**电子技术实验报告**

**——电梯控制器的设计与分析**

班级： 计算机 73

姓名： 杨志远

学号： 2176112484

日期：2019年6月4日

联系电话：13572281956

**目 录**

[一．实验目的 3](#_Toc10556595)

[二．项目设计 3](#_Toc10556596)

[1. 系统设计要求 3](#_Toc10556597)

[三．系统设计方案 4](#_Toc10556598)

[1. 电梯控制器DTKZQ的输入输出 4](#_Toc10556599)

[2.电梯控制器的内部组成结构图 4](#_Toc10556600)

[3.主要的VHDL源程序 5](#_Toc10556601)

[四．测试结果及分析 12](#_Toc10556602)

[电路模拟仿真测试 12](#_Toc10556603)

[电路硬件测试 12](#_Toc10556604)

[五．实验总结 13](#_Toc10556605)

[六．结束语 14](#_Toc10556606)

# 一．实验目的

电子技术专题实验是对“数字逻辑”课程内容的全面、系统的总结、巩固和提高的一项课程实践活动。根据数字逻辑的特点，选择相应的题目，在老师的指导下，由学生独立完成。目的是通过实验使学生掌握数字逻辑电路设计的基本方法和技巧，正确运用QuartusⅡ软件及实验室多功能学习机硬件平台，完成所选题目的设计任务，并掌握数字逻辑电路测试的基本方法，训练学生的动手能力和思维方法。通过实验，一方面提高运用数字逻辑电路解决实际问题的能力，另一方面使学生更深入的理解所学知识，为以后的计算机硬件课程的学习奠定良好的基础。

# 二．项目设计

## 1. 系统设计要求

随着社会的发展，电梯的使用越来越普遍，已从原来只在商业大厦、宾馆使用，过渡到在办公楼、居民楼等场所使用，并且对电梯功能的要求也不断提高，相应地其控制方式也在不停地发生变化。对于电梯的控制，传统的方法是使用继电器—接触器控制系统进行控制，随着技术的不断发展，微型计算机在电梯控制上的应用日益广泛，现在已进入全微机化控制的时代。

**2. 设计方案**

电梯的微机化控制主要有以下几种形式：① PLC控制；② 单板机控制；③ 单片机控制；④ 单微机控制；⑤ 多微机控制；⑥ 人工智能控制。随着EDA技术的快速发展，CPLD/FPGA已广泛应用于电子设计与控制的各个方面。本设计就是使用一片CPLD/FPGA来实现对电梯的控制的。

# 三．系统设计方案

## 1. 电梯控制器DTKZQ的输入输出

根据系统设计要求，并考虑到系统的可验证性，整个系统的输入输出接口设计如图9.1所示：系统工作用2 Hz基准时钟信号CLKIN，楼层上升请求键UPIN，楼层下降请求键DOWNIN，楼层选择键入键ST\_CH，提前关门输入键CLOSE，延迟关门输入键DELAY，电梯运行的开关键RUN\_STOP，电梯运行或停止指示键LAMP，电梯运行或等待时间指示键RUN\_WAIT，电梯所在楼层指示数码管ST\_OUT，楼层选择指示数码管DIRECT。



图9.1 电梯控制器DTKZQ的输入输出接口图

## 2.电梯控制器的内部组成结构图

电梯的控制状态包括运行状态、停止状态及等待状态，其中运行状态又包含向上状态和向下状态。主要动作有开、关门，停靠和启动。乘客可通过键入开、关门按钮，呼唤按钮，指定楼层按钮等来控制电梯的行与停。

据此，整个电梯控制器DTKZQ应包括如下几个组成部分：① 时序输出及楼选计数器；② 电梯服务请求处理器；③ 电梯升降控制器；④ 上升及下降寄存器；⑤ 电梯次态生成器。该电梯控制器设计的关键是确定上升及下降寄存器的置位与复位。整个系统的内部组成结构图如图9.2所示。



图9.2 电梯控制器的内部组成结构图

## 3.主要的VHDL源程序

--Vhdl1.VHD

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

USE IEEE.STD\_LOGIC\_ARITH.ALL;

