**北京邮电大学**

**数字电路与逻辑设计实验报告**

Digital Electronics Design Projects and Innovation

学院：信息与通信工程学院

班级：2014211113班

姓名：程潇

学号：2014210360

班内序号：07

指导老师：高英

日期：2016年5月11日

一、实验部分

实验一

■实验名称

Quartus II 原理图输出法设计

（一）半加器

1、实验任务要求

用逻辑门设计实现一个半加器，仿真验证其功能，并生成新的半加器图形模块单元。

2、设计思路和过程

●设计思路

半加器电路是指对两个输入数据位进行加法，输出一个结果位和进位，不产生进位输入的加法器电路，是实现两个一位二进制数的加法运算电路。

数据输入：被加数A、加数B 数据输出：半加和S、进位C

●设计过程

(1) 列出真值表

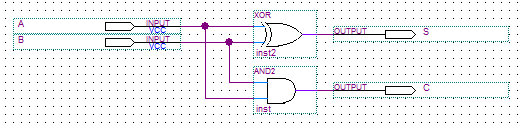
|  |  |  |  |
| --- | --- | --- | --- |
| input | | output | |
| A | B | S | C |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

(2)根据真值表写出输出逻辑表达式

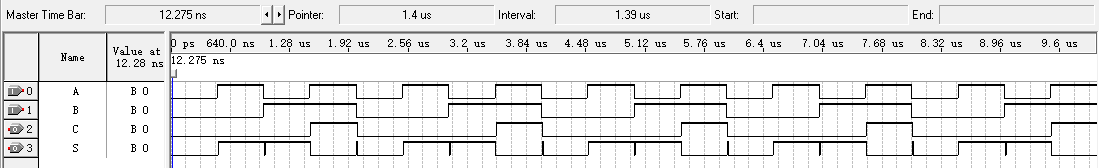
该电路有两个输出端，属于多输出组合数字电路，电路的逻辑表达式如下：

S=A⊕B,C=AB。所以，可以用一个两输入异或门和一个两输入与门实现。

3、实验原理图



4、仿真及波形分析



根据仿真波形对比半加器真值表，可以确定电路实现了半加器的功能。但我们也可以发现输出S出现了静态功能冒险，要消除该冒险可以加入相应的选通脉冲。

（二）全加器

1、实验任务要求

用实验内容1中生成的半加器模块和逻辑门设计实现一个全加器，仿真验证其功能，并下载到实验板测试，要求用拨码开关设定输入信号，发光二极管显示输出信号。

2、设计思路和过程

●设计思路

全加器与半加器的区别在于全加器有一个低进位CI，从外部特性来看，它是一个三输入两输出的器件。

●设计过程

(1)全加器的真值表如下：

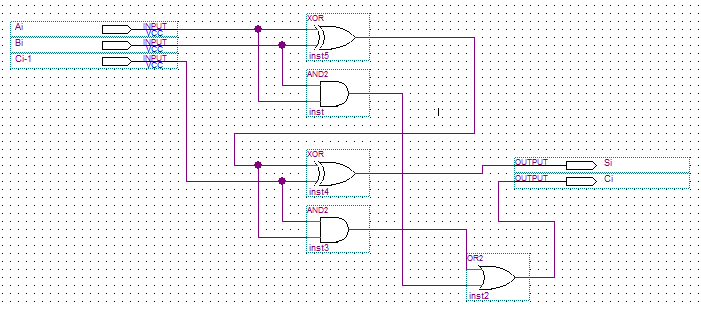
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| input | | | output | |
| Ai | Bi | Ci-1 | Si | Ci |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

