# 《数字电路与FPGA》课程综合设计实验报告

频率计设计

含弘学院 通信工程**专业** 2018**级**袁隆平**班** **学号**222018316210108 **姓名**黄彬山

## 实验目的

运用数字电路知识，用VHDL语言对FPGA器件进行编程，连接BASYS3板子上的外围电路：四位七段数码管、100MHz的时钟和I/O扩展口，完成一个数字电路综合系统设计。设计一个测频范围是1~9999Hz的频率计，用数码管显示结果。

## 实验器材

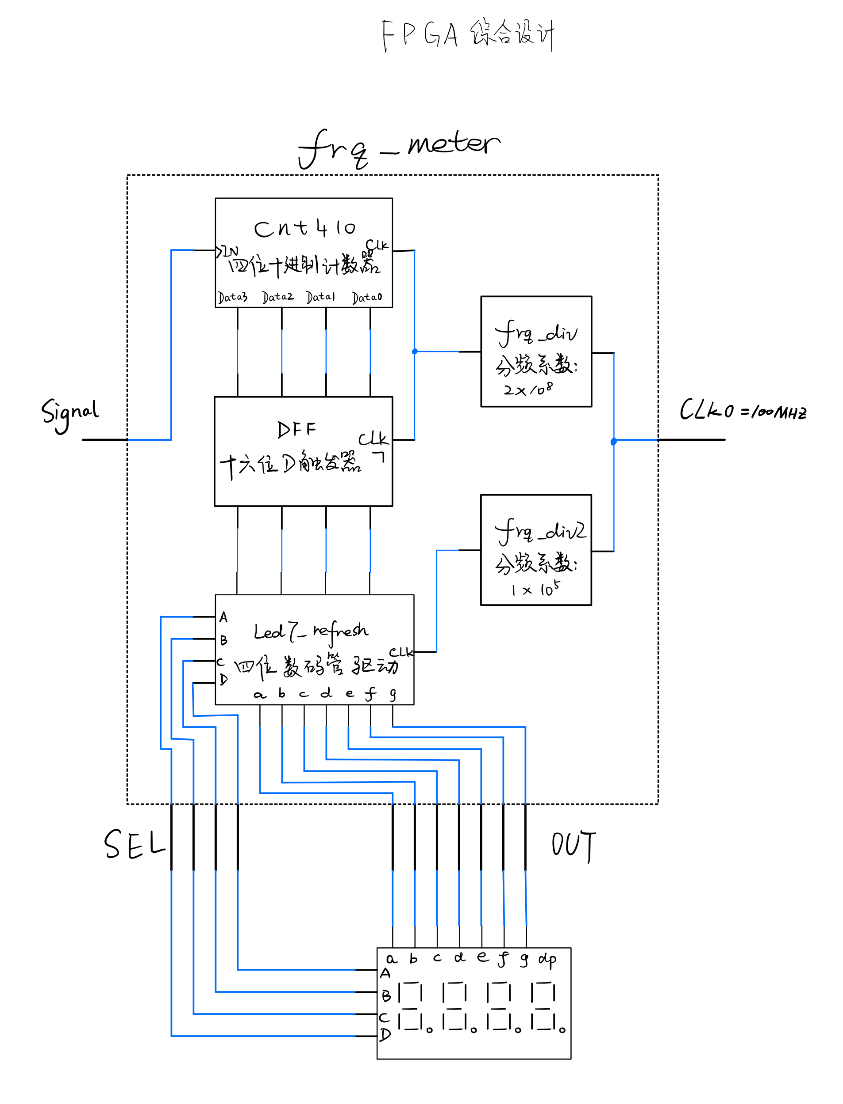
Digilent BASYS3 FPGA平台、Xilinx Vivado 2017.4（FPGA开发工具）、一根杜邦线。

