|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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4：状态机电路设计 | | | | | | | | |

1. **实验目的**

1.通过本实验掌握典型状态机电路的功能和特点;掌握摩尔型和米利型状态机的基本分析方法和设计方法;掌握使用硬件描述语言设计状态机电路的方法;巩固和加深对课程基本理论知识的理解。

2.通过交通灯、流水灯、序列检测器等电路的设计与测试，掌握状态机电路的分析方法和设计方法;学会使用Verilog HDL设计状态机电路

1. **任务一设计与实现**
2. 要求
3. 输出用LED显示，显示模式为LED灯从左至右或从右至左轮流点亮，也可自定义显示模式。根据实现模式，画出状态图。
4. 用Verilog编写状态机程序
5. 设计思路

任务一因为分频器只需要分出1hz信号，所以对原先的分频器代码进行简化，只输出1hz信号。流水灯根据书上的提示进行书写，我们流水灯设置的是，初始状态全灭，然后由右向左流水。



图1.设计思路

1. 详细设计

表1.详细设计

|  |  |
| --- | --- |
| 输入 | 输出 |
| reset | LED灯 |
| 0 | 1111\_1111\_1111\_1111 |
| 1 | 1111\_1111\_1111\_1111 |
| 1 | 1111\_1111\_1111\_1110 |
| 1 | 1111\_1111\_1111\_1101 |
| 1 | 1111\_1111\_1111\_1011 |
| 1 | 1111\_1111\_1111\_0111 |
| 1 | 1111\_1111\_1110\_1111 |
| 1 | 1111\_1111\_1101\_1111 |
| 1 | 1111\_1111\_1011\_1111 |
| 1 | 1111\_1111\_0111\_1111 |
| 1 | 1111\_1110\_1111\_1111 |
| 1 | 1111\_1101\_1111\_1111 |
| 1 | 1111\_1011\_1111\_1111 |
| 1 | 1111\_0111\_1111\_1111 |
| 1 | 1110\_1111\_1111\_1111 |
| 1 | 1101\_1111\_1111\_1111 |
| 1 | 1011\_1111\_1111\_1111 |
| 1 | 0111\_1111\_1111\_1111 |

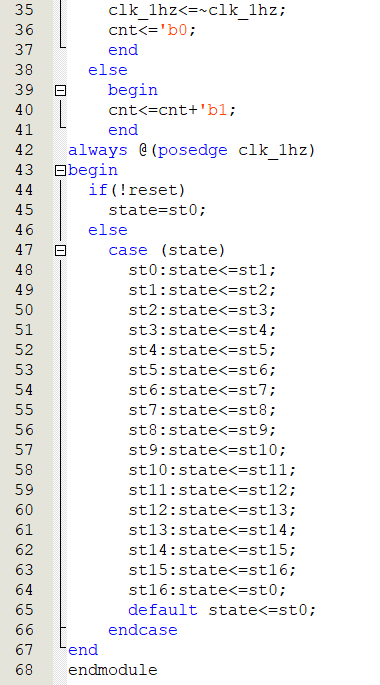
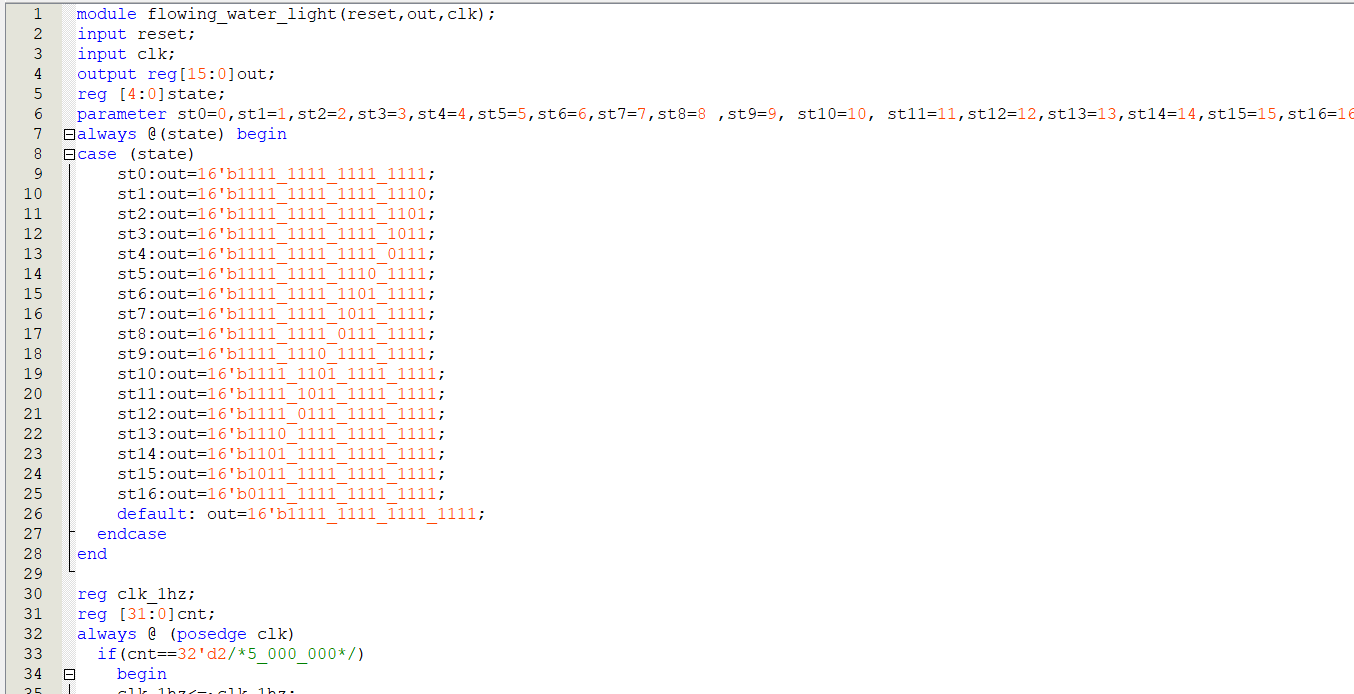