(2)根据真值表写出逻辑表达式：

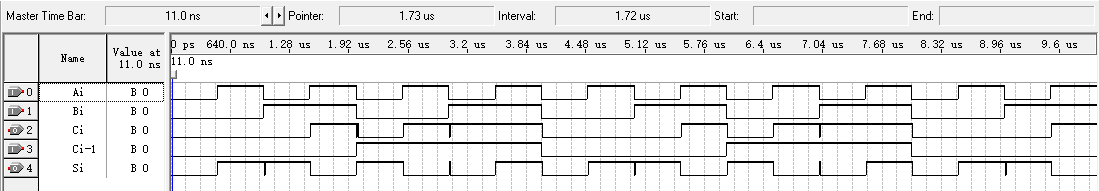
Si=Ai⊕Bi⊕Ci-1，Ci=（Ai⊕Bi）Ci-1 + AiBi

根据逻辑表达式，可以知道只要在半加器的基础上再加入一个异或门、一个两输入与门和两输入或门即可实现全加器。

3、实验原理图：



4、仿真及波形分析：



根据仿真波形对比全加器真值表，可以确定电路实现了全加器的功能。

（三）3线—8线译码器

1、实验任务要求

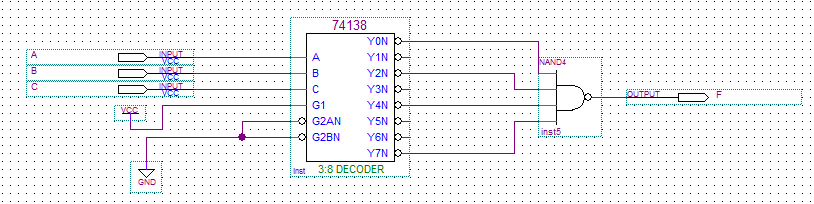
用3线—8线译码器（74LS138）和逻辑门设计实现函数

,仿真验证其功能，并下载到实验板测试。要求用拨码开关设定输入信号，发光二极管显示输出信号。

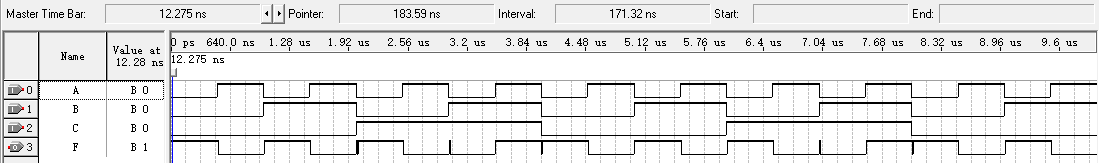
2、设计思路

74LS138是一个3线—8线的译码器，其输出为低电平有效，使能端G1为高电平有效，G2、G3为低电平有效，当其中一个为高电平，输出端全部为1。在中规模集成电路中译码器的几种型号里，74LS138使用最广泛, 只要将相应输出用一个四输入与非门实现即可。

3、实验原理图



4、仿真及波形分析



**实验二**

■实验名称

用VHDL语言实现设计组合逻辑电路

（一）数码管译码器

1、实验任务要求

用VHDL语言设计实现一个共阴极7段数码管译码器，仿真验证其功能，并下载到实验板测试。要求用拨码开关设定输入信号，7段数码管显示输出信号。

2、设计思路和过程

VHDL程序：

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

ENTITY seg7\_1 IS

PORT(a:IN STD\_LOGIC\_VECTOR(3 downto 0);

b:OUT STD\_LOGIC\_VECTOR(6 downto 0);

c:OUT STD\_LOGIC\_VECTOR(7 downto 0));

END seg7\_1;

ARCHITECTURE one OF seg7\_1 IS

BEGIN

PROCESS(a)

BEGIN

CASE a IS

WHEN"0000" =>b<="1111110";

WHEN"0001" =>b<="0110000";

WHEN"0010" =>b<="1101101";

WHEN"0011" =>b<="1111001";

WHEN"0100" =>b<="0110011";

WHEN"0101" =>b<="1011011";

WHEN"0110" =>b<="1011111";

WHEN"0111" =>b<="1110000";

WHEN"1000" =>b<="1111111";

WHEN"1001" =>b<="1111011";

WHEN OTHERS =>b<="0000000";

END CASE;

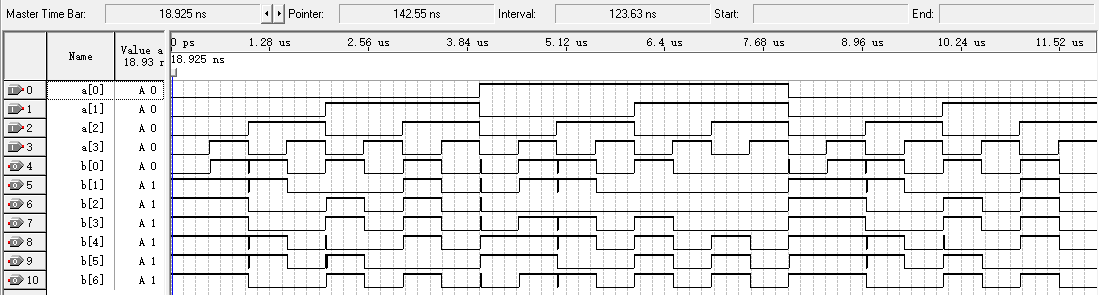
END PROCESS;

c<="11011111";

