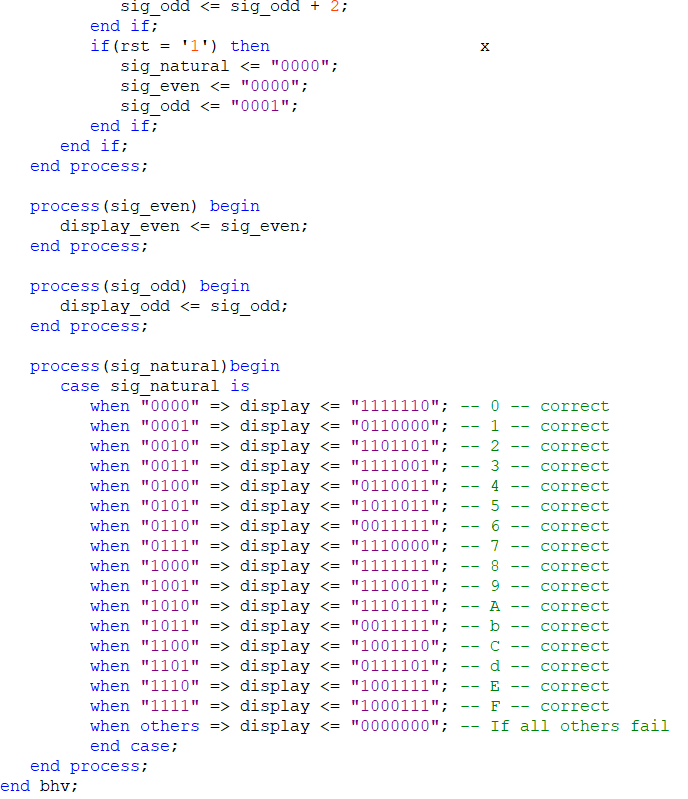
通过testbench，很容易看出写的代码按预期操作。

TESTBENCH



1. 第二个任务：设计一个数码管，显示数列（奇数，偶数，自然数）。可以利用实验平台上所有接口和按键（包括clk和rst）

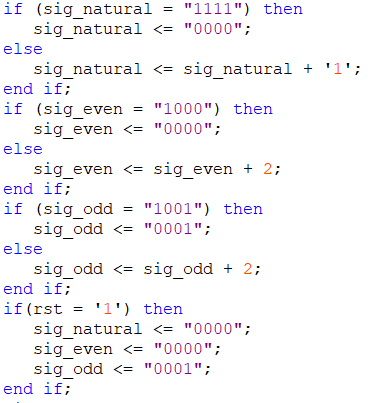




与第一个程序不同，第一个程序使用4个单独的输入来模拟“键”以根据用户的输入来点亮数字管，而该源程序则使用clk来创建一个自动计数器。 该程序使用三种不同的七段数码管显示，一种用于奇数，一种用于偶数，一种用于所有数字（0到F）。

这个程序有三个输出，display对应不带译码的七段数码管，所以有它可以存7个bit，对应七段数码管得七个部分，对应a,b,c,d,e,f,g. a表示高位，g表示最低位。display\_even和display\_odd 对应两个带译码的七段数码管。所以就需要存4个bit，对应D，C，B，A ，或 BCD码的8421。

这个程序有两个输入，clk和rst。 rst表示“reset” 当其值变为“1”的时，它把三个七段数码管变成它们最低的值。例如，他把display和display\_even变成“0000” 或0，也把display\_even变成“0001”或1. 然后在下一次clk事件中，数字显示器将继续计数。

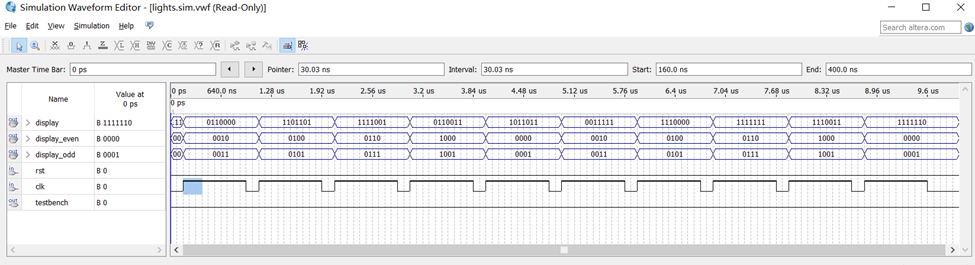
此部（左图）分控制程序的“计数器”部分。 此过程由clk控制。 并且对于每个时钟周期，每个数码管器的数量都会增加。 奇数和偶数计数器每次都将增加2，而自然数计数器将每次增加1。

