

dd project report

1. 开发计划

1.1 任务划分

1.2 进度安排

1.3 责任人

1.4 执行记录:

1.4.1 确定选题: 12月7日

1.4.2 确认大致思路: 12月8日

1.4.3 确认分工及进度安排: 12月9日

1.4.4 进行基本实现, 商量优化的实现: 12月9日——12月14日

1.4.5 把各自部分完善: 12月14日——12月21日

1.4.6 解决蜂鸣器和按键消抖的问题: 12月22日

1.4.7 合并并 debug: 12月24日

1.4.8 商讨如何展示和报告的书写问题: 12月25日

1.4.9 完成报告, 写使用文档: 1月2日

2. 设计

2.1 需求分析

2.1.1 系统功能

2.1.2 系统输入输出

2.1.3 端口规格

2.2 系统结构图

2.3 系统执行流程

2.3.1 数据流向和状态转移图

2.3.2 伪代码

2.4 子模块代码

2.5 约束文件

3. 测试

3.1 testbench 文件

3.2 子模块的仿真波形及测试结果

4. 总结

4.1 开发过程中遇到的问题及解决方案

4.1.1 在设置部分遇到的问题

4.1.2 在抢答部分遇到的问题

4.1.3 由于没有配合好遇到的问题

4.2 当前系统特色

4.3 优化方向

dd project report

made by 11811026 张望, 11811011 商际豪

presentation之后改动: 蜂鸣器支持主持人按键提示, 复核部分更加稳定。

1. 开发计划

1.1 任务划分

张望: 做抢答步骤以及这步的显示, 写 report

商际豪: 做设置步骤以及这步的显示, 写 report

1.2 进度安排

主要根据两个 DDL：展示 DDL——12.26，report 提交 DDL——1.4

- 12.9 之前确认选题和思路，讨论接口规范。
- 12.13 之前进行基本的实现，讨论是否需要换选题，如果不换，讨论更优的实现。
- 12.19 之前把各自的部分做好，互相展示。
- 12.24 之前把各自的部分合并完全。
- 12.25 商讨如何展示和报告的具体书写。
- 1.4 之前把报告写好

1.3 责任人

每天互相交流做了什么内容，互为监督（由于是室友每天都可以线下交流，很方便）。

1.4 执行记录：

1.4.1 确定选题：12 月 7 日

任务一：考虑到学习成本可能会比较高，小组只有两个人，人手不够，这方面本来得到的信息就有限，有做不出来的风险，所以放弃选择这个题目。

任务二：看题之后能够想象要怎么去完成这个项目，同时对于抢答有生活体验，便于设计。

任务三：过于简单，同时没法拿到满分，不予考虑。

任务四：生活体验不足，无法想象项目如何构建，相较之下任务二更加适合。

综合上面的分析，选择任务二——抢答器为我们的 project 题目。

1.4.2 确认大致思路：12 月 8 日

商讨实现的具体方案，大体有想法，但是由于一些细节有待思考和协商，故未能直接确认分工，决定择日确认。

1.4.3 确认分工及进度安排：12 月 9 日

分析：抢答器主要有设置，抢答和显示三个步骤，设置和抢答可以是两个模块，设置模块传给抢答模块必要的参数即可，两个模块都需要显示，所以分工主要根据设置、抢答、显示来进行分工，由于只有两个人，所以决定一人负责一个模块，同时写出逻辑和显示。

具体分工见 1.1 任务划分。根据分工和 DDL 得到的进度安排见 1.2 进度安排。

1.4.4 进行基本实现，商量优化的实现：12 月 9 日——12 月 14 日

开始做 project 的第一个周末，两人都完成了基本的逻辑实现，通过 testbench 测试符合预期，确定继续做这个选题，都暂时没有写显示模块，烧板子没法展示效果。

交流之后，觉得设置步骤使用键盘代替效果会更好，而抢答步骤继续做即可，所以设置部分接下来需要搞懂键盘的使用，两人都需要做出显示。

1.4.5 把各自部分完善：12 月 14 日——12 月 21 日

由于 dsaa 的学习压力，到 21 日才各自做完各自的部分，两人的部分基本已经完成，抢答部分由于不熟悉蜂鸣器的使用暂时没有做出来，设置部分没有解决按键消抖的问题。

