

组合逻辑作业

何尔恒 20172333090 通信 6 A

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity keyseg7scan is
PORT(  clk : IN STD_LOGIC; key3 : IN STD_LOGIC; key1 : IN STD_LOGIC; key2 :
IN STD_LOGIC; key4 : IN STD_LOGIC; key5 : IN STD_LOGIC;
seg: out std_logic_vector(7 downto 0); --数码管的段数据, 包括小数点共 8 段 LED, 低
电平点亮
dig: out std_logic_vector(3 downto 0)--数码管的选通信号,共 4 个数码管, 低电平选通
);
END keyseg7scan;
ARCHITECTURE behav OF keyseg7scan IS
--数码管扫描显示用到的信号
SIGNAL scan : STD_LOGIC_VECTOR(1 DOWNT0 0);
SIGNAL divclk : STD_LOGIC; --由基准时钟分频得到的扫描时钟
signal data_r : integer range 0 to 10;
signal num : integer range 0 to 10 ; signal num1 : integer range 0 to 10 ; signal num2 :
integer range 0 to 10 ; signal num3 : integer range 0 to 10 ;

signal key1filt1 : std_logic; --按键 key1 消抖后得到的信号
signal key1cnt1 : integer range 0 to 50000000; --用于对 key1 按键输入有效时间进行计数
signal key1filt2 : std_logic; --按键 key1 消抖后得到的信号
signal key1cnt2 : integer range 0 to 50000000;
signal key1filt3 : std_logic; --按键 key1 消抖后得到的信号
signal key1cnt3 : integer range 0 to 50000000;
signal key1filt4 : std_logic; --按键 key1 消抖后得到的信号
signal key1cnt4 : integer range 0 to 50000000;
signal key1filt5 : std_logic; --按键 key1 消抖后得到的信号
signal key1cnt5 : integer range 0 to 50000000;
begin
--*****

--数码管译码显示部分
--*****

--分频得到扫描时钟
process(clk)
```

```

constant clkfrq :integer := 50000000; --设基准时钟频率 clkfrq 为 50Mhz
constant scanfrq :integer := 50; --设扫描频率 scanfrq 为 50Hz
variable count:integer range 0 to 50000000;

begin
if clk'event and clk = '1' then
if count = clkfrq /(scanfrq*4)-1 then -- 设扫描频率 scanfrq 为 50Hz, 基准时钟频率
clkfrq 为 50Mhz
count := 0;
divclk <= '1'; --得到扫描时钟
else
count := count + 1; divclk <= '0';
end if;
end if;
end process;

process(divclk)
begin

if divclk'event and divclk = '1' then--elsif
scan <= scan + '1';
end if;
end process;

process(scan)
begin
case scan is
when "00" => dig <= "1110"; --选通第一个数码管
data_r <= num; --同时把第一个数码管显示的数据送给总线
when "01" => dig <= "1101"; --选通第二个数码管
data_r <= num1; --同时把第二个数码管显示的数据送给总线
when "10" => dig <= "1011"; --选通第三个数码管
data_r <= num2; --同时把第三个数码管显示的数据送给总线
when "11" => dig <= "0111"; --选通第四个数码管
data_r <= num3; --同时把第四个数码管显示的数据送给总线

end case;
end process;

process(data_r)
begin
case data_r is
when 0 => seg <= "11000000"; -- 0
when 1 => seg <= "11111001"; -- 1

```

```

when 2 => seg <= "10100100"; -- 2
when 3 => seg <= "10110000"; -- 3
when 4 => seg <= "10011001"; -- 4
when 5 => seg <= "10010010"; -- 5
when 6 => seg <= "10000010"; -- 6
when 7 => seg <= "11111000"; -- 7
when 8 => seg <= "10000000"; -- 8
when 9 => seg <= "10010000"; -- 9
when others => seg <="10001110";
end case;
end process;

```