## 实验原理

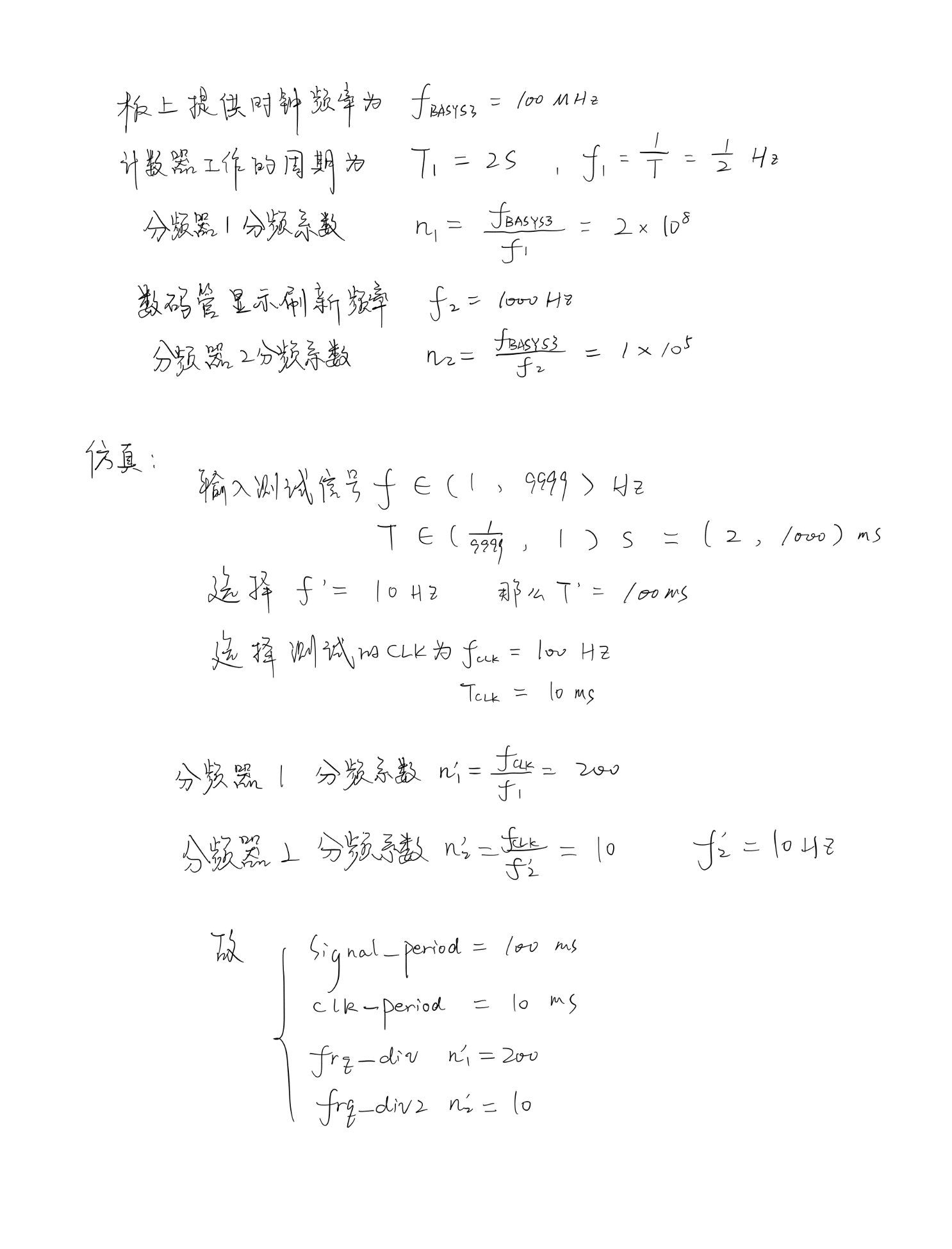
* 1. 总体

利用BASYS3板上提供的100MHz的时钟分频得到0.5Hz的时钟信号，在时钟的高电平期间（持续一秒钟）利用四位十进制计数器计算输入端的脉冲数目，即为输入信号的频率。将结果输入到低电平触发的D触发器暂存。在时钟下降沿时，D触发器将数据传输到数码管驱动模块，由数码管驱动控制四位数码管显示出待测信号的频率。

使用vivado 2017.4平台，采用模块化的设计思想，自顶向下地进行设计



设计草图



* 1. 分频器

分频器初始化一个n值（n=分频系数），将n值赋给变量cnt，在探测到一个输入时钟clkin上升沿就将其减一，cnt大于n/2时，分频后的时钟信号clkout输出为‘0’；否则输出为‘1‘。当cnt减为-1时，重新赋值为n。调整n的取值可以获得分频系数不一样的分频器。

本实验所设计的数字系统用到两个分频器，在使用100MHz的BASYS3板载时钟时：分频器1分频系数为两亿，用于驱动计数器和触发器工作；分频器2分频系数为九十万，用于动态刷新数码管显示。

* 1. D触发器

一个时钟下降沿触发的D触发器，当下降沿到来时，将输入的值传递给输出。用于暂存计数器的结果。

* 1. 四位十进制计数器

设计为四个同步的十进制计数器级联，使能端为高电平有效，在使能端为零时，计数器各位置零。将使能端接到分频后的时钟信号上，计数的时间为1秒。计数的结果用四组四位的信号并行传输到D触发器。

* 1. 数码管驱动程序

接收从D触发器传过来的结果，利用分频器2提供的时钟信号，以约30Hz（一位）的频率刷新数码管。数码管刷新驱动程序由一个4选1的数据选择器，一个二进制加计数器和一个显示译码器组成。当时钟上升沿到来时，计数器从“00”增加到“11”，一共四个状态，代表数码管的四个位，作为sel信号传输给数据选择器，而数据选择器依次将四组数据传输到译码显示器，由它译码为数码管的显示信号输出，于此同时，因为数码管是共阳极的，第二个译码器将“00”~“11”四个状态译码为“0111”~“1110”，作为数码管显示的位选信号。

## 实验内容

* 1. 各个模块独立编写

-- --------------------------------用于驱动核心工作计数---------------------------------

entity frq\_div is

generic(n:integer:=200000000); -- 分频系数 200000000 分频后0.5hz 用于驱动核心工作计数

Port ( clk : in STD\_LOGIC;

clkout : out std\_logic);

end frq\_div;

architecture Behavioral of frq\_div is

begin

process(clk)

variable count:integer range n-1 downto 0:=n-1;

begin

if (clk'event AND clk='1') then

count:=count-1;

if (count>=n/2) then

clkout<='0';

else

clkout<='1';

-- count:=0;

end if;

if count<=0 then

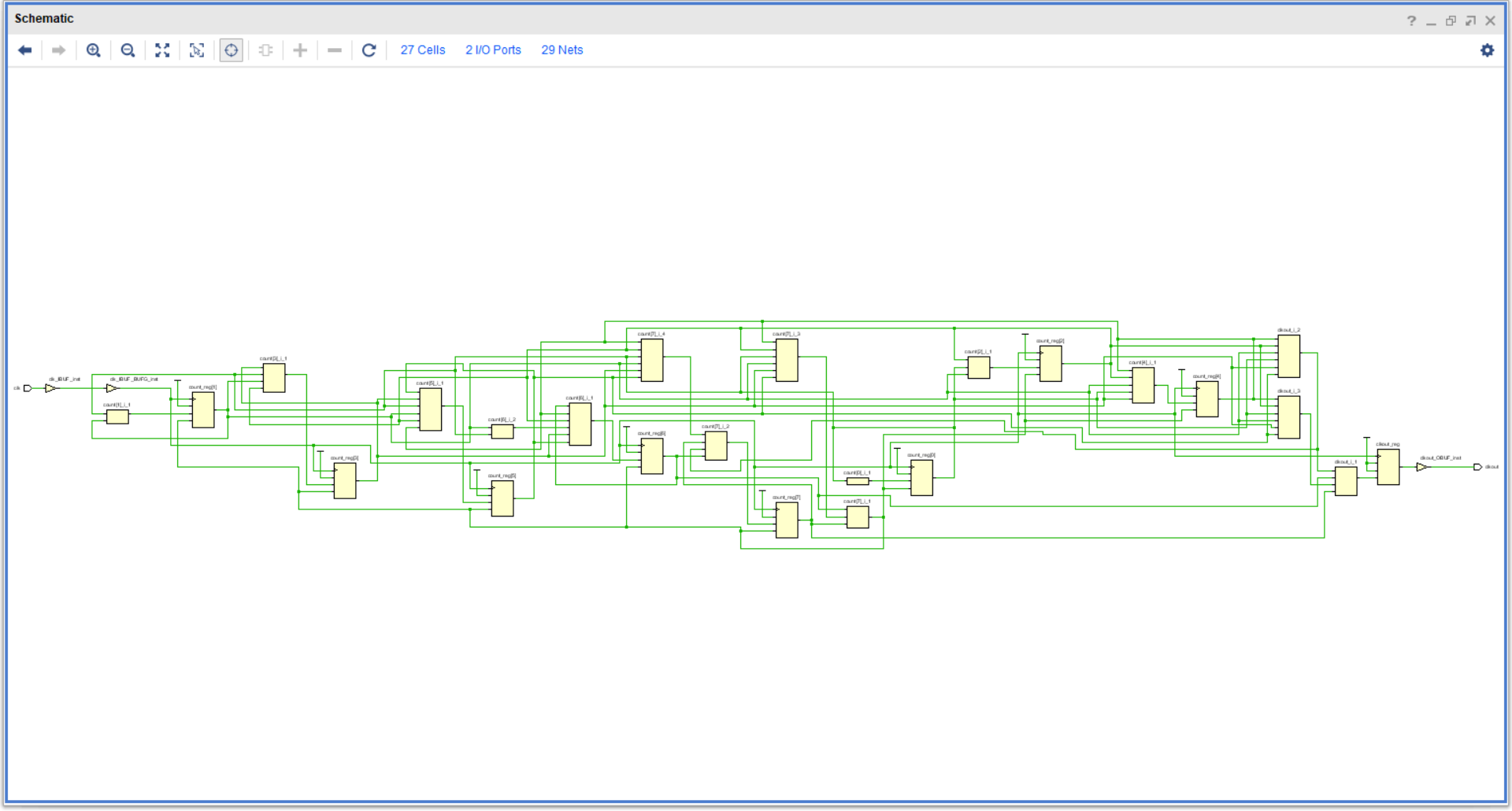
count:=n-1;

end if;

end if;

end process;

end Behavioral;



