Lab2：三位十进制计数器

高政 11912015

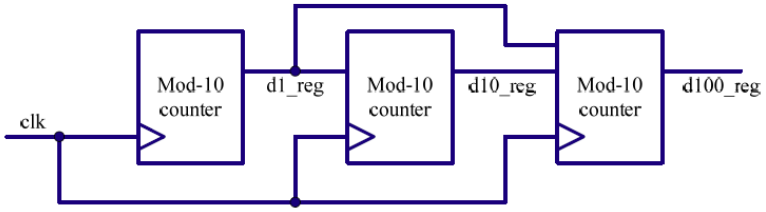
1. 简述

(a) 实验目的

本次实验通过设计VHDL代码实验三位十进制计数器来学习VHDL语言中的条件并发语句和if嵌套语句。

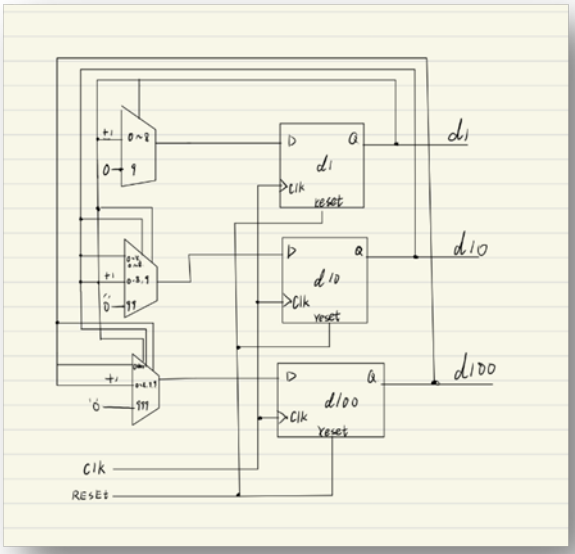
1. (a) 十进制计数器

在十进制计数体制中，每位数都可能是0，1，2，...，9十个数码中的任意一个，且“逢十进一“。根据计数器的构成原理，必须由四个触发器的状态来表示一位十进制数的四位二进制编码。本次实验使用8421BCD码的编码方式来表示一位十进制数。



通过D触发器实现的三位十进制计数器逻辑狂徒如下：

1. 实验准备



通过D触发器实现的三位十进制计数器电路图如下：

1. VHDL代码与test bench代码

实验中使用条件并发语句实现三位十进制计数器：

```
library IEEE;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
entity three_digit_decimal_counter is
Port ( CLK : in STD_LOGIC;
RESET : in STD_LOGIC;
d1 : out STD_LOGIC_VECTOR (3 downto 0);
d10 : out STD_LOGIC_VECTOR (3 downto 0);
d100 : out STD_LOGIC_VECTOR (3 downto 0));
end three_digit_decimal_counter;
architecture Behavioral of three_digit_decimal_counter is
signal d1_reg, d10_reg, d100_reg: std_logic_vector (3 downto 0);
signal d1_next, d10_next, d100_next: std_logic_vector (3 downto 0);
```

```
begin
– register
process (CLK, RESET) is
begin
if RESET = '1' then
d1_reg <="0000";
d10_reg <="0000";
d100_reg <="0000";
elsif CLK'event and CLK='1' then
d1_reg <= d1_next;
d10_reg <= d10_next;
d100_reg <= d100_next;
end if;
end process;
– next-state logic
d1_next <= "0000" when d1_reg = 9 else d1_reg+1;
d10_next <= "0000" when (d1_reg = 9 and d10_reg = 9) else
d10_reg+1 when d1_reg = 9 else d10_reg;
d100_next <= "0000" when (d1_reg=9 and d10_reg=9 and d100_reg=9)
else
d100_reg+1 when (d1_reg=9 and d10_reg=9) else d100_reg;
– Output logic
d1 <= d1_reg; d10 <= d10_reg; d100 <= d100_reg;
end Behavioral;
```