1.4.6 解决蜂鸣器和按键消抖的问题：12 月 22 日

由于只有蜂鸣器和按键消抖问题，聚在一起查查资料讨论讨论很快就解决了，此时两个人的各自的部分没有 bug。

1.4.7 合并并 debug：12 月 24 日

把两个人的部分合并，由于前期商量的时候接口规范讨论充分，同时两个人写代码的时候十分重视为最后的对接做准备，在这个 part 没有遇到太多问题，主要是一些小细节导致最后出现 bug，很快就把 bug de 完了。

1.4.8 商讨如何展示和报告的书写问题：12 月 25 日

展示前一天把需要展示的内容商量了一下，报告的开发计划部分平时就有写，交流了本次 project 的心得，互相反思了本次 project 的不足之处，把总结部分写了。同时商量数字逻辑期末考试之后再一起把报告的剩余部分写了。

1.4.9 完成报告，写使用文档：1 月 2 日

将报告的设计和测试部分写了，同时写了使用文档，录制了使用视频。

2. 设计

2.1 需求分析

2.1.1 系统功能

主要需要设计一个抢答器系统，根据日常生活的经验，可以想象到，需要能够设置参与者人数，倒计时时间，加分减分数值的模块，需要模拟抢答时选手抢答加分减分的模块，同时需要最后有选手胜利时供展示和复核的模块。

所以这就是我们系统需要实现的功能，根据这个功能去分工和做就能实现智力抢答器系统。

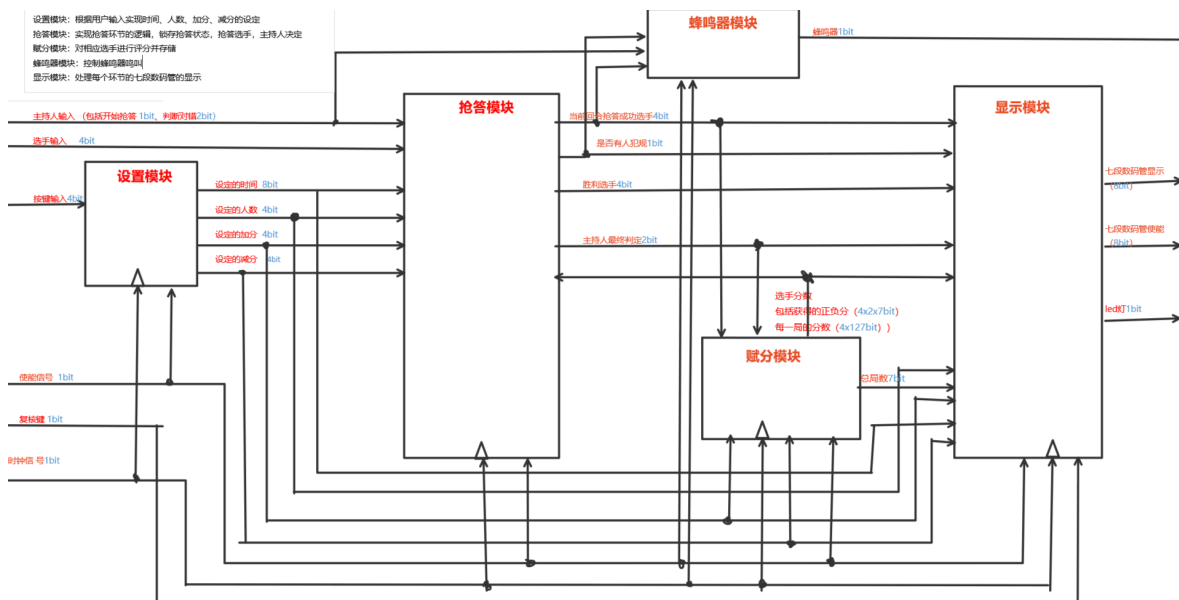
2.1.2 系统输入输出

使用矩阵键盘来进行用户的设置输入，因为这样更加符合用户的习惯，上手即用。使用开关来进行抢答阶段的选手输入和主持人输入。使用键盘作为输出设备，因为选手和用户需要能够看到得分这些东西。