分频器1综合电路图

-- --------------------------分频器2用于刷新显示器-----------------------------------

entity frq\_div2 is

generic(n:integer:=905000); -- 分频系数 905000 ，分频后刷新率约30 hz

Port ( clk : in STD\_LOGIC;

clkout : out std\_logic);

end frq\_div2;

architecture Behavioral of frq\_div2 is

begin

process(clk)

variable count:integer range n-1 downto 0:=n-1;

begin

if (clk'event AND clk='1') then

count:=count-1;

if (count>=n/2) then

clkout<='0';

else

clkout<='1';

-- count:=0;

end if;

if count<=0 then

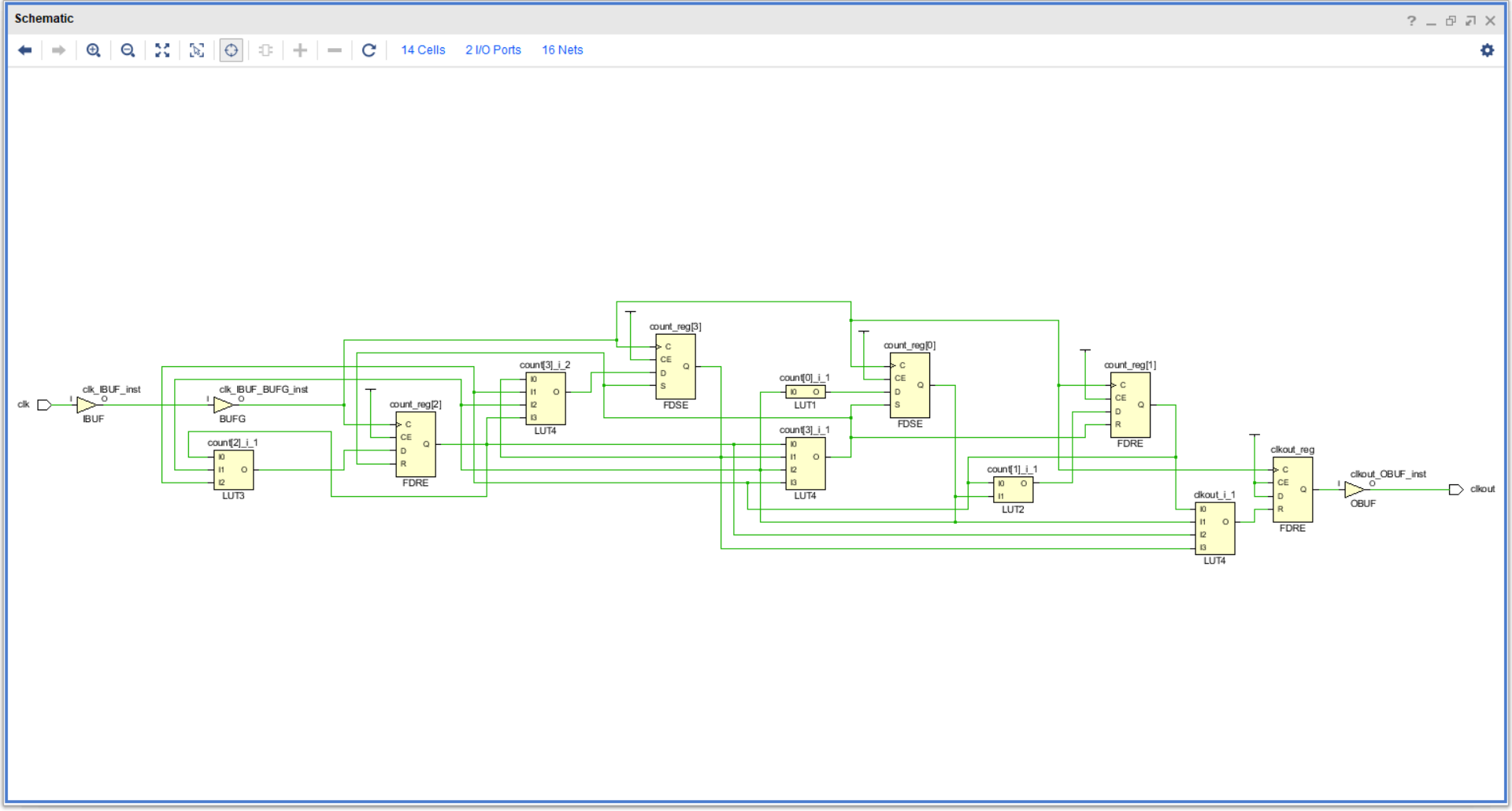
count:=n-1;

end if;

end if;

end process;

end Behavioral;



分频器2综合电路图

------------------------------- D触发器用于暂存计数结果---------------------------------

entity DFF is

Port ( D : in STD\_LOGIC\_VECTOR(3 downto 0);

CLK\_DFF : in STD\_LOGIC;

Q : out STD\_LOGIC\_VECTOR(3 downto 0));

end DFF;

architecture Behavioral of DFF is

begin

process(CLK\_DFF)

begin

if(CLK\_DFF'event and CLK\_DFF='0') then -- 下降沿触发

Q <= D;

end if;

end process;

end Behavioral;

entity cnt10 is

port(

CIN:in std\_logic;

