|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **学 期** | 2022-2023学年第1学期 | | | | **实验日期** | | 2023/2/21 | | |
| **学 院** | 信息学部 | | | | **专 业** | | 计算机科学与技术（实验班） | | |
| **班 级** | 210710 | | **学 号** | | 21071003 | | **姓 名** | | 高立扬 |
| **组 号** | 43 | | **学 号** | | 21071004 | | **姓 名** | | 石昊阳 |
| **评 阅 内 容** | | | | | | | | | |
| **任务一** | | **任务二** | | **总结** | | **格式** | | **成 绩** | |
|  | |  | |  | |  | |  | |
|  | | | | | | | | | |
| **题 目** | 实验5：简单接口电路设计 | | | | | | | | |

1. **实验目的**

通过本实验内容的学习，掌握如何利用基本组合逻辑电路和基本时序电路解决实际问题；掌握常用简单接口电路的工作原理及控制方法，为后续综合设计的实现做好准备；学习在数字电路设计中，按模块设计电路的思想。

1. 通过基本逻辑电路的应用，掌握键盘扫描电路、动态数码管显示、直流电机控制、步进电机控制等电路的设计；学会使用Verilog HDL设计简单接口的控制器。

2. 掌握在EDA软件中利用图形方式和硬件描述语言进行模块调用的方法。

1. **任务一设计与实现**

1. 要求：

(1)系统时钟为50MHz，使用之前设计的分频器模块。

(2)完成动态扫描模块的设计。

(3)将个人学号显示在动态数码管上。（需要8个数码管，轮流点亮8个数码管）

2.设计思路

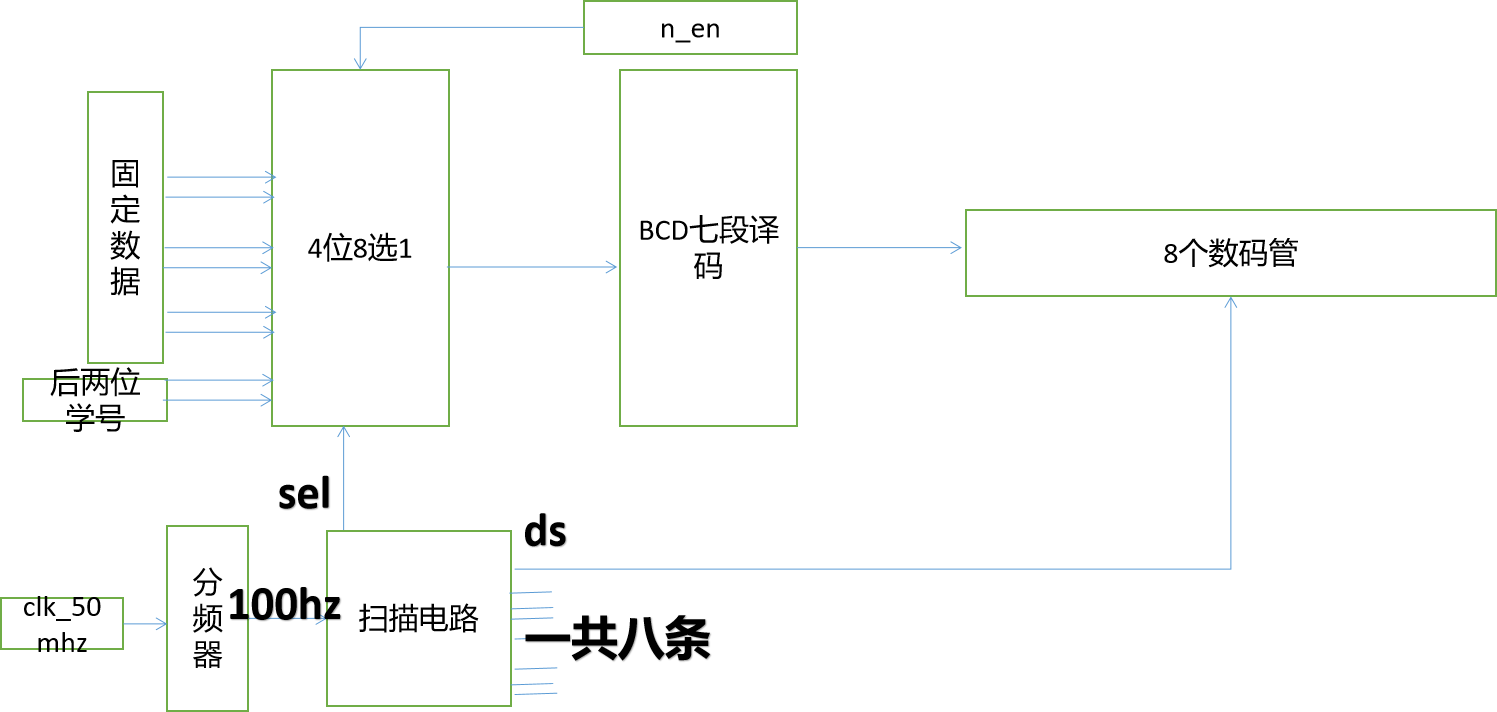
任务一要求设计一个能同时在数码管上显示出8位学号的电路。通过分析，我们发现可以将学号作为输入输入进4位8选1器中，再用扫描电路选择4位8选1的输出，并且决定某段数码管显示，再用bcd七段译码器将4位8选1的输出转换为数码管上的显示的数字，即可实现该电路。

图1.设计思路

1. 详细设计

表1.扫描电路

|  |  |
| --- | --- |
| Sel | Ds |
| 0 | 1111\_1110 |
| 1 | 1111\_1101 |
| 2 | 1111\_1011 |
| 3 | 1111\_0111 |
| 4 | 1110\_1111 |
| 5 | 1101\_1111 |
| 6 | 1011\_1111 |
| 7 | 0111\_1111 |

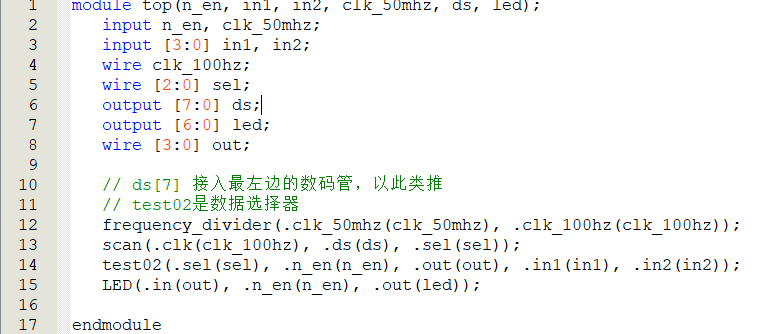
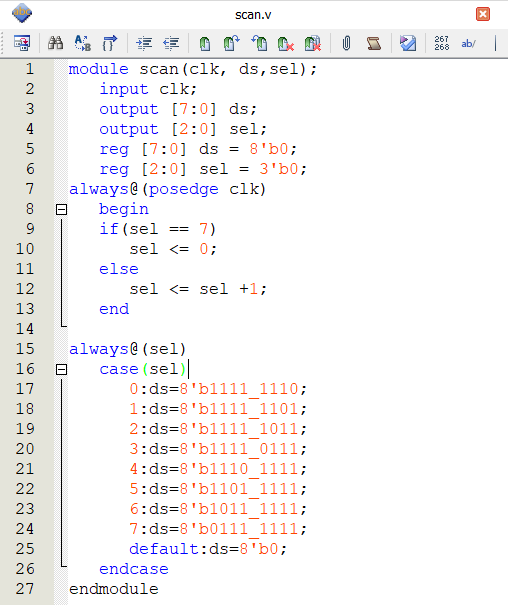