END one;

由真值表使用CASE语句枚举出结果

3、仿真波形图



根据仿真波形对比数码管译码器的真值表，可以确定实现了数码管译码器的功能。

（二）8421码到格雷码

1、实验任务要求

用VHDL语言设计实现一个8421码转换为格雷码的代码转换器，仿真验证其功能，并下载到实验板测试。要求用拨码开关设定输入信号，发光二极管显示输出信号。

2、设计思路和过程

由真值表写程序

VHDL程序：

library ieee;

use ieee.std\_logic\_1164.all;

entity t8421\_gley is

port

(

i1,i2,i3,i4:in std\_logic;

o1,o2,o3,o4:out std\_logic

);

end t8421\_gley;

architecture t8421\_gley\_arc of t8421\_gley is

begin

process (i1,i2,i3,i4)

variable input:std\_logic\_vector (3 downto 0);

begin

input:=i1&i2&i3&i4;

case input is

when "0000" =>o1<='0';o2<='0';o3<='0';o4<='0';

when "0001" =>o1<='0';o2<='0';o3<='0';o4<='1';

when "0010" =>o1<='0';o2<='0';o3<='1';o4<='1';

when "0011" =>o1<='0';o2<='0';o3<='1';o4<='0';

when "0100" =>o1<='0';o2<='1';o3<='1';o4<='0';

when "0101" =>o1<='0';o2<='1';o3<='1';o4<='1';

when "0110" =>o1<='0';o2<='1';o3<='0';o4<='1';

when "0111" =>o1<='0';o2<='1';o3<='0';o4<='0';

when "1000" =>o1<='1';o2<='1';o3<='0';o4<='0';

when "1001" =>o1<='1';o2<='0';o3<='0';o4<='0';

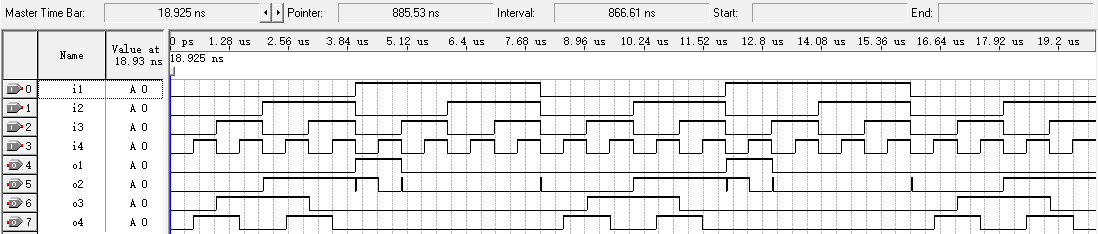
when others =>o1<='0';o2<='0';o3<='0';o4<='0';

end case;

end process;

end;

3、仿真截图



（三）4人表决器

1、实验任务要求

用VHDL语言设计实现一个4人表决器，多数人赞成决议则通过，否则决议不通过，仿真验证其功能，并下载到实验板上测试。要求用数码开关设定输入信号，发光二极管显示输出信号。

2、设计思路和过程：

由于只有4人，可列举情况不多，故考虑使用枚举法，且由题意同意的绝对多才通过，故只需要列举通过的情况简化代码。

Vhdl源代码：

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

ENTITY vote4 IS

PORT

(

v1,v2,v3,v4:in std\_logic;

result:out std\_logic

);

end vote4;

architecture vote4\_arc of vote4 is

begin

process(v1,v2,v3,v4)

variable vote:std\_logic\_vector (3 downto 0);

begin

vote:=v1&v2&v3&v4;

case vote is

when "1111"=>result<='1';

when "1110"=>result<='1';

when "0111"=>result<='1';

when "1011"=>result<='1';

when "1101"=>result<='1';

