

# 硬件课程设计实验指导书

## 一、课程性质与目的

《硬件课程设计》是计算机科学与技术专业的专业必修核心课程。适用于计算机科学与技术专业。是计算机科学与技术专业重要的实践环节。通过本课程的学习，使学生基本掌握硬件编程语言 Verilog，并熟悉 Vivado 平台的使用方法。通过设计、编程、调试过程，培养学生的设计和实验动手能力。

## 二、教学基本内容及基本要求

- 1、学习现代硬件系统设计基本方法。
- 2、学习使用硬件描述语言 Verilog。
- 3、掌握 Verilog 分层次设计方法。
- 4、能够将一个程序封装成 IP，并能够调用已封装好的 IP。
- 5、能够用有限状态机描述应用问题。
- 6、熟练使用 Vivado 平台。
- 7、详细的设计任务要求见附件 1。

## 三、时间安排

课程设计教学与实验时间共 5 周。

第 19 周～第 20 周：理论教学与 Vivado 平台指导。

第 25 周～第 27 周：课程设计与 Verilog 程序调试和验证。

## 四、参考书目

1. 王金明，数字系统设计与 Verilog HDL（Vivado 版）。电子工业出版社。2020，3
2. 张冬冬等，数字逻辑与组成原理实践教程。清华大学出版社。2018，8
3. 高小鹏，计算机组成与实现。高等教育出版社。2018，12
4. 雷思磊，自己动手写 CPU。电子工业出版社。2014，9
5. 汪文祥，CPU 设计实战。机械工业出版社。2021，4
6. 吴继华、王诚，Verilog 设计与验证。人民邮电出版社。2006，8

## 五、成绩评定

成绩由平时听课与实验状况、设计与调试过程和结果、实验报告三部分构成。

平时成绩 50%=平时考勤+平时实验+平时成绩 50%×(50%~80%)(完成选做 16 道题情况)

期末成绩 50%=调试+验机+实验报告

总评=平时成绩 50%+期末成绩 50%

考核依据：

(1) 自己选做 16 道题，需调试和功能仿真通过，提交程序和仿真程序。

(2) 实验与设计结果。该部分由学生独立完成具体的设计项目，完成后由教师验收，根据验收情况和设计结果给出相应的成绩。

(3) 设计报告。在实验完成的基础上，学生根据自己设计的实验与调试过程，每人写出自己的设计报告，根据设计报告的情况由指导老师评定并给出得分。

## 六、要求

在课程设计阶段，严格考勤，根据题目要求认真独立完成设计与调试，要求学生全程参与课程设计过程，完成理论到实验的全过程训练。在实验过程中，独立完成设计任务并调试，根据设计结果和设计过程中的努力程度，由指导教师给出设计成绩。在实验过程完成的基础上，撰写设计报告，并于设计结束时将报告上交。根据情况，具体做如下规定：

1) 课程设计成绩以百分制计。

2) 要求学生独立完成自己的设计和实验任务。有互相拷贝和抄袭的，涉及到的相关同学均在该部分得 0 分。

3) 完成实验和设计后，各自独立完成设计报告的撰写工作。发现设计报告有相互拷贝的，设计报告一项一律记为 0 分。

计算机硬件课程设计指导组 吴磊 修订 2022-6-12

## 附件 1 硬件课程设计任务

本次硬件课程设在集中学习 Verilog 语言编程基础上，每人独立完成实验和调试，在 Vivado 环境下使用 Verilog 语言，编程调试并对 FPGA 编程。每位同学需独立完成下列 5 个必做的实验项目。

### 硬件课程设计题目

- 1 在 7 段数码管上显示自己的最后三位学号数字。
- 2 在点阵上显示自己的名字。
- 3 能将程序封装为 IP，既能调用自己封装的 IP，也能调用库中的 IP。

学号 尾数	题目
0	调用自编写并封装的两位数二进制加法器的 IP 核，生成多位数的加法器。
1	调用自编写并封装的两位数二进制的减法器的 IP 核，生成多位数的减法器。
2	调用自编写并封装的两位数二进制的乘法器的 IP 核，生成多位数的乘法器。
3	调用自编写并封装的两位数二进制的多路选择器的 IP 核，生成多位数的多路选择器。
4	调用自编写并封装的两位数二进制的 2—4 译码器的 IP 核，生成多位数的译码器。
5	调用自编写并封装的两位数二进制的比较器的 IP 核，生成多位数的比较器。
6	调用自编写并封装的两位数二进制的偶校验器的 IP 核，生成多位数的偶校验器。
7	调用自编写并封装的两位数二进制的计数器的 IP 核，生成多位数的计数器。
8	调用自编写并封装的两位数十进制的加法器的 IP 核，生成多位数的十进制的加法器。
9	调用自编写并封装的两位数十进制的减法器的 IP 核，生成多位数的十进制的减法器。