图2.扫描电路以及顶层文件

4.仿真验证

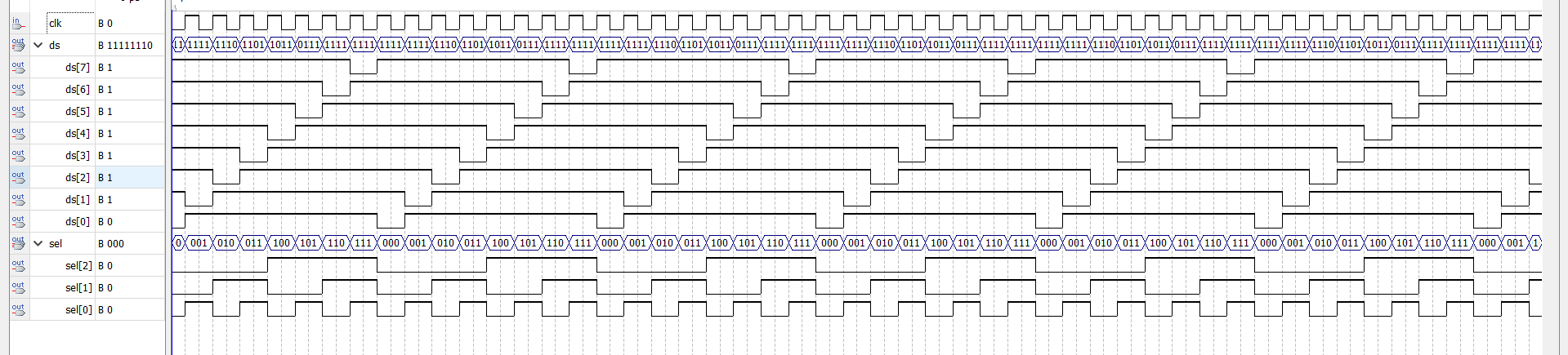


图3.扫描电路波形图

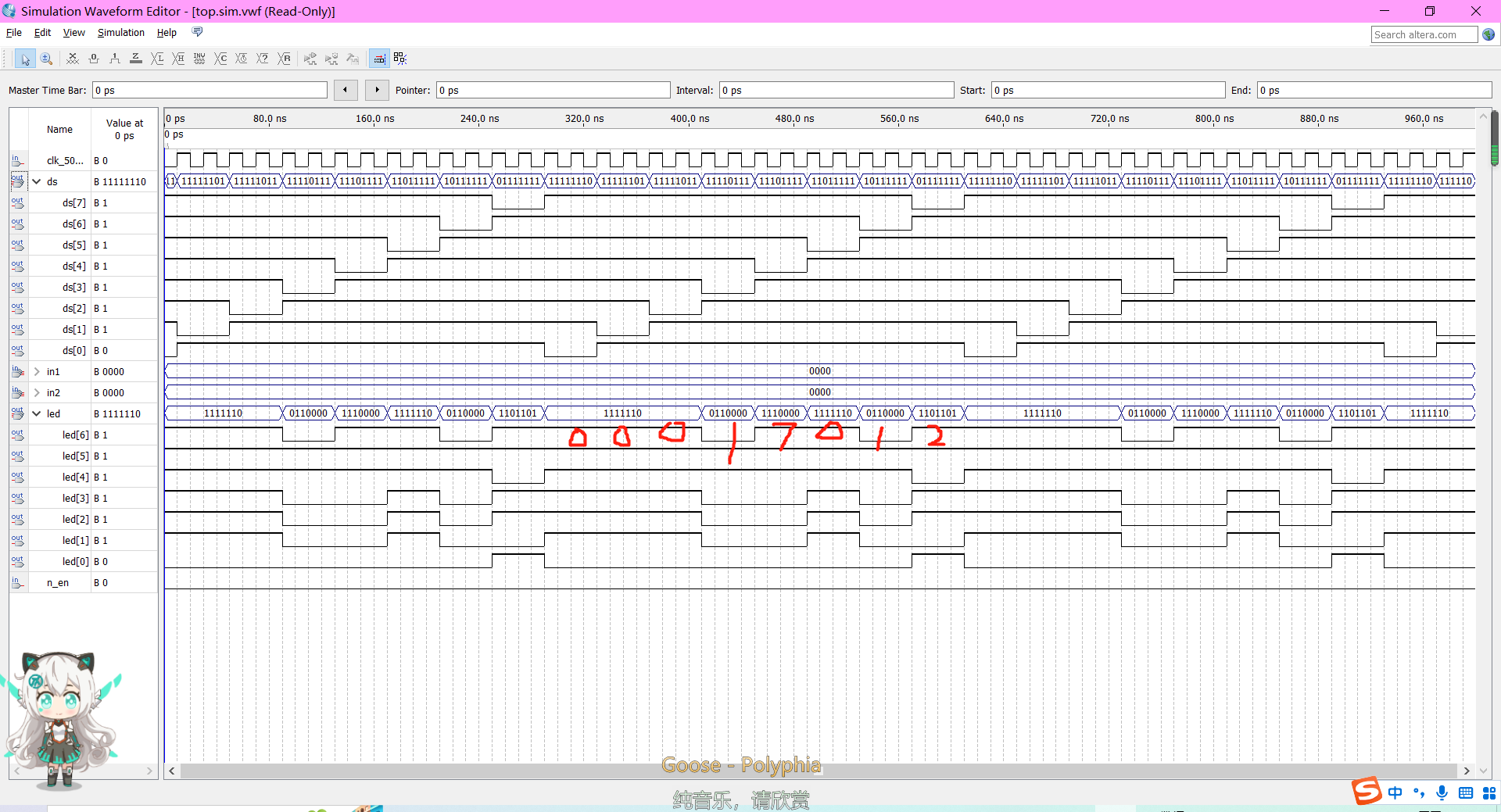


图4.顶层文件波形图

如图3可知ds和sel的扫描电路准确无误。由图4.，可找出初始学号被正确打印。

5.引脚分配

表2.引脚分配

|  |  |  |
| --- | --- | --- |
| 端口名称 | 输入端 | |
| 时钟信号clk | n\_en |
| 引脚编号 | T1 | A3 |
| 平台端口 | T1 | SW16 |

表2.续表

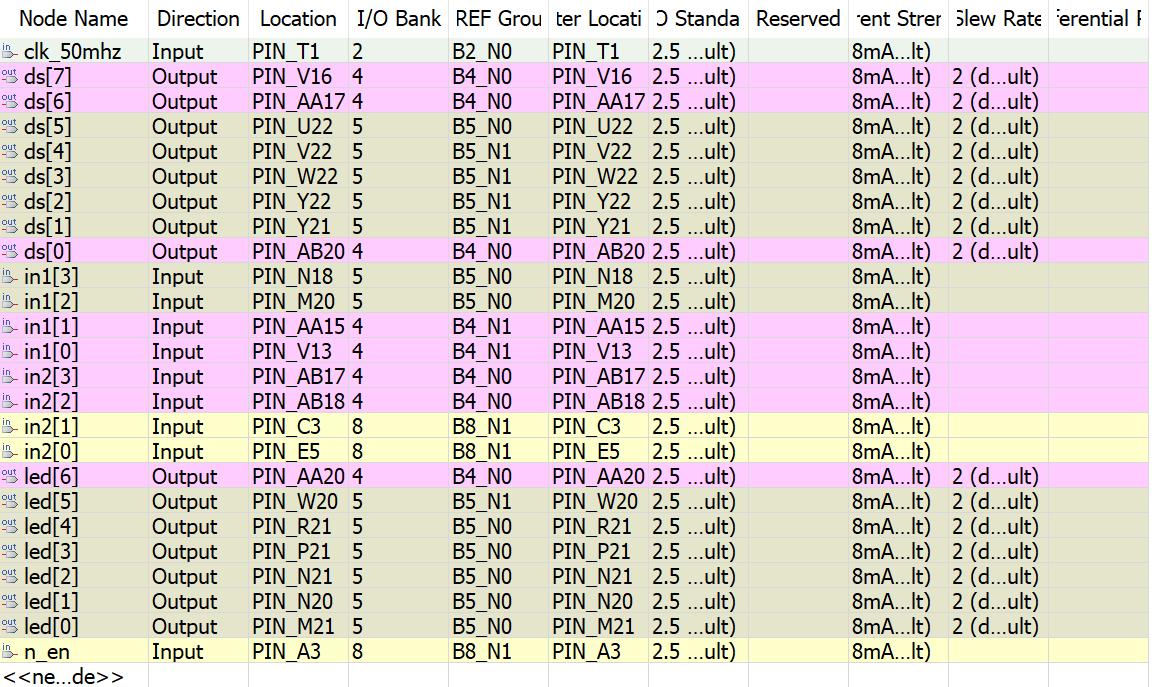
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 输入端 | | | | | | | |
| in1 | | | | in2 | | | |
| in1[3] | in1[2] | in1[1] | in1[0] | in2[3] | in2[2] | in2[1] | in2[0] |
| N18 | M20 | AA15 | V13 | AB17 | AB18 | C3 | E5 |
| SW1 | SW2 | SW3 | SW4 | SW9 | SW10 | SW11 | SW12 |