```

process(divclk,key1filt1,key1filt5,key1filt2,key1filt4,key1filt3) begin
if divclk'event and divclk = '1' then
if key1filt1 = '1' then
num <= 4;
end if;
if key1filt2 = '1' then
num <= 3;
end if;
if key1filt3 = '1' then
num <= 2;
end if;
if key1filt4 = '1' then
num <= 1;
end if;
if key1filt5 = '1' then
num <= 5;
end if;
end if;
end process;

```

```

process (key5)
constant N :integer := 20;  --消抖时间，对于 50Mhz 的基准时钟，这相当于 0.1S

begin
if divclk'event and divclk = '1' then
if key5 = '0' then  --当 key1 输入低电平，即按键按下
if key1cnt5 /= N then --一直计数到 N
key1cnt5 <= key1cnt5 + 1;
end if;
if key1cnt5 = N-1 then --最后一个计数时输出 key1filt 脉冲
key1filt5 <= '1';

```

```

else
key1filt5 <= '0';
end if;
else    --若 key1 输入高电平，表明按键被释放
key1cnt5 <= 0;
end if;
end if;
end process;

```

```

process (key1)
constant N :integer := 20;  --消抖时间，对于 50Mhz 的基准时钟，这相当于 0.1S
begin
if divclk'event and divclk = '1' then
if key1 = '0' then  --当 key1 输入低电平，即按键按下
if key1cnt1 /= N then --一直计数到 N
key1cnt1 <= key1cnt1 + 1;
end if;
if key1cnt1 = N-1 then --最后一个计数时输出 key1filt 脉冲
key1filt1 <= '1';
else
key1filt1 <= '0';
end if;
else    --若 key1 输入高电平，表明按键被释放
key1cnt1 <= 0;
end if;
end if;
end process;

```

```

process (key2)
constant N :integer := 20;  --消抖时间，对于 50Mhz 的基准时钟，这相当于 0.1S
begin
if divclk'event and divclk = '1' then
if key2 = '0' then  --当 key1 输入低电平，即按键按下
if key1cnt2 /= N then --一直计数到 N
key1cnt2 <= key1cnt2 + 1;
end if;
if key1cnt2 = N-1 then --最后一个计数时输出 key1filt 脉冲
key1filt2 <= '1';
else
key1filt2 <= '0';
end if;
else    --若 key1 输入高电平，表明按键被释放

```

```

key1cnt2 <= 0;
end if;
end if;
end process;