图2.流水灯

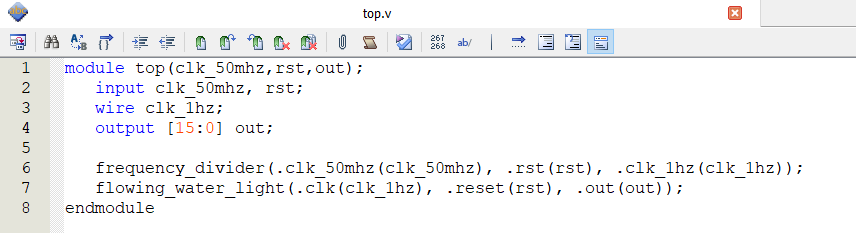


图3.顶层文件

1. 仿真验证

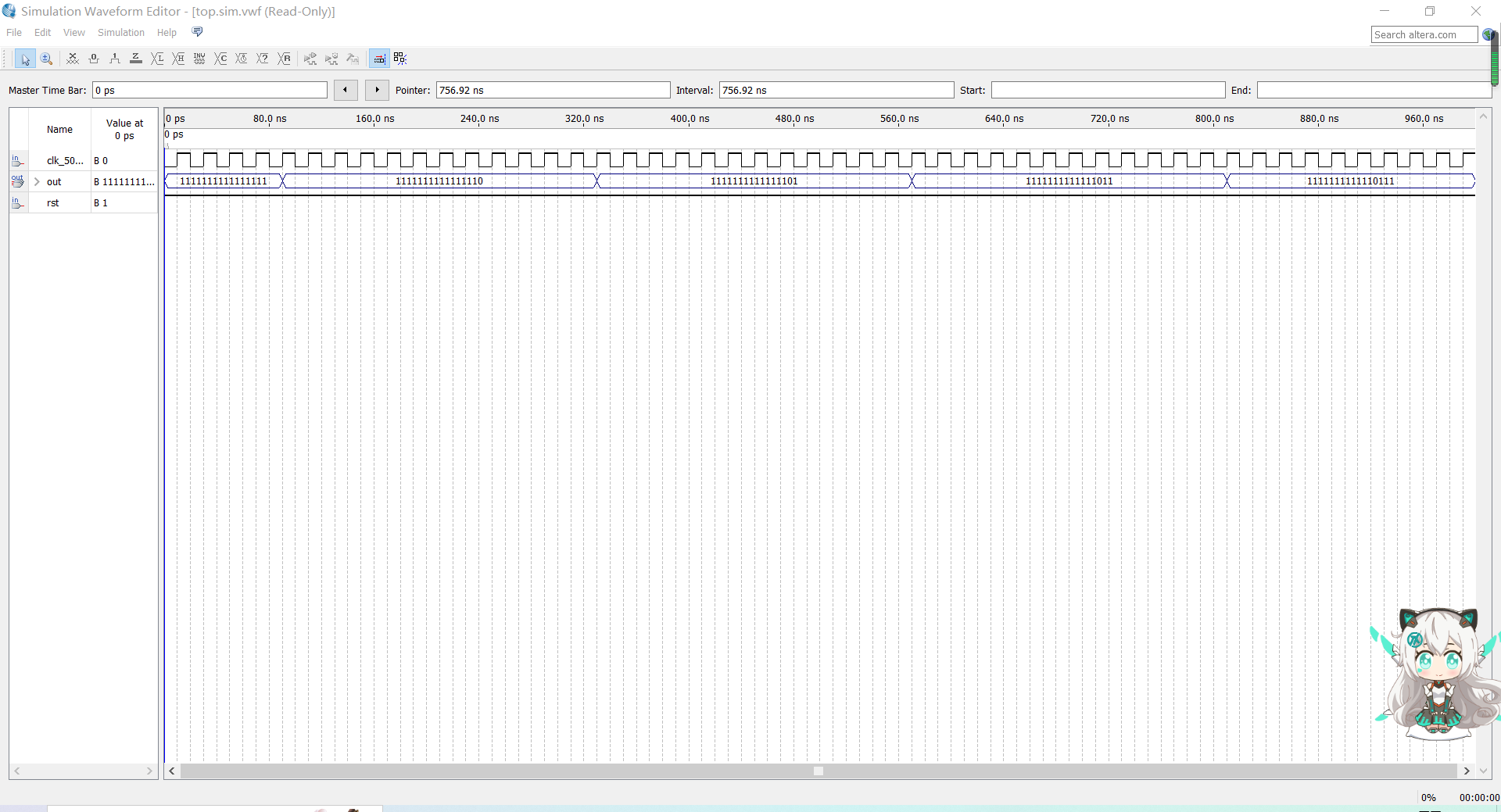


图4.波形图

显而易见，流水灯正常工作。

1. 引脚分配

表2.引脚分配

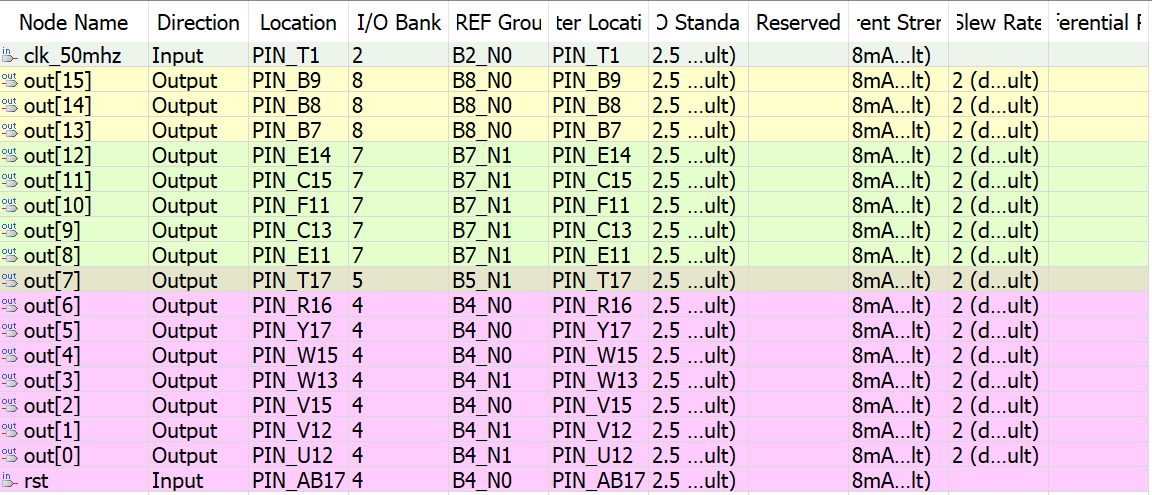
|  |  |  |
| --- | --- | --- |
| 端口名称 | 输入端 | |
| 时钟信号clk | reset |
| 引脚编号 | T1 | SW9 |
| 平台端口 | T1 | AB17 |

表2.续表

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 输出端 | | | | | | | | | | | | |
| LED | | | | | | | | | | | | |
| out[15] | out[14] | out[13] | out[12] | | out[11] | | out[10] | | out[9] | | out[8] | |
| B9 | B8 | B7 | E14 | | C15 | | F11 | | C13 | | E11 | |
| LED16 | LED15 | LED14 | | LED13 | | LED12 | | LED11 | | LED10 | | LED9 |

表2.续表

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 输出端 | | | | | | | | | | | | |
| LED | | | | | | | | | | | | |
| out[7] | out[6] | out[5] | out[4] | | out[3] | | out[2] | | out[1] | | out[0] | |
| T17 | R16 | Y17 | W15 | | W13 | | V15 | | V12 | | U12 | |
| LED8 | LED7 | LED6 | | LED5 | | LED4 | | LED3 | | LED2 | | LED1 |



1. 实验现象

程序开始执行后，工作台上的LED灯从左到右间隔一秒依次闪烁。在拨动rst开关后，程序重新执行，从第一个LED灯开始闪烁。本次实验符合所有输出预期。

1. **任务二设计与实现**
2. 要求
3. 设计一个1010序列检测器(不考虑序列重叠)或一个8位序列的序列检测器，序列为同组两位同学学号末两位相加，如相加后低于16，需在和的基础上加30(检测序列为结果对应的8421BCD码，如结果为34，检测序列为00110100)。
4. 采用Verilog实现状态机。
5. 设计思路