表2.续表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 输出端 | | | | | | | |
| ds | | | | | | | |
| ds[7] | ds[6] | ds[5] | ds[4] | ds[3] | ds[2] | ds[1] | ds[0] |
| V16 | AA17 | U22 | V22 | W22 | Y22 | Y21 | AB20 |
| DS8 | DS7 | DS6 | DS5 | DS4 | DS3 | DS2 | DS1 |

表2.续表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 输出端 | | | | | | |
| led | | | | | | |
| led[6] | led[5] | led[4] | led[3] | led[2] | led[1] | led[0] |
| AA20 | W20 | R21 | P21 | N21 | N20 | M21 |
| LA | LB | LC | LD | LE | LF | LG |



6.实验现象

LED屏上从左到右依次输出21071000。若改变输入端，输出的学号也会随输入端的改变而改变。如果改变en开关，LED屏被关闭。与实验预期相符。

1. **任务二设计与实现**
2. 要求

(1)系统时钟为50MHz，使用之前设计的分频器模块。

(2)实现4×4键盘识别。

(3)将按键显示在数码管上。（只需要1位数码管，按下某个键时数码管显示对应数字）

1. 设计思路

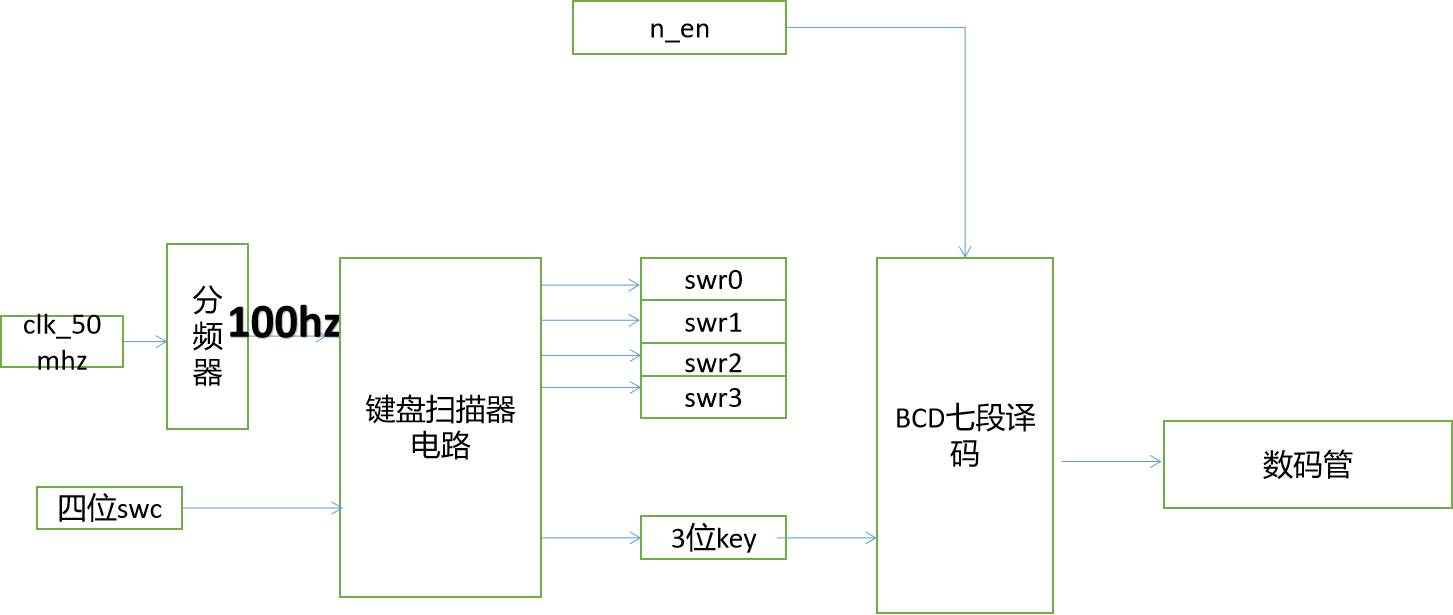
任务二要求设计一个用4\*4键盘识别的数码管显示器。通过分析发现，该任务难点在与如何判断键盘上哪个按键被摁下。我们发现可以将键盘分别以行和列分组，当按键被摁下，其对应的行线和列线变为低电平。用高频扫描其状态即可判断哪一按键被摁下。

图5.设计思路

1. 详细设计

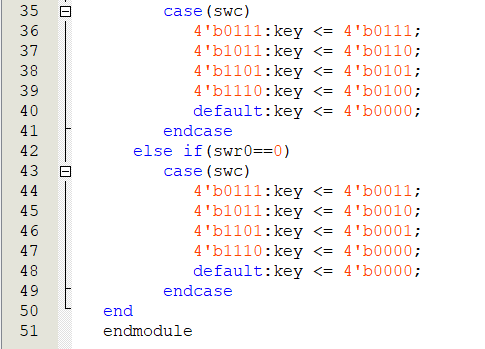
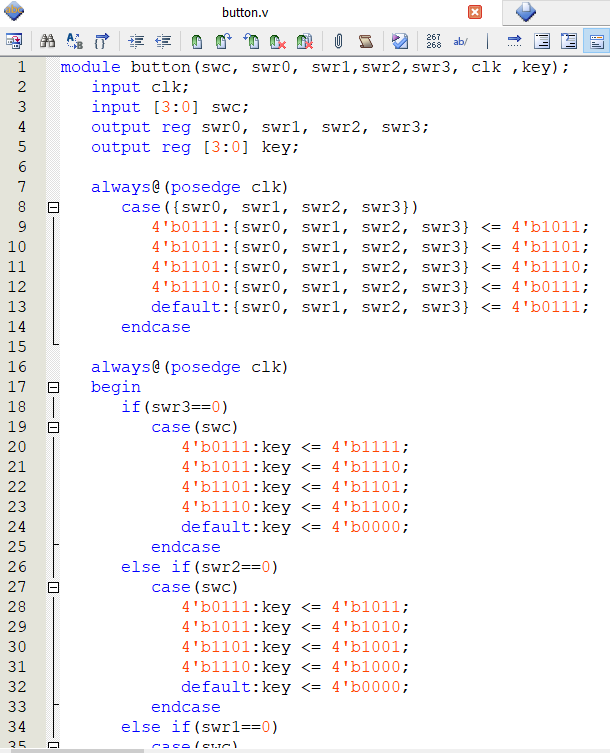


图6.键盘扫描电路

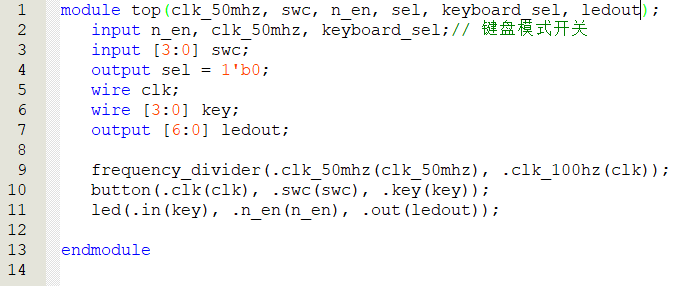


图7.顶层文件

表3.键盘扫描

|  |  |  |
| --- | --- | --- |
| Swr[3:0]低->高位（in） | Swc[3:0]高->低位（out） | Key（out）高->低位 |
| 0111 | 1110 | 0000 |
| 1101 | 0001 |
| 1011 | 0010 |
| 0111 | 0011 |
| 1011 | 1110 | 0100 |
| 1101 | 0101 |
| 1011 | 0110 |
| 0111 | 0111 |
| 1101 | 1110 | 1000 |
| 1101 | 1001 |
| 1011 | 1010 |
| 0111 | 1011 |
| 1110 | 1110 | 1100 |
| 1101 | 1101 |
| 1011 | 1110 |
| 0111 | 1111 |