### 4 序列检测器

根据学号尾数，用有限状态机，分别设计带重复判断和不带重复判断的两个简单的序列检测器。

学号尾数	题目
0	检测特定序列的二进制数 001
1	检测特定序列的二进制数 010
2	检测特定序列的二进制数 011
3	检测特定序列的二进制数 100
4	检测特定序列的二进制数 101
5	检测特定序列的二进制数 110
6	检测特定序列的二进制数 111
7	检测特定序列的二进制数 01
8	检测特定序列的二进制数 10
9	检测特定序列的二进制数 11

### 5 简单 CPU 的设计

## 附件 2 实验设计参考及实验报告要求

### 一、在 7 段数码管上显示自己的最后三位学号数字

#### 1、实验任务

要求将自己的最后三位学号数字在数码管上显示。

#### 2、实验现象与结果

应能看见自己学号稳定的在数码管上显示，并用手机拍摄记录下来。

#### 3、实验报告

- (1) 简单陈述在数码管上显示数字的原理。
- (2) 画出编写 Verilog 程序的功能框图。
- (3) 说明改变扫描频率的软件方法，及对显示的影响。
- (4) 附上 Verilog 实验源程序，并对关键或重要的程序语句给出简要的注释和说明。
- (5) 将显示的学号截图附上。
- (6) 如果调试过程遇到了问题，应将问题截图，并给出解决问题的办法和对应的截图。

#### 4、参考提示

- ①. 王金明，数字系统设计与 Verilog HDL（Vivado 版）。电子工业出版社。2020，3
- ②. 张冬冬等，数字逻辑与组成原理实践教程。清华大学出版社。2018，8
- ③. 吴继华、王诚，Verilog 设计与验证。人民邮电出版社。2006，8

### 二、在点阵上显示自己的名字

#### 1、实验任务

在 16×16 点阵上，编写 Verilog 程序实现循环显示你自己的“名字”的汉字。

#### 2、实验结果与现象

在点阵模块上循环依次显示你自己的“名字”，每个汉字显示的时间约为 0.5 秒。

#### 3、问题与思考

- (1) 怎样让汉字左旋转和右旋转。
- (2) 怎样让汉字左移动和右移动。
- (3) Verilog 程序中哪部分，影响每一列的显示时间？
- (4) Verilog 程序中哪部分，影响每一个字的显示时间？

#### 4、实验报告

- (1) 简单陈述在 16×16 点阵上显示汉字的原理。

- (2) 画出编写 Verilog 程序的功能框图。
- (3) 附上实验源程序，并对关键或重要的 Verilog 程序语句给出简要的注释和说明。
- (4) 对影响每一列扫描快慢的 Verilog 关键程序部分给予简要的解释和说明。
- (5) 对影响名字扫描快慢的 Verilog 关键程序部分给予简要的解释和说明。
- (6) 附上显示的名字截图。
- (7) 如果调试过程遇到了问题，应将问题截图，并给出解决问题的办法和对应的截图。

## 5、参考提示

- ①. 王金明，数字系统设计与 Verilog HDL（Vivado 版）。电子工业出版社。2020，3
- ②. 张冬冬等，数字逻辑与组成原理实践教程。清华大学出版社。2018，8
- ③. 吴继华、王诚，Verilog 设计与验证。人民邮电出版社。2006，8

## 三、能将程序封装为 IP，既能调用自己封装的 IP，也能调用库中的 IP。

### 1、实验题目

根据不同的学号尾数，做相应的题目。

学号 尾数	题目
0	调用自编写并封装的两位数二进制加法器的 IP 核，生成多位数的加法器。
1	调用自编写并封装的两位数二进制的减法器的 IP 核，生成多位数的减法器。
2	调用自编写并封装的两位数二进制的乘法器的 IP 核，生成多位数的乘法器。
3	调用自编写并封装的两位数二进制的多路选择器的 IP 核，生成多位数的多路选择器。
4	调用自编写并封装的两位数二进制的 2—4 译码器的 IP 核，生成多位数的译码器。
5	调用自编写并封装的两位数二进制的比较器的 IP 核，生成多位数的比较器。
6	调用自编写并封装的两位数二进制的偶校验器的 IP 核，生成多位数的偶校验器。
7	调用自编写并封装的两位数二进制的计数器的 IP 核，生成多位数的计数器。
8	调用自编写并封装的两位数十进制的加法器的 IP 核，生成多位数的十进制的加法器。
9	调用自编写并封装的两位数十进制的减法器的 IP 核，生成多位数的十进制的减法器。

