

实验 35 推箱子游戏的设计

一、实验目的

- (1) 掌握复杂数字系统设计方法。
- (2) 掌握用图层设计动态图像界面。

二、点灯游戏简介

经典的推箱子是一个来自日本的古老游戏，目的是在训练你的逻辑思考能力。在一个狭小的仓库中，要求把所有的箱子都推到目标位置上。游戏规则主要如下。

1. 箱子只能推动而不能拉动，且一次只能推动一个箱子。
2. 搬运工不能跨越箱子。

推箱子游戏的规则虽简单,但稍不小心就会出现箱子无法移动或者通道被堵住的情况，所以需要巧妙的利用有限的空间和通道，合理安排移动的次序和位置，才能顺利的完成任务。

三、实验任务

1. 基本要求

(1) 界面设计如图35.1 所示，包括墙体在内，仓库最大面积为 8 行×8 列，界面主要元素有墙体、通道、箱子、搬运工、目标位置标记、信息区及命令按钮等。注意，当箱子位于目标位置时，箱子要变色以示区别。另外，界面的背景图中未画出，由读者自行设计。

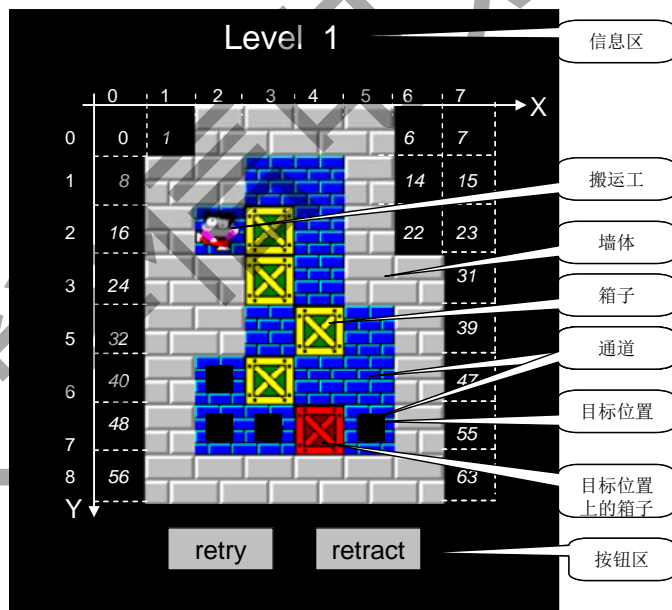


图 35.1 点灯游戏的界面

注意，本实验提供的墙体、通道、箱子、搬运工等图片 coe 文件均为 32 像素×32 像素。

(2) 游戏共有 4 关（其中 Level 0 为测试关），全部过关为胜利。信息区在游戏过程中显示游戏所处的关数 Level n (n 为 0~3)，全部过关后显示 Win。

(3) VGA 显示模式为 800×600@75Hz。

(4) 用 PS2 鼠标控制游戏进程。

①当箱子两边分别为搬运工和通道，即满足箱子移动条件时，用鼠标左键点击箱子，箱子和搬运工均前进一步。

②为设计简单起见，搬运工每次只能移动一步，用鼠标左键点击搬运工旁边的通道时，搬运工即移动到该位置。

③ 游戏至少可“悔”三步，鼠标左键点击“retract”按钮可“悔”一步。

④设置“retry”按钮，鼠标左键点击重玩本关。

⑤用鼠标右键可复位游戏。

2. 个性化要求

(1) 本实验对界面没有严格要求，可根据自己喜好设计界面。

(2) 增加信息显示内容，使游戏操作更为方便。

(3) 用键盘控制游戏进程。

四、实验原理

首先在游戏设置如图 35.1 所示的坐标，给每个小方格一个编号 n , $n = \{Y, X\}$ 。设置 wall、way、box、destination、man 等变量表示游戏所示的状态，各变量的含义表示如表 35.1 所示。

表 35.1 各游戏状态变量的含义

变量名称	变量说明	备注
wall [63:0]	wall [n]=1, 表示编号为 n 的小方格为墙体	每关初始化时更新
destination [63:0]	destination [n]=1, 表示编号为 n 的小方格为目标位置	每关初始化时更新
box 63:0]	box [n]=1, 表示编号为 n 的小方格为箱子	
way [63:0]	way [n]=1, 表示编号为 n 的小方格为通道, 注意, 未被箱子占居的目标位置也为通道	
man [5:0]	搬运工的位置, 即搬运工所处小方格的编号	