1. 仿真验证

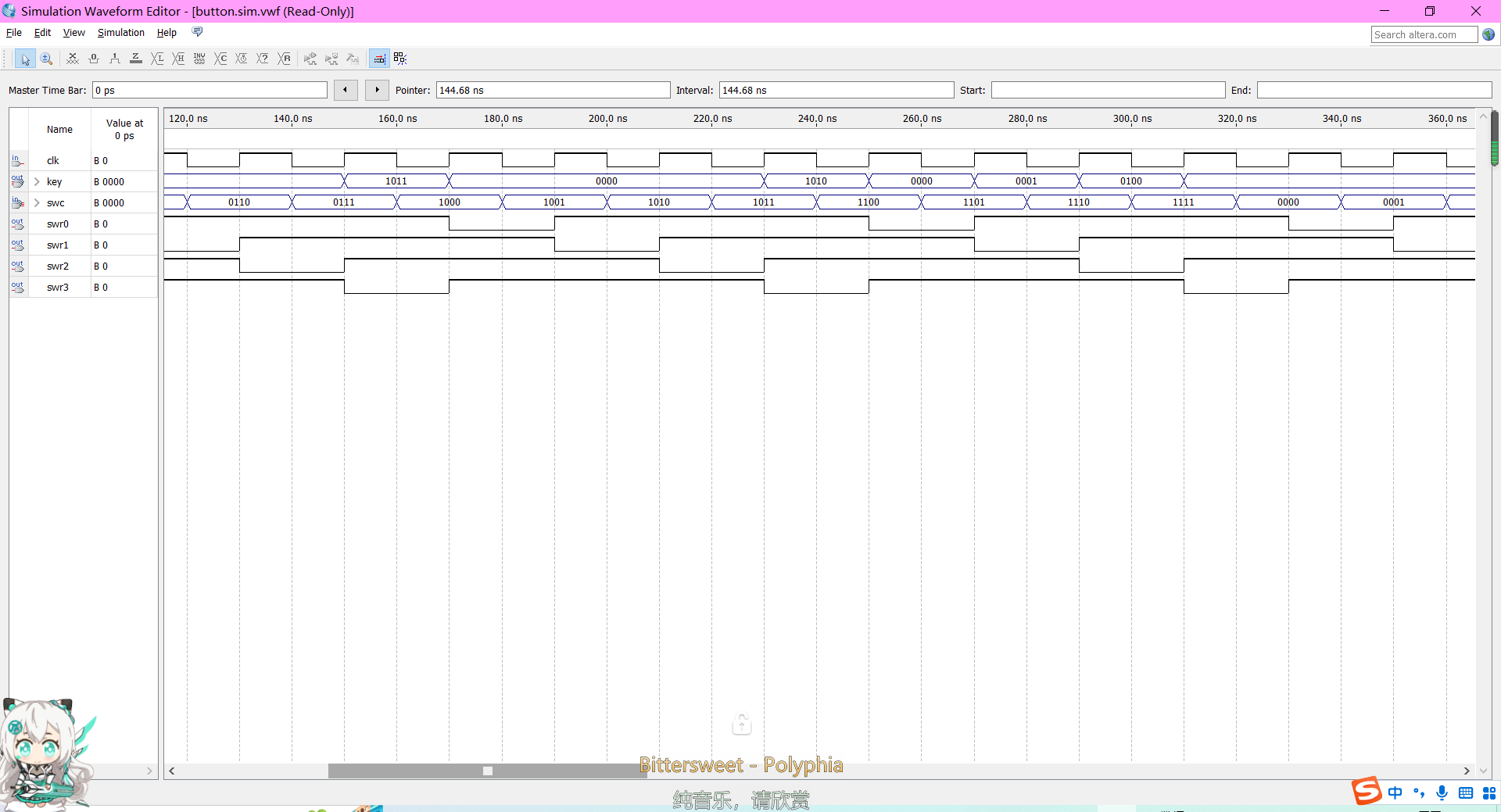


图8.波形图

由图8.，再根据代码所写和3.中的功能表，可以看出键盘扫描得到的结果准确无误，与键盘上数字对应关系一致。

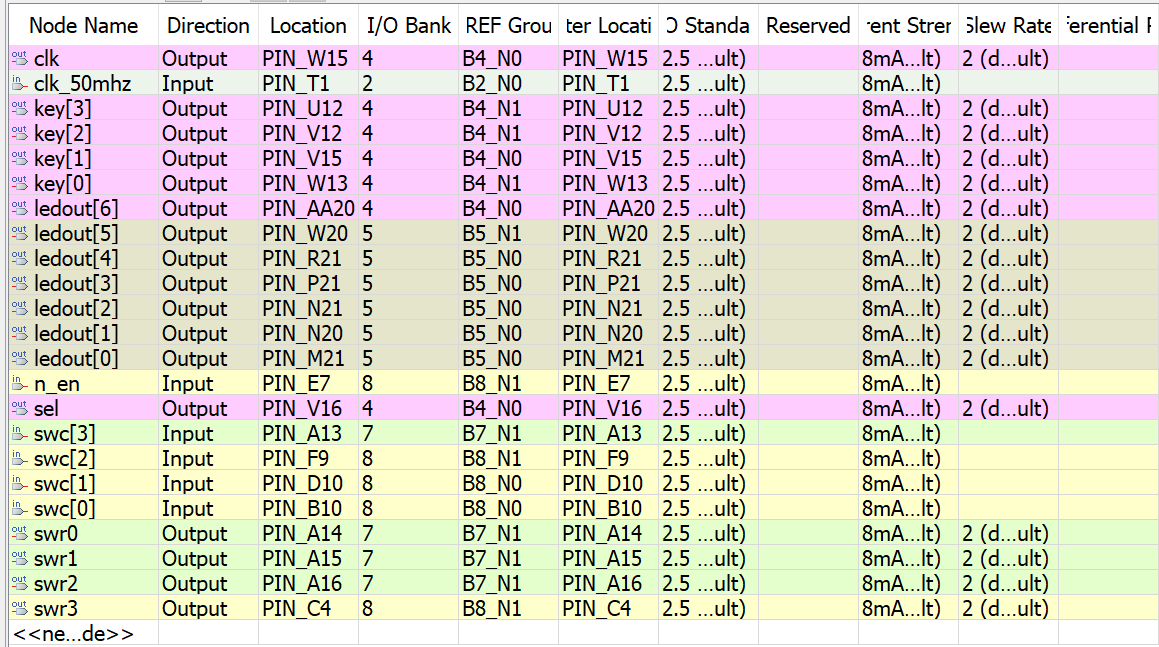
1. 引脚分配

表4.引脚分配

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 端口名称 | 输入端 | | | | | | |
| swc | | | | n\_en | clk | sel |
| swc[3] | Swc[2] | Swc[1] | Swc[0] |
| 引脚编号 | A13 | F9 | D10 | B10 | E7 | T1 | V16 |
| 平台端口 | SWC3 | SWC2 | SWC1 | SWC0 | SW7 | T1 | DS8 |

表4.续表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 输出端 | | | | | | | | | |
| led | | | | | | | | | |
| led[6] | led[5] | | led[4] | led[3] | | led[2] | led[1] | | led[0] |
| AA20 | W20 | | R21 | P21 | | N21 | N20 | | M21 |
| LA | LB | | LC | LD | | LE | LF | | LG |
| Swr0 | | Swr1 | | | Swr2 | | | Swr3 | |
| A14 | | A15 | | | A16 | | | C4 | |
| Swr0 | | Swr1 | | | Swr2 | | | Swr3 | |



1. 实验现象

摁下键盘的某一按钮，其对应的数字可以在LED屏上显示出来。如果改变en开关，LED屏被关闭。与实验预期相符。

1. **扩展实验**

1.设计思路

我们希望将任务一和二结合起来，设计一个可以使用户通过键盘扫描，自行决定数码管后两位的学号，于是我们将任务一和二相连，稍作修改，添加了num\_sel来决定键盘输入的是两位中的哪一位，新增了key1和key2寄存器分别负责记录in1和in2，修改case中的default使得不按下键盘的时候，key保留上一次的输入