### 2、实现步骤：

先用 Verilog 实现一个器件，并封装成 IP。然后，调用刚才的 IP 核，生成新的器件。

### 3、实验要求：

对新生成的器件，

- ① 进行仿真测试并截图；
- ② 管脚绑定，下载到试验箱进行验证，并将验证结果截图保留。

### 4、问题与思考

(1) 在 Verilog 程序中如何调用库中的 IP 核。

## 5、实验报告

(1) 根据题目要求，编写要生成器件的 Verilog 程序。

(2) 附上 Verilog 程序对应的电路图。

(3) 编写仿真程序，并对仿真波形用线段或带箭头的线段标注输出与各个输入之间的关系，并给予简单的解析和说明。

(4) 调用刚才的 IP 核，生成新的器件。然后，管脚绑定，下载到试验箱进行验证，并将验证结果截图保留。

(5) 如果调试过程遇到了问题，应将问题截图，并给出解决问题的办法和对应的截图。

## 6、参考提示

①. 王金明，数字系统设计与 Verilog HDL（Vivado 版）。电子工业出版社。2020，3

②. 张冬冬等，数字逻辑与组成原理实践教程。清华大学出版社。2018，8

4.4 IP 核封装及模块化设计 P119~P146

③. 吴继华、王诚，Verilog 设计与验证。人民邮电出版社。2006，8

# 四、序列检测器

## 1、序列检测器原理

1) 序列检测器在很多数字系统中都不可缺少，尤其是在通信系统当中。序列检测器的作用就是从一系列的码流中找出希望出现的序列，序列可长可短。比如在通信系统中，数据流帧头的检测就属于一个序列检测器。序列检测器的类型有很多种，有逐个比特比较的，有逐个字节比较的，也有其它的比较方式，实际应用中采用何种比较方式，主要是看序列的多少以及系统的延时要求。

2) 逐个比特比较的序列检测器，是将一个二进制码流与特定序列比较。首先将二进制码流与特定序列的第一个码比较，如果二进制码流与特定序列的第一个码相同，那么再和特定的序列的第二个码相比较，依次比较下去，直到二进制码流和特定序列相一致，就认为检测到一个特定序列。

3) 逐个比特比较的序列检测分为两种情形：

①不进行重复判断

假设，通过拨挡开关 K1~K8，输入一个八位二进制数据 X=10101010。

检测在  $X$  中包含了几个不重复的特定序列  $Y=101$ , 并在输出  $Z$  的指定位置上显示。

$X = 10101010$

$Z = 00100010$

也就是说, 在  $Z$  中包含了几不重复的特定序列  $Y$  有 2 个。

## ②进行重复判断

假设, 通过拨挡开关  $K1 \sim K8$ , 输入一个八位二进制数据  $X=10101010$ 。

检测在  $X$  中包含了几个重复的特定序列  $Y=101$ , 并在输出  $Z$  的指定位置上显示。

$X = 10101010$

$Z = 00101010$

也就是说, 在  $Z$  中包含了几重复的特定序列  $Y$  有 3 个。

## 2、状态机 (State Machine)

### 1) 为什么使用状态机

- (1) 有限状态机克服了纯硬件数字系统顺序方式控制不灵活的缺点。
- (2) 设计方案相对固定, 结构模式简单, 可定义符号化枚举类型的状态。
- (3) 状态机的 Verilog 描述层次分明, 结构清晰, 易读易懂。
- (4) 状态机容易构成性能良好的同步时序逻辑模块。
- (5) 基于有限状态机技术设计的控制器其工作速度大大优于 CPU。
- (6) 就可靠性而言, 基于有限状态机技术设计的控制器其可靠性优于 CPU。

### 2) 状态机是一类很重要的时序电路

时序电路是很多数字电路的核心部件, 是大型电子设计的基础。

状态机相当于一个控制器, 它将一项功能的完成分解为若干步, 每一步对应于二进制的状态, 通过预先设计的顺序在各状态之间进行转换, 状态转换的过程就是实现逻辑功能的过程。

状态机图是指用图形的方式来表示一个设计实体的各种工作状态、内部各工作状态转换的条件以及各工作状态对应的输出信号序列。

## 3、用状态机描述如何进行序列检测

例子: 设计检测二进制输入序列的检测电路, 当输入序列中连续输入 4 位数码均为 1 时, 电路输出为 1, 其它为 0。

### 第一步: 建立原始状态图



首先，根据设计要求，分析清楚电路的输入和输出，确定有多少种输入信息需要“记忆”，对每一种需“记忆”的输入信息用一种状态来表示，根据输入条件和输出要求确定各状态之间的关系，从而构成原始状态图。