任务二采用任务一的分频器，时钟信号输入到数据发生器，发生器产生的单个编码传入到检测器检测序列。检测器采用Moore型。

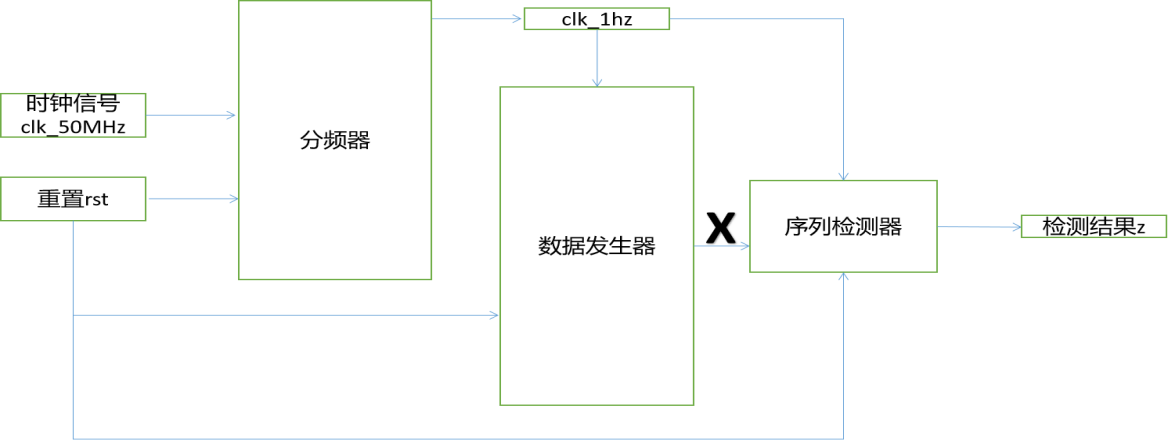


图5.设计思路

1. 详细设计

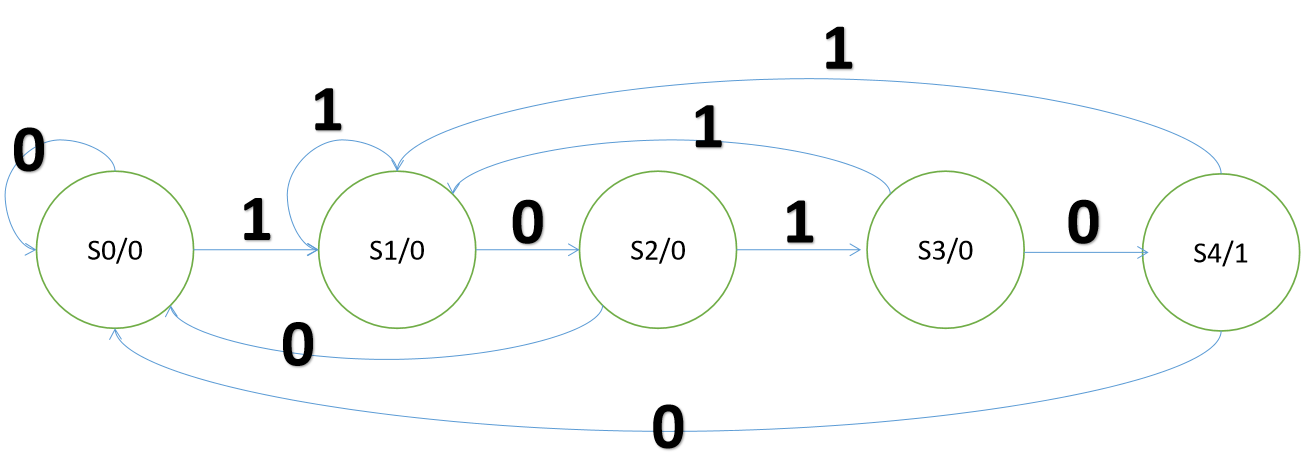


图6.状态图

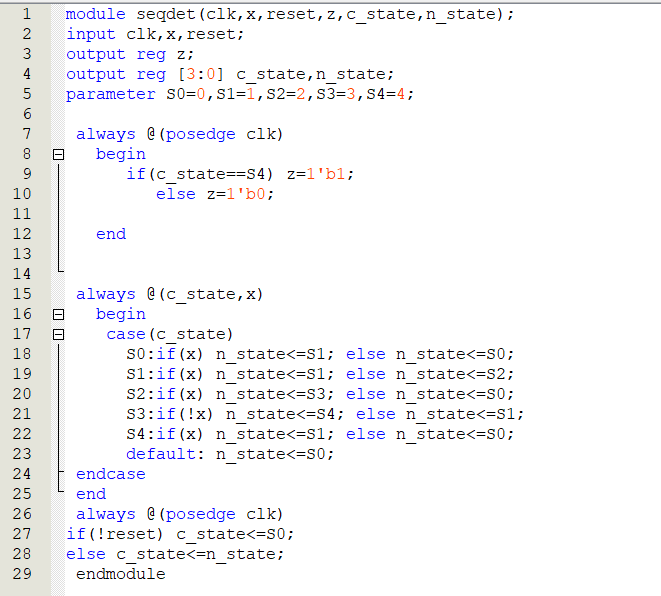
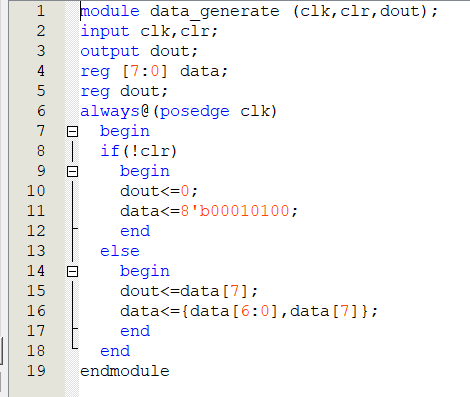