2.1.3 端口规格

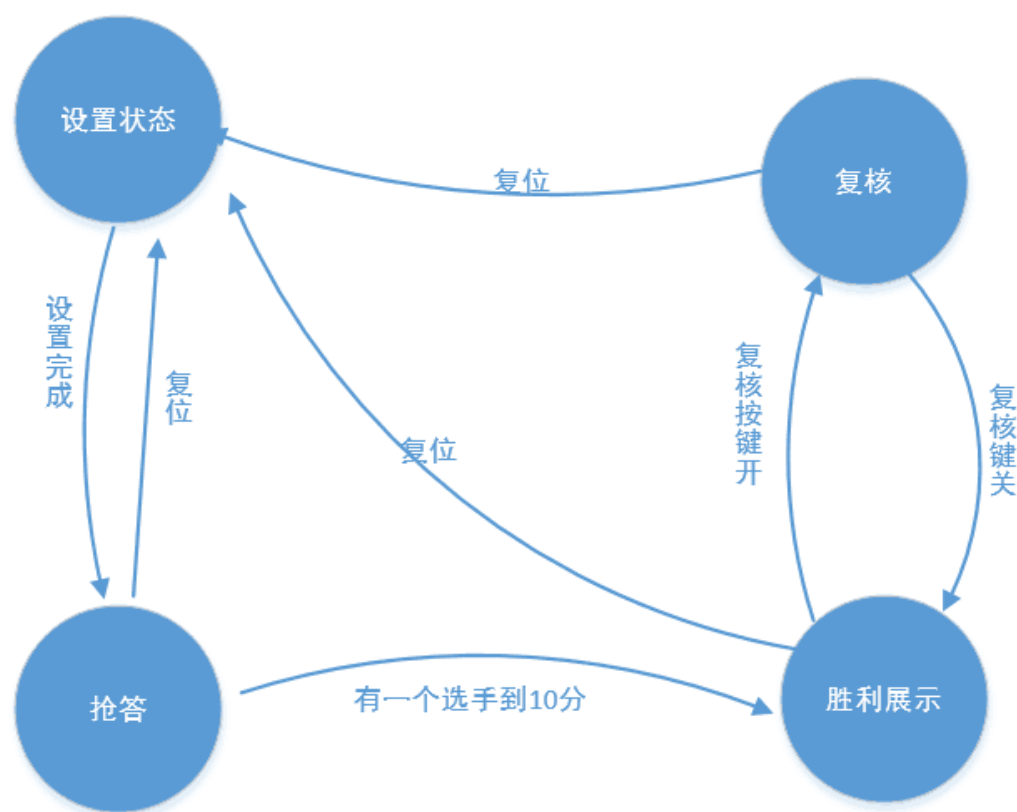
端口规格主要从板子文档说明和方便代码实现的角度来定制，具体看 2.2 的系统结构图，这里不多赘述。

2.2 系统结构图



2.3 系统执行流程

2.3.1 数据流向和状态转移图



2.3.2 伪代码

由于不太会写伪代码，这边直接大致描述一下功能的实现：

1. 胜利显示阶段：实质是一个模4计数器，当到4的时保持显示胜利选手。
2. 复核阶段：实质是一个模4计数器和模轮数计数器的组合，模4计数器相当于低位，模轮数计数器相当于高位。每一个小周期以此显示4位选手在第i论得分（大周期），显示当前周期。
3. 蜂鸣器功能：
 1. 在选手犯规时长鸣，直到选手复位，低音
 2. 在主持人开始按下（上升沿），鸣笛半秒，中音（演示时不知道要做，后来做了）
 3. 在可以抢答时（上升沿），选手抢答后，鸣笛半秒，高音
4. 赋分模块：
 1. 敏感于主持人赋分isAdd信号上升沿。
 2. 根据胜利的人数和主持人赋分进行加分，轮数记录。
5. 分数存储：
 1. 用8个8位宽的变量来分别存储四个选手的加减分，正分减负分为总分。
 2. 用4个128位宽的变量分别标记每个选手每轮得分，对于每个变量，每一轮先左移位2bit，再给抢答选手进行得分标记。
6. 设置阶段：调试好键盘输入即可，按照输入进入不同的阶段，设置即可。