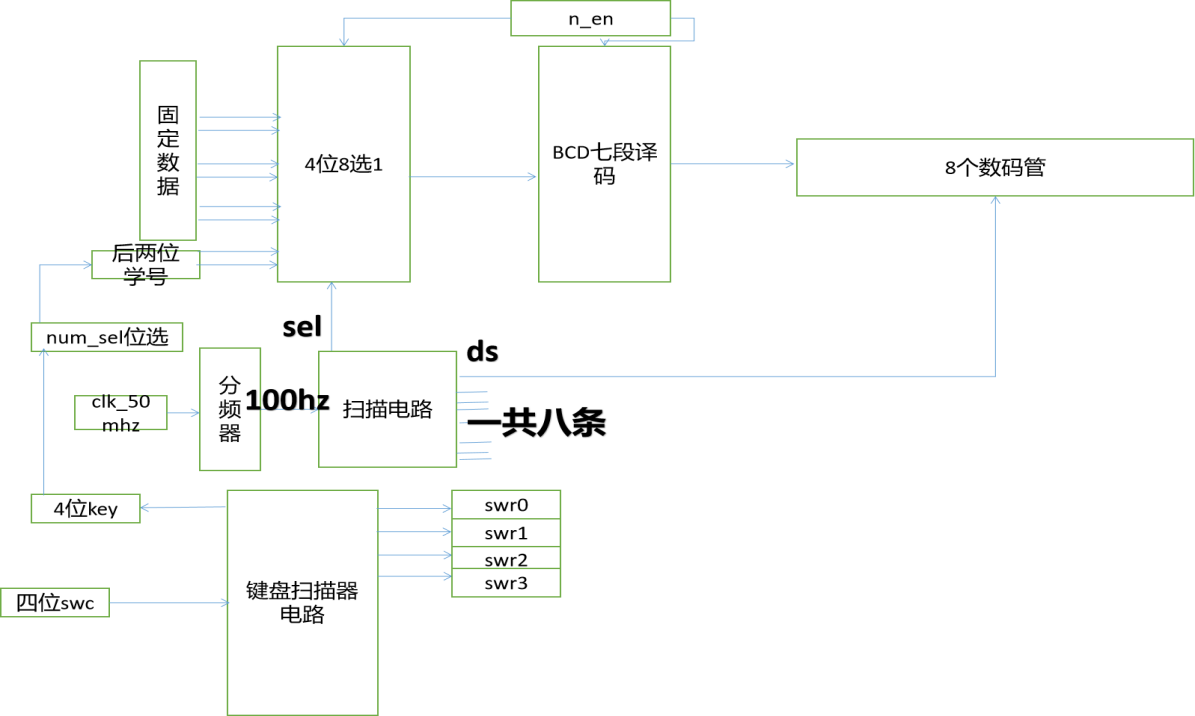


图9.设计思路

1. 详细设计

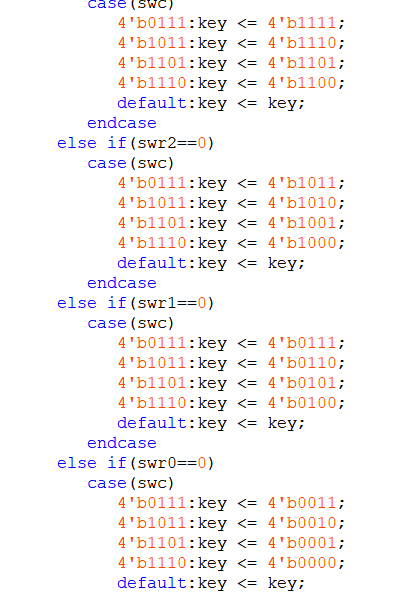
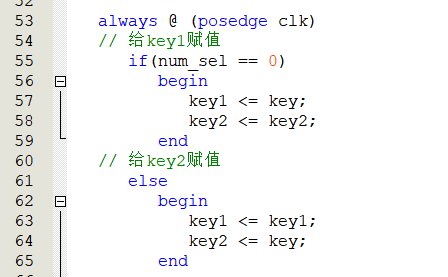