图7.数据发生器 图8.序列检测器

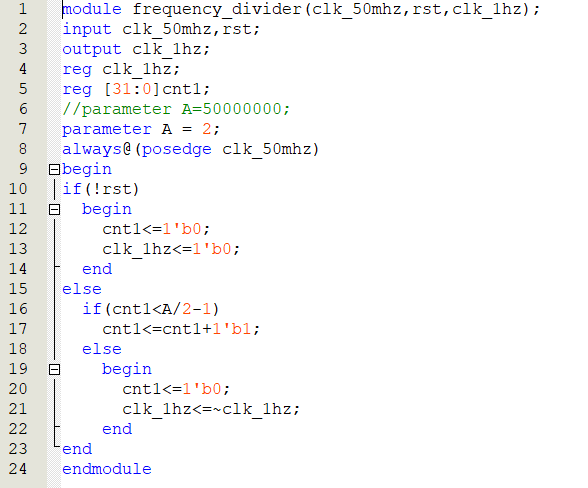


图9.分频器

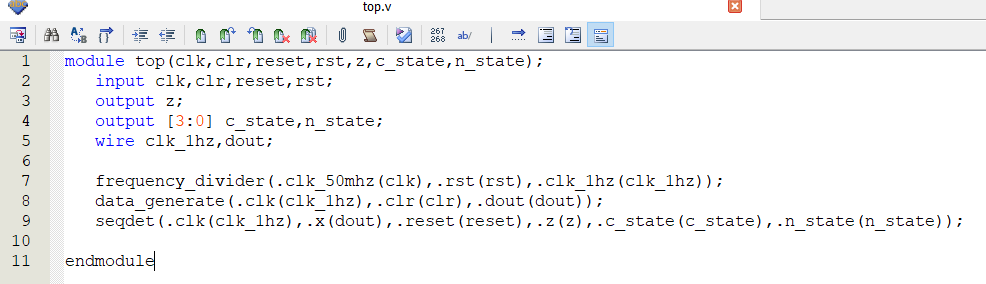


图10.顶层文件

1. 仿真验证

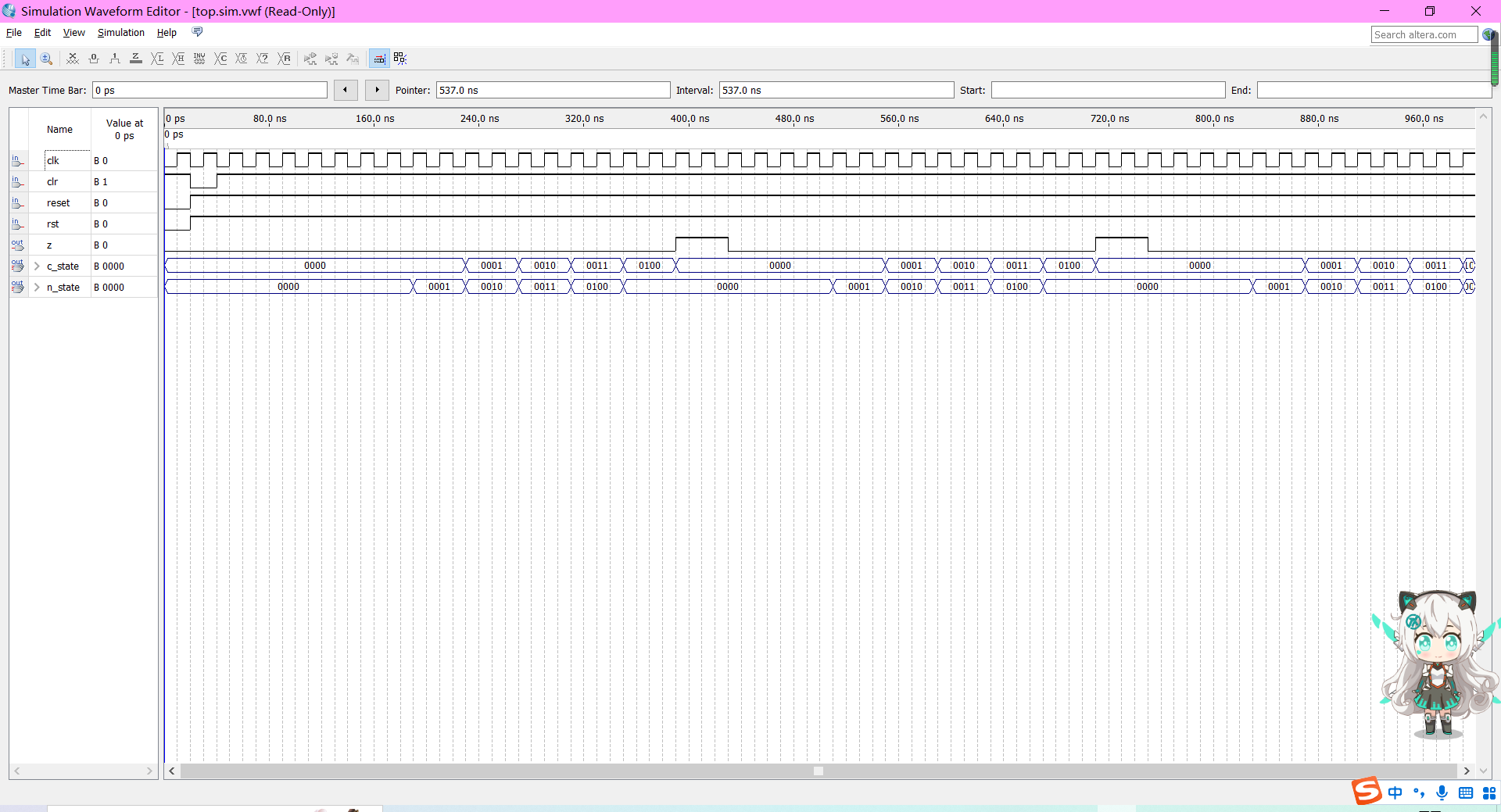


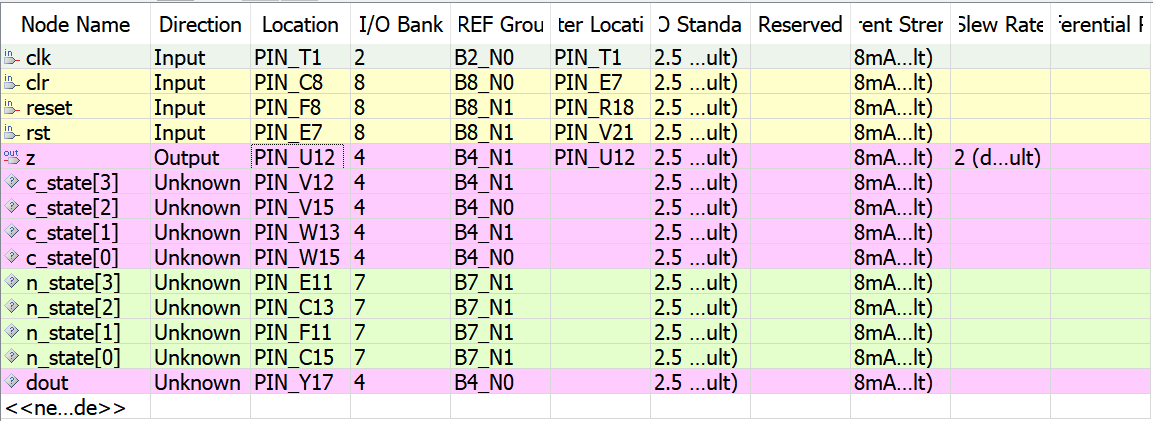
图11.波形图

可见z在状态机进入最后状态的时候变为1了，此时序列也正是检测到了1010.

