《数字电路与数字系统实验》实验报告

**实验8: 状态机及键盘输入**

**姓名:** 尹浚宇

**学号:** 161130118

**班级:** 2018-2019第一学期数字电路与数字系统实验2班

**邮箱:** [908664035@qq.com](mailto:908664035@qq.com)

**实验时间:** 2018-11-06

1. **实验目的**
2. 掌握状态机的工作原理.
3. 了解状态机的编码方式.
4. 利用PS/2键盘输入实现简单状态机的设计.
5. **实验原理**
6. 状态机

有限状态机FSM(Finite State Machine)简称状态机, 是⼀个在有限个状态间进⾏转换和动作的计算模型. 有限状态机含有⼀个起始状态、⼀个输⼊列表(列表中包含了所有可能的输⼊信号序列)、⼀个状态转移函数和⼀个输出端,状态机在⼯作时由状态转移函数根据当前状态和输⼊信号确定下⼀个状态和输出. 状态机⼀般都从起始状态开始, 根据输⼊信号由状态转移函数决定状态机的下⼀个状态.

有限状态机是数字电路系统中⼀种⼗分重要的电路模块,是⼀种输出取决于过去输⼊和当前输⼊的时序逻辑电路, 他是组合逻辑电路和时序逻辑电路的⼀种组合. 其中组合逻辑分为两个部分, ⼀个是⽤于产⽣有限状态机下⼀个状态的次态逻辑, 另⼀个是⽤于产⽣输出信号的输出逻辑,次态逻辑的功能是确定有限状态机的下⼀个状态; 输出逻辑的功能是确定有限状态机的输出. 除了输⼊和输出外, 状态机还有⼀组具有”记忆”功能的寄存器, 这些寄存器的功能是记忆有限状态机的内部状态, 常被称作状态寄存器.

在实际应⽤中, 有限状态机被分为两种: Moore(摩尔)型有限状态机和Mealy(米里)型有限状态机. Moore型有限状态机的输出信号只与有限状态机的当前状态有关, 与输⼊信号的当前值⽆关, 输⼊信号的当前值只会影响到状态机的次态, 不会影响状态机当前的输出. 即Moore型有限状态机的输出信号2是直接由状态寄存器译码得到. Moore型有限状态机在时钟CLK信号有效后经过⼀段时间的延迟, 输出达到稳定值. 即使在这个时钟周期内输⼊信号发⽣变化, 输出也会在这个完整的时钟周期内保持稳定值⽽不变. 输⼊对输出的影响要到下⼀个时钟周期才能反映出来. Moore有限状态机最重要的特点就是将输⼊与输出信号隔离开来. Mealy状态机与Moore有限状态机不同, Mealy有限状态机的输出不仅仅与状态机的当前状态有关, ⽽且与输⼊信号的当前值也有关. Mealy有限状态机的输出直接受输⼊信号的当前值影响, ⽽输⼊信号可能在⼀个时钟周期内任意时刻变化, 这使得Mealy有限状态机对输⼊的响应发⽣在当前时钟周期, 比Moore有限状态机对输⼊信号的响应要早⼀个周期. 因此, 输⼊信号的噪声可能影响到输出的信号.

1. PS/2接口控制器及键盘输入

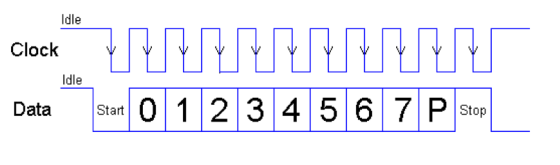
PS/2是个⼈计算机串行I/O接口的⼀种标准, 因其⾸次在IBM PS/2(Personal System/2)机器上使⽤⽽得名, PS/2接口可以连接PS/2键盘和 PS/2鼠标.

键盘处理器在识别到有键按下或放开时都会从PS2\_DAT引脚送出扫描

码. 当按键被按下时送出的扫描码被称为”通码(Make Code)”, 当按键被释放时送出的扫描码称为”断码(Break Code)”. 以”W”键为例, “W”键的通码是1Dh,如果”W”键被按下, 则PS2\_DAT引脚将输出1Dh, 如果”W”键⼀直没有释放, 则不断输出扫描码1Dh…, 直到有其他键按下或者”W”键被放开. 某按键的断码是F0h加此按键的通码, 如释放”W”键时输出的断码为F0h 1Dh.

多个键被同时按下时, 将逐个输出扫描码, 如: 先按左”Shift”键(扫描码为 12h)、再按”W”键、放开”W”键、再放开左”Shift”键, 则此过程送出的全部扫描码为: 12h 1Dh F0h 1Dh F0h 12h.