本例电路有一个输入端 X，接收被检测的串行输入的二进制序列；有一个输出端 Z。电路的工作与输入序列必须同步。



根据检测要求，当连续输入 4 个 1 的二进制序列时，输出为 1，其余情况输出皆为 0。

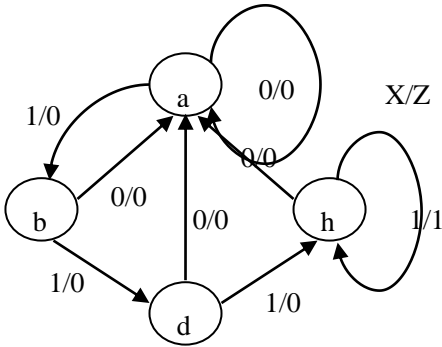
所以，该电路必须“记忆”3 位连续输入的序列：

一共有 8 种情况，即 000、100、010、110、001、101、011、111。

只有当 3 位连续输入为 111，第 4 位也输入 1 时，输出才为 1。

### 第二步：用状态图描述

用状态图描述如下图所示：



或

### 第二步：用状态转移表描述

次态/Z 初态 X	a	b	d	h
0	a/0	a/0	a/0	a/0
1	b/0	d/0	h/0	h/1

### 第三步：对状态进行编码

一般有热独(ONE-HOT)编码和二进制编码。

本例中，进行热独编码：a 为 0001，b 为 0010，c 为 0100，d 为 1000。

#### 第四步：状态机选型

从输出时序上看，Mealy 型状态机的输出是当前状态和所有输入信号的函数，它的输出是在输入变化后立即发生的，不依赖时钟的同步。Moore 型状态机的输出仅为当前状态的函数，状态机的输入变化必须与状态机的时钟同步。

#### 第五步：通过 Verilog 代码描述

### 4、通过时序图进行验证

时序图反映了输入输出信号随时间变化的情况。可以帮助理解模块的功能，直观的反映信号的变化及其相互关系。

#### 时序图的问题：

①要将模块的所有功能描述出来，时序图要把输入信号和输出信号的所有可能状态都反映出来，这样需要时序图的周期个数要尽可能的多些，输入信号的变化尽可能的多些。②时序图一般用来辅助理解模块的功能，模块一般需要文字或状态机的功能描述。③时序图不便于 HDL 语言的描述。

### 5、实验内容

#### 输入方式与输出显示提示：

本实验就是要设计一个序列检测器，要求检测的序列长度为 8 位，实验中用拨挡开关的 K1~K8 作为外部二进制码流的输入，与特定序列逐个比较。在 LEDi 灯上显示检测的结果。同时用按键模块的 S1 作为启动检测信号，每按下 S1 一次，检测器检测一次。

例如：用拨挡开关 K1~K8 随机输入二进制码流 1 0 1 0 1 0 1 1

#### 检测特定序列 1 0 1

① 带重复检测结果是：

LED1	LED2	LED3	LED4	LED5	LED6	LED7	LED8
灭	灭	亮	灭	亮	灭	亮	灭

在七段码灯上显示检测到 3 个特定的二进制数字。



② 不带重复检测结果是：

LED1 LED2 LED3 LED4 LED5 LED6 LED7 LED8

灭 灭 亮 灭 灭 灭 亮 灭

在七段码灯上显示检测到 2 个特定的二进制数字。



③ 如果检测结果为零，即随机输入的二进制码流中不包含特定序列二进制数字，那么 LEDi 显示如下：

LED1 LED2 LED3 LED4 LED5 LED6 LED7 LED8

灭 灭 灭 灭 灭 灭 灭 灭

在七段码灯上显示 0。



在八位连续数据检测结束之后，需要对状态机内部进行复位操作。同时用按键模块的 S1 作为启动检测信号，每按下 S1 一次，检测器检测一次。

## 6、问题与思考

- 1) 如何保证状态机在初始时状态为 S0?
- 2) 在状态机跑飞，即脱离有效状态 (S0—S3) 时，如何使状态机能恢复工作。

## 7、实验报告

- (1) 分别简单陈述带重复判断和不带重复判断检测特定二进制数字系列的原理。
- (2) 分别简单陈述用有限状态机描述带重复判断和不带重复判断检测特定二进制数字的原理。并分别给出对应的两个状态迁移图和两个状态迁移表。
- (3) 画出程序功能框图，然后，根据程序框图，编写 Verilog 程序。
- (4) 对 Verilog 程序关键或重要的程序语句给出简要的注释和说明。
- (5) 分别对带重复判断和不带重复判断检测特定二进制数字系列进行仿真，对仿真波形用线段或带箭头的线段标注输出与各个输入之间的关系，并给予简单的解析和说明。
- (6) 如果调试过程遇到了问题，应将问题截图，并给出解决问题的办法和对应的截图。