1. 引脚分配

表3.引脚分配

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 端口名称 | 输入端 | | | | | | 输出端 |
| rst | clr | clk | | reset | | z | |
| 引脚编号 | C8 | E7 | T1 | | F8 | | U12 | |
| 平台端口 | SW6 | SW7 | | T1 | | SW8 | LED1 | |



1. 实验现象

先将Reset开关与rst开关置1，将clr开关先置0，再置1，发现LED灯每隔8秒闪烁一次。将Reset开关或rst开关置0再置1后，LED灯闪烁间隔时间重置。本次实验符合所有输出预期。

1. **扩展实验**

1.设计思路

我们希望设计一个用户可以自行输入希望序列检测器检测的数据，于是我们在任务二的基础上增加了一个模块以实现该功能。

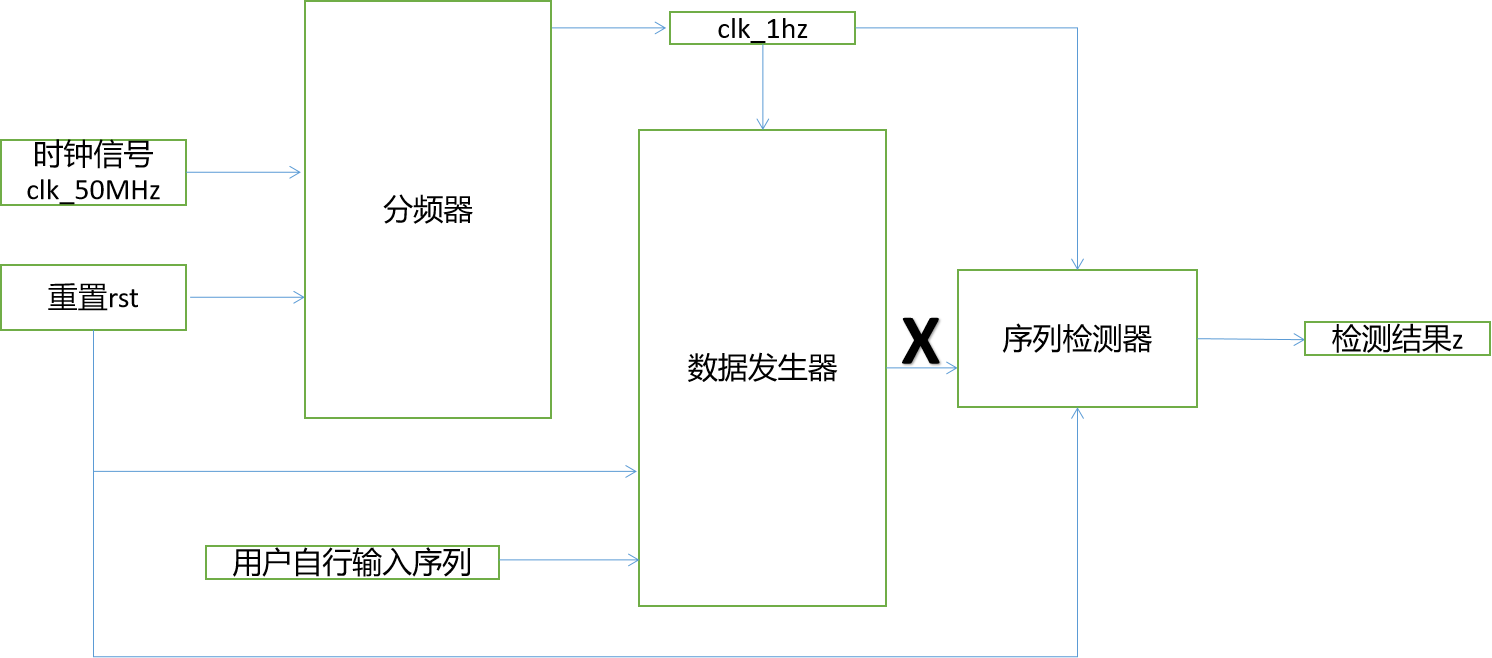
****

图12.设计思路

2.详细设计

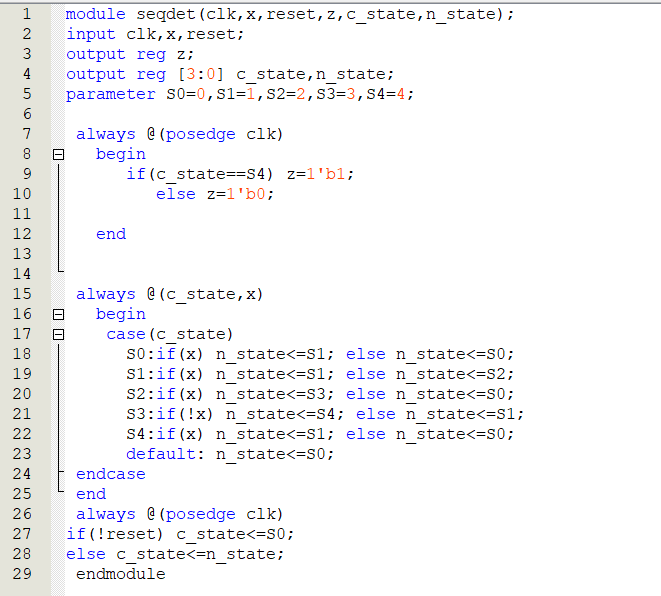
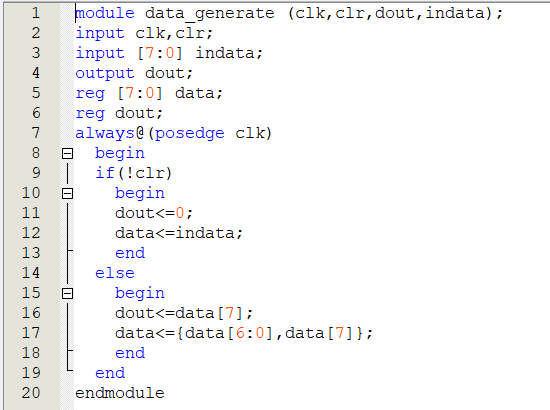