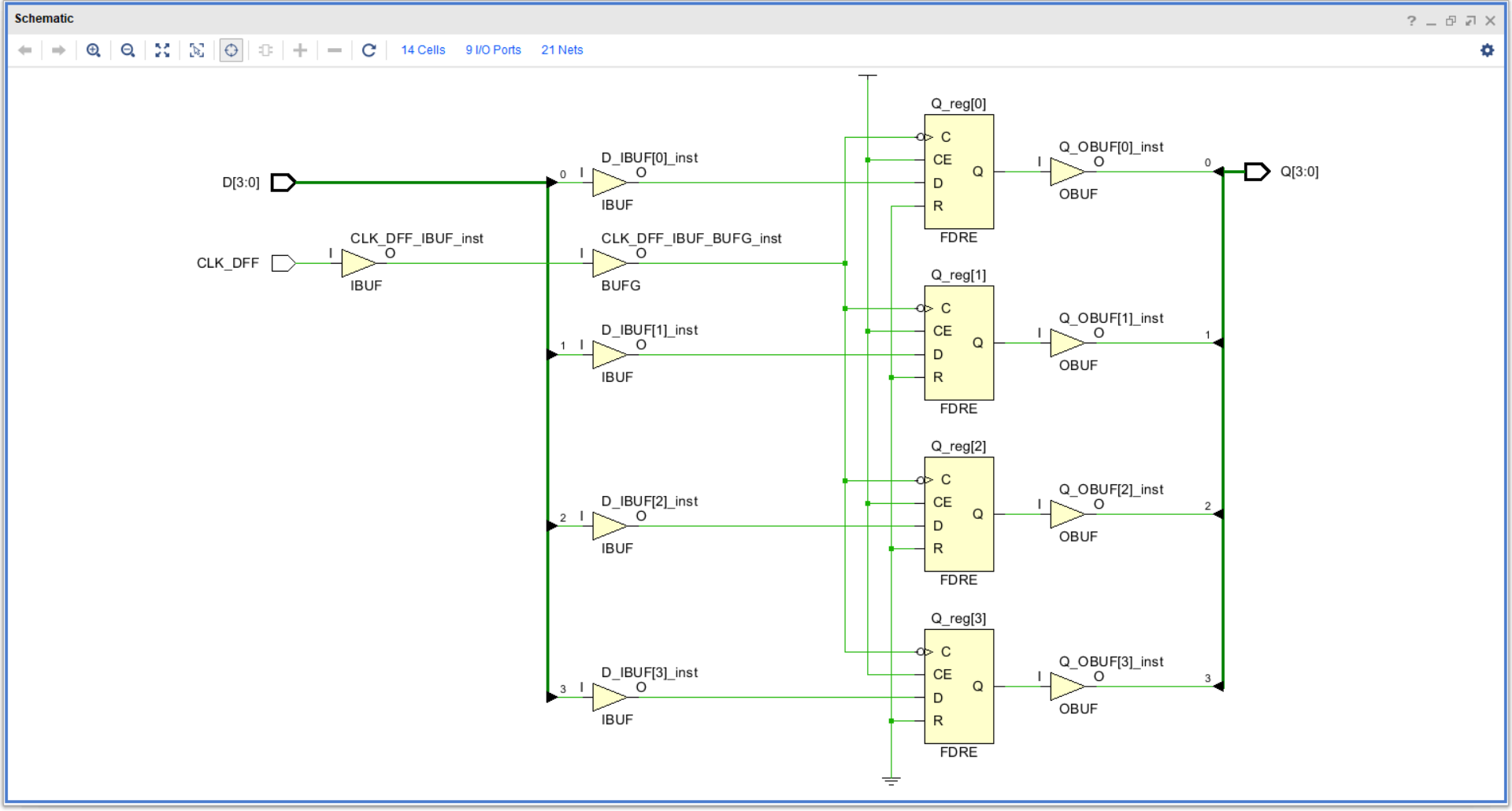
CLK: in std\_logic;

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

COUT : OUT std\_logic -- 进位信号

);

end cnt10;



4组16位D触发器电路图

-- --------------------十进制计数器，组成同步四位十进制计数器的模块----------------------

architecture behav of cnt10 is

begin

process(CIN,CLK)

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

begin

if CLK = '0' then Q := "0000";

elsif CLK = '1' then

if CIN 'event and CIN = '1' then

if Q = "1001" then

Q := "0000";

COUT <= '1';

else

Q := Q+1;

COUT <= '0';

end if;

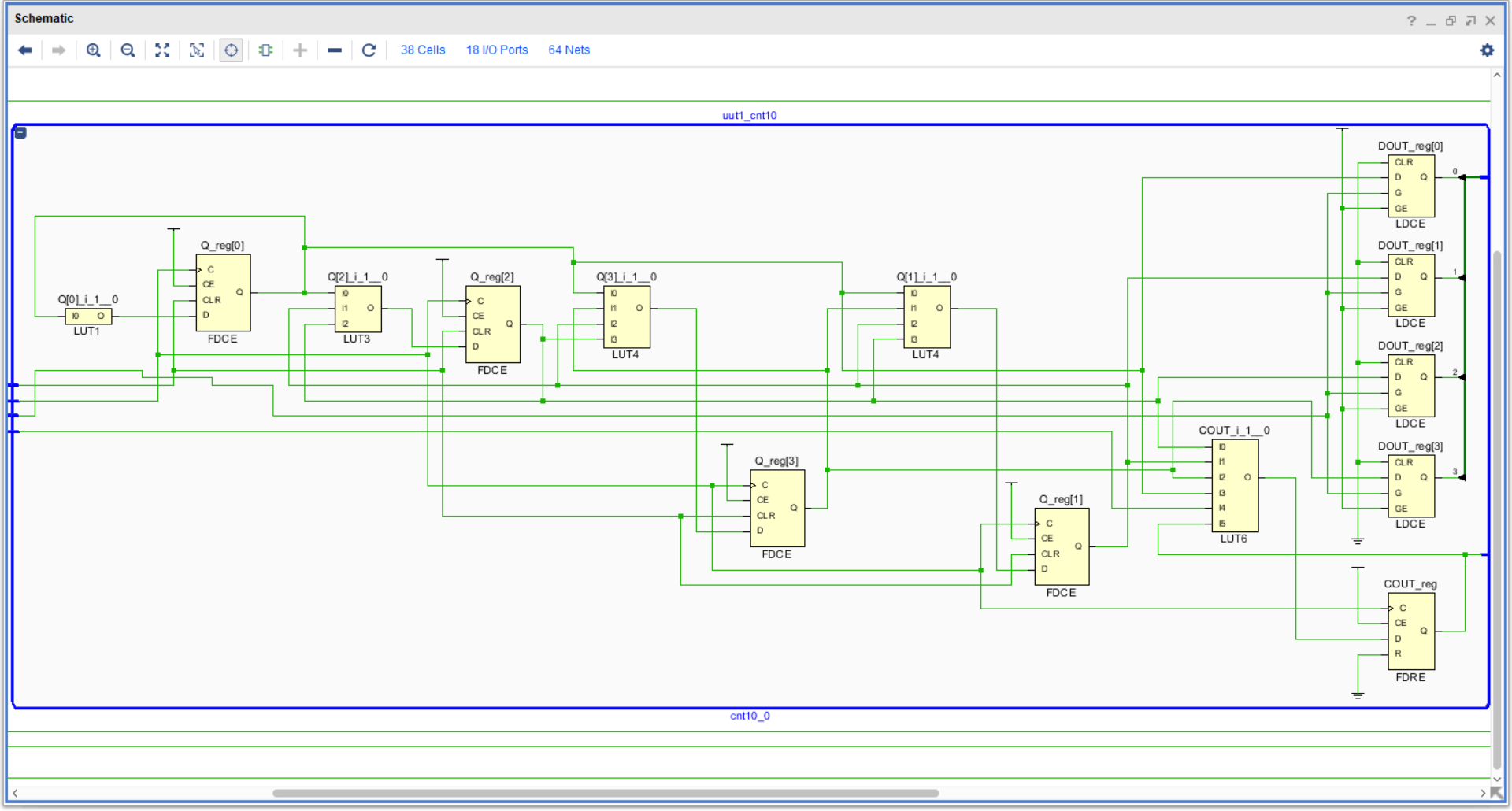
end if;