图10.键盘扫描电路新增代码

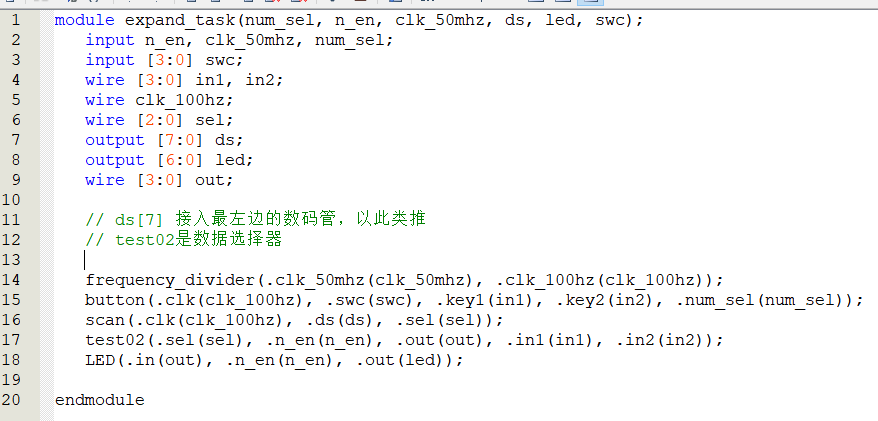


图11.顶层文件

1. 仿真验证

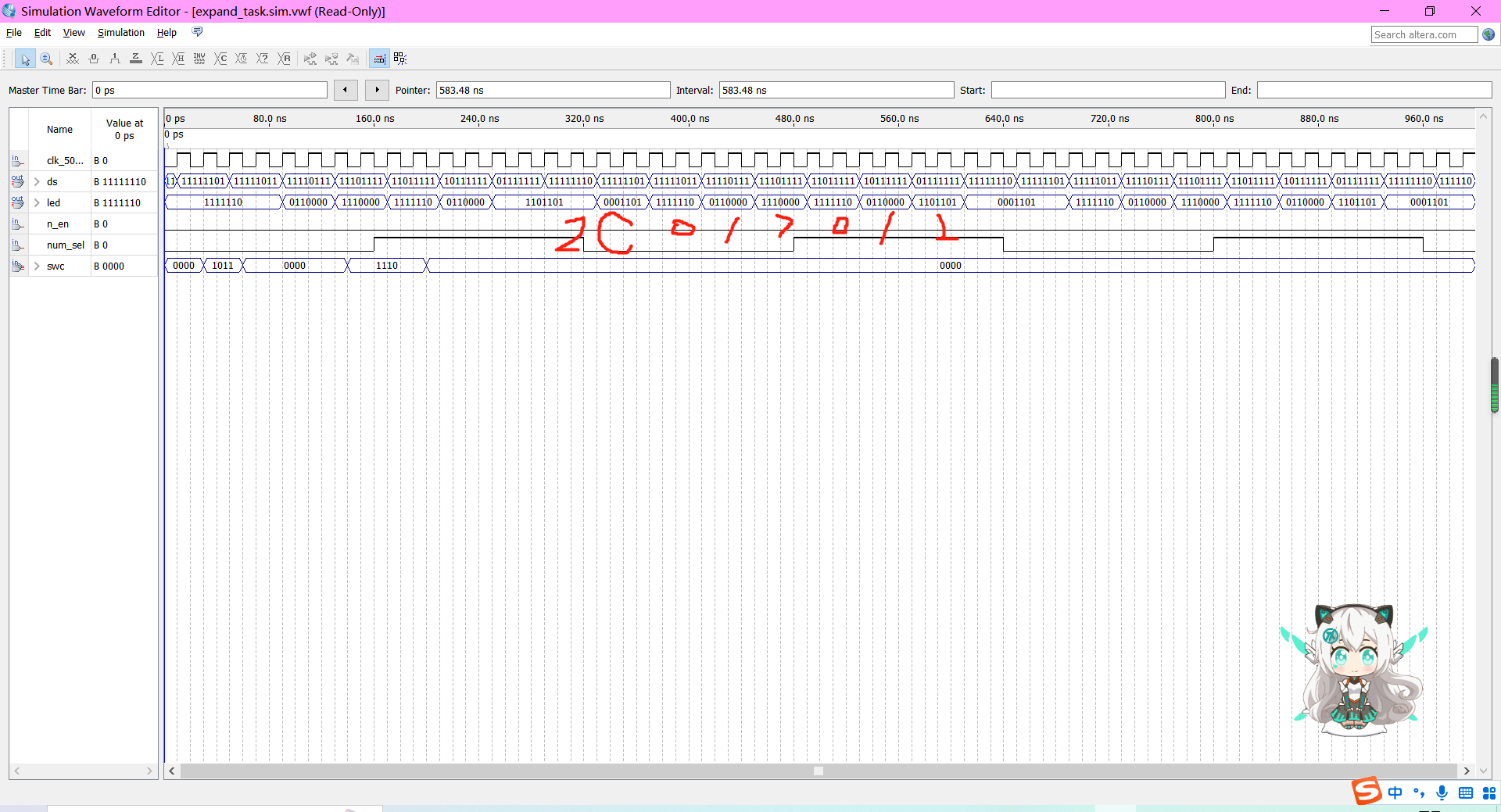


图12.仿真验证

如图12.，我们通过num\_sel分别控制键盘输入后两位学号，我们将两位学号分别写入为C和2，可见学号随即打印为210710C2，证明了准确性，也证明了修改case中的default使得最近一次键盘输入确实被保留。由于扫描频率很高，因此键盘颤动带来的影响可以忽略。

4.引脚分配

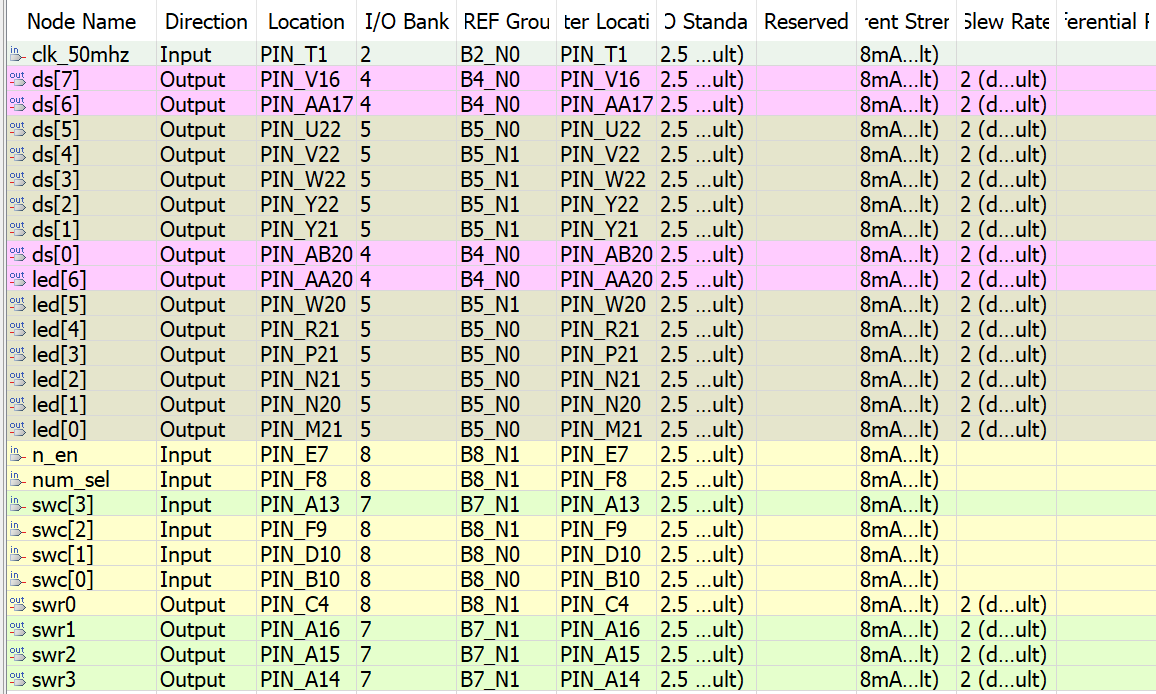
表5.引脚分配

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 端口名称 | 输入端 | | | | | | |
| swc | | | | n\_en | clk | Num\_sel |
| swc[3] | Swc[2] | Swc[1] | Swc[0] |
| 引脚编号 | A13 | F9 | D10 | B10 | E7 | T1 | F8 |
| 平台端口 | SWC3 | SWC2 | SWC1 | SWC0 | SW15 | T1 | SW8 |

表5.续表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 输出端 | | | | | | | |
| ds | | | | | | | |
| ds[7] | ds[6] | ds[5] | ds[4] | ds[3] | ds[2] | ds[1] | ds[0] |
| V16 | AA17 | U22 | V22 | W22 | Y22 | Y21 | AB20 |
| DS8 | DS7 | DS6 | DS5 | DS4 | DS3 | DS2 | DS1 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 输出端 | | | | | | | | | |
| led | | | | | | | | | |
| led[6] | led[5] | | led[4] | led[3] | | led[2] | led[1] | | led[0] |
| AA20 | W20 | | R21 | P21 | | N21 | N20 | | M21 |
| LA | LB | | LC | LD | | LE | LF | | LG |
| Swr0 | | Swr1 | | | Swr2 | | | Swr3 | |
| A14 | | A15 | | | A16 | | | C4 | |
| Swr0 | | Swr1 | | | Swr2 | | | Swr3 | |



5.实验现象

LED屏上从左到右依次输出21071000。若通过键盘改变输入端，输出的学号也会随输入端的改变而改变。可以通过sel选择希望改变的位。如果改变en开关，LED屏被关闭。与实验预期相符。

1. **总结**

在任务一中，我们第一次将代码下载到到实验台上的现象是LED屏幕上显示乱码。于是我们重新下载了程序，并重启了实验台，让实验台显示恢复了正常

在任务二中，由于疫情过后对软件操作生疏，在引脚分配的时候没有设置swr和clk，导致实验出错。经过此次实验，我们在实验六和七中每次都详细检查引脚分配错误和少分配的问题，极大地避免了因为此错误导致的实验结果错误。

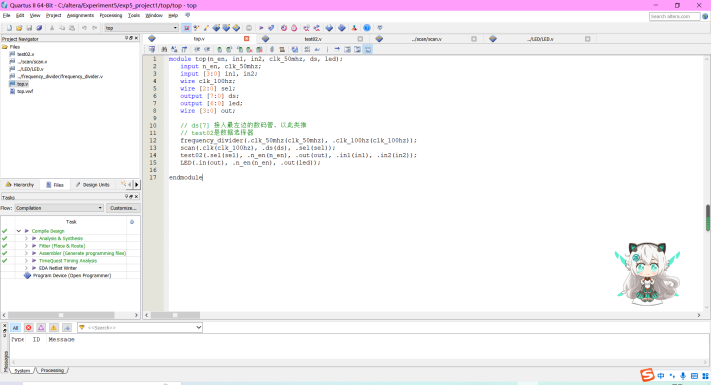
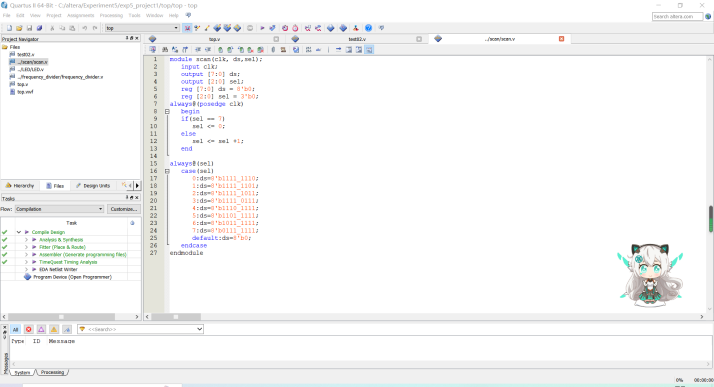