## 8、参考提示

- ①. 王金明，数字系统设计与 Verilog HDL (Vivado 版)。电子工业出版社。2020，3
- ②. 张冬冬等，数字逻辑与组成原理实践教程。清华大学出版社。2018，8
- ③. 吴继华、王诚，Verilog 设计与验证。人民邮电出版社。2006，8

## 五、简单 CPU 的设计

### 1、实验报告要求

#### (1) 简单陈述 CPU 指令系统

设计 CPU 就是从设计或分析 CPU 指令系统开始的，因此，对 CPU 指令系统理解的越深入，越有助于 CPU 的实现。

- ① 对 CPU 指令进行分类并画出对应的指令分类表
- ② 对 CPU 的操作码进行详细的描述
- ③ 对 CPU 的功能码进行详细的描述
- ④ 对 CPU 的操作数进行分类并画出对应的操作数分类表
- ⑤ 对 CPU 的操作数涉及的寻址方式进行分类并画出对应的寻址方式分类表
- ⑥ 对存储器进行存、取的过程进行简单的描述

#### (2) 每一类的一条 CPU 指令的实现

在仔细分析 CPU 指令系统的基础上，用 Verilog 编程实现每一类的一条指令。

##### ① 用 Verilog 实现每一类的一条指令，通常需要陈述下面一些基本的内容：

- 1) 指令名称：
- 2) 指令描述：
- 3) 指令功能：
- 4) 指令格式：①编码形式②操作码和功能码及操作数的长度
- 5) 寻址方式：
- 6) 指令的执行流程，给出指令执行流程图

#### (3) 画出指令实现的数据流图

#### (4) 画出主要的功能部件，

如 PC、IR、MAR、MDR 等功能部件的框图，需要对功能部件的输入/输出引脚一一标注清楚。

#### (5) 每一类的一条 CPU 指令的测试

编写测试每一类的一条 CPU 指令的代码，对每条指令测试并截图；对测试波形用线段或带箭头的线段标注，并给予简单的解析和说明。

#### (6) 每一类的一条 CPU 指令的仿真

对每一类的一条 CPU 指令进行仿真并截图；对仿真波形用线段或带箭头的线段标注，并给予简单的解析和说明。

(7) 如果调试过程遇到了问题，应将问题截图，并给出解决问题的途径和方法及对应的截图。

(8) 附上 Verilog 程序源代码，标注实现的功能，对 Verilog 程序关键或重要的语句给出简要的注释和说明。

## 2、参考提示

- ①. 王金明，数字系统设计与 Verilog HDL（Vivado 版）。电子工业出版社。2020，3
- ②. 张冬冬等，数字逻辑与组成原理实践教程。清华大学出版社。 2018，8
- ③. 汪文祥，CPU 设计实战。 机械工业出版社。2021，4
- ④. 高小鹏，计算机组成与实现。高等教育出版社。2018，12
- ⑤. 雷思磊，自己动手写 CPU。电子工业出版社。 2014，9
- ⑥. 吴继华、王诚，Verilog 设计与验证。 人民邮电出版社。2006，8

## 六、选做题

### 1 序列检测器

#### 输入方式与输出显示提示

①用 8 位拨动开关，随机设置 K1 K2 K3 K4 K5 K6 K7 K8 八个拨动开关，第一次输入表示低八位，第二次输入表示高八位，两次输入组成一个十六位的随机二进制数。