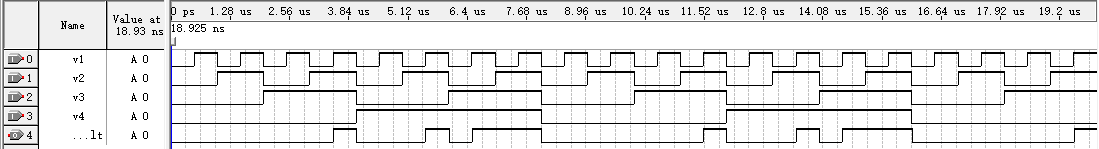
when others =>result<='0';

end case;

end process;

end;

3、实验仿真图



**实验三**

■实验名称

用VHDL实现设计时序逻辑电路

（一）8421码十进制计数器

1、实验任务要求

用VHDL语言设计实现一个带异步复位的8421码十进制计数器，仿真验证其功能，并下载到实验板测试。要求用按键设定输入信号，发光二极管显示输出信号。

2、实验设计思路

使用if判断语句判断时钟沿，当计数达到十个时要将输出归零重新进行计数，而异步功能写在优先级最高的if判断语句内即可。

VHDL源代码：

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity cout\_8421 is

port

(

clk,reset:in std\_logic;

q:out std\_logic\_vector (3 downto 0)

);

end cout\_8421;

architecture cout\_8421\_arc of cout\_8421 is

signal q\_temp: std\_logic\_vector (3 downto 0);

begin

process (clk,reset)

begin

if (reset='1') then q\_temp<="0000";

elsif (clk'event and clk='1') then

if (q\_temp="1001") then q\_temp<="0000";

else q\_temp<=q\_temp+1;

end if;

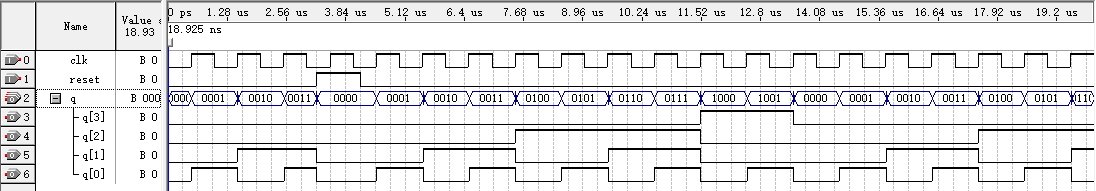
end if;

end process;

q<=q\_temp;

end cout\_8421\_arc;

1. 实验仿真结果



（二）分频器

1、实验任务要求

用VHDL语言设计实现一个分频系数为12，分频输出信号占空比为50%的分频器。

2、实验设计思路

在程序中引入一个计数的信号量temp，检测到时钟沿时让其加一，通过合适的判断达到12分频。

VHDL源代码：

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity div\_12 is

port

(

clk:in std\_logic;

q:out std\_logic

);

end div\_12;

architecture div\_12\_arc of div\_12 is

signal temp:integer range 0 to 5;

signal q\_temp:std\_logic;

begin

process(clk)

begin

if clk'event and clk='1' then

if temp=5 then temp<=0 ;q\_temp<= not q\_temp;

else temp<=temp+1;

end if;

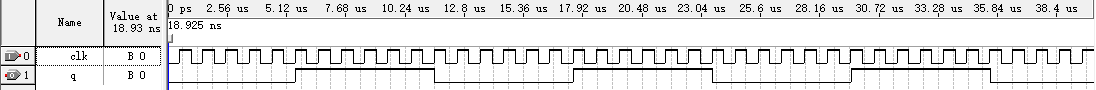
end if;

end process;

q<=q\_temp;

end div\_12\_arc;

1. 实验仿真图



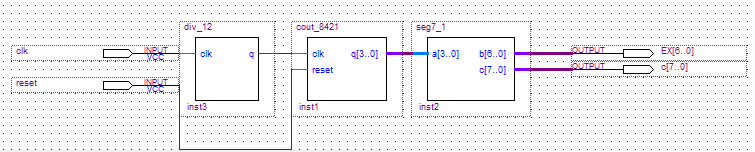
（三）链接元件

1、实验任务要求

将分频器，8421十进制计数器和译码管3个电路进行链接，并下载到实验板显示技术结果

2、实验设计思路

先列出级联的原理图：



**端口说明：**12分频器的输入端接时钟clk，8421计数器的复位端接复位的输入信号，12分频器的输出Q作为8421计数器的时钟clk，12分频器的输出q[3..0]作为译码管的输入端a[3..0],译码管的输出b[6..0]作为最后的输出EX[6..0],C[7..0]作为公共阴极输出端。