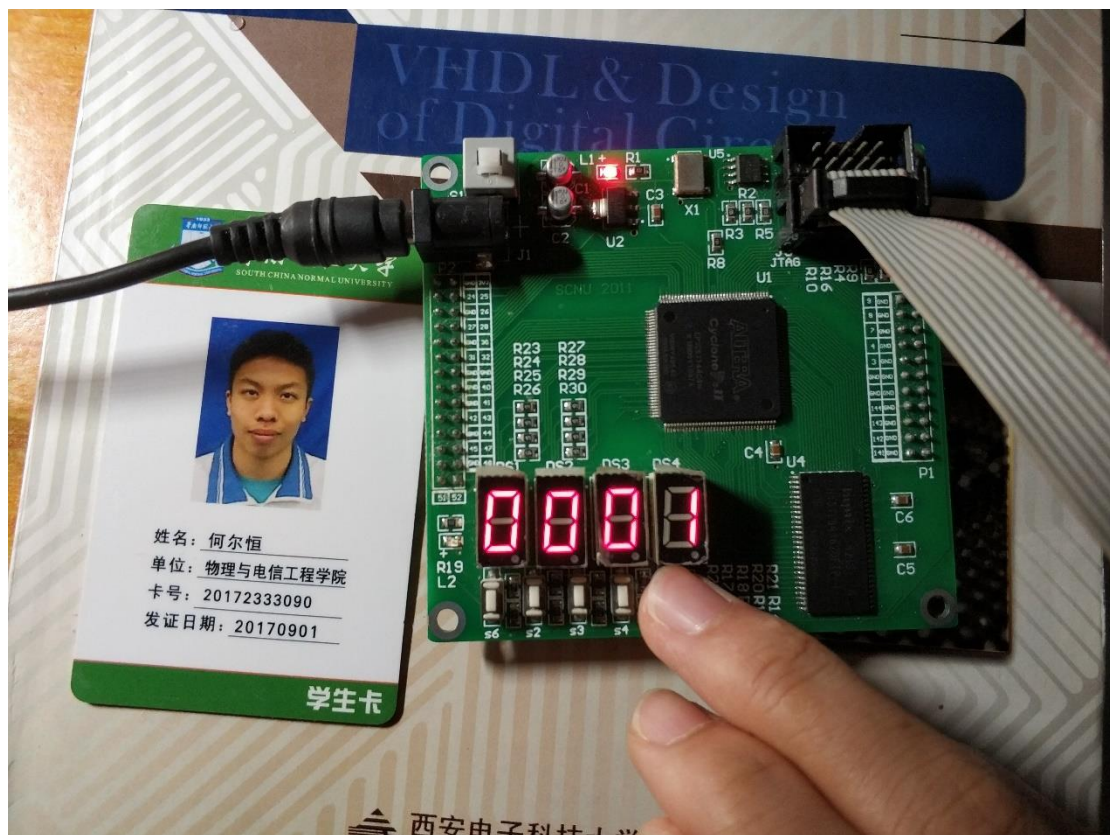
process (key3)
constant N :integer := 20;  --消抖时间, 对于 50Mhz 的基准时钟, 这相当于 0.1S
begin
if divclk'event and divclk = '1' then
if key3 = '0' then  --当 key1 输入低电平, 即按键按下
if key1cnt3 /= N then --一直计数到 N
key1cnt3 <= key1cnt3 + 1;
end if;
if key1cnt3 = N-1 then --最后一个计数时输出 key1filt 脉冲
key1filt3 <= '1';
else
key1filt3 <= '0';
end if;
else  --若 key1 输入高电平, 表明按键被释放
key1cnt3 <= 0;
end if;
end if;
end process;

process (key4)
constant N :integer := 20;  --消抖时间, 对于 50Mhz 的基准时钟, 这相当于 0.1S
begin
if divclk'event and divclk = '1' then
if key4 = '0' then  --当 key1 输入低电平, 即按键按下
if key1cnt4 /= N then --一直计数到 N
key1cnt4 <= key1cnt4 + 1;
end if;
if key1cnt4 = N-1 then --最后一个计数时输出 key1filt 脉冲
key1filt4 <= '1';
else
key1filt4 <= '0';