图13.数据发生器 图14.序列检测器

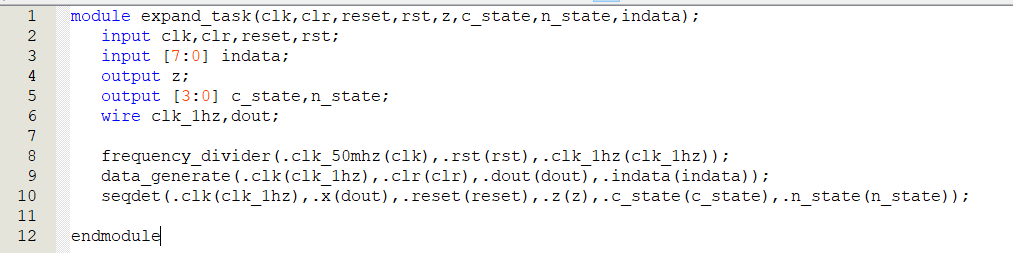


图15.顶层文件

1. 仿真验证

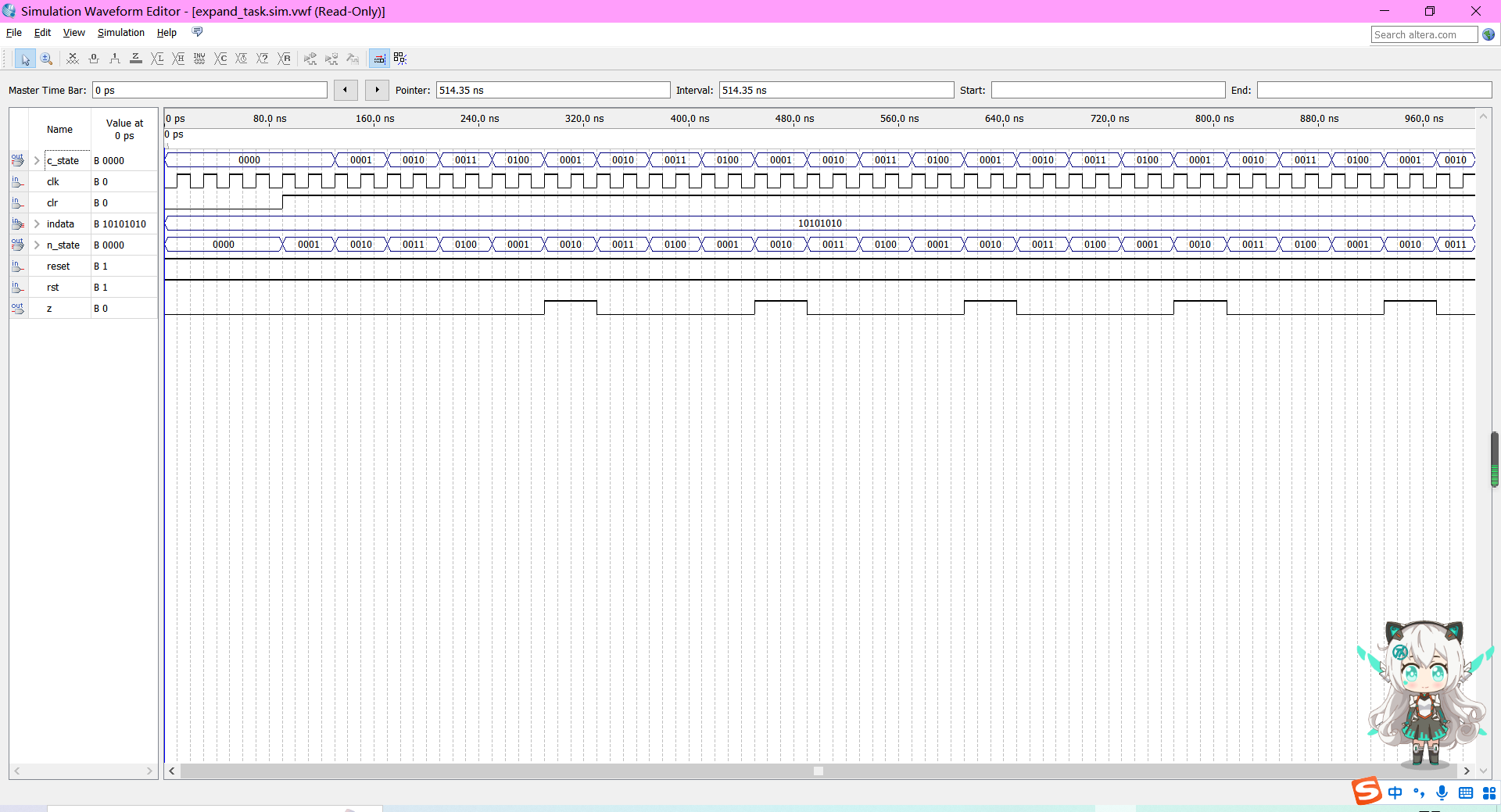


图16.仿真验证

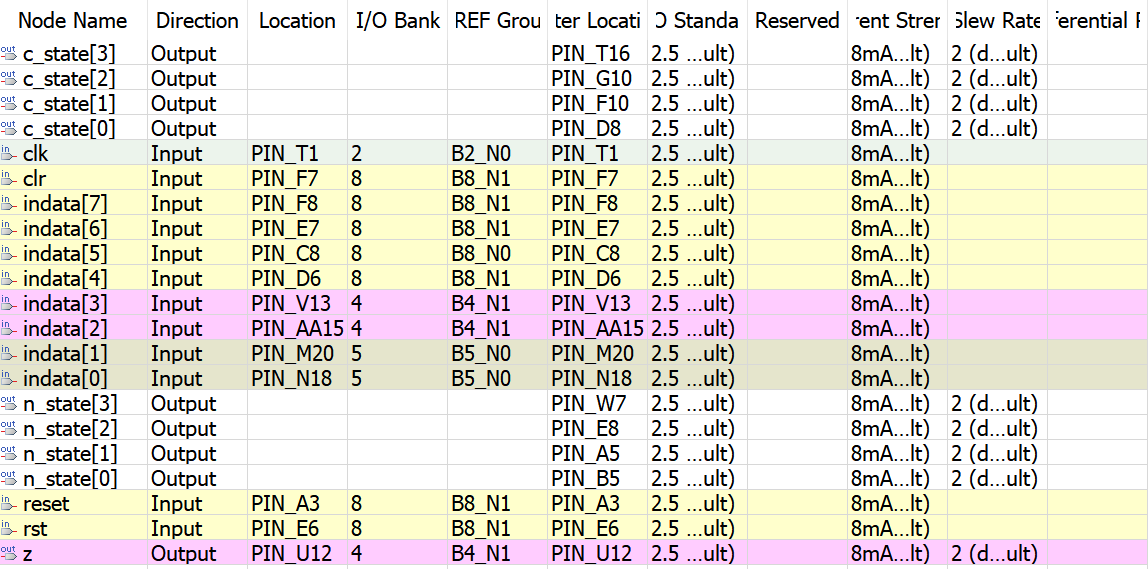
z的频率变高，因为序列被设置为了10101010。这个思路其实从另一种角度来说，构造出了新的分频器，如果控制1010出现的频率，分频器出的频率也可以随之改变。

1. 引脚分配

表4.引脚分配

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 端口名称 | 输入端 | | | | | | | |
| indata[7] | indata[6] | indata[5] | indata[4] | indata[3] | indata[2] | indata[1] | indata[0] |
| 引脚编号 | F8 | E7 | C8 | D6 | V13 | AA15 | M20 | N18 |
| 平台端口 | SW8 | SW7 | SW6 | SW5 | SW4 | SW3 | SW2 | SW1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 端口名称 | 输入端 | | | | 输出端 |
| rst | clr | clk | reset | z | |
| 引脚编号 | E6 | F7 | T1 | A3 | U12 | |
| 平台端口 | SW14 | SW15 | T1 | SW16 | LED1 | |



1. 实验现象

我们将序列设置为10101010，先将Reset开关与rst开关置1，将clr开关先置0，再置1，发现LED灯每隔4秒闪烁一次。将Reset开关或rst开关置0再置1后，LED灯闪烁间隔时间重置。本次实验符合所有输出预期。