DOUT <= Q;

end if;

end process;

end behav;



一位十进制计数器综合电路图

-- -------------------------------------四位十进制计数器--------------------------------

entity cnt410 is

port(

signal\_in:in std\_logic;

clk\_in: in std\_logic;

OUT3 : out std\_logic\_vector (3 downto 0);

OUT2 : out std\_logic\_vector (3 downto 0);

OUT1 : out std\_logic\_vector (3 downto 0);

OUT0 : out std\_logic\_vector (3 downto 0));

end cnt410;

architecture Behavioral of cnt410 is

component cnt10

port(

CIN:in std\_logic;

CLK: in std\_logic;

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

COUT : OUT std\_logic);

end component;

signal a:std\_logic;

signal b:std\_logic;

signal c:std\_logic;

signal d:std\_logic;

begin

uut0\_cnt10:cnt10 port map

( CIN => signal\_in,

CLK => clk\_in,

DOUT => OUT0,

COUT => a);

uut1\_cnt10:cnt10 port map

( CIN => a,

CLK => clk\_in,

DOUT => OUT1,

COUT => b);

uut2\_cnt10:cnt10 port map

( CIN => b,

CLK => clk\_in,

DOUT => OUT2,

COUT => c);

uut3\_cnt10:cnt10 port map

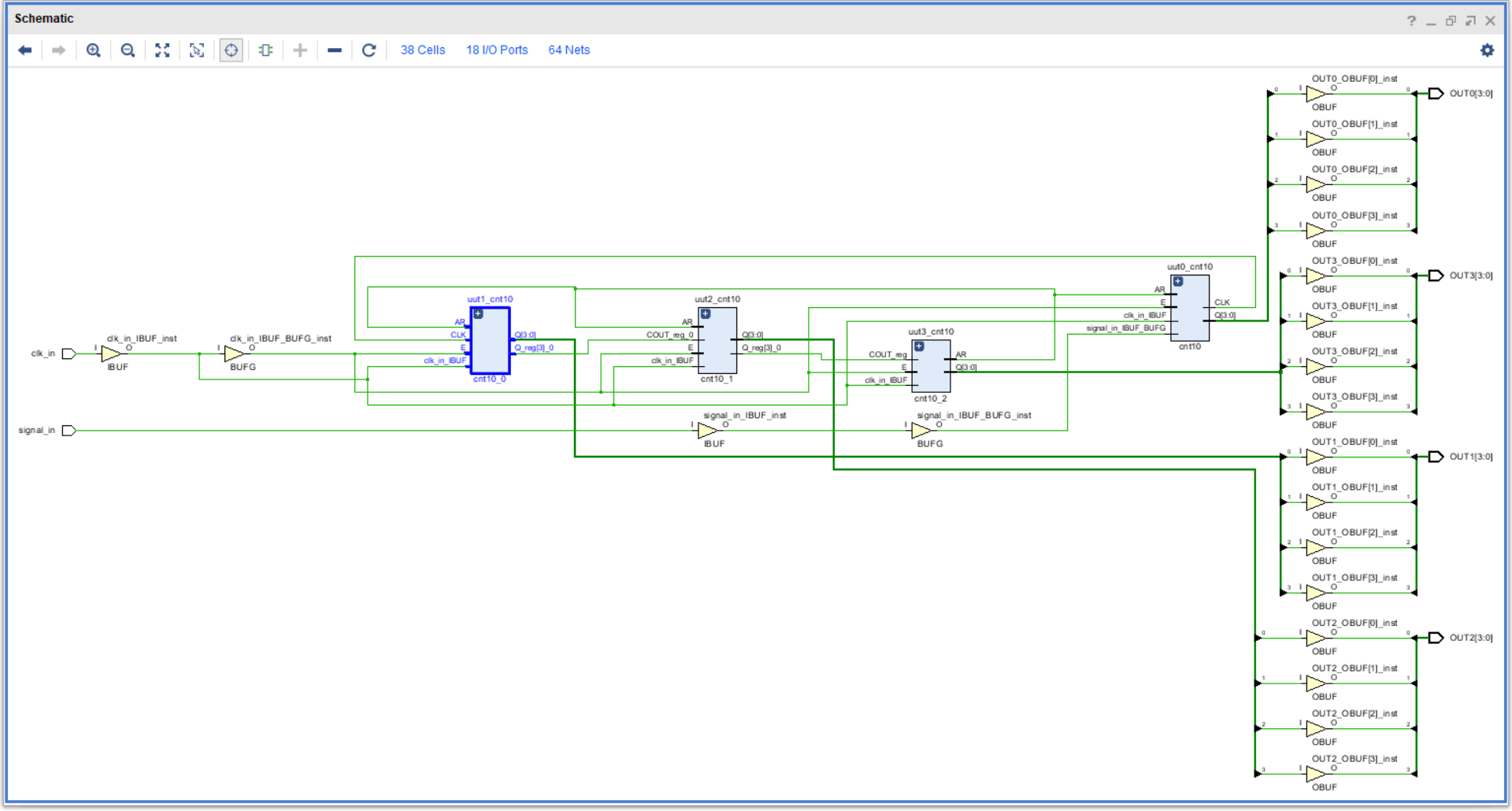
( CIN => c,

CLK => clk\_in,

DOUT => OUT3,

COUT => d);

end Behavioral;



四位十进制计数器综合电路图

-- ------------------------------------七段数码管译码器--------------------------------

Entity led7 is

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

a:out std\_logic;

b:out std\_logic;

c:out std\_logic;

d:out std\_logic;

e:out std\_logic;

f:out std\_logic;

g:out std\_logic);

end led7;

Architecture LED of led7 is

signal y: STD\_LOGIC\_VECTOR(6 DOWNTO 0);

begin

process(DATA)

begin

case DATA is

when "0000"=> y<="0000001" ;

when "0001"=> y<="1001111" ;

when "0010"=> y<="0010010" ;

when "0011"=> y<="0000110" ;

when "0100"=> y<="1001100" ;

when "0101"=> y<="0100100" ;

when "0110"=> y<="0100000" ;

when "0111"=> y<="0001111" ;

when "1000"=> y<="0000000" ;

when "1001"=> y<="0000100" ;

when others=>null;