end if;
else  --若 key1 输入高电平, 表明按键被释放
key1cnt4 <= 0;
end if;
end if;
end process;

```

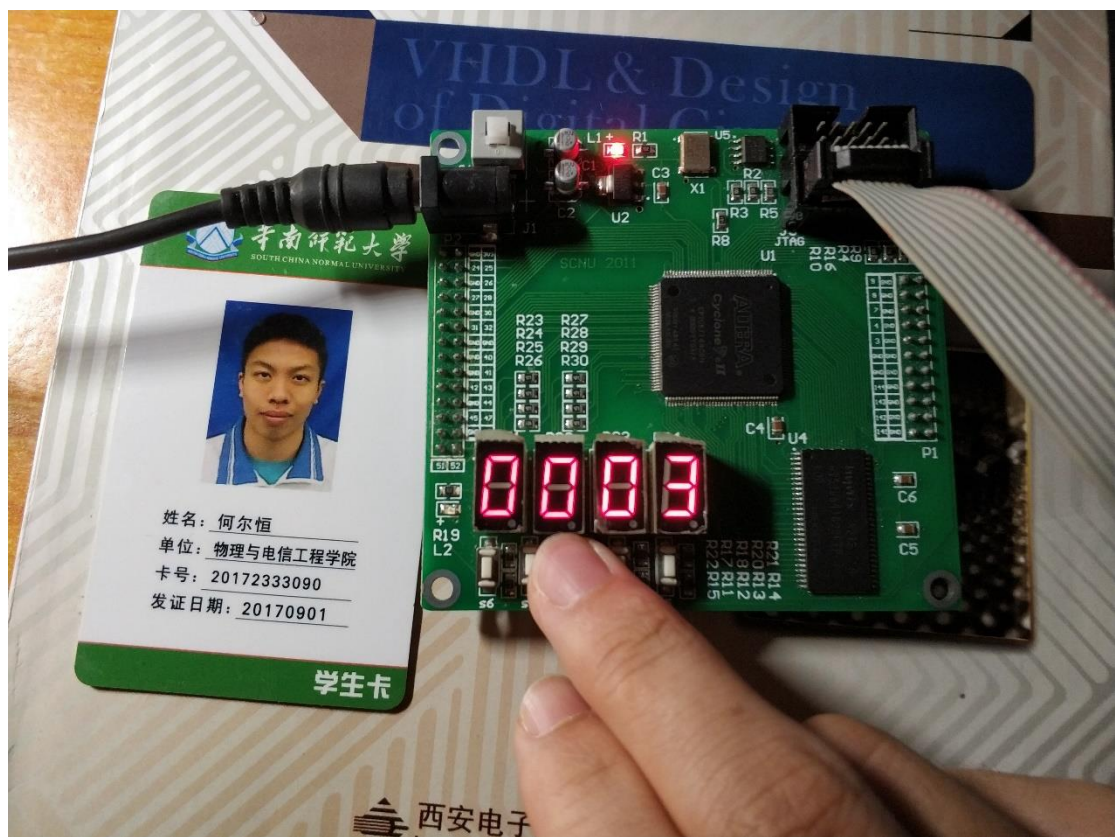
end behav;