第一次拨动 K1~K8 开关	第二次拨动 K1~K8 开关
----------------	----------------

例如：

第一次输入表示低八位：0 1 0 1 0 1 0 0

第二次输入表示高八位：1 0 1 0 1 0 1 1

两次输入组成一个十六位的随机二进制数：

1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 0

②并在点阵的第一行上显示随机输入的十六位二进制数。

显示情况是：亮 灭 亮 灭 亮 灭 亮 亮 灭 亮 灭 亮 灭 亮 灭 灭

③在点阵的第二行上显示需要检测的特定的二进制数字。

例如检测 1 0 1

显示情况是：亮 灭 亮 灭 灭 灭 灭 灭 灭 灭 灭 灭 灭 灭 灭 灭

④在点阵的第三行上显示检测到特定的二进制数字所在的位置。

(1)带重复检测结果是：

输入的随机数：1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 0

检测到特定数的位置：0 0 1 0 1 0 1 0 0 1 0 1 0 1 0 0

显示情况是：灭 灭 亮 灭 亮 灭 亮 灭 灭 亮 灭 亮 灭 亮 灭 灭

在 LED 灯上显示检测到 6 个特定的二进制数字。



(2)不带重复检测结果是：

输入的随机数：1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 0

检测到特定数的位置：0 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0

显示情况是：灭 灭 亮 灭 灭 灭 亮 灭 灭 亮 灭 灭 灭 亮 灭 灭

在 LED 灯上显示检测到 4 个特定的二进制数字。



(3)没有检测到特定的二进制数字

点阵第三行上的 LED 灯全部是熄灭的。

显示情况是：灭 灭 灭 灭 灭 灭 灭 灭 灭 灭 灭 灭 灭 灭 灭

在 LED 灯上显示的是 0 个特定的二进制数字。



在每个周期十六位数据检测结束之后，需要对状态机内部进行复位操作。同时用按键模块的 S1 作为启动检测信号，每按下 S1 一次，检测器检测一次。

### 实验报告要求

- (1) 分别简单陈述用二个进程或三个进程描述带重复判断和不带重复判断检测特定二进制数字的原理。并分别画出对应的两个进程或三个进程的程序框图。
- (2) 分别画出带重复判断和不带重复判断检测特定二进制数字系列的程序功能框图。
- (3) 附上 Verilog 程序，分别说明两个进程或三个进程各自完成什么任务，两个进程或三个进程之间的关系。
- (4) 对程序关键或重要的程序语句给出简要的注释和说明。
- (5) 分别对带重复判断和不带重复判断检测特定二进制数字系列进行仿真，波形波形用线段或带箭头的线段标注输出与各个输入之间的关系，并给予简单的解析和说明。
- (6) 如果调试过程遇到了问题，应将问题截图，并给出解决问题的办法和对应的截图。

## 2 自动售货机

### 1、设计方案

- (1) 自动售货机仅销售一种商品矿泉水，价格是 1.5 元，假设机器中的商品数量无限。
- (2) 机器有一个投币孔，假设任意时刻只可以每次投入一枚 1 元或 5 角硬币，机器能自动识别 5 角和 1 元两种硬币。两个输出口，分别输出购买的商品和找零钱。
- (3) 当投入累积为 1.5 元硬币后，机器自动给出一瓶矿泉水；当投入累积为 2 元硬币后，在给出一瓶矿泉水的同时找回一枚 5 角的硬币；若投入的币值小于商品的单价时，则机器一直处于等待状态，不退硬币。
- (4) 一次只能购买一瓶矿泉水，若需要购买更多的矿泉水，则需重复操作。
- (5) 另外设置一复位按钮，当复位按钮按下时，自动售货机回到初始状态。

## 2、状态转换图与状态转换表

### 第一种状态转换图形式

根据设计方案可知，输出仅与状态有关，所以选用 Moore 型状态机设计自动售货机控制模块，状态转换图如图 1 所示。

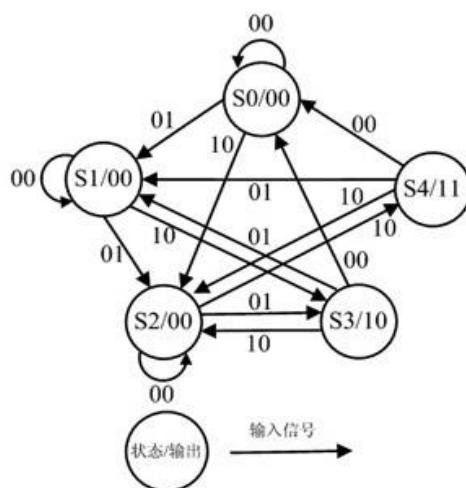


图 1 自动售货机状态转换图

1) 状态定义: S0 表示初态, S1 表示投入 5 角硬币的状态, S2 表示投入累积为 1 元的状态, S3 表示投入累积为 1 元 5 角的状态, S4 表示投入累积为 2 元的状态。

2) 输入信号: 设投币信号为输入逻辑变量, 用两位的矢量 inputs 表示。

输入仅有三种可能: 00, 01, 10。

inputs (00) 表示没有投入硬币, 或投币总值不够, 处于等待状态。

inputs (10) 表示投入 1 元硬币,

inputs (01) 表示投入 5 角硬币,

3) 输出信号: 给出矿泉水和找零为两个输出变量, 用两位的矢量 outputs 表示。outputs (10) 表示输出商品, outputs (11) 表示输出商品并找 5 角硬币。

**需要说明的是:**

(1) 在 S2 状态下, 再投入 5 角硬币, 则进入到 S3 状态。在 S3 状态下, 输出矿泉水后:

①可以回到 S0 的初始状态;

②此时, 如果投入了 5 角硬币, 则进入 S1 状态;