再通过component语句调用各个元件并使用map进行链接

VHDL源代码：

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY EX3 IS

PORT(

clk,reset:IN STD\_LOGIC;

EX:OUT STD\_LOGIC\_VECTOR(6 DOWNTO 0);

c:OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0)

);

END EX3;

ARCHITECTURE EX3\_arch OF EX3 IS

component cout\_8421 is

PORT(

clk,reset:IN STD\_LOGIC;

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

END component;

component div\_12 IS

PORT(

clk:IN STD\_LOGIC;

q:OUT STD\_LOGIC);

END component;

component seg7\_1 IS

PORT(

a:IN STD\_LOGIC\_VECTOR(3 DOWNTO 0);

b:OUT STD\_LOGIC\_VECTOR(6 DOWNTO 0);

c:OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0)

);

END component;

signal tmp1:STD\_LOGIC;

SIGNAL tmp2:STD\_LOGIC\_VECTOR(3 DOWNTO 0);

SIGNAL CLKTMP:STD\_LOGIC;

BEGIN

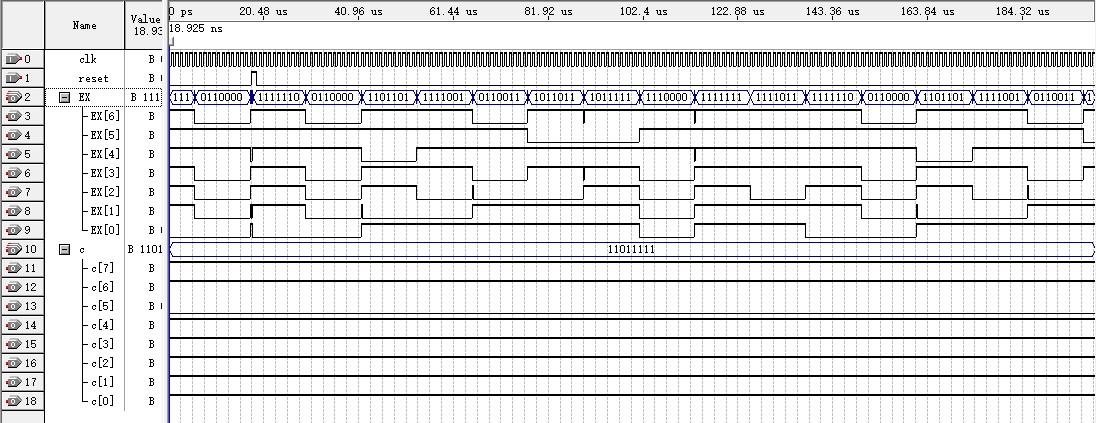
u1: div\_12 port map(clk,tmp1);

u2: cout\_8421 port map (tmp1,reset,tmp2);

u3: seg7\_1 port map(tmp2,EX,c);

END EX3\_arch;

3、实验仿真图



**实验四**

■实验名称

用VHDL实现与设计相关电路

●实验目的

1、 进一步了解时序电路设计方法

2、 掌握多个数码管动态扫描显示的原理及设计方法

●实验仪器

1、计算机

2、直流稳压电源

3、数字系统与逻辑设计实验开发板

●实验原理

多个数码管动态扫描显示，是将所有数码管的相同段并联在一起，通过选通信号分时控制各个数码管的公共端，循环一次点亮多个数码管，利用人眼睛的视觉暂留现象，只要扫描频率大于50Hz，将看不到闪烁。

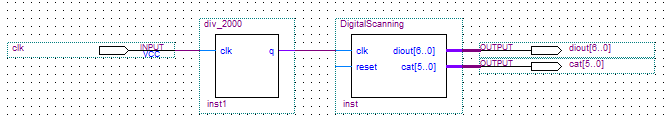
当闪烁显示的发光二极管闪烁频率较高时，我们将观察到持续点亮的现象。同理，当多个现实数码管一次显示，但切换速度足够快时，我们将观察到所有数码管都是同时显示。一个数码管要稳定显示要求显示频率>50Hz，那么6个数码管则需要50×6=300Hz以上才能看到持续稳定点亮的现象。

对于第一个实验,可以用一个时钟信号同时控制着两个输出,一个是选择的数码管(管脚0-5),一个是选择的显示数字(管脚0-6),在时钟的作用下这两个变量在移动,只要令时钟变化的足够快就可以实现六个数码管同时显示的状态.