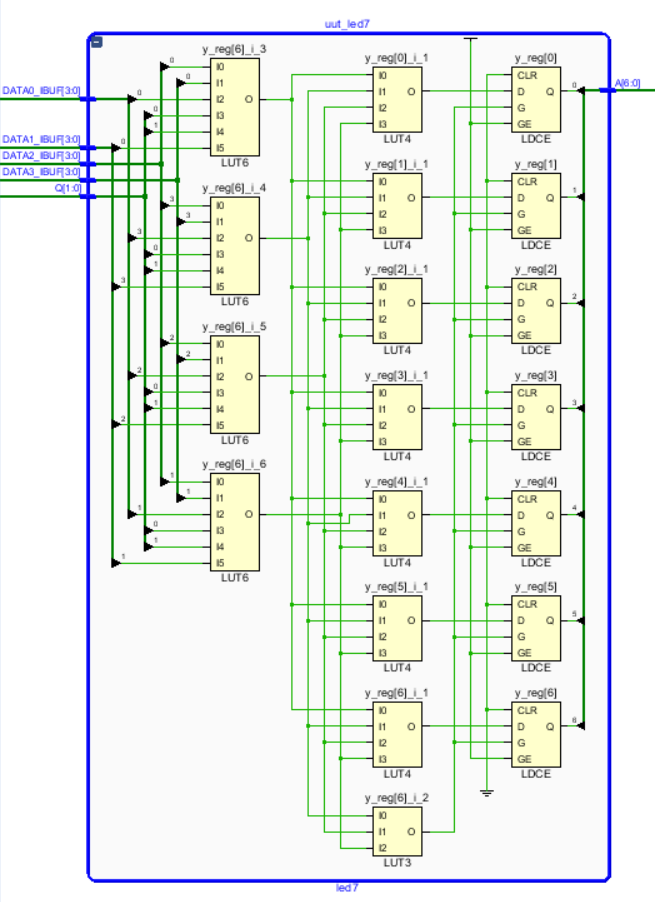
end case;

a<=y(6);b<=y(5);c<=y(4);

d<=y(3);e<=y(2);f<=y(1);g<=y(0);

end process;

end;



七段数码管译码器综合电路图

-- -----------接收四组四位的信号和时钟信号，输出位选和对应数据控制数码管动态显示---------

entity led7\_refresh is

Port (

CLK : in STD\_LOGIC;

DATA3: in STD\_LOGIC\_VECTOR(3 downto 0);

DATA2: in STD\_LOGIC\_VECTOR(3 downto 0);

DATA1: in STD\_LOGIC\_VECTOR(3 downto 0);

DATA0: in STD\_LOGIC\_VECTOR(3 downto 0);

SEL : out STD\_LOGIC\_VECTOR(3 downto 0);

A:out std\_logic;

B:out std\_logic;

C:out std\_logic;

D:out std\_logic;

E:out std\_logic;

F:out std\_logic;

G:out std\_logic);

end led7\_refresh;

architecture Behavioral of led7\_refresh is

--signal cnt:std\_logic\_vector(1 downto 0);

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

signal clock:std\_logic;

component led7

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

a:out std\_logic;

b:out std\_logic;

c:out std\_logic;

d:out std\_logic;

e:out std\_logic;

f:out std\_logic;

g:out std\_logic);

end component;

component frq\_div2 -- 调用frq\_div2的时钟

Port ( clk : in STD\_LOGIC;

clkout : out STD\_LOGIC);

end component;

begin

uut\_led7:led7 port map

( DATA => data,

a => A,b => B,c => C,d => D,e => E,f => F,g => G);

uut\_frq\_div:frq\_div2 port map

( clk => CLK,

clkout => clock);

process(clock)

variable cnt:std\_logic\_vector(1 downto 0):="00";

begin

if (clock'event and clock='1') then

cnt := cnt + 1;

end if;

case cnt is

when "00"=> data<=DATA3;

when "01"=> data<=DATA2;

when "10"=> data<=DATA1;

when "11"=> data<=DATA0;

when others=> null;

end case;

-- end process;

-- process(cnt)

-- begin

case cnt is

when "00"=> SEL<="0111";

when "01"=> SEL<="1011";

when "10"=> SEL<="1101";

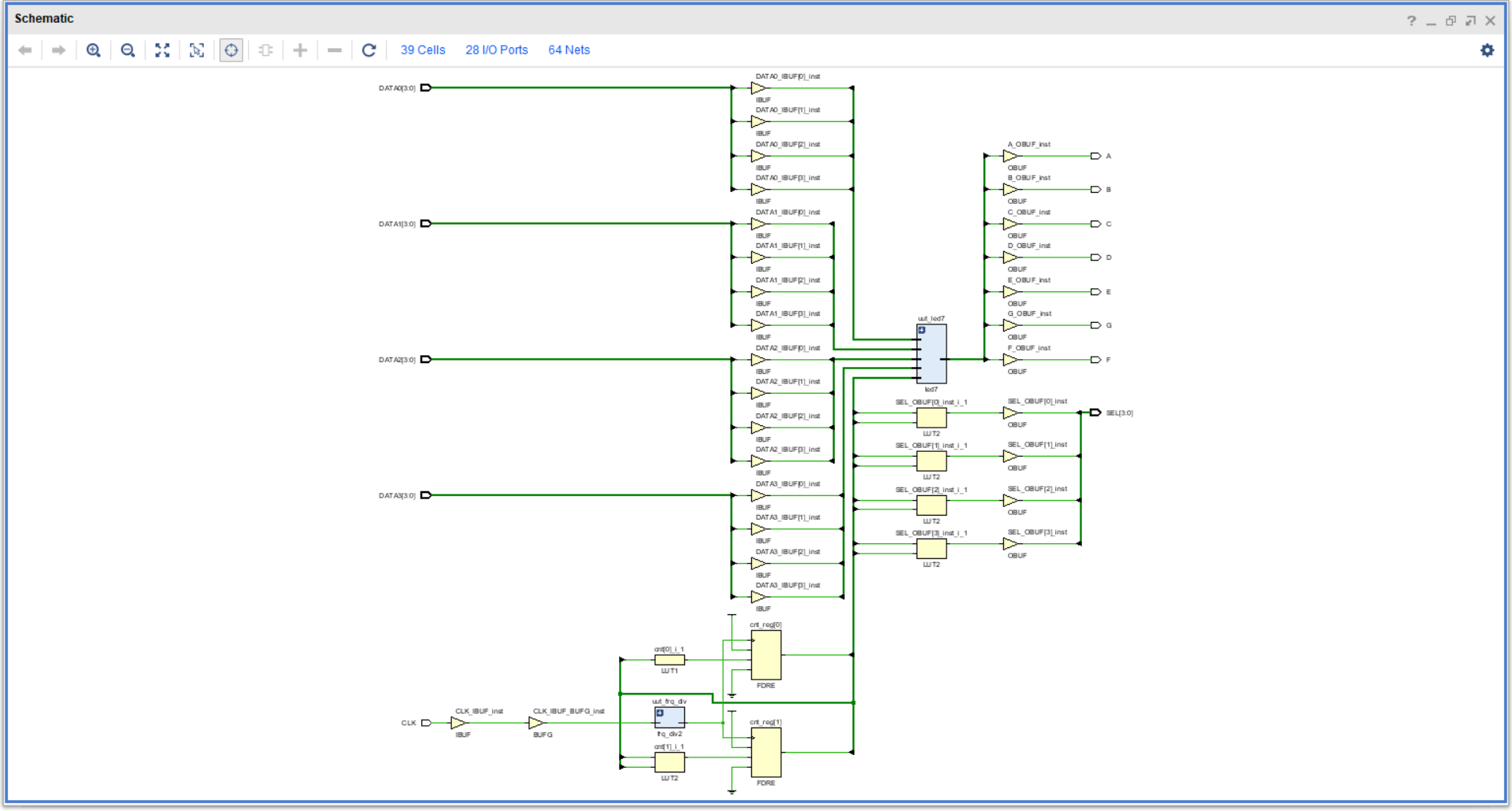
when "11"=> SEL<="1110";