③此时, 如果投入了一元硬币, 则进入 S2 状态。



(2) 在 S2 状态下，再投入一元硬币，则进入到 S4 状态。在 S4 状态下，输出矿泉水和找 5 角零钱后：

- ①可以回到 S0 的初始状态；
- ②此时，如果投入了 5 角硬币，则进入 S1 状态；
- ③此时，如果投入了一元硬币，则进入 S2 状态。

自动售货机状态转换表一

原状态	输入		次状态	输出	
S0	0.5 元		S1		
S0		一元	S2		
S1	0.5 元		S2		
S1		一元	S3		
S3			S0	矿泉水	
S3	0.5 元		S1	矿泉水	
S3		一元	S2	矿泉水	
S2		一元	S4	矿泉水	0.5 元
S4	0.5 元		S1		
S4		一元	S2		

第二种状态转换图形式

根据上面的分析可得到状态转换图如图 2 所示。

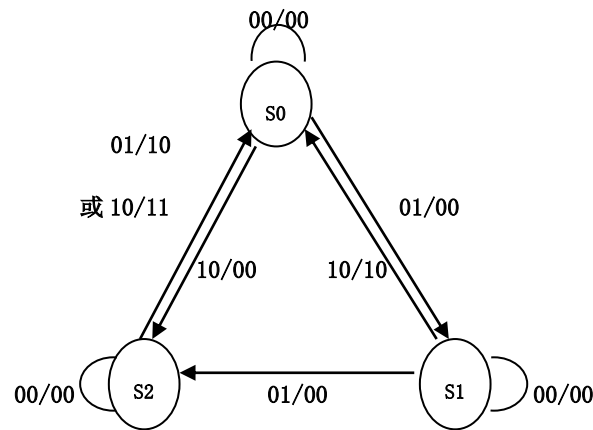


图 2 自动售货机状态转换图

需要说明的是：

- (1) 在 S2 状态下，再投入 5 角硬币，输出商品后，进入到 S0 状态。
- (2) 在 S2 状态下，再投入一元硬币，输出商品并找 5 角零钱后，进入到 S0 状态。

自动售货机状态转换表二

原状态	输入		次状态	输出	
S0	0.5 元		S1		
S0		一元	S2		
S1	0.5 元		S2		
S1		一元	S0	矿泉水	
S2	0.5 元		S0	矿泉水	
S2		一元	S0	矿泉水	0.5 元

### 3、输入方式与输出显示提示：

#### 八个七段码灯显示固定的信息内容

- ①用 4\*4 键盘输入商品的价格，并在第八个 LED8 七段码灯上显示商品的价格。
- ②可在 4\*4 键盘上输入 3 种硬币值，分别是 1、2、5 元硬币，并分别在 LED1、LED2、LED3 三个七段码灯上固定显示；如果输入其它值的硬币，则拒绝接受和拒绝在 LEDi 七段码灯上显示。
- ③在第四个 LED4 和第五个 LED5 七段码灯上显示输入的硬币之和，LED4 表示的是高位值，显示的是十位数；LED5 表示的是低位值，显示的是个位数。
- ④在第六个 LED6 七段码灯上显示找零的硬币值。
- ⑤在第七个 LED7 七段码灯上显示出货的数量。

例如：商品单价是 4 元，分别输入了 1、2 和 5 元硬币，输入的硬币之和为 8。

1

2

5

LED1

LED2

LED3

LED4

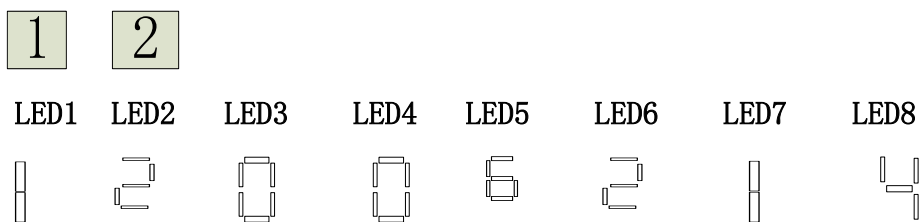
LED5

LED6

LED7

LED8

例如：商品单价是 4 元，分别输入了二次 1 和 2 元硬币，输入的硬币之和为 6。



每次购买结束之后，需要对状态机内部进行复位操作。同时用按键模块的 S1 作为启动新的一次购买的启动信号。

#### 4、问题与思考

- 1): 第一种与第二种状态转换图形式的各自特点是什么？
- 2): 在状态机跑飞，即脱离有效状态时，如何使状态机能恢复工作。

#### 5、实验报告

- (1) 简单陈述带找零的简单自动售货机的原理。
- (2) 给出带找零的简单自动售货机的状态迁移图和状态迁移表。
- (3) 附上将状态迁移图或状态迁移表自动转化为 VHDL 的程序，并根据程序画出程序功能框图。
- (4) 对程序关键或重要的程序语句给出简要的注释和说明。
- (5) 对生成的程序进行仿真，对仿真波形用线段或带箭头的线段标注输出与各个输入之间的关系，并给予简单的解析和说明。
- (6) 如果调试过程遇到了问题，应将问题截图，并给出解决问题的办法和对应的截图。