根据实验任务可画出系统的原理框图, 如图 35.2 所示。PS2 接口电路和鼠标数据处理电路负责与 PS2 鼠标接口通信并翻译鼠标移动及按键状态; 游戏进程控制是电路的核心, 在鼠标的控制下完成箱子、搬运工等游戏过程的控制, DCM 模块为二分频电路, 产生 50MHz 像素时钟, 该时钟同时作为系统的工作时钟; SVGA 视频同步发生器和显示电路负责将游戏进程及结果显示在 VGA 显示器上。

1. PS2 接口电路和鼠标数据处理电路

PS2 接口电路即为实验 14 介绍的 PS2 鼠标接口模块, 鼠标数据处理电路需完成下列功能。

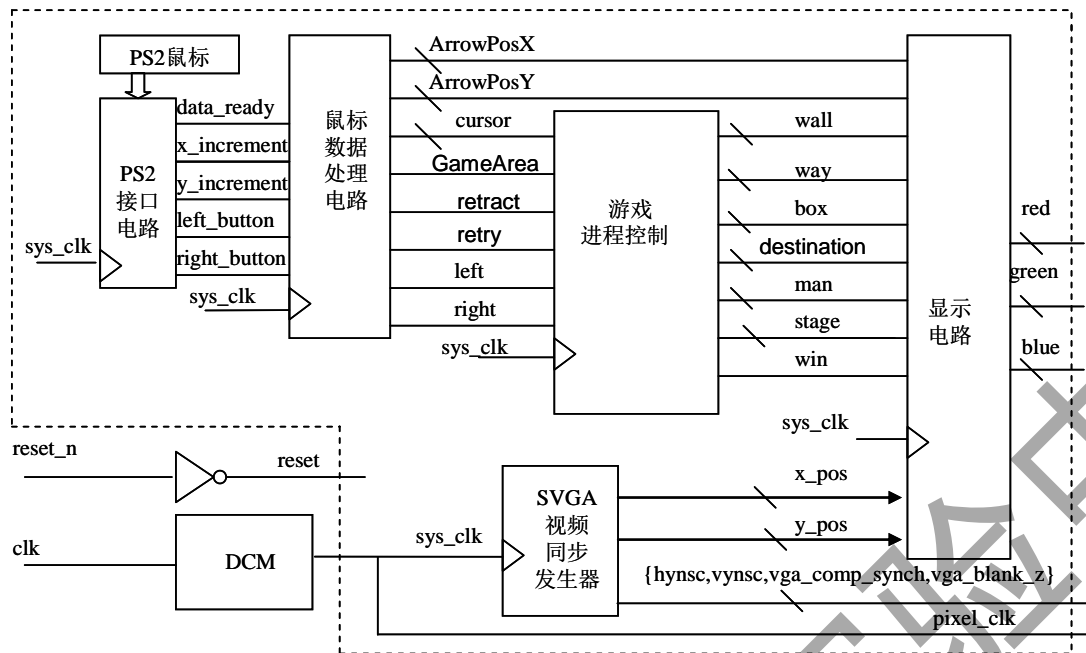
(1) 根据鼠标移动距离修改鼠标光标位置坐标 (ArrowPosX, ArrowPosY)。

(2) 根据鼠标光标位置给出 cursor、GameArea、retract 及 retry 变量的值, 上述各变量的含义如表 35.2 所示。

表 35.2 mouse 控制变量的含义

变量名称	变量说明	备注
cursor [5:0]	当鼠标光标处在游戏区域时, 鼠标光标所处的小方格编号。	当光标处在游戏区域外, 该变量无效
GameArea	当鼠标光标处在游戏区域时, GameArea=1, 否则 GameArea=0	
retract	当鼠标光标处在 retract 按钮时, retract=1, 否则 retract=0	
retry	当鼠标光标处在 retract 按钮时, retry=1, 否则 retry=0	