2.4 子模块代码

见赋上的 codes 压缩包中 design file 文件夹中的文件

2.5 约束文件

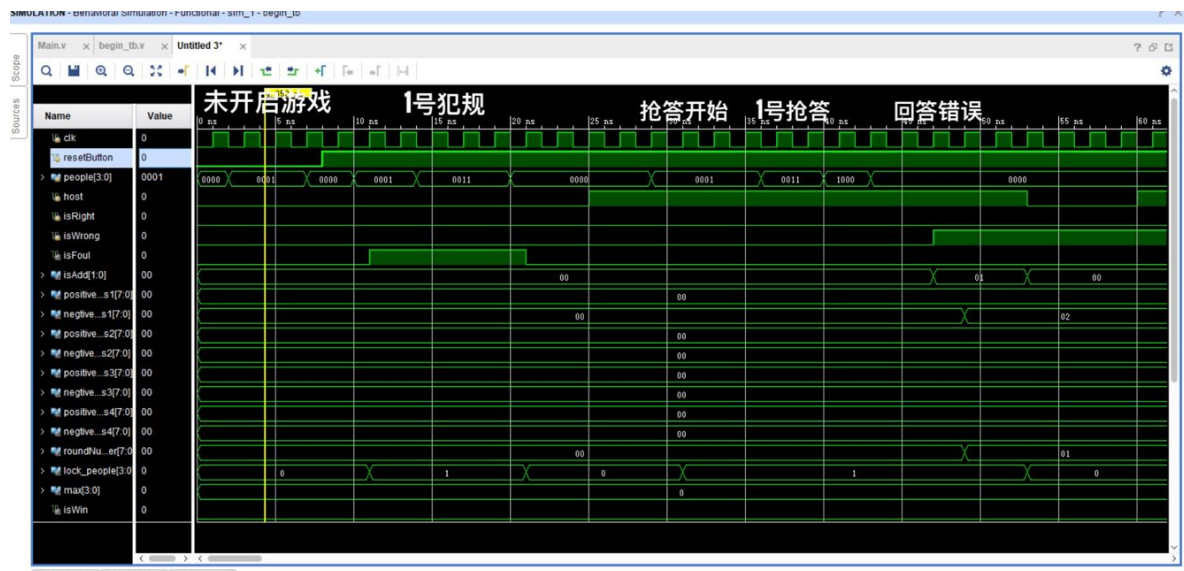
见赋上的 codes 压缩包中 constraint file 文件夹中的文件

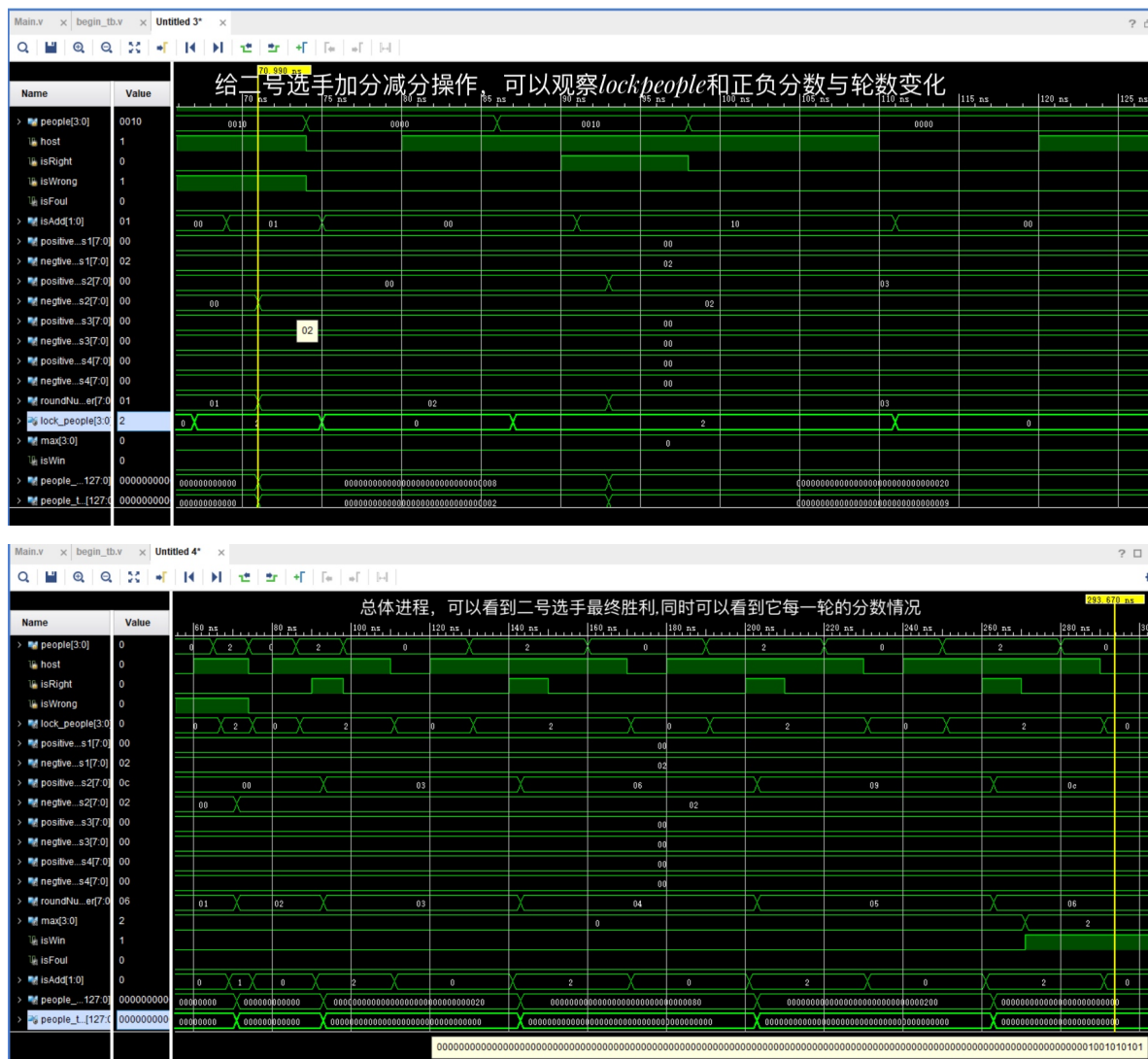
3. 测试

3.1 testbench 文件

见赋上的 codes 压缩包中 testbench 文件夹中的文件

3.2 子模块的仿真波形及测试结果





4. 总结

4.1 开发过程中遇到的问题及解决方案

分为各自的部分的难点和配合过程中遇到的问题来说：

4.1.1 在设置部分遇到的问题

1. **键盘使用的问题：**给的板子文档关于键盘使用没有做详细说明，花了一段时间琢磨如何使用键盘，发现给的文档没法用之后和别人讨论、自己琢磨琢磨之后知道了怎么用键盘。
2. **按键抖动的问题：**按键调试了很久都出现预料之外的结果，起初不知道是什么原因，后面知道了按键抖动之后因为有了方向很快就通过**按键消抖解决**，具体是通过 Google 和别的组交流知道如何进行按键消抖。