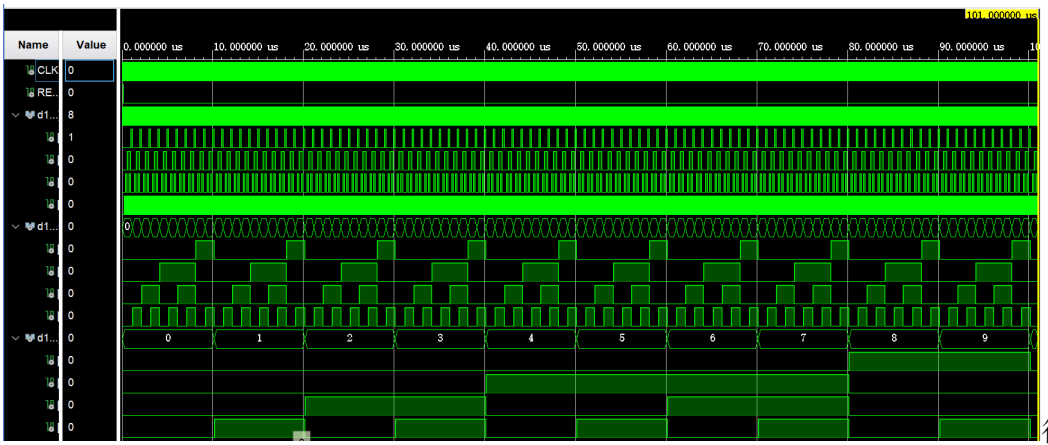
test bench文件如下：

```
library IEEE;
use ieee.std_logic_1164.all;
USE ieee.numeric_std.ALL;
use ieee.std_logic_unsigned.all;
entity tb_three_digit_decimal_counter is
end tb_three_digit_decimal_counter;
architecture Behavioral of tb_three_digit_decimal_counter is
component three_digit_decimal_counter
```

```
Port ( CLK : in STD_LOGIC;
RESET : in STD_LOGIC;
d1 : out STD_LOGIC_VECTOR (3 downto 0);
d10 : out STD_LOGIC_VECTOR (3 downto 0);
d100 : out STD_LOGIC_VECTOR (3 downto 0));
end component;
–Inputs
signal CLK : std_logic := '0';
signal RESET : std_logic := '0';
–Outputs
signal d1 : std_logic_vector(3 downto 0):="0000";
signal d10 : std_logic_vector(3 downto 0):="0000";
signal d100 : std_logic_vector(3 downto 0):="0000";
begin
– Instantiate the Unit Under Test (UUT)
uut: three_digit_decimal_counter PORT MAP (
CLK => CLK,
RESET => RESET,
d1 => d1,
d10 => d10,
d100 => d100
);
clock_gen: process
constant period : time := 100 ns;
begin
CLK <= '0';
wait for period/2;
CLK <= '1';
wait for period/2;
end process;
reset_process : process
begin
RESET <= '1';
```

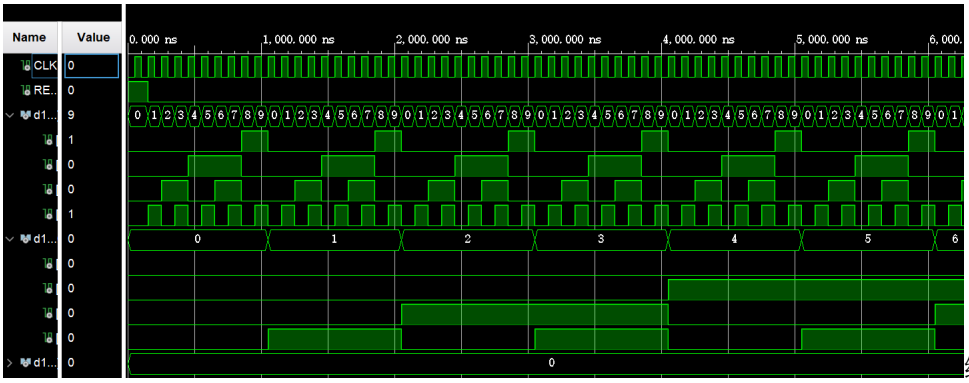
```
for i in 1 to 2 loop
wait until CLK = '1';
end loop;
RESET <= '0';
wait;
end process;
monitor : process
constant n : integer := 1000;
variable number : integer range 0 to 999 :=0;
begin
wait until reset <= '0';
wait for 1 ns;
for i in 0 to n loop
    number:=to_integer(unsigned(d100))*100+to_integer(unsigned(d10))*10+
    to_integer(unsigned(d1));
assert number = i mod n
report "count of " & integer'image(i mod n) & " failed"
severity error;
wait until clk = '1';
wait for 1 ns;
end loop;
wait;
end process;
end Behavioral;
```

1. 实验结果与分析

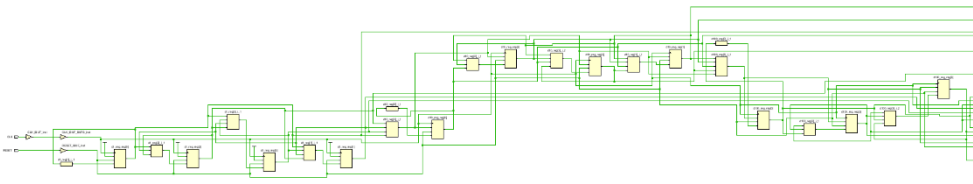
(a) 