(3) 当鼠标有左键 (右键) 按下, 给出一个时钟周期宽度的 left (right) 脉冲。

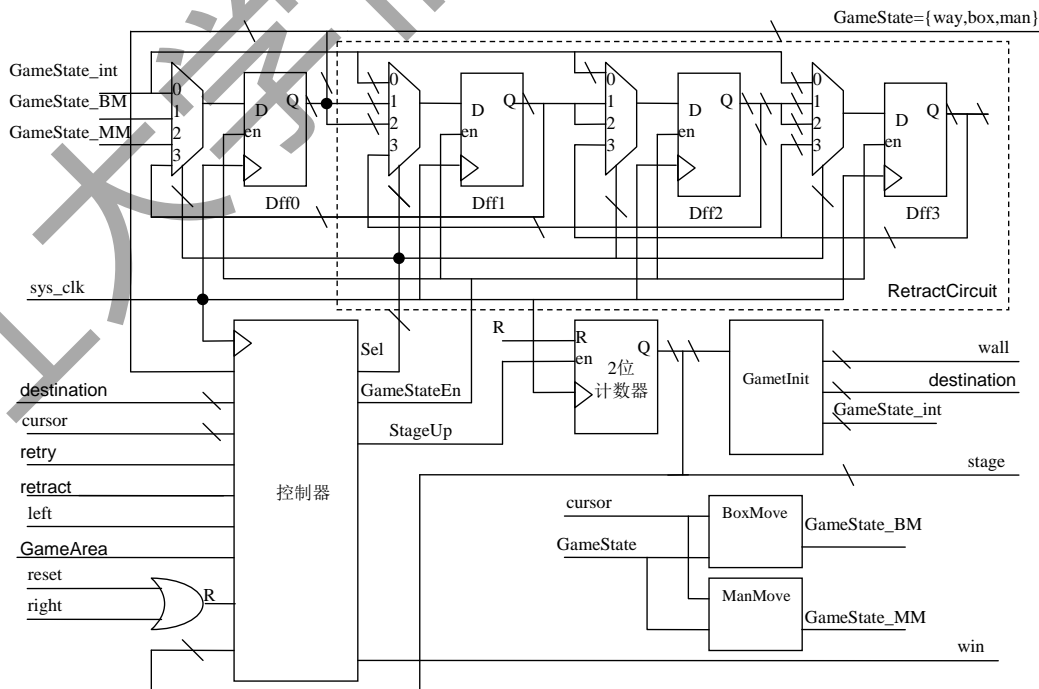


2. 游戏进程控制模块的设计

游戏进程控制模块原理框图如图 35.3 所示, 控制器是电路的核心, 控制着游戏的进程; 初始化电路 (GameInit)、搬箱次态电路 (BoxMove)、搬运工移动次态电路 (ManMove) 及悔棋电路 (RetractCircuit) 给出游戏下一个各种可能状态, 控制器根据鼠标操作输出相应的 Sel、GameStateEn 来更新游戏状态。

根据游戏规则和实验要求,可画出图 35.4 所示的控制器的算法流程图。复位并初始后进入后等待按键输入状态 (Wait), 然后判断是否过关, 若未过关且鼠标左键键入, 则根据鼠标所在的区域进入相应功能处理状态, 再回到 Wait 状态; 若过关, 则根据游戏的进程进入暂停状态 (Pause) 或游戏结束状态 (Over)。如在 Pause 状态, 则需要在按 Enter 键后进入下一关初始化状态 (Next)。

这里特别说明一下，ASM 中的 Reset 和 Interim 两个状态是非必要，仅仅为了 ASM 图的可读性及 Verilog HDL 代码编写方便而增加的。



$$\left\{ \begin{array}{l} \text{wall} = 64'h3828_2fe1_87f4_141c; \\ \text{destination} = 64'h0010_0002_4000_0800; \\ \text{way_int} = 64'h0010_001A_5008_0800; \\ \text{box_int} = 64'h0000_1004_2800_0000; \\ \text{man_int} = \{3'o4, 3'o4\} \end{array} \right. \quad (35.1)$$

```

wall      = 64'h7e42_4246_6622_263c;
destination = 64'h003c_0400_0000_0000;
way_int    = 64'h002c_3428_1014_1800;
box_int    = 64'h0010_0810_0808_0000;
man_int    = {3'o2,3'o2};

```

(35.2)