4.1.2 在抢答部分遇到的问题

1. **显示部分的分频问题：**最初七段数码管分频不当，结果一直没有显示。
2. **复合部分分数显示问题：**最初总是会显示分数出问题，一局可能显示多个选手的分数，或者有些局不会显示分数，这个问题延续到了最后，后来发现每个选手分数显示其实是一个组合逻辑，将它单独列出来，最终问题得到解决。
3. **时序问题：**有些需要锁存的使用时序电路的设计的没有用时序电路，比如isAdd，最初只是将它写组合逻辑部分，发现出现了很多混乱，后来用D锁存器锁存状态，解决问题，还有很多类似错误。
4. **分数储存：**最初一直在思考分数存储问题，主要是负分和每一轮选手得分怎么存，一开始想的是要使用sign reg来存，由于时间原因导致没有怎么弄懂，选择了用一个更低效的办法：将正分和负分

分别存，将正分减负分和负分减正分也分别存，最后通过判断得出结果。对于每一轮选手的得分，最初想的是通过二维数组来存，后来写了写没有实现，改用了分别给每个选手一个变量，然后难点在动态储存，就是每次轮数不同，而进行的轮数可能很多，无法以此列出，而在verilog中数组下标只能是常量，后来学到了移位寄存器，想到可以用移位来空出低比特位，通过总游戏轮数来反解出每一回合的分数。

5. **赋分不稳定**：最初赋分要么就有时赋分失败，有时就赋分错误，虽然基本稳定，但还是会有少量错误，后来通过网上学习，自己思考了一下，可以让赋分模块敏感于锁存的赋分情况的上升沿，结果就没有问题了。

4.1.3 由于没有配合好遇到的问题

1. **接口命名和规范问题**：在 12 月 14 日初步相互展示的过程中发现两个人的变量命名和 bit 位数对不上，发现是交流不充分不有效造成的，**发现这个问题之后两人对于这方面相当重视，一旦有问题就发消息交流，后面就没有出现这方面的问题。**
2. **分工问题**：这个其实也不算是问题，主要是反思，分工的时候两个人其实做了挺多同样的工作，也就是同一份工作做了两份工（比如分频模块，比如显示模块），这个首先是分工的时候没有分好工，其次是写的过程中互相没有交流，这次 project 没有机会改进了，但为下次的 project 积累了宝贵经验。

4.2 当前系统特色

1. **人性化的定制功能**：在设置步骤中，用户可以很简单知道如何进行设置，只需像用户说明每一个步骤是在设置什么，用户直接按键盘就可以设置，如果不是受限于显示屏有限其实可以更加人性化，每一步都可以提示。同时支持很大的可能定制：时间支持 0~99 s，参与者支持 2~4 人，加分支持 1~9 分（如果 ≤ 0 分将没有人能赢，所以，如果高于 9 分一轮就赢了，能做，但是我们感觉不好），减分支持 0~9 分。
2. **输入防错功能**：为了防止用于乱按按键，在设置步骤中，如果用户选择了不支持的值，将自动设置为默认值。
3. **人性化的显示**：比赛过程中分数滚动显示，很好了利用了有限的屏幕，同时符合实际场景。
4. **犯规设定**：联系生活中的抢答器，会有犯规设定，所以做了合理的拓展，其功能与实际相符合。
5. **蜂鸣器多功能鸣叫**：蜂鸣器可以支持主持人、选手抢答、选手犯规三种提示。
6. **按键合理性**：对于设置阶段，我们采用数字键盘输入，对于只有 0 和 1 的输入，我们采用开关进行输入。

4.3 优化方向

1. 抢答的步骤可以进一步实现按键抢答。
2. 设置步骤可以使用某些方式提示用户需要输入什么，这样就真的可以上手即用，符合我们最开始这样设计设置环节的目的。觉得可以用上板子的声音模块实现，播放录音什么的。
3. 代码可以重构一下，把两个人重复做的工作整合起来，更加精细的模块化。这样不仅便于后面加功能，还可以使得程序逻辑十分清楚。
4. 显示模块太冗杂，可以将显示模块分得更细一点特别是每个阶段的显示应该分模块写，其实本来是分模块写的，后来发现一个变量最后不要在多个 always 里实现，就把所有的显示加在一起，做完之后，发现可以在各个阶段给 seg 赋值不同的变量，而这些变量就可以通过 module 来传出。