# 课程设计说明模板

## 1、功能设计

### 1.1 概述

*（整体功能介绍，介绍包含哪些功能）*

### 1.2 详细描述

*（对功能进行分解，每项功能详细介绍，包括输入/操作、处理流程、输出/结果等）。*

## 2、体系结构设计

### 2.1 硬件体系结构

*（对课程设计项目的硬件体系结构进行描述，要有图，并进行详细解释）*

### 2.2 软件体系结构

*（对课程设计项目的软件体系结构进行描述，要有图，并进行详细解释）*

## 3、硬件模块设计

### 3.1模块1

输入：*（对系统输入进行介绍）*

输出：*（对系统输出进行介绍）*

处理过程：*（对硬件模块功能进行介绍）*

### 3.2模块2

输入：*（对系统输入进行介绍）*

输出：*（对系统输出进行介绍）*

处理过程：*（对硬件模块功能进行介绍）*

### 3.3模块3

输入：*（对系统输入进行介绍）*

输出：*（对系统输出进行介绍）*

处理过程：*（对硬件模块功能进行介绍）*

### 3.4……

## 4、软件模块设计

### 4.1 状态更新器时钟(clk\_50Hz)

输入：板载的时钟信号clk

输出：50Hz的时钟信号update\_clk

处理过程：

1、创建寄存器cnt；

2、每遇到clk的上升沿，若cnt大于等于1000000，则清零，且切换update\_clk的状态；否则将cnt加一。

### 4.2 VGA工作时钟(clk\_25MHz)

输入：板载的时钟信号clk

输出：25MHz的时钟信号vga\_clk

处理过程：

1、创建寄存器cnt；

2、每遇到clk的上升沿，若cnt大于等于2，则清零，且切换vga\_clk的状态；否则将cnt加一。

### 4.3 PS/2键盘输入驱动(keyboard\_driver)

输入：时钟信号clk，PS/2数据线ps2\_data，PS/2时钟线ps2\_clk，游戏状态state[1:0]

输出：游戏状态state[1:0]，方向键按键信号keys[1:0]

处理过程：

1. 检测PS/2数据线的信号，输出左右方向键的按键信号；
2. 若游戏状态为游戏开始前（0），且检测到任意键按下，则将游戏状态改为游戏进行中（1）；
3. 若游戏状态为游戏失败（2），且检测到任意键按下，则将游戏状态改为游戏进行中（1）；
4. 若游戏状态为游戏通关（3），且检测到任意键按下，则将游戏状态改为游戏开始前（0）；
5. 若游戏状态为游戏进行中，则进行以下操作：

* 若检测到左方向键按下，则在RAM中将角色的方向设为向左，keys[0]设为1；
* 若检测到右方向键按下，则在RAM中将角色的方向设为向右，keys[1]设为1；
* 若检测到左方向键松开，则将keys[0]设为0；
* 若检测到右方向键松开，则将keys[1]设为0；
* 若检测到跳跃键按下，且剩余跳跃次数大于0，则在RAM中将角色的垂直速度设为某个正整数值，垂直移动设为True，剩余跳跃次数减一；
* 若检测到射击键按下，则根据角色的当前位置和方向，在RAM中新增一个子弹实例，初始化其位置和水平速度，将子弹的数量加一。

6、无论处于何种游戏状态，若检测到返回键按下，则将游戏状态改为游戏开始前。

### 4.4 初始化实例(instance\_initializer)

输入：游戏状态state[1:0]

处理过程：

1、若游戏状态从游戏开始前变为游戏进行中，则在RAM中进行以下操作：

* 初始化角色的位置，设置其方向向右，垂直移动为False，垂直速度为0，剩余跳跃次数为2；
* 初始化子弹的数量为0；
* 初始化各水果的位置、触发方向与触发状态；
* 初始化通关按钮的位置；
* 初始化死亡次数为0；

2、若游戏状态从游戏失败变为游戏进行中，则进行除了上述最后一条外的所有操作。

### 4.5 位置更新模块(position\_updater)

输入：状态更新时钟update\_clk，游戏状态state[1:0]，方向键按键信号keys[1:0]

处理过程：每遇到update\_clk的上升沿，若游戏状态为进行中，则进行以下操作：

1. 若keys全为0或全为1，则不改变角色的横向坐标，否则使角色的横向坐标加上或减去一个固定值；
2. 从RAM中获取角色的垂直速度和垂直移动状态，若垂直移动状态为True，则根据其垂直速度改变角色的纵向坐标，并将垂直速度减去一个固定值，否则保持不变；

3、从RAM中获取每个子弹的位置和水平速度，据此改变其在RAM中的位置；

4、从RAM中获取每个水果的触发状态，若已触发，则根据其触发方向改变位置；若未触发，则根据角色的位置和水果的触发方向判断是否需要改变触发状态。

### 4.6 碰撞检测模块(collision\_checker)

输入：状态更新时钟update\_clk，游戏状态state[1:0]

输出：游戏状态state[1:0]

处理过程：每遇到update\_clk的上升沿，若游戏状态为进行中，则进行以下操作：

1、从RAM中获取角色的位置，若与地面保持碰撞，则将垂直速度设为0，垂直移动设为False，剩余跳跃次数设为2。若未与地面碰撞，则将垂直移动设为True；

2、从RAM中获取每个水果的位置，若与角色发生碰撞，则将游戏状态设为游戏失败，并将死亡次数加一；

3、从RAM中button的位置，若与角色发生碰撞，则将游戏状态设为游戏通关。

### 4.7 VGA屏幕输出(render)

输入：VGA工作时钟vga\_clk，游戏状态state[1:0]

输出：水平同步信号hsync，垂直同步信号vsync，颜色数据rgb\_data

处理过程：每遇到clk\_20ms的上升沿，则进行以下操作：

1. 若游戏状态为游戏开始前（0），则渲染游戏开始界面；
2. 若游戏状态为游戏进行中（1），则根据RAM中的位置、动作等数据，依次渲染游戏背景、陷阱、子弹和人物；
3. 若游戏状态为游戏失败（2），则依次渲染游戏失败界面和死亡次数；
4. 若游戏状态为游戏通关（3），则依次渲染游戏通关界面和死亡次数。

## 5、进度计划

*（对进度计划进行介绍……）*

包含各检查点要求

## 6、人员与分工

*（对团队成员及分工进行介绍……）*