ENTITY Vhdl1 IS

PORT(CLKIN:IN STD\_LOGIC; --2 Hz时钟输入信号

UPIN:IN STD\_LOGIC; --楼层上升请求键

DOWNIN:IN STD\_LOGIC; --楼层下降请求键

ST\_CH:IN STD\_LOGIC;

--结合DIRECT完成楼层选择的键入

CLOSE:IN STD\_LOGIC; --提前关门输入键

DELAY:IN STD\_LOGIC; --延迟关门输入键

RUN\_STOP:IN STD\_LOGIC; --电梯运行的开关键

LAMP:OUT STD\_LOGIC; --电梯运行或停止指示键

RUN\_WAIT:OUT STD\_LOGIC\_VECTOR(3 DOWNTO 0);

--结合LAMP指示电梯运行或等待时间

ST\_OUT:OUT STD\_LOGIC\_VECTOR(3 DOWNTO 0); --电梯所在楼层指示数码管

DIRECT:OUT STD\_LOGIC\_VECTOR(3 DOWNTO 0)); --楼层选择指示数码管

END ENTITY Vhdl1;

ARCHITECTURE ART OF Vhdl1 IS

SIGNAL UR,DR:STD\_LOGIC\_VECTOR(16 DOWNTO 1);

--上升或下降楼层请求寄存器

SIGNAL DIR,LIFTOR:STD\_LOGIC\_VECTOR(3 DOWNTO 0);

--楼选指示及楼层数计数器

SIGNAL WAI\_T:STD\_LOGIC\_VECTOR(3 DOWNTO 0);

--运行或等待计数器

SIGNAL DIVID,HAND:STD\_LOGIC;

--时钟2分频和楼选复位变量

SIGNAL LADD:STD\_LOGIC\_VECTOR(1 DOWNTO 0);

--电梯运行状态变量

SIGNAL CLOSEX,DELAYX,LAMP0:STD\_LOGIC;

--提前关门及延迟变量

BEGIN

--内部信号值的输出

DIRECT<=DIR+1;

ST\_OUT<=LIFTOR+1;

RUN\_WAIT<=WAI\_T;

P0:PROCESS(CLKIN) IS

BEGIN

IF (CLKIN'EVENT AND CLKIN='1') THEN

LAMP<=LADD(1);

END IF;

END PROCESS P0;

HAND<=WAI\_T(2)AND (NOT WAI\_T(1))AND WAI\_T(0);

CLOSEX<=CLOSE AND (NOT LADD(1));

DELAYX<=DELAY AND (NOT LADD(1));

--分频及楼选信号产生进程

P1:PROCESS(ST\_CH)

BEGIN

IF (ST\_CH='1') THEN

IF (DIR="1111") THEN DIR<="0000";

ELSE DIR<=DIR+1;

END IF;

END IF;

END PROCESS;

--楼层请求寄存器的置位与复位进程

P2:PROCESS(UR, DR, UPIN, DOWNIN, ST\_CH, LIFTOR, WAI\_T, RUN\_STOP, HAND)

VARIABLE NUM,T:INTEGER RANGE 0 TO 16 ;

BEGIN

NUM:=CONV\_INTEGER(LIFTOR)+1;

T:=CONV\_INTEGER(DIR)+1;

IF (RUN\_STOP='1') THEN

IF (((T>NUM)AND (ST\_CH='1'))OR (UPIN='1'))THEN

CASE T IS

WHEN 1 => UR(1)<='1';

WHEN 2 => UR(2)<='1';

WHEN 3 => UR(3)<='1';

WHEN 4 => UR(4)<='1';

WHEN 5 => UR(5)<='1';

WHEN 6 => UR(6)<='1';

WHEN 7 => UR(7)<='1';

WHEN 8 => UR(8)<='1';

WHEN 9 => UR(9)<='1';

WHEN 10 => UR(10)<='1';

WHEN 11 => UR(11)<='1';

WHEN 12 => UR(12)<='1';

WHEN 13 => UR(13)<='1';

WHEN 14 => UR(14)<='1';

WHEN 15 => UR(15)<='1';