对于第二个和第三个实验,利用同样的思路,只不过这个时候需要利用分频器再造出一时钟信号来控制整个系统.

●实验总体思路

由于人的视觉暂留效应以及实验室时钟频率的限制，需要先做一个分频器，级联图见下图：



**端口说明**：输入端接CLK时钟，前级为2000分频器，后级为数码显示器，前级的输出Q作为后级的时钟clk，diout[6..0]作为数码管输出端，cat[5..0]作为公共阴极的输出端。

其中2000分频器的VHDL源码为：

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity div\_2000 is

port

(

clk:in std\_logic;

q:out std\_logic

);

end div\_2000;

architecture div\_2000\_arc of div\_2000 is

signal temp:integer range 0 to 999;

signal q\_temp:std\_logic;

begin

process(clk)

begin

if clk'event and clk='1' then

if temp=999 then temp<=0 ;q\_temp<= not q\_temp;

else temp<=temp+1;

end if;

end if;

end process;

q<=q\_temp;

end div\_2000\_arc;

**（注：之后只对数码显示器进行分析和设计）**

●实验内容

1、 用VHDL语言设计并实现六个数码管串行扫描电路，要求同时显示0、1、2、3、4、5

这6个不同的数字图形到6个数码管上，仿真并下载验证其功能。

☛设计思路

cat1-6对应于6个共阴极数码管的公共端，当cat（i）=‘0’时，其对应的数码管被点亮。通过控制cat1~6，在时钟信号的第一个周期开启第一个数码管，第二个周期开启第二个，以此类推，6个数码管依次点亮，由于扫描频率很高，我们会看到连续亮的数码管，1-6号管分别显示012345.

VHDL源代码（本代码根据后面选做部分进行了部分的修改）：

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity DigitalScanning is

port

(

clk,reset:in std\_logic;

diout:out std\_logic\_vector (6 downto 0);

cat:out std\_logic\_vector (5 downto 0)

);

end DigitalScanning;

architecture DigitalScanning\_arch of DigitalScanning is

signal tmp:integer range 0 to 5;

signal count:integer range 0 to 5:=0;

signal tmpplus:integer range 0 to 10:=0;

signal cat\_tmp:std\_logic\_vector (5 downto 0);

begin

p:process(clk,reset)

begin

if reset='1' then tmp<=0;count<=0;

elsif clk'event and clk='1' then

if tmp=5 then

tmp<=0;

count<=count+1;

else

tmp<=tmp+1;

end if;

if count=6 then count<=0;

end if;

for i in 0 to 5 loop

if tmp=i then cat\_tmp(i)<='0';

else cat\_tmp(i)<='1';

end if;

end loop;

for i in 0 to 5 loop

cat(i)<=cat\_tmp(i);

end loop;

tmpplus<=tmp;

