**数字系统（课程）设计**

**实 验 报 告**

|  |  |
| --- | --- |
| **课程名称：** | **数字系统设计** |
| **学生姓名：** | **江泽群** |
| **学生学号：** | **201530371299** |
| **学生专业：** | **电子科学与技术** |
| **开课学期：** | **5** |
| **实验成绩：** |  |

# 实验四 按键控制的状态机设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **地 点：** | 31号 楼 | 312 房； | **实验台号：** | 42 |
| **实验日期与时间：** | 2017年12月1日 | | **评 分：** |  |
| **预习检查纪录：** |  | | **批改教师：** |  |

报告内容：

一，实验要求：

运用状态机设计按键控制数码管显示的电路系统

1. 实验内容：
2. 按键控制数码管显示，当按下 S2时，Q4数码管显示1；再次按下 S2，Q4数码管显示2；第三次按下 S2， Q4数码管显示3，第四次按下 S2，Q4数码管显示4；第五次按下 S2时 Q4又从1开始显示，如此反复。
3. 检测按键是否按下需要消抖，消抖程序请同学们参考附录程序，参考程序是利用四个按键控制四个 LED 灯的亮灭。请同学们在读懂参考程序的基础上将其改编为符合要求1的程序。
4. 参考程序是普通的控制程序，本实验要求同学们利用状态机来编写程序。
5. 实验设计原理；