WHEN 16 => UR(16)<='1';

WHEN OTHERS=>NULL;

END CASE;

ELSIF (HAND='1')THEN

CASE NUM IS

WHEN 1 => UR(1)<='0';

WHEN 2 => UR(2)<='0';

WHEN 3 => UR(3)<='0';

WHEN 4 => UR(4)<='0';

WHEN 5 => UR(5)<='0';

WHEN 6 => UR(6)<='0';

WHEN 7 => UR(7)<='0';

WHEN 8 => UR(8)<='0';

WHEN 9 => UR(9)<='0';

WHEN 10 => UR(10)<='0';

WHEN 11 => UR(11)<='0';

WHEN 12 => UR(12)<='0';

WHEN 13 => UR(13)<='0';

WHEN 14 => UR(14)<='0';

WHEN 15 => UR(15)<='0';

WHEN 16 => UR(16)<='0';

WHEN OTHERS=>NULL;

END CASE;

END IF;

IF (((T<NUM)AND (ST\_CH='1'))OR(DOWNIN='1')) THEN

CASE T IS

WHEN 1 => DR(1)<='1';

WHEN 2 => DR(2)<='1';

WHEN 3 => DR(3)<='1';

WHEN 4 => DR(4)<='1';

WHEN 5 => DR(5)<='1';

WHEN 6 => DR(6)<='1';

WHEN 7 => DR(7)<='1';

WHEN 8 => DR(8)<='1';

WHEN 9 => DR(9)<='1';

WHEN 10 => DR(10)<='1';

WHEN 11 => DR(11)<='1';

WHEN 12 => DR(12)<='1';

WHEN 13 => DR(13)<='1';

WHEN 14 => DR(14)<='1';

WHEN 15 => DR(15)<='1';

WHEN 16 => DR(16)<='1';

WHEN OTHERS=>NULL;

END CASE;

ELSIF (HAND='1') THEN

CASE NUM IS

WHEN 1 => DR(1)<='0';

WHEN 2 => DR(2)<='0';

WHEN 3 => DR(3)<='0';

WHEN 4 => DR(4)<='0';

WHEN 5 => DR(5)<='0';

WHEN 6 => DR(6)<='0';

WHEN 7 => DR(7)<='0';

WHEN 8 => DR(8)<='0';

WHEN 9 => DR(9)<='0';

WHEN 10 => DR(10)<='0';

WHEN 11 => DR(11)<='0';

WHEN 12 => DR(12)<='0';

WHEN 13 => DR(13)<='0';

WHEN 14 => DR(14)<='0';

WHEN 15 => DR(15)<='0';

WHEN 16 => DR(16)<='0';

WHEN OTHERS=>NULL;

END CASE;

END IF;

ELSE

UR<="0000000000000000";

DR<="0000000000000000";

END IF;

END PROCESS;

--电梯运行次态的控制进程

P3:PROCESS(UR,DR,LIFTOR,LADD,WAI\_T,RUN\_STOP,ST\_CH)

BEGIN

IF (RUN\_STOP='1') THEN

IF ((UR OR DR)="0000000000000000") THEN

LADD(1)<='0'; --初始状态

ELSE

CASE LIFTOR IS

WHEN "0000"=>IF ((UR(1)OR DR(1))>'0') THEN LADD(1)<='0'; --等待状态

ELSE LADD<="11"; --上升状态

END IF;

WHEN "0001"=>IF ((UR(2)OR DR(2))>'0') THEN LADD(1)<='0'; --等待状态

ELSIF ((( LADD(0)='1')AND ((UR(16 DOWNTO 3)

OR DR(16 DOWNTO 3))>"00000000000000"))

OR((UR(1)OR DR(1))='0')) THEN

LADD <="11"; --上升状态

ELSE LADD<="10"; --下降状态

END IF;

WHEN "0010"=>IF ((UR(3) OR DR(3))>'0') THEN LADD(1)<='0';

ELSIF ((( LADD(0)='1')AND ((UR(16 DOWNTO 4)

OR DR(16 DOWNTO 4))>"0000000000000"))

OR((UR(2 DOWNTO 1)