1. **总结**

我们小组由于疫情原因，本实验没能在上学期做完，而是在本学期去实验台上完成的实验。因为时间比较紧张，我们小组也出现了较多的低级错误。

①实验一中，流水灯模块是由石昊阳完成，而顶层文件则是由高立扬完成。石昊阳已经在流水灯中写入了分频器代码，而高立扬在写顶层文件的时候又加入了分屏器模块，因此是流水灯闪烁的时间间隔缩短了一倍。我们在之后的合作中会更加积极的向对方讲解自己的想法与代码，这大大的提高了我们的团队合作能力。

②实验一中，我们发现后八个LED灯不能闪烁。通过查找资料得知后八个LED灯需要在模式二下才能正常工作。

③实验二中，我们发现实验台上的现象与波形图不一致。老师告诉我们可以将顶端中的一些变量输出到实验台上，以便查找错误。我们用这个方法顺利的找到并修改了错误。

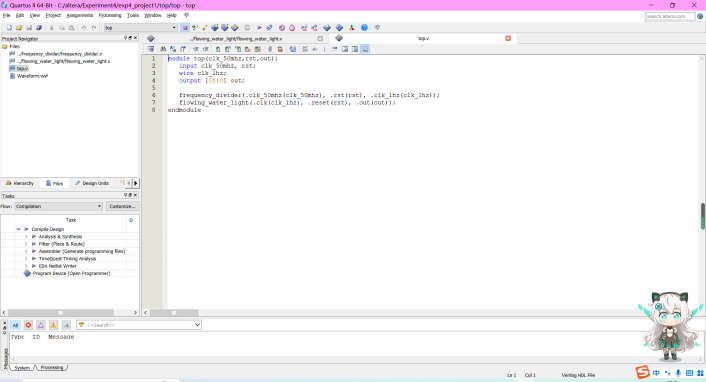
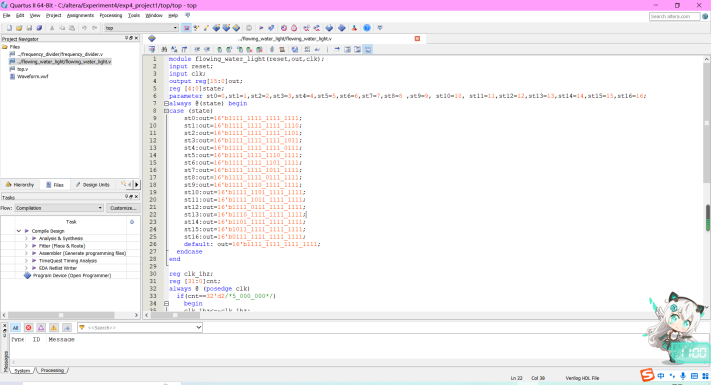


图1-2.流水灯代码和顶层文件代码

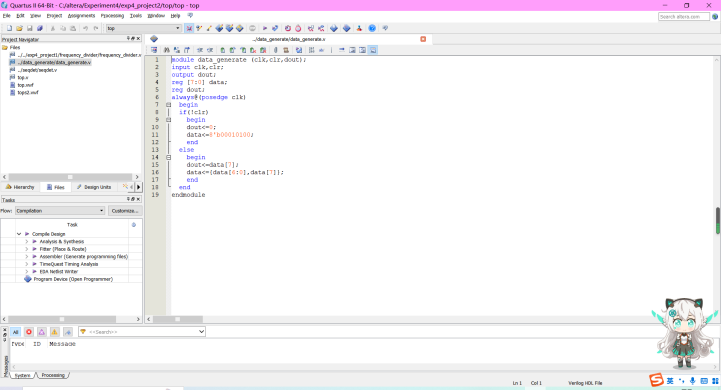
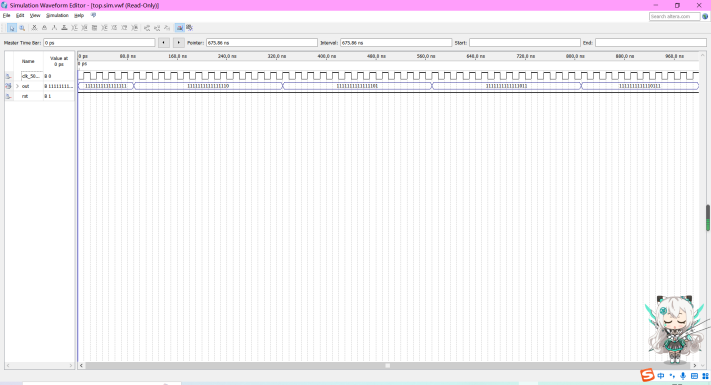


图3.流水灯波形图 图4.状态机序列生成器

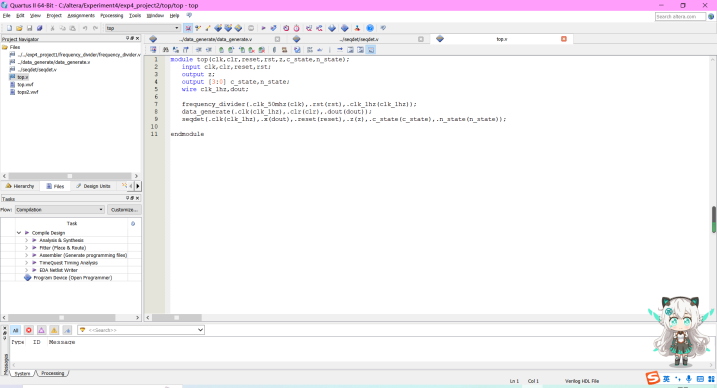
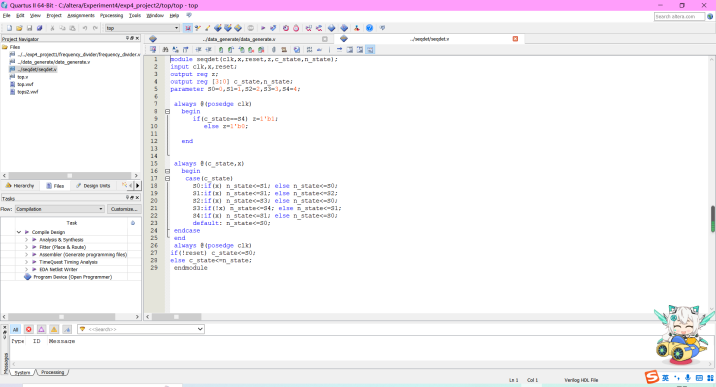