case tmpplus is

when 0 =>diout<="1111110"; --0

when 1 =>diout<="0110000"; --1

when 2 =>diout<="1101101"; --2

when 3 =>diout<="1111001"; --3

when 4 =>diout<="0110011"; --4

when 5 =>diout<="1011011"; --5

when 6 =>diout<="1111110"; --0

when 7 =>diout<="0110000"; --1

when 8 =>diout<="1101101"; --2

when 9 =>diout<="1111001"; --3

when 10 =>diout<="0110011"; --4

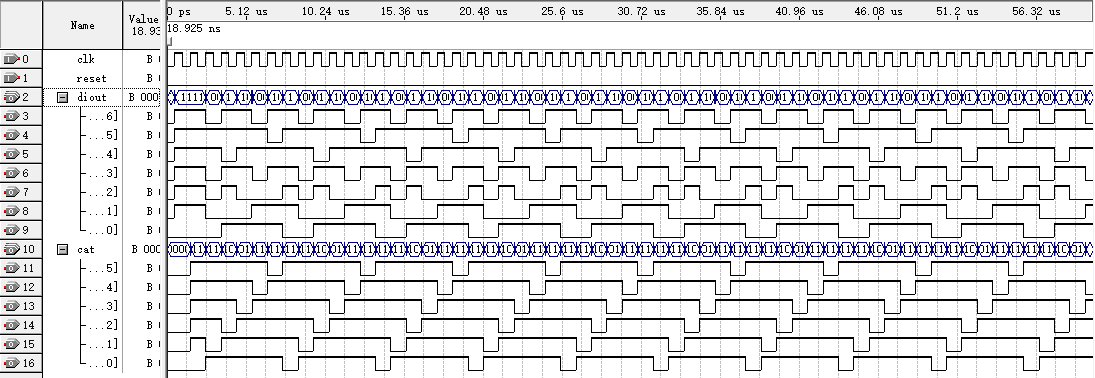
end case;

end if;

end process p;

end DigitalScanning\_arch;

☛实验仿真图



2、 用VHDL语言设计并实现六个数码器滚动显示电路。

（1） 循环左滚动，始终点亮6个数码管，左出右进。状态为：012345——123450——

234501——345012——450123——501234——012345

☛VHDL源代码

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity DS is

port

(

clk,reset:in std\_logic;

diout:out std\_logic\_vector (6 downto 0);

cat:out std\_logic\_vector (5 downto 0)

);

end DS;

architecture DigitalScanning\_arch of DS is

signal tmp:integer range 0 to 5;

signal count:integer range 0 to 5:=0;

signal count2:integer range 0 to 20:=0;

signal tmpplus:integer range 0 to 10:=0;

signal cat\_tmp:std\_logic\_vector (5 downto 0);

begin

p:process(clk,reset)

begin

if reset='1' then tmp<=0;count<=0;

elsif clk'event and clk='1' then

if tmp=5 then

tmp<=0;

count2<=count2+1;

else

tmp<=tmp+1;

end if;

if count2=21 then count2<=0;count<=count+1;

end if;

if count=6 then count<=0;

end if;

for i in 0 to 5 loop

if tmp=i then cat\_tmp(i)<='0';

else cat\_tmp(i)<='1';

end if;

end loop;

for i in 0 to 5 loop

cat(i)<=cat\_tmp(i);

end loop;

tmpplus<=tmp+count;

case tmpplus is

when 0 =>diout<="1111110"; --0

when 1 =>diout<="0110000"; --1

when 2 =>diout<="1101101"; --2

when 3 =>diout<="1111001"; --3

when 4 =>diout<="0110011"; --4

when 5 =>diout<="1011011"; --5

when 6 =>diout<="1111110"; --0

when 7 =>diout<="0110000"; --1

when 8 =>diout<="1101101"; --2

when 9 =>diout<="1111001"; --3

when 10 =>diout<="0110011"; --4

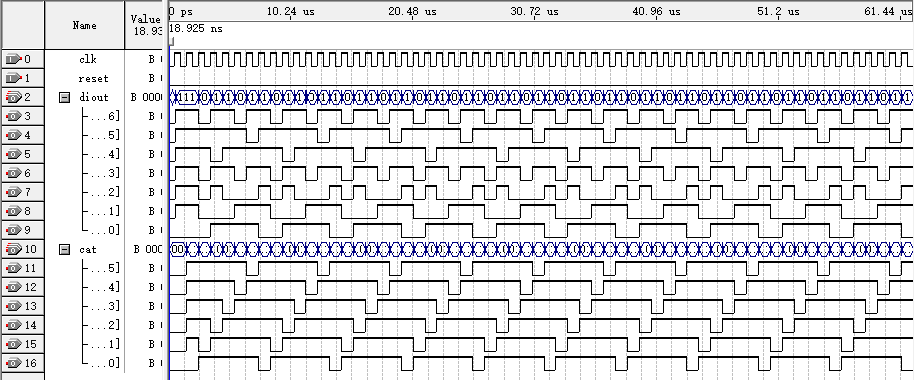
end case;

end if;

end process p;

end DigitalScanning\_arch;

☛实验仿真图及输出分析



波形输出分析：

因为要实现的是循环左滚动，且状态转移的顺序是012345->123450->234501->345012->450123->501234->012345共六个状态.从图中可以清楚地看出这六个状态转移过程中,发光二极管管脚的变化情况. 输入时钟信号后,6个数码管的选通控制端在同一时刻始终保持一个低电平其余高电平,在0-5对应的数码管上输出实验设定的数字,并且由地位向高位依次循环移动.