OR DR(2 DOWNTO 1))="00")) THEN LADD<="11";

ELSE LADD<="10";

END IF;

WHEN "0011"=>IF ((UR(4) OR DR(4))>'0') THEN LADD(1)<='0';

ELSIF (((LADD(0)='1')AND ((UR(16 DOWNTO 5)

OR DR(16 DOWNTO 5))>"000000000000"))

OR((UR(3 DOWNTO 1)

OR DR(3 DOWNTO 1))="000")) THEN LADD<="11";

ELSE LADD<="10";

END IF;

WHEN "0100"=>IF ((UR(5) OR DR(5))>'0') THEN LADD(1)<='0';

ELSIF (((LADD(0)='1')AND ((UR(16 DOWNTO 6)

OR DR(16 DOWNTO 6))>"00000000000"))

OR((UR(4 DOWNTO 1)

OR DR(4 DOWNTO 1))="0000")) THEN LADD<="11";

ELSE LADD<="10";

END IF;

WHEN "0101"=>IF ((UR(6) OR DR(6))>'0') THEN LADD(1)<='0';

ELSIF (((LADD(0)='1')AND ((UR(16 DOWNTO 7)

OR DR(16 DOWNTO 7))>"0000000000" ))

OR((UR(5 DOWNTO 1)

OR DR(5 DOWNTO 1))="00000")) THEN LADD<="11";

ELSE LADD<="10";

END IF;

WHEN "0110"=>IF ((UR(7) OR DR(7))>'0') THEN LADD(1)<='0';

ELSIF (((LADD(0)='1')AND ((UR(16 DOWNTO 8)

OR DR(16 DOWNTO 8))>"000000000"))

OR((UR(6 DOWNTO 1)

OR DR(6 DOWNTO 1))="000000"))

THEN LADD<="11";

ELSE LADD<="10";

END IF;

WHEN "0111"=>IF ((UR(8) OR DR(8))>'0') THEN LADD(1)<='0';

ELSIF (((LADD(0)='1')AND ((UR(16 DOWNTO 9)

OR DR(16 DOWNTO 9))>"00000000" ))

OR((UR(7 DOWNTO 1)

OR DR(7 DOWNTO 1))="0000000"))

THEN LADD<="11";

ELSE LADD<="10";

END IF;

WHEN "1000"=>IF ((UR(9) OR DR(9))>'0') THEN LADD(1)<='0';

ELSIF ((( LADD(0)='1')AND ((UR(16 DOWNTO 10)

OR DR(16 DOWNTO 10))>"0000000" ))

OR((UR(8 DOWNTO 1)

OR DR(8 DOWNTO 1))="00000000"))

THEN LADD<="11";

ELSE LADD<="10";

END IF;

WHEN "1001"=>IF ((UR(10) OR DR(10))>'0') THEN LADD(1)<='0';

ELSIF ((( LADD(0)='1')AND ((UR(16 DOWNTO 11)

OR DR(16 DOWNTO 11))>"000000" ))

OR((UR(9 DOWNTO 1)

OR DR(9 DOWNTO 1))="000000000"))

THEN LADD<="11";

ELSE LADD<="10";

END IF;

WHEN "1010"=>IF ((UR(11) OR DR(11))>'0') THEN LADD(1)<='0';

ELSIF (((LADD(0)='1')AND ((UR(16 DOWNTO 12)

OR DR(16 DOWNTO 12)) >"00000"))

OR((UR(10 DOWNTO 1)

OR DR(10 DOWNTO 1))="0000000000"))

THEN LADD<="11";

ELSE LADD<="10";

END IF;

WHEN "1011"=>IF ((UR(12) OR DR(12))>'0') THEN LADD(1)<='0';

ELSIF ((( LADD(0)='1')AND ((UR(16 DOWNTO 13)

OR DR(16 DOWNTO 13))>"0000" ))

OR((UR(11 DOWNTO 1)

OR DR(11 DOWNTO 1))="00000000000"))

THEN LADD<="11";

ELSE LADD<="10";

END IF;

WHEN "1100"=>IF ((UR(13) OR DR(13))>'0') THEN LADD(1)<='0';