图5.状态机本体 图6.任务二顶层文件

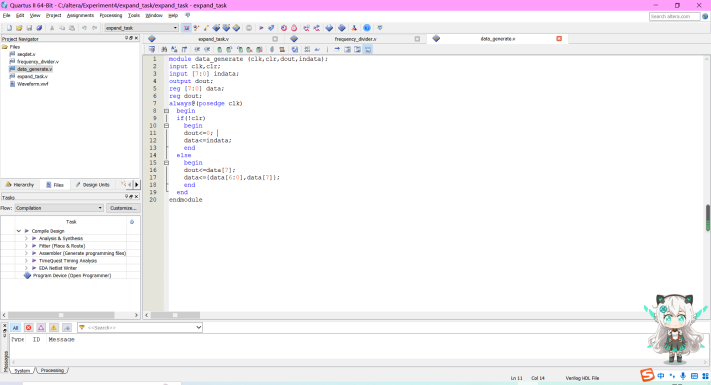
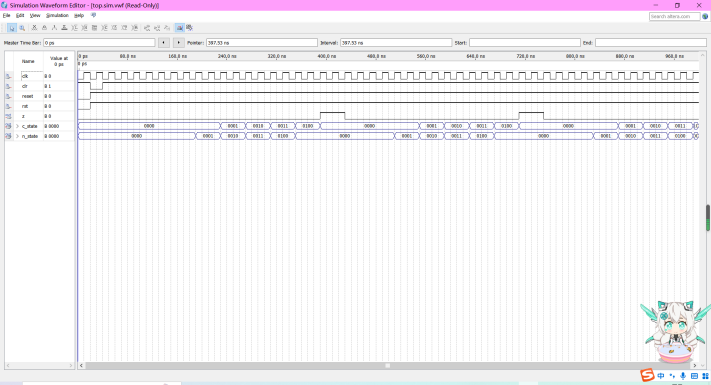


图7.任务二波形图 图8.扩展任务序列生成器改版

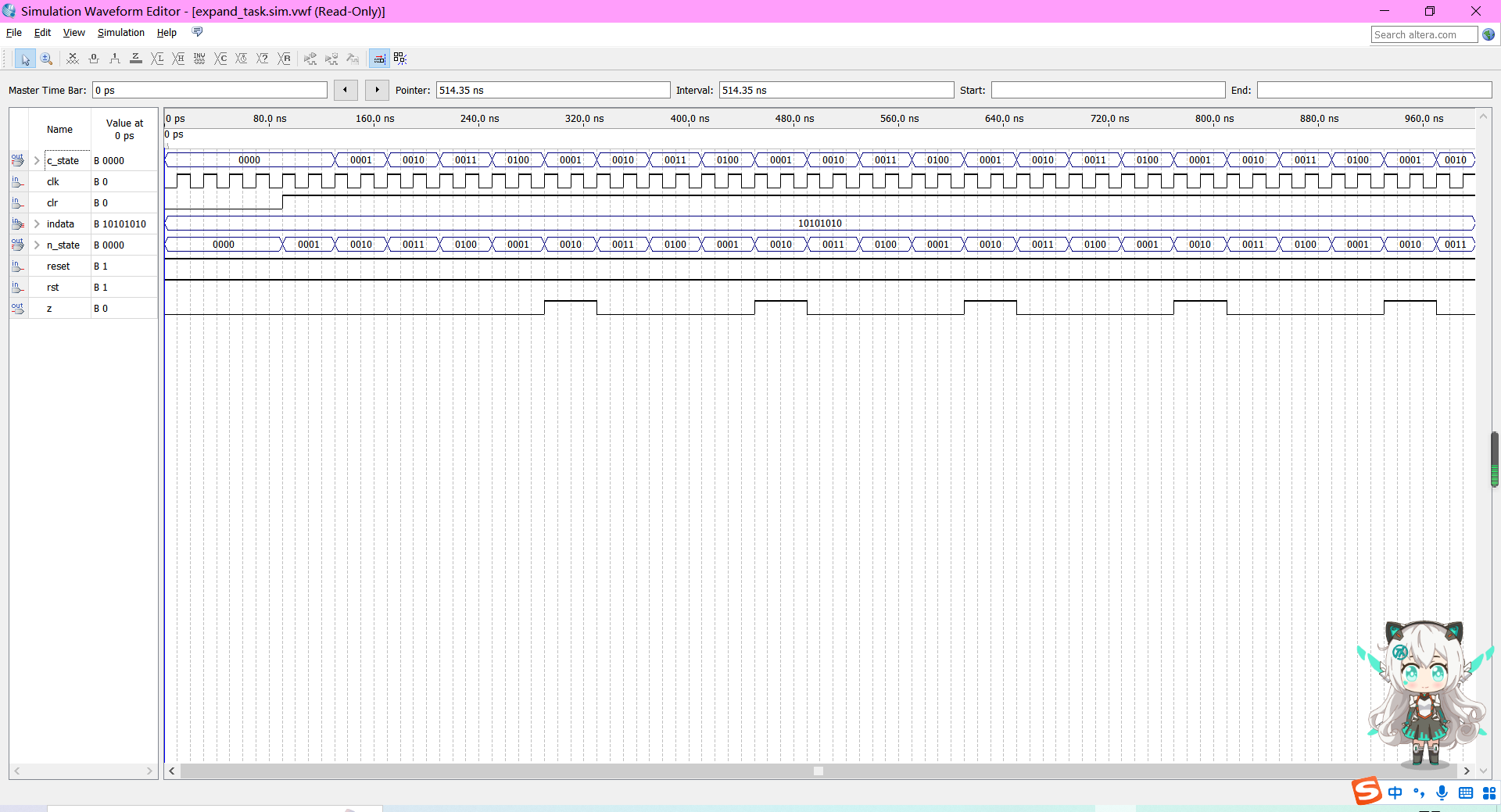
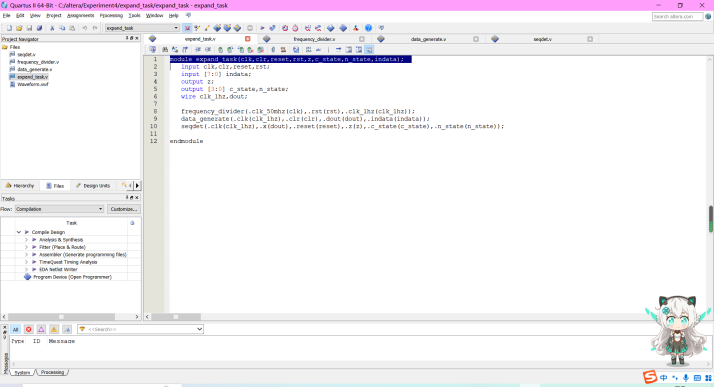


图9.任务二波形图 图10.扩展任务序列生成器改版

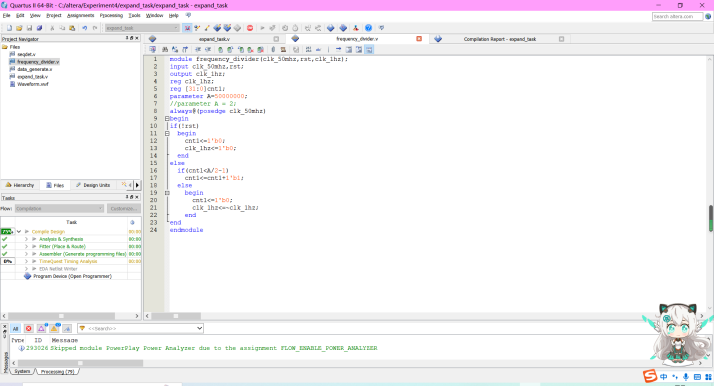


图11.公用分频器