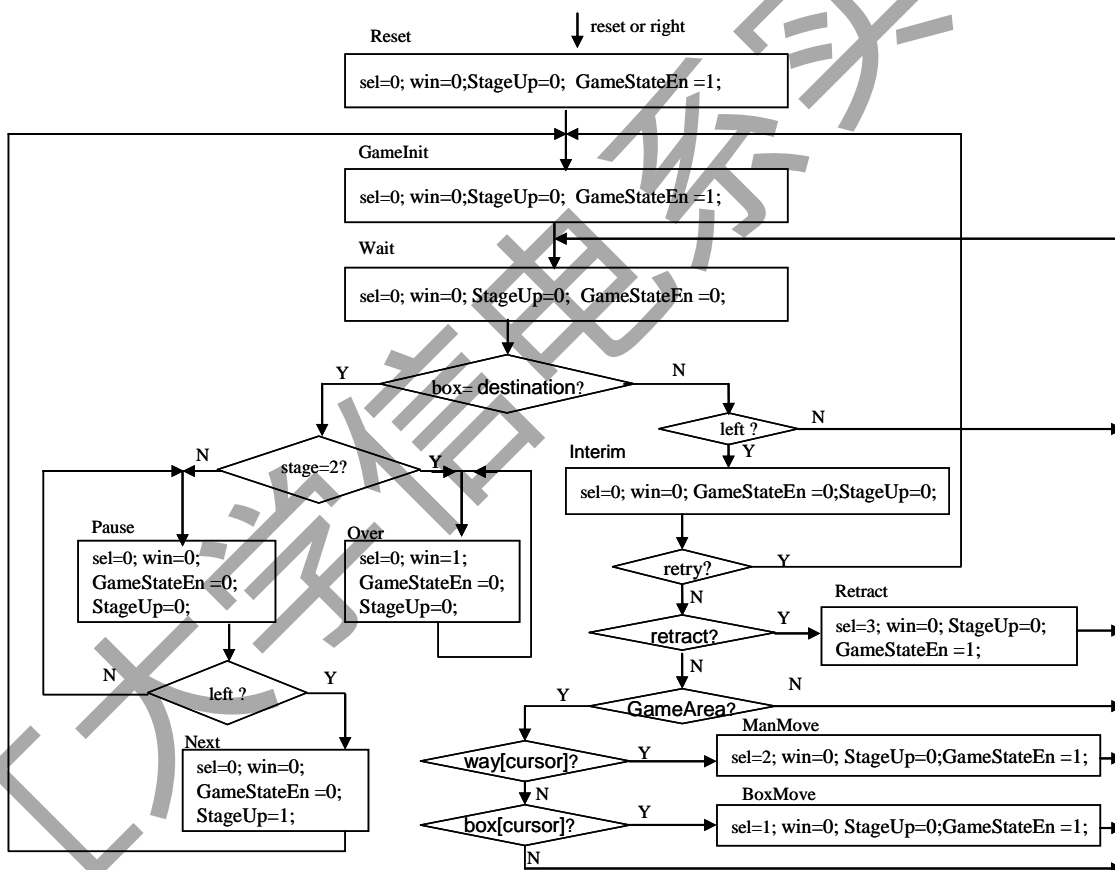


图 35.4 控制器的 ASM 图

3. 显示电路的设计

采用多图层概念可简化显示电路的设计。图层划分方法较多,在此作者建议将图 35.1 所示的界面划分为四个图层,底层为信息区和两命令按钮;中间 1 层包括墙体、通道、箱子、目标位置标记按钮;中间 2 层为搬运工,如读者能力较强,可将搬运工设计为动态显示(般运动作);而顶层则鼠标光标(箭头)。这里说明一下,在中间 1 层应将搬运工所在位置显示为通道。优先级为从高到低分别为顶层、中间 2 层、中间 1 层和底层。

根据上述图层划分方法,可画出显示模块的原理框图,如图 35.5 所示。各图层的设计方法均已在前面的实验中有所涉及,这里不再赘述。

注意,界面中的各元素大小和颜色不作规定,读者可根据自己的喜处理。

五、实验设备

- (1) 装有 ISE、ModelSim SE 和 ChipScope Pro 软件的计算机。
- (2) XUP Virtex-II Pro 开发系统一套。
- (3) PS2 鼠标一个, SVGA 显示器一台。