when others=> null;

end case;

end process;

end Behavioral;



数码管驱动模块综合电路图

* 1. 各个模块独立仿真

**对以上各个模块的功能分别进行了时序仿真验证，结果内容较多，附于附录。**

* 1. 模块调用，组合成整体

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use ieee.std\_logic\_unsigned.all;

use ieee.std\_logic\_arith.all;

entity frq\_meter is

Port(

clk\_frq\_meter : in STD\_LOGIC;

signal\_frq\_meter : in STD\_LOGIC; -- 注释后内输出测试

sel\_frq\_meter : out STD\_LOGIC\_VECTOR (3 downto 0);

a : out STD\_LOGIC;

b : out STD\_LOGIC;

c : out STD\_LOGIC;

d : out STD\_LOGIC;

e : out STD\_LOGIC;

f : out STD\_LOGIC;

g : out STD\_LOGIC);

end frq\_meter;

ARCHITECTURE Behavioral of frq\_meter is

component frq\_div

Port( clk : in STD\_LOGIC;

clkout : out std\_logic);

end component;

--component frq\_div2

-- Port( clk : in STD\_LOGIC;

-- clkout : out std\_logic);

--end component;

component cnt410

port(

signal\_in:in std\_logic;

clk\_in: in std\_logic;

OUT3 : out std\_logic\_vector (3 downto 0);

OUT2 : out std\_logic\_vector (3 downto 0);

OUT1 : out std\_logic\_vector (3 downto 0);

OUT0 : out std\_logic\_vector (3 downto 0));

end component;

component DFF

Port ( D : in STD\_LOGIC\_VECTOR(3 downto 0);

CLK\_DFF : in STD\_LOGIC;

Q : out STD\_LOGIC\_VECTOR(3 downto 0));

end component;

component led7\_refresh

Port(

CLK : in STD\_LOGIC;

DATA3: in STD\_LOGIC\_VECTOR(3 downto 0);

DATA2: in STD\_LOGIC\_VECTOR(3 downto 0);

DATA1: in STD\_LOGIC\_VECTOR(3 downto 0);

DATA0: in STD\_LOGIC\_VECTOR(3 downto 0);

SEL : out STD\_LOGIC\_VECTOR(3 downto 0);

A:out std\_logic;

B:out std\_logic;

C:out std\_logic;

D:out std\_logic;

E:out std\_logic;

F:out std\_logic;

G:out std\_logic);

end component;

signal m:std\_logic;

--signal n:std\_logic;

signal dffin3:STD\_LOGIC\_VECTOR(3 downto 0);

signal dffin2:STD\_LOGIC\_VECTOR(3 downto 0);

signal dffin1:STD\_LOGIC\_VECTOR(3 downto 0);

signal dffin0:STD\_LOGIC\_VECTOR(3 downto 0);

signal dffout3:STD\_LOGIC\_VECTOR(3 downto 0);

signal dffout2:STD\_LOGIC\_VECTOR(3 downto 0);

signal dffout1:STD\_LOGIC\_VECTOR(3 downto 0);

signal dffout0:STD\_LOGIC\_VECTOR(3 downto 0);

BEGIN

uut\_frq\_div:frq\_div port map

( clk => clk\_frq\_meter,

clkout => m);

--uut\_frq\_div2:frq\_div2 port map

--( clk => clk\_frq\_meter,

-- clkout => n);

uut\_cnt410:cnt410 port map

( signal\_in => signal\_frq\_meter, -- 记得改回signal\_frq\_meter

clk\_in => m,

OUT3 =>dffin3,

OUT2 =>dffin2,

OUT1 =>dffin1,

OUT0 =>dffin0);

uut\_DFF3:DFF port map

( D => dffin3,

CLK\_DFF => m,

Q => dffout3);

uut\_DFF2:DFF port map

( D => dffin2,

CLK\_DFF => m,

Q => dffout2);

uut\_DFF1:DFF port map

( D => dffin1,

CLK\_DFF => m,

Q => dffout1);

uut\_DFF0:DFF port map

( D => dffin0,

CLK\_DFF => m,

Q => dffout0);