键盘以每帧11位的格式传送数据给主机. 第⼀位是开始位(逻辑0), 后⾯跟8位数据位(低位在前), ⼀个奇偶校验位(奇校验)和⼀位停⽌位(逻辑1)每位都在时钟的下降沿有效, 下图显⽰了键盘传送⼀字节数据的时序.



1. **实验环境/器材**

系统环境是window10, 硬件环境是DE10-Standard开发板, 软件环境是Quartus.

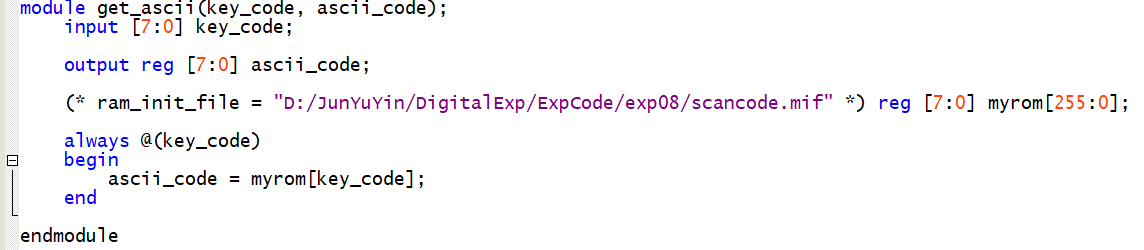
1. **程序代码**

整个工程的设计思路如下:

整个工程分为四个模块完成, 三个子模块功能分别为: 读取键盘扫描码, 根据扫描码生成对应的ascii码, 将结果映射到数码管上. 顶层模块是一个输入为当前扫描码的Mealy型状态机.

对于读取键盘扫描码模块, 完全按照教程所给方法设计, 未加改动.

扫描码转ascii码模块利用了一个rom来实现, 其地址为扫描码, 数据为扫描码对应的ascii码, 使用.mif文件初始化该rom, 代码见下图.



结果显示模块分别将按键次数, ascii码, 扫描码从高到低依次显示在数码管上, 每一项使用2根数码管. 为了实现松开按键时低四根数码管全灭, 额外加了一个参数pressing, 当pressing为1时, 低四根数码管才有效.

对于顶层模块里的状态机, 这里采用Mealy型状态机. 一共有两个状态, 分别为Release和Press, 分别代表没有键按下和有键按下. 输入为键盘扫描码, 输出为按键次数, 当前键盘扫描码, 和当前键盘扫描码对应的ascii码. 当前状态为Release, 且输入为通码时, 计数加一, 输出情况如上所述, 同时下一状态为Press, 当前状态为Press时, 若输入为通码则检测和之前的通码是否一致, 若一致说明按下某个键后一直未松开, 输出和状态均不变, 若检测到断码, 则状态变为Release, 后四位全灭.

具体代码见压缩包内文件.

1. **实验步骤**

首先设计了一个mealy型状态机并将其作为顶层模块, 而后分别设计了各个子模块来完成整个工程的设计(具体思路见四), 通过编译后又进行了仿真模拟(见六), 仿真模拟通过后分配了引脚(见下方), 最后将生成的二进制烧写文件导入开发板进行硬件验证, 并通过了助教的查验.

引脚分配思路如下:

采用内置时钟, ps/2时钟, ps/2数据口, nextdata\_n分配给一个开关, 过程中始终置为0, clr\_n分配给一个按钮, 数码管按照官方手册逐一分配.

1. **测试方法**

采用仿真验证和硬件验证的测试方法.

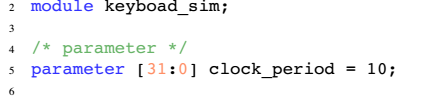
仿真代码的编写方法按照讲义, 未加改动.

硬件验证结果难以用图片文字呈现, 且验收已通过, 可将压缩包内的.sof文件写入开发版内进行验证, 此处不在赘述.

1. **实验结果**

成功实现了实验的基本功能, 且通过验收.

1. **实验中遇到的问题及解决方案**
2. 在仿真模拟时始终没有结果, 后面发现教程中给的仿真代码的模块为keyboad\_sim, 而仿真设置里为keyboard\_sim, 修改后成功仿真.



1. **实验得到的启示**
2. 不要复制粘帖教程代码, 如需要使用可以手打一遍.
3. 对于复杂工程而言, 分模块测试是一个必备的好方法.
4. **意见和建议**
5. 建议给出一个mealy型状态机的示例代码.