### 附件 3 硬件课程设计报告格式

统一封面如下

序号:



# 硬件课程设计报告 (2020-2021 学年)

姓 名: \_\_\_\_\_

学 号: \_\_\_\_\_

班 级: \_\_\_\_\_

指导教师: \_\_\_\_\_

2021 年 9 月 日

题 号	一	二	三	四	五	总 分
得 分						
满 分	10	10	10	10	60	

评语:

指导教师签字:

日期:

## 硬件课程设计报告正文格式

一、 第一页为目录

二、 ①1 级标题和 2 题标题均为小四号宋体加粗；

②正文为五号宋体不加粗，1.5 倍行间距；

③程序用五号 Times New Roman 不加粗，单倍行间距。

三、 每个题目均从新的页开始，表和框图应在同一页内。

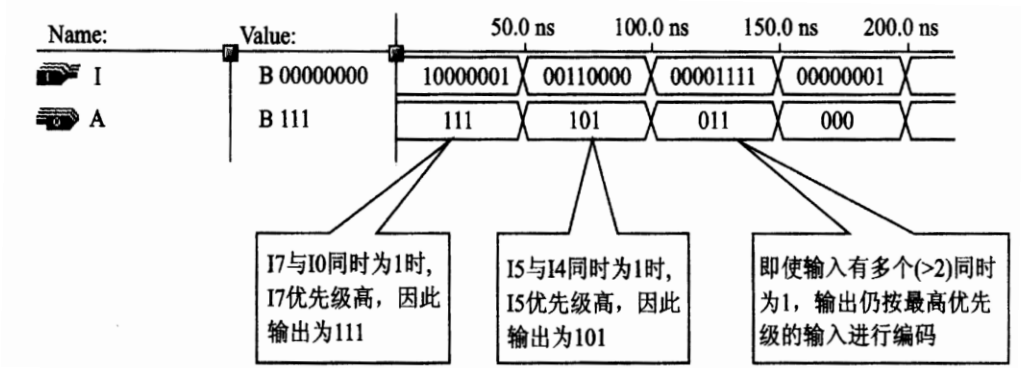
四、 第二部分：设计个人体会

叙述在本次课程设计中的收获和不足，要求至少 200 字

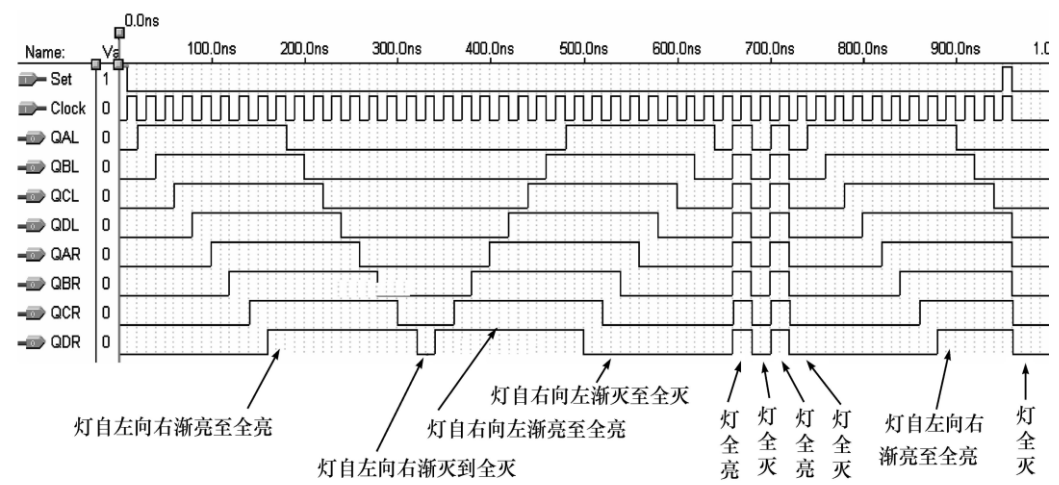
五、 用订书机订起来，不要用夹子夹起来。

# 附件 4 仿真测试截图的解释和说明

仿真截图及注释例子 1



仿真截图及带注释和说明形式的例子 2



设计一个带复位端且对输入时钟 `clk` 进行二分频模块，并说明仿真波形。

设计要求：复位信号为同步、高电平有效，时钟的下降沿触发。

```
module m2(out,clk,reset);
input  reset,clk;
output out;
reg out;
always @(negedge clk)
begin
if(reset)
out<=0;
else
out<=~out;
end
endmodule
```