最后一个过程，负责BCD 的8421码根第一部分一样的。

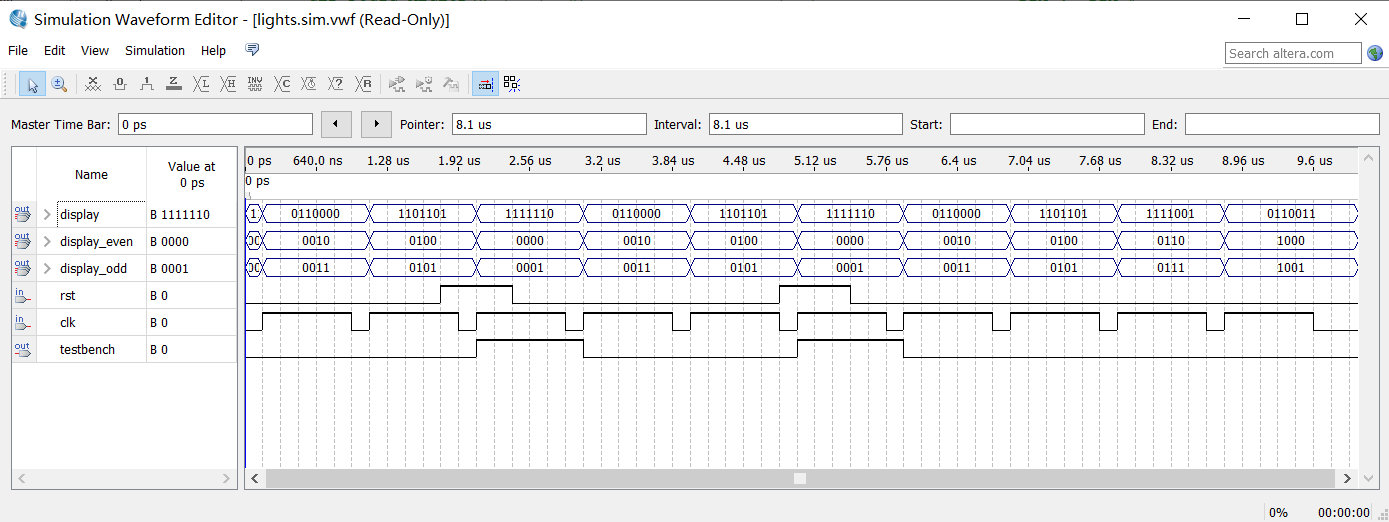
下面包括waveform和testbench文档。

Waveform file 和 Testbench

Edit → Set End Time: 10μs

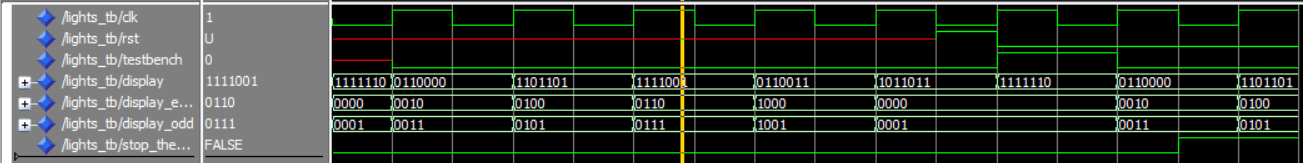
第一个波形文件不包括rst信号的使用, 可以看出程序逻辑时正确的。在第5个时钟cycle, display\_even和display\_odd的值分别返回其基本值“ 0000”和“ 0001”。display, 从0到F的所有自然数的显示，继续计数。

第二个波形包括rst信号使用，可以看出rst信号等于“1”的时候，下一个时钟cycle, display，display\_even, 和display\_odd 的值返回初始值。

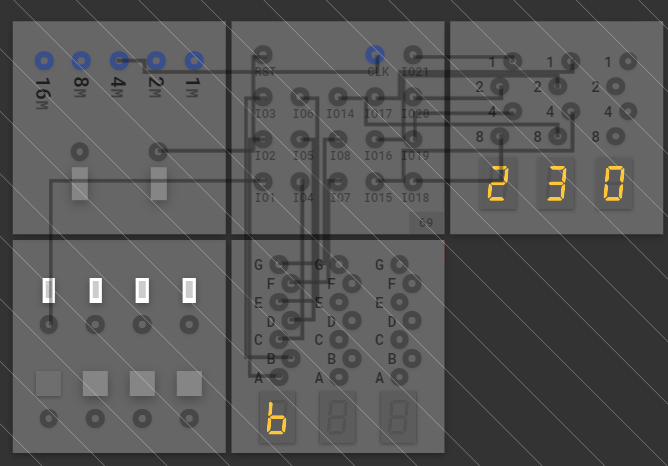


ModelSim-Altera TestBench

Testbench也



JieLabs



三，实验结论

由于该实验相对简单，其主要目的是引入VHDL，并使我们对EDA软件更加熟悉，因此在该实验室的编程方面不会发生很多错误或复杂情况。但是，花了很多时间和精力才能完成第一个实验。阅读和分析“数字逻辑实验指导”文档花费了很多时间；建立EDA软件并熟悉Quartus2平台也花费了一些时间。学习VHDL的基础知识并不太困难，因为它的结构例如“ for”循环和while循环的用法类似于先前学习的编码语言。

但是，在VHDL中，与其他顺序编程语言不同，“进程”（类似于其他语言中的函数）并行运行。分配不再立即发生，而是在流程或周期结束时发生。

尽管在这种相对简单的逻辑设计中，测试台和波形的使用并不十分重要，但我看到了它的重要性，并且相信它将对以后的实验有所帮助。