图1-2扫描电路和顶层文件

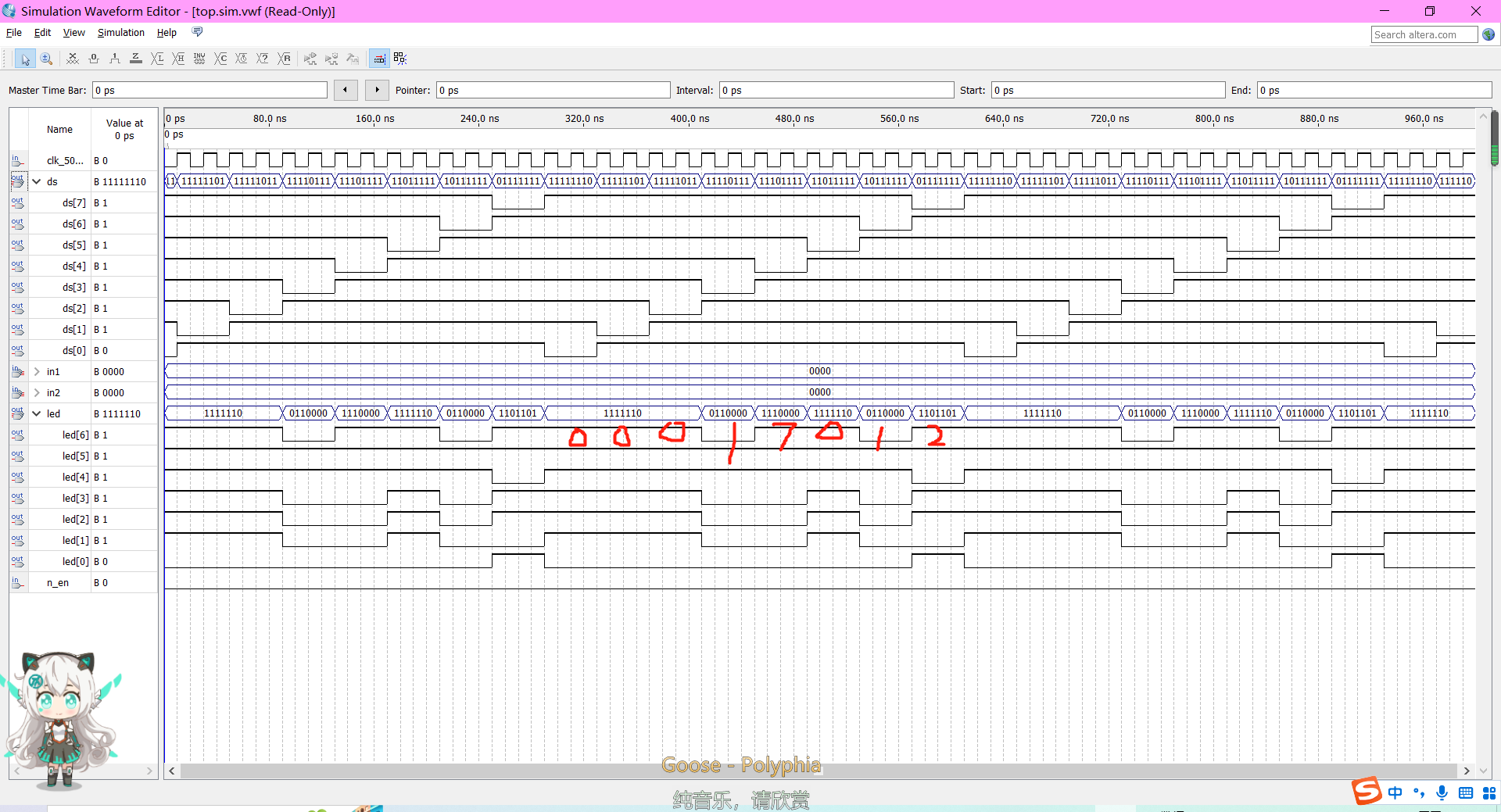
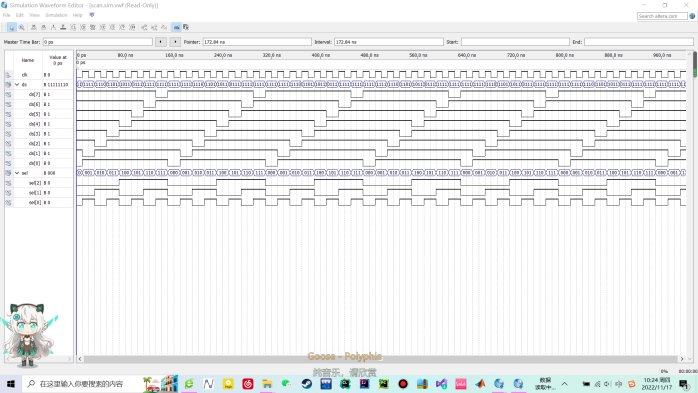


图3-4扫描电路和顶层文件波形图

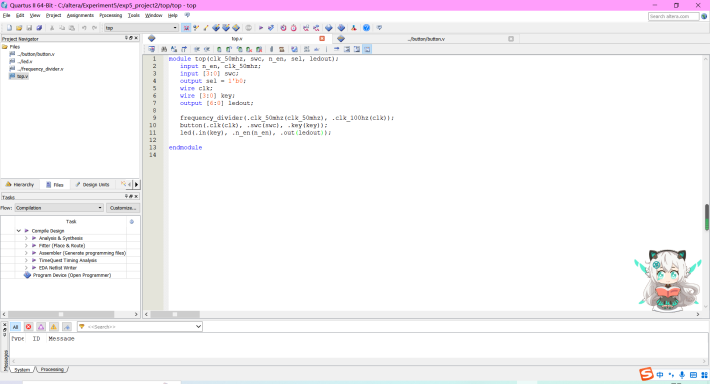
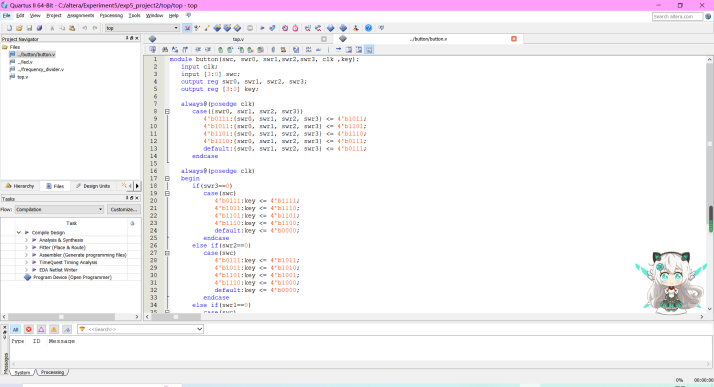