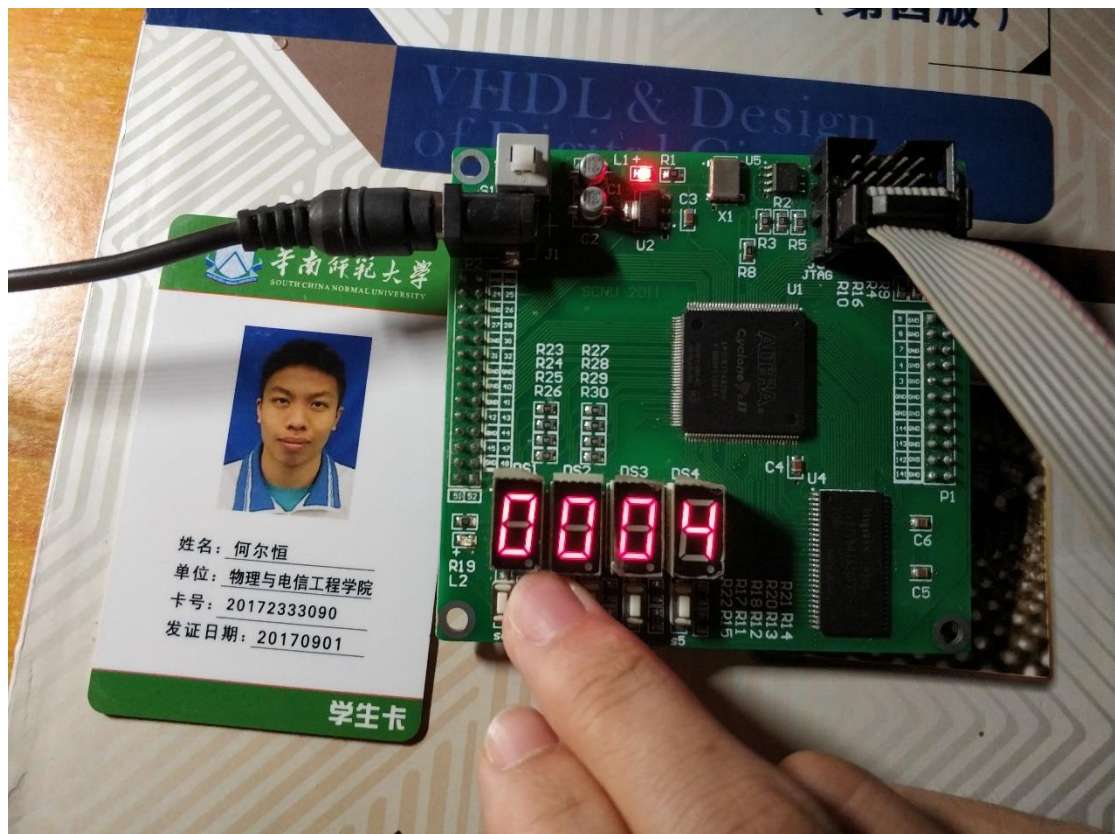
按下第一个按键



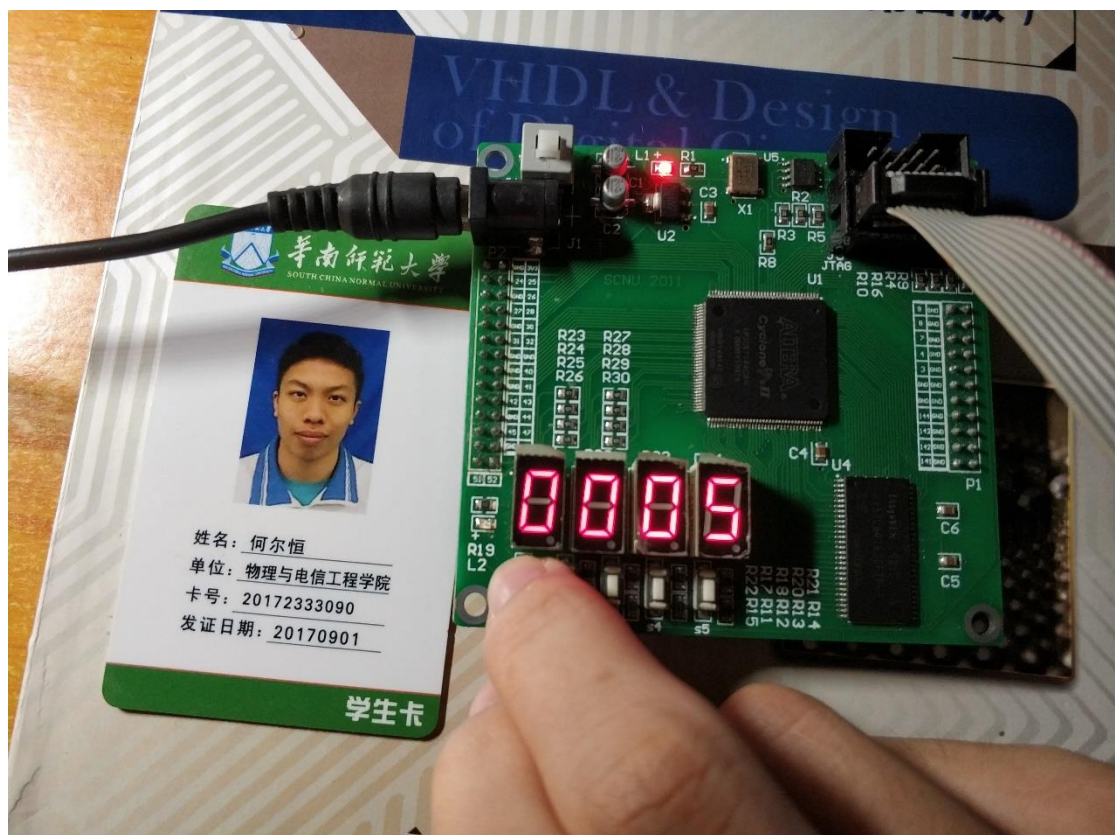
按下第二个按键



按下第三个按键



按下第四个按键



按下第五个按键