根据状态机设计方法，先画出 ASM 图，然后结合参考程序设计按键控制数码管显示电路。

1. 实验过程记录

按键控制数码管显示电路 ASM 图如图4-1所示。

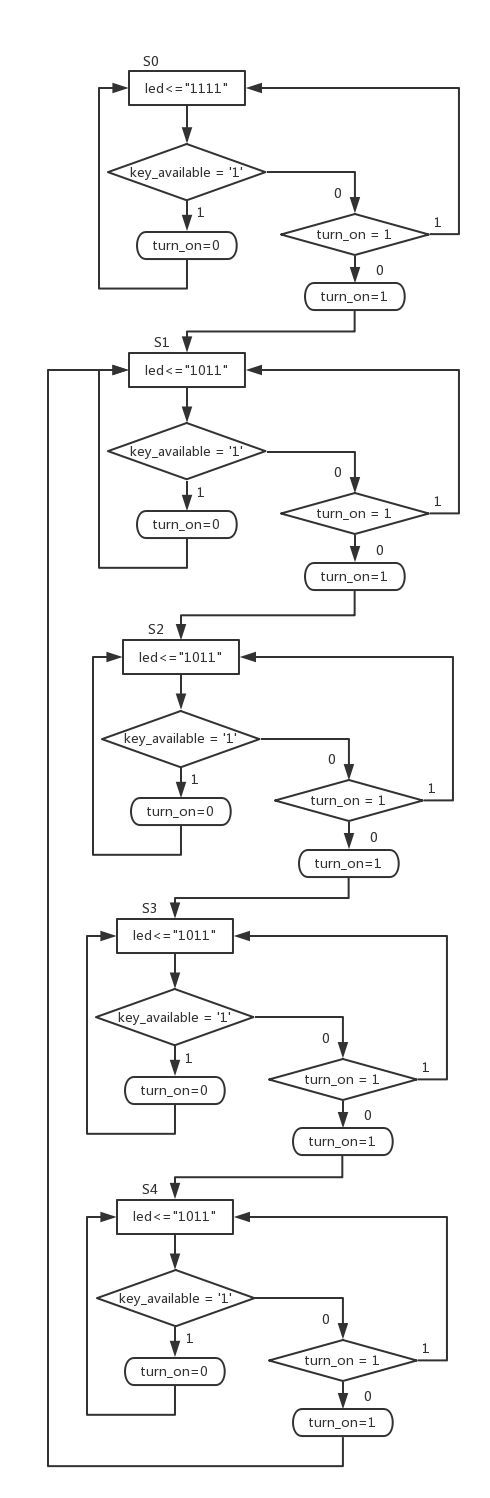


图4-1

1. 实验结果和分析

按键控制数码管显示电路模块图如图5-1所示。

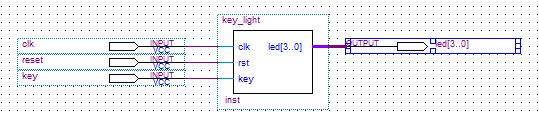


图5-1

按键控制数码管显示电路仿真波形图如图5-2所示。

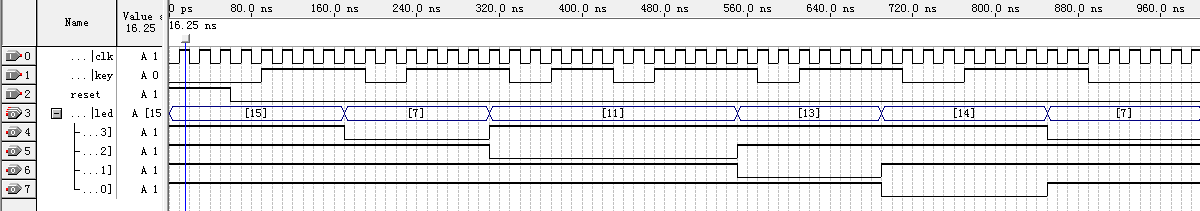


图5-2

实验结果符合要求。

六，附录

按键控制数码管显示电路实验代码如下：

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_arith.all;

entity key\_light is

port(clk: in std\_logic;

rst: in std\_logic;

key: in std\_logic;

led: out std\_logic\_vector(3 downto 0));

end entity;

architecture behav of key\_light is

begin

process (clk, rst, key)

type state\_type is(S0, S1, S2, S3, S4);

variable present\_state: state\_type;

variable key\_available: std\_logic;

variable turn\_on : integer range 1 downto 0;

variable cnt: integer range 10 downto 0;

variable key\_rst: std\_logic;

variable key\_rst\_an: std\_logic;

variable key\_rst\_r: std\_logic;

variable low\_sw: std\_logic;

variable low\_sw\_an: std\_logic;

variable low\_sw\_r: std\_logic;

begin

if rst='1' then

present\_state:=S0;

key\_available := '0';

led <= "1111";

turn\_on := 0;

key\_rst\_r := '0';

cnt := 0;

elsif rising\_edge(clk) then

key\_rst := key;

key\_rst\_an := (key\_rst and key\_rst\_r);

if key\_rst\_an = '1' then

cnt := cnt + 1;

else

cnt := 0;

end if;

key\_rst\_r := key\_rst;

if cnt = 3 then

key\_available := '1';

cnt := 2;

else

key\_available := '0';

end if;

case present\_state is

when S0 =>

led <= "1111";

if (key\_available = '1') then

if (turn\_on = 1) then

present\_state := S0;

else

present\_state := S1;

end if;

turn\_on := 1;

else

turn\_on := 0;

end if;

when S1 =>

led <= "0111";

if (key\_available = '1') then

if (turn\_on = 1) then

present\_state := S1;

else

present\_state := S2;

end if;

turn\_on := 1;

else

turn\_on := 0;

end if;

when S2 =>

led <= "1011";

if (key\_available = '1') then

if (turn\_on = 1) then

present\_state := S2;

else

present\_state := S3;

end if;

turn\_on := 1;

else

turn\_on := 0;

end if;

when S3 =>

led <= "1101";

if (key\_available = '1') then

if (turn\_on = 1) then

present\_state := S3;

else

present\_state := S4;

end if;

turn\_on := 1;

else

turn\_on := 0;

end if;

when S4 =>

led <= "1110";

if (key\_available = '1') then

if (turn\_on = 1) then

present\_state := S4;

else

present\_state := S1;

end if;

turn\_on := 1;

else

turn\_on := 0;

end if;

end case;

end if;

end process;

end behav;