图5-6键盘扫描电路和顶层文件

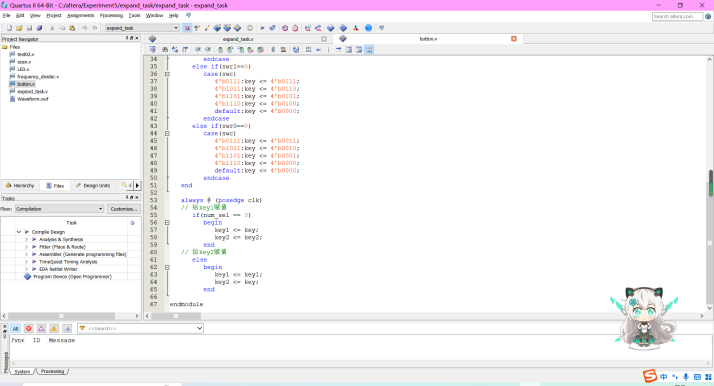
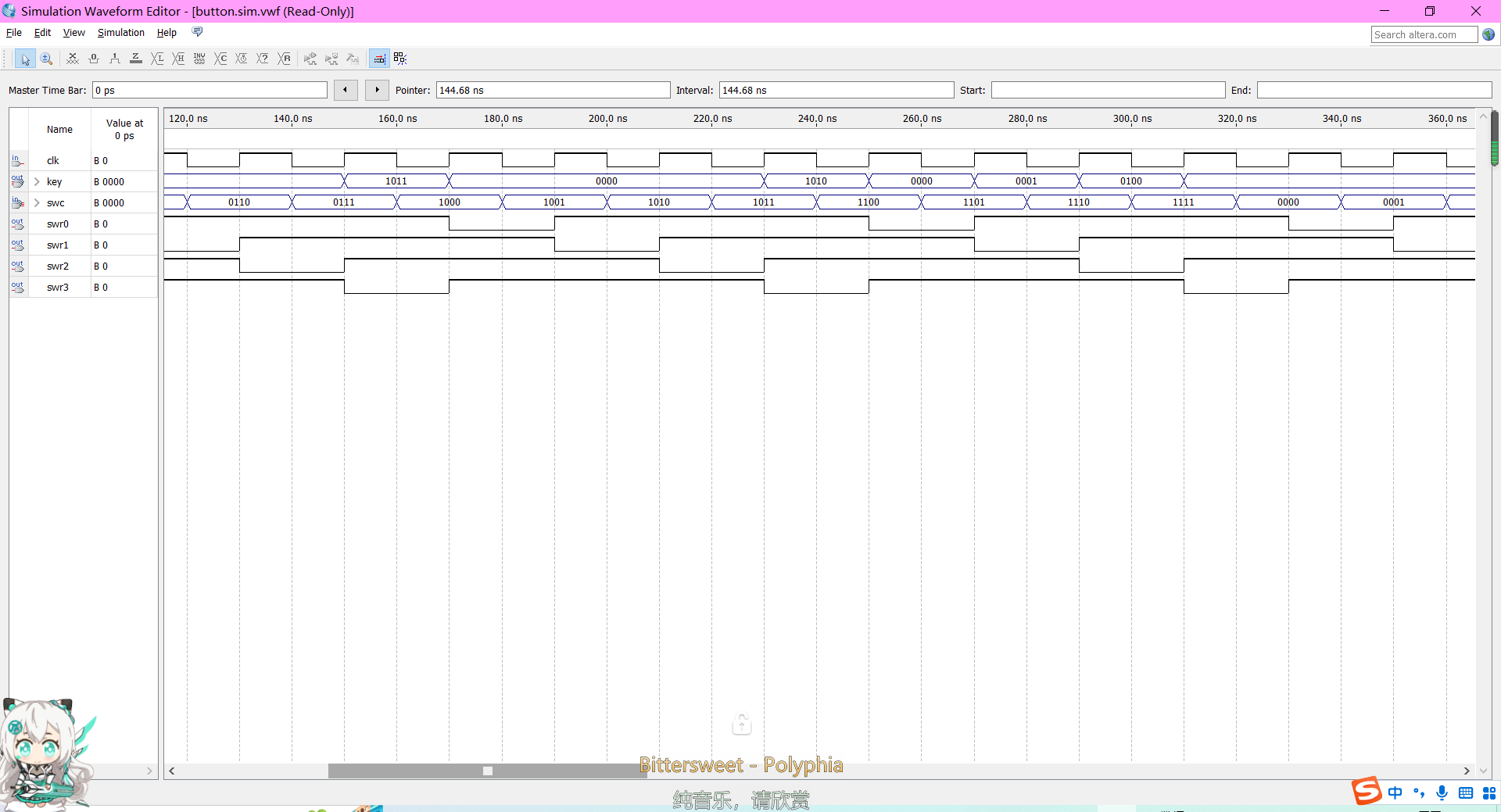


图7.顶层文件波形图 图8.扩展任务键盘扫描

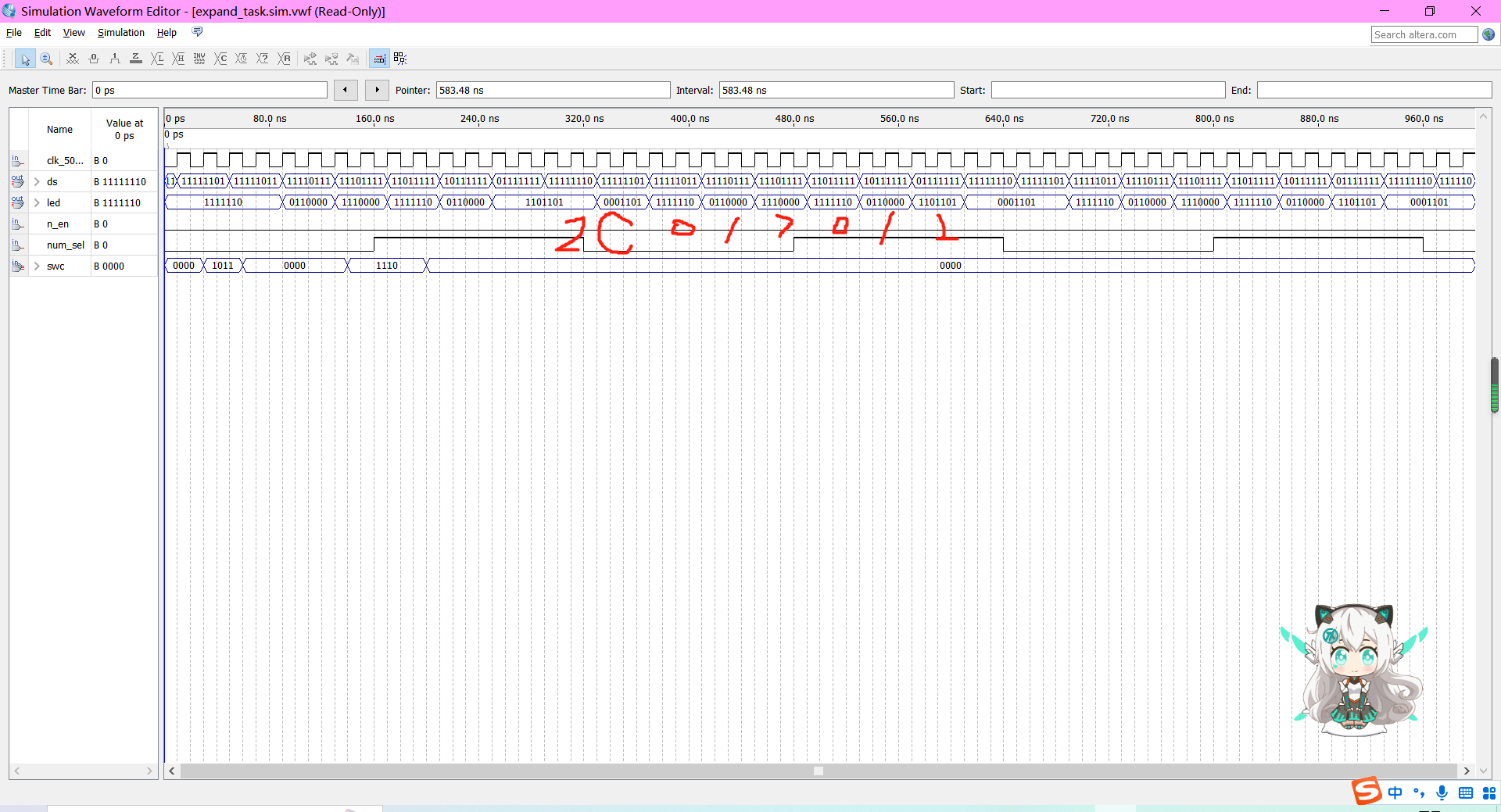
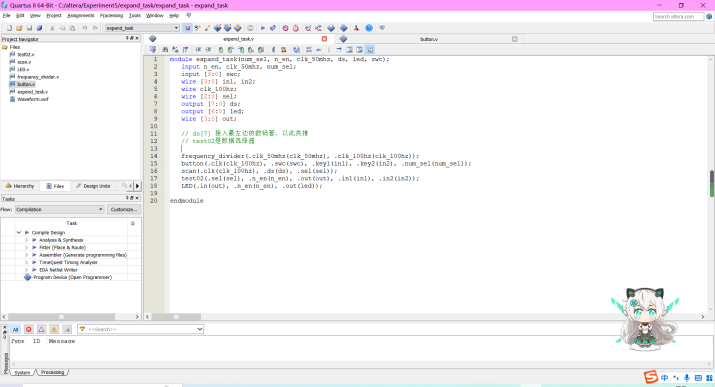


图9.扩展任务顶层文件 图10.扩展任务仿真验证