（2）向左滚动，用全灭的数码管填充右边，直至全部变灭，然后再依次从右边一个一个的点亮。状态为：012345——12345X——2345XX——345XXX——45XXXX——5XXXXX——XXXXXX——XXXXX0——XXXX01——XXX012——XX0123——X01234——012345，其中X表示数码管不显示。

☛VHDL源代码

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity DigitalScanning2 is

port

(

clk,reset:in std\_logic;

diout:out std\_logic\_vector (6 downto 0);

cat:out std\_logic\_vector (5 downto 0)

);

end DigitalScanning2;

architecture DigitalScanning2\_arch of DigitalScanning2 is

signal tmp:integer range 0 to 5;

signal count:integer range 0 to 5:=0;

signal count2:integer range 0 to 11:=0;

signal tmpplus:integer range 0 to 10:=0;

signal cat\_tmp:std\_logic\_vector (5 downto 0);

begin

p:process(clk,reset)

begin

if reset='1' then tmp<=0;count<=0;count2<=0;

elsif clk'event and clk='1' then

if tmp=5 then

tmp<=0;

count<=count+1;

count2<=count2+1;

else

tmp<=tmp+1;

end if;

if count=6 then count<=0;

if count2=12 then count2<=0;

end if;

end if;

for i in 0 to 5 loop

if tmp=i then cat\_tmp(i)<='0';

else cat\_tmp(i)<='1';

end if;

end loop;

for i in 0 to 5 loop

cat(i)<=cat\_tmp(i);

end loop;

case count2 is

when 1=>cat(5)<='1';

when 2=>cat(5)<='1';cat(4)<='1';

when 3=>cat(5)<='1';cat(4)<='1';cat(3)<='1';

when 4=>cat(5)<='1';cat(4)<='1';cat(3)<='1'; cat(2)<='1';

when5=>cat(5)<='1';cat(4)<='1';cat(3)<='1';cat(2)<='1';cat(1)<='1';

when6=>cat(5)<='1';cat(4)<='1';cat(3)<='1';cat(2)<='1';cat(1)<='1';cat(0)<='1';

when7=>cat(4)<='1';cat(3)<='1';cat(2)<='1';cat(1)<='1';cat(0)<='1';

when 8=>cat(3)<='1'; cat(2)<='1';cat(1)<='1';cat(0)<='1';

when 9=>cat(2)<='1';cat(1)<='1';cat(0)<='1';

when 10=>cat(1)<='1';cat(0)<='1';

when 11=>cat(0)<='1';

when 0=>

for i in 0 to 5 loop

cat(i)<=cat\_tmp(i);

end loop;

end case;

tmpplus<=tmp+count;

case tmpplus is

when 0 =>diout<="1111110"; --0

when 1 =>diout<="0110000"; --1

when 2 =>diout<="1101101"; --2

when 3 =>diout<="1111001"; --3

when 4 =>diout<="0110011"; --4

when 5 =>diout<="1011011"; --5

when 6 =>diout<="1111110"; --0

when 7 =>diout<="0110000"; --1

when 8 =>diout<="1101101"; --2

when 9 =>diout<="1111001"; --3

when 10 =>diout<="0110011"; --4

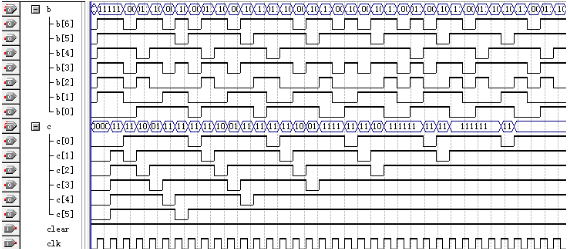
end case;

end if;

end process p;

end DigitalScanning2\_arch;

☛仿真图及分析



波形分析:

由波形图可以清楚地看到12个状态的波形,当时钟clk变化一些周期之后,波形就转到下一个状态.如图所示.状态转移关系:

012345->12345X->2345XX->345XXX->45XXXX->5XXXXXX->XXXXXX->XXXXX0 ->XXXX01->XXX012->XX0123->X01234->012345

二、故障及问题分析

三、总结和结论