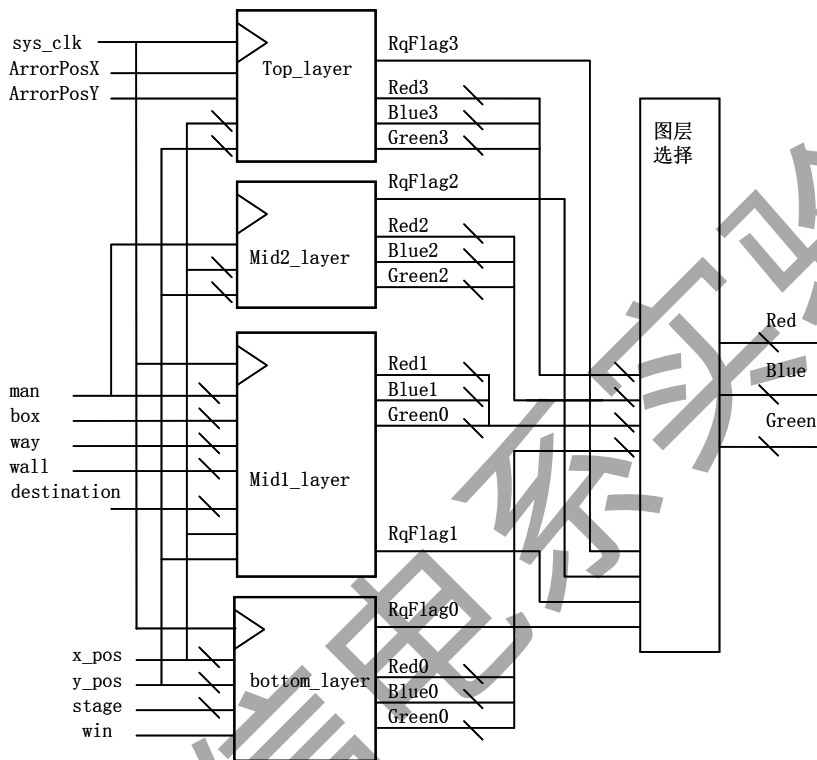


图 35.5 显示模块的原理框图

六、提供的文件

- (1) 小字库 zk.coe, 依次存放“0”、“1”、“2”、“3”、“4”、“5”、“L”、“R”、“W”、“a”、“c”、“e”、“i”、“l”、“n”、“r”、“t”、“v”和“y”等 18 个字符的 16×8 点阵。
- (2) wall.coe、way.coe、box.coe 和 man.coe 四个文件分别墙体、通道、箱子和搬运工等图片 coe 文件, 注意 man.coe 文件中 FFFFCC 表示透明色。

七、预习

- (1) 查阅相关资料, 了解鼠标工作原理和动态显示工作原理。
- (2) 查阅相关资料, 了解 coe 文件的作用和生成方法。

八、实验内容

- (1) 编写各模块的 Verilog HDL 代码及其测试代码, 并用 ModelSim 仿真。
- (2) 编写推箱子实验的 top 文件及其测试代码, 并用 ModelSim 仿真产生 rgb 文件, 模拟显示复位后的一帧图像。
- (3) 建立推箱子游戏实验 ISE 工程, 并对工程进行综合、约束、实现并下载至实验开发板中。FPGA 引脚约束内容如表 35.3 所示。
- (4) 将 SVGA 显示器、键盘接入 XUP Virtex-II Pro 开发实验板中, 试玩游戏并验证实验结果。

表 35.3 FPGA 引脚约束内容

引脚名称	I/O	引脚编号	说明
clk	Input	AJ15	系统 100MHz 主时钟
reset_n	Input	AG5	Enter 按键
red [0]	Ouput	G8	红基色
red [1]	Ouput	H9	
red [2]	Ouput	G9	红基色
red [3]	Ouput	F9	
red [4]	Ouput	F10	
red [5]	Ouput	D7	
red [6]	Ouput	C7	
red [7]	Ouput	H10	
green[0]	Ouput	G10	绿基色
green[1]	Ouput	E10	
green[2]	Ouput	D10	
green[3]	Ouput	D8	
green[4]	Ouput	C8	
green[5]	Ouput	H11	
green[6]	Ouput	G11	
green[7]	Ouput	E11	
blue[0]	Output	D15	蓝基色
blue[1]	Ouput	E15	
blue [2]	Ouput	H15	
blue [3]	Ouput	J15	
blue [4]	Ouput	C13	
blue [5]	Ouput	D13	
blue [6]	Ouput	D14	
blue [7]	Ouput	E14	
pixel_clk	Ouput	H12	像素时钟
vga_comp_synth	Ouput	G12	
vga_blank _z	Ouput	A8	消隐信号
hsync	Ouput	B8	行同步信号
vsync	Ouput	D11	帧同步信号
ps2_clk	inout	Ad6	
ps2_data	inout	Ad5	

九、实验报告要求

- (1) 写出设计原理、列出 Verilog HDL 代码并对设计作适当说明。
- (2) 记录 ModelSim 仿真波形，并对仿真波形作适当解释，分析是否符合预期功能。
- (3) 记录实验结果，分析设计是否正确。
- (4) 记录实验中碰到的问题和解决方法。

十、思考题

1. 怎样修改向量中其中一位？
2. 本实验方案中，对一些如有左击箱子但推不动箱子等无效操作，也进行无必要的游戏状态的保存。怎样修改设计，以减少这些不必要的资源浪费。