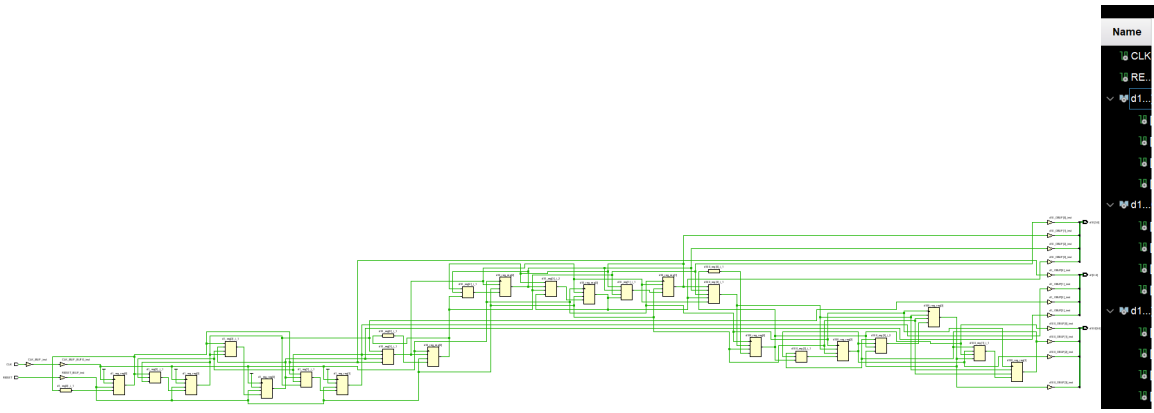
行为仿真

行为仿真结果如图，代码完整实现三位十进制计数器的功能。

1. (a) 

综合仿真

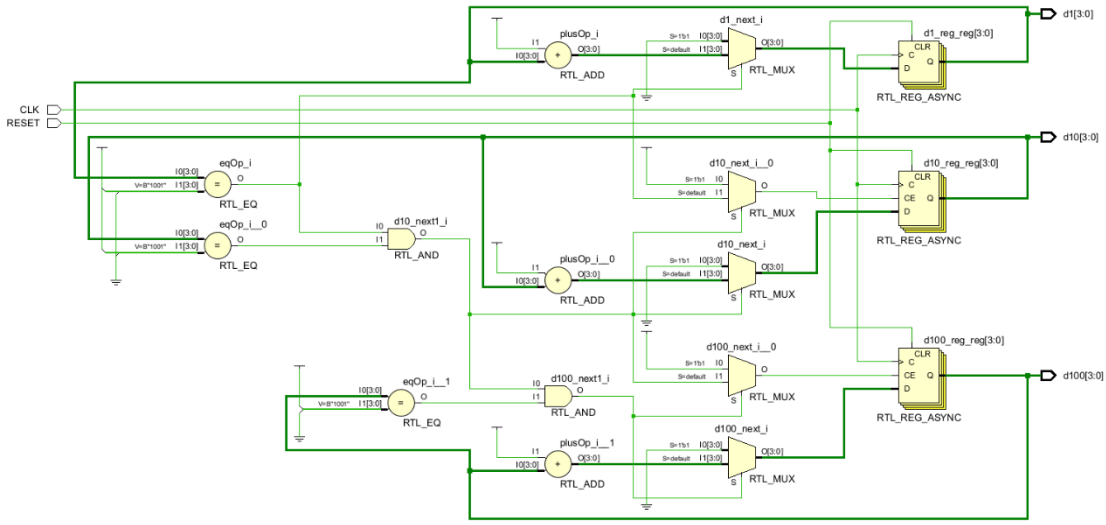
综合仿真结果如上。 

1. (a) 

序仿真

时序仿真结果及原理图如上。

1. (a) RTL分析



RTL原理图如下，可以发现，根据代码生成的电路原理图与实验前绘制的电路原理图基本一致。

1. 实验结论

通过本次实验我学习了VHDL语言中的条件并发语句使用，并成功实现了三位十进制计数器的仿真。实验过程中遇到了诸如 Error: No suitable operator definition found for "*". 等问题，再浏览相关文章后均找到了解决方案。但由于未知原因，我的FPGA板

暂不能连接到仿真器，故无法继续验证在实际情况下程序的时序等项目，目前我让在努力尝试解决FPGA板与仿真器的连接问题。