ELSIF ((( LADD(0)='1')AND ((UR(16 DOWNTO 14)

OR DR(16 DOWNTO 14))>"000" ))

OR((UR(12 DOWNTO 1)

OR DR(12 DOWNTO 1))="000000000000"))

THEN LADD<="11";

ELSE LADD<="10";

END IF;

WHEN "1101"=>IF ((UR(14) OR DR(14))>'0') THEN LADD(1)<='0';

ELSIF ((( LADD(0)='1')AND ((UR(16 DOWNTO 15)

OR DR(16 DOWNTO 15))>"00" ))

OR((UR(13 DOWNTO 1)

OR DR(13 DOWNTO 1))="0000000000000"))

THEN LADD<="11";

ELSE LADD<="10";

END IF;

WHEN "1110"=>IF ((UR(15)OR DR(15))>'0') THEN LADD(1)<='0';

ELSIF ((( LADD(0)='1')AND ((UR(16)OR DR(16))>'0'))

OR ((UR(14 DOWNTO 1)

OR DR(14 DOWNTO 1))="00000000000000"))

THEN LADD<="11";

ELSE LADD<="10";

END IF;

WHEN "1111"=>IF ((UR(16) OR DR(16))>'0') THEN LADD(1)<='0';

ELSE LADD<="10";

END IF;

WHEN OTHERS=>NULL;

END CASE;

END IF;

ELSE LADD<="00";

END IF;

END PROCESS;

--电梯运行楼层计数及提前/延迟关门控制进程

P4:PROCESS(WAI\_T,LADD,CLOSEX,DELAYX,DIVID)

BEGIN

IF (CLKIN'EVENT AND CLKIN='1') THEN

IF (WAI\_T="000" OR CLOSEX='1') THEN WAI\_T<="1111";

ELSE

IF (DELAYX='0')THEN WAI\_T<=WAI\_T-1;

ELSE WAI\_T<="0010";

END IF;

IF (LADD="11") THEN LIFTOR<=LIFTOR+1;

ELSIF (LADD="10") THEN LIFTOR<=LIFTOR-1;

END IF;

END IF;

END IF;

END PROCESS;

END ARCHITECTURE ART;

# 四．测试结果及分析

## 电路模拟仿真测试

仿真结果符合预期，未保存仿真实验结果图。

## 电路硬件测试

测试结果大致符合预期，观察到的现象如下：

1. 通过ST\_CH输入键选定目标楼层，在等到WAIT时间由F跳转到0或者直接先将CLOSE键置1，再将RUN\_STOP键置1，就能使电梯立即启动，实际楼层开始发生跳转，同时运行时间从F开始逐次跳转到0，如果实际楼层到了目标楼层，则运行时间变为等待时间置F，实际楼层与目标楼层数相同不动。
2. 如果要再次开始测试，则需先将CLOSE与RUN\_STOP键置0，再重复上述操作。
3. 如果按下DELAY键，等待时间会直接置2，电梯无法运行，这是在程序里面写好的。
4. 如果不进行任何操作，电梯的等待时间会从F跳转到0，再知道跳转到F，循环进行。
5. 按照以上的操作来看，电梯可以实现人为控制目标楼层而达到控制电梯的目的。

出现的问题及问题分析如下：

有些时候ST\_CH输入键无法正常工作，即输入目标楼层时（每一次置到高电平）目标楼层会随机跳转，不能实现一层一层的控制，但是等多调试几次后就会稳定下来，而且换台实验箱就不会发生这个问题。推测出现这个现象的原因是实验箱器件老化或损坏，换台实验箱就可以解决这个问题。

# 五．实验总结

这个实验的难度很大，经过3次实验课的探究，最终修改完成了对源程序的大幅度删改，先将修改源程序的思路介绍如下：

1. P2子程序里 VARIABLE NUM,T:INTEGER RANGE 0 TO 15 ;

应该是1-16（更规范）或者0-16，不建议为了保留0-15，而将when16的情况删除，这样即是把整个楼层数降到15楼，而且要同步修改后面所有的地方。

1. P4子程序里的按了CLOSE后的等待—运行时间置为运行时间，是6（110），但实际上运行时间应该是16（1111），因为运行时间是6的话，运行6层就会停止运行，有可能到不了目标楼层，运行最大时间16保证了可以覆盖所有的楼层！（注：这个地方改动后，牵连WAIT格式改动的地方有3处，这里不再详细列举）
2. （最关键）这个是难度最大而且修改幅度最大的地方，原先ppt中给出的程序的思路是用时钟源CLK控制电梯的目标楼层（原P1程序里），这样的思路是完全错误的！如果按照这个思路进行仿真实验可以得到完全正确的结果，但是在进行硬件测试时，发现如下的问题：
3. 完全无法控制目标楼层，违反了电梯使用的原则，即目标楼层全程由时钟源控制（从0-F根据时钟源的频率自动跳转），人为无法指定楼层。
4. 无法解决的**数据延迟**问题（data delay），在程序编译时就会报错，报错指令大致翻译如下：“程序可能无法运行，因为时钟跳转已经小于数据延迟”，实际操作时表现在，按下开关使电梯运行，实际楼层无法跳转到目标楼层，因为目标楼层仍在做人为无法控制的跳转，程序无法识别到底要跳转到哪一层楼去，而且存在的数据延迟使系统完全处于人为无法观察控制的失控状态，所以要么实际楼层完全不动，要么仅运行一层后就不动，运行时间在F和E之间来回跳转（即运行1层后就恢复等待时间，再置于运行时间，循环往复）。

在面对以上的问题后，现对源程序的思路和程序进行修改，大致修改点列举如下：

* 1. 修改ST\_CH输入的功能，让ST\_CH不仅仅如ppt里给出的实现一小部分功能，而将其置为一个可以操纵目标楼层的输入流，取代原有时钟源控制目标楼层的输入。
  2. 原有的P3里面if条件中删去**wait为某值时改变运行状态的条件**，这个条件使实际楼层完全静止不动，实属画蛇添足，是完全没有逻辑的条件。
  3. 将P4里面(DIVID'EVENT AND DIVID='1') 变成(CLKIN'EVENT AND CLKIN='1')，因为前面程序的修改，这里不再需要用寄存器来控制。

综上所述，修改完以上全部的修改点，就能得到本实验报告中给出的源程序，这个源程序可以实现在实验结果中描述的功能。

仍存在的问题与不足：

1. UP和DOWN功能仍无法正常使用，即从电梯外部还是不能控制电梯实际楼层的变化。
2. 硬件测试的时候还是存在ST\_CH有可能无法正常工作的情况，这种情况在几台实验箱测试后发现，有些实验箱可以正常工作，有些不行，证明是实验箱的问题，一部分实验箱发生了老化和损坏影响了实验结果。

实验后给出的建议：

1. 建议调整这道题的难度，电梯控制器和其他4道题来比明显难了好多。
2. 难度来源我认为是原ppt给出的程序，思路是完全错误的，有明显的误导倾向。
3. 实验室的实验箱好多已经存在硬件老化和损坏的情况，导致硬件测试有些时候完全查找不到错误的来源，换台实验箱则不会发生问题，影响学生的实验。

# 六．结束语

数字逻辑专题实验对我们进一步理解所学理论知识、深化对数字逻辑电路的认识有着不可替代的重要作用。计算机本身就属于工科专业，要求学生具备较高的动手实践操作能力，不能仅仅局限于课本知识的学习。数字逻辑专题实验与数字逻辑课程一脉相承，在我们刚学完理论知识后就能得到相应的实践能力训练，把所学知识应用到自己的设计题目过程中去，是一个难得的机会。

专题实验不仅使我对所学知识有了更加深入的理解，也让我了解了数字逻辑电路的基本设计过程和调试方法，并能够利用QuartusⅡ软件及实验室提供的多功能学习机硬件平台完成一些比较简单的数字逻辑电路题目的设计任务。同时，对数字逻辑电路的深入理解也为我以后的计算机硬件课程的学习奠定了良好的基础。另一方面，实验增强了我的团队合作意识和能力，这对我今后的发展大有裨益。