uut\_led7\_refresh:led7\_refresh port map

( CLK => clk\_frq\_meter,

DATA3 => dffout3,

DATA2 => dffout2,

DATA1 => dffout1,

DATA0 => dffout0,

SEL => sel\_frq\_meter,

A => a,

B => b,

C => c,

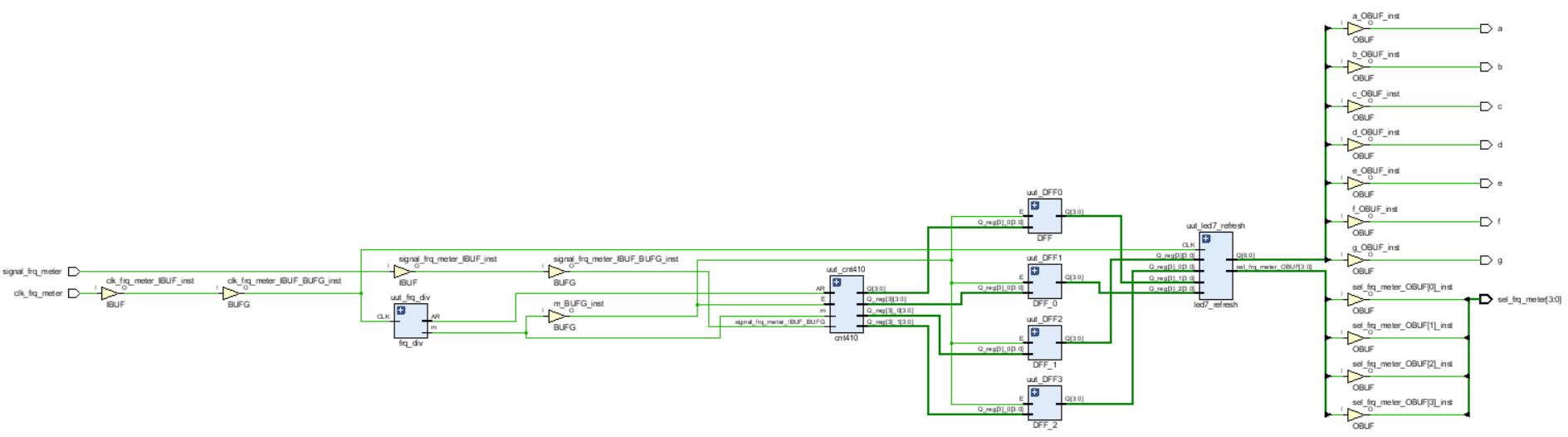
D => d,

E => e,

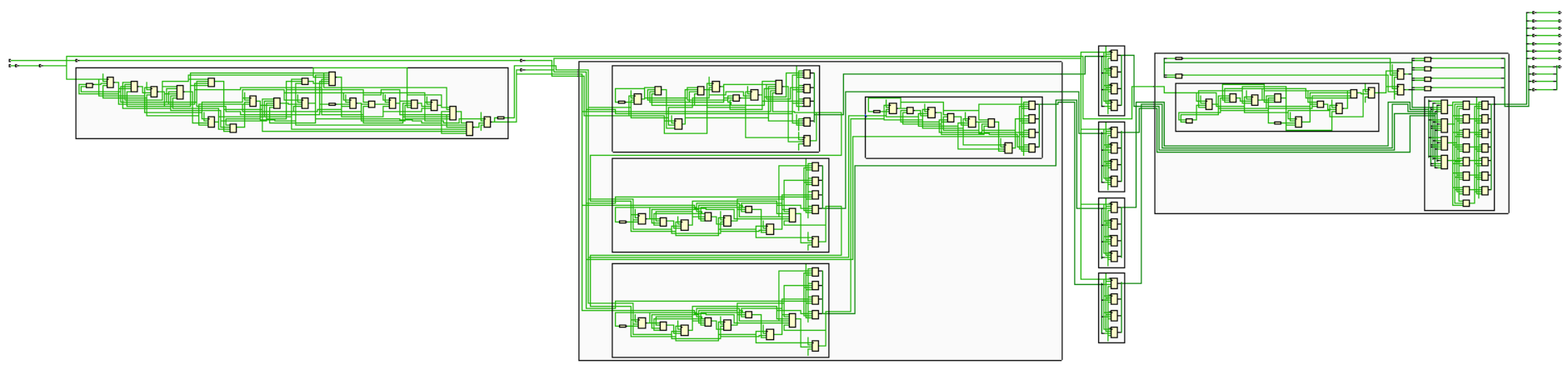
F => f,

G => g);

end Behavioral;



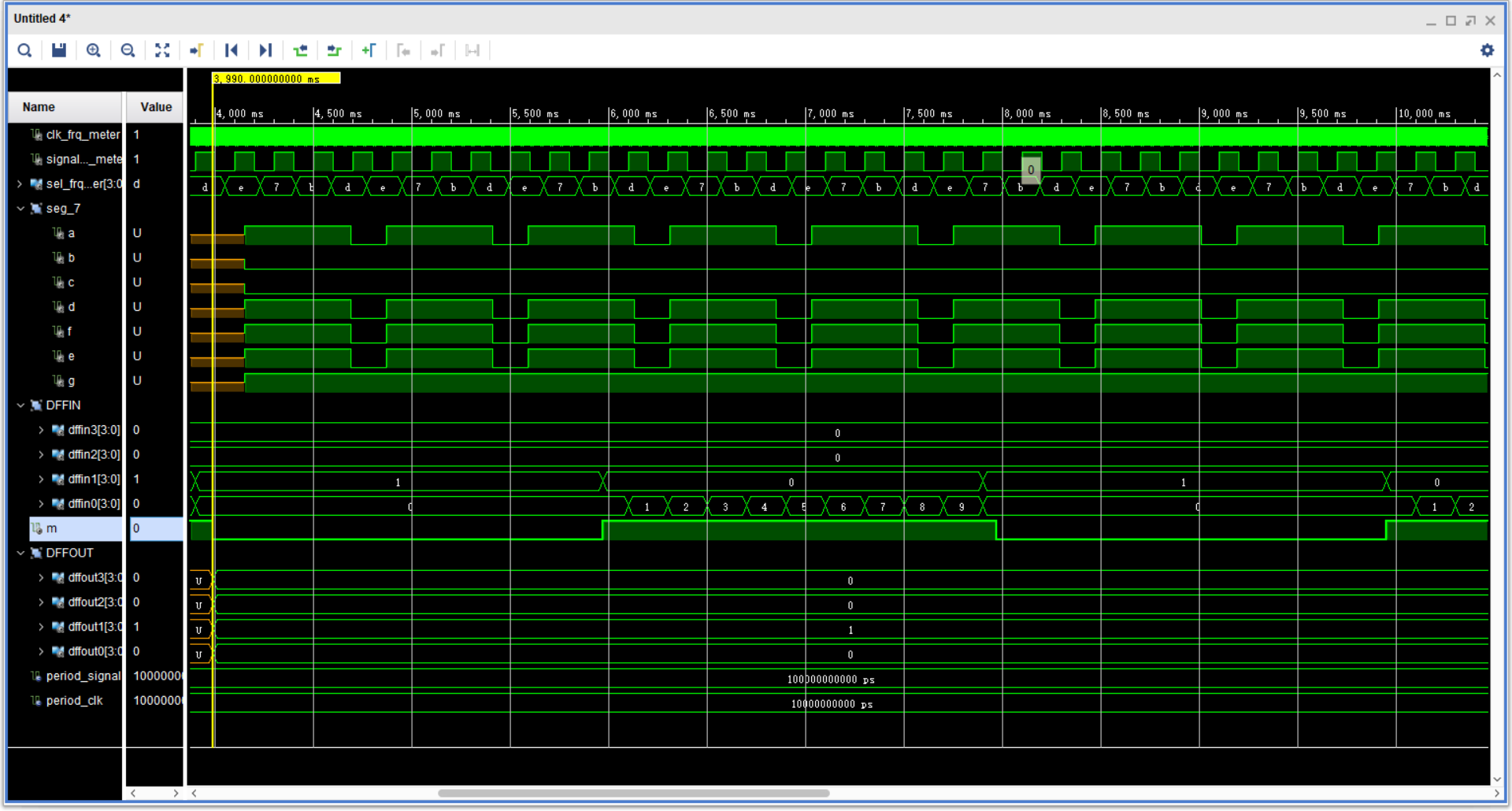
频率计系统总体综合电路图



频率计系统综合电路图展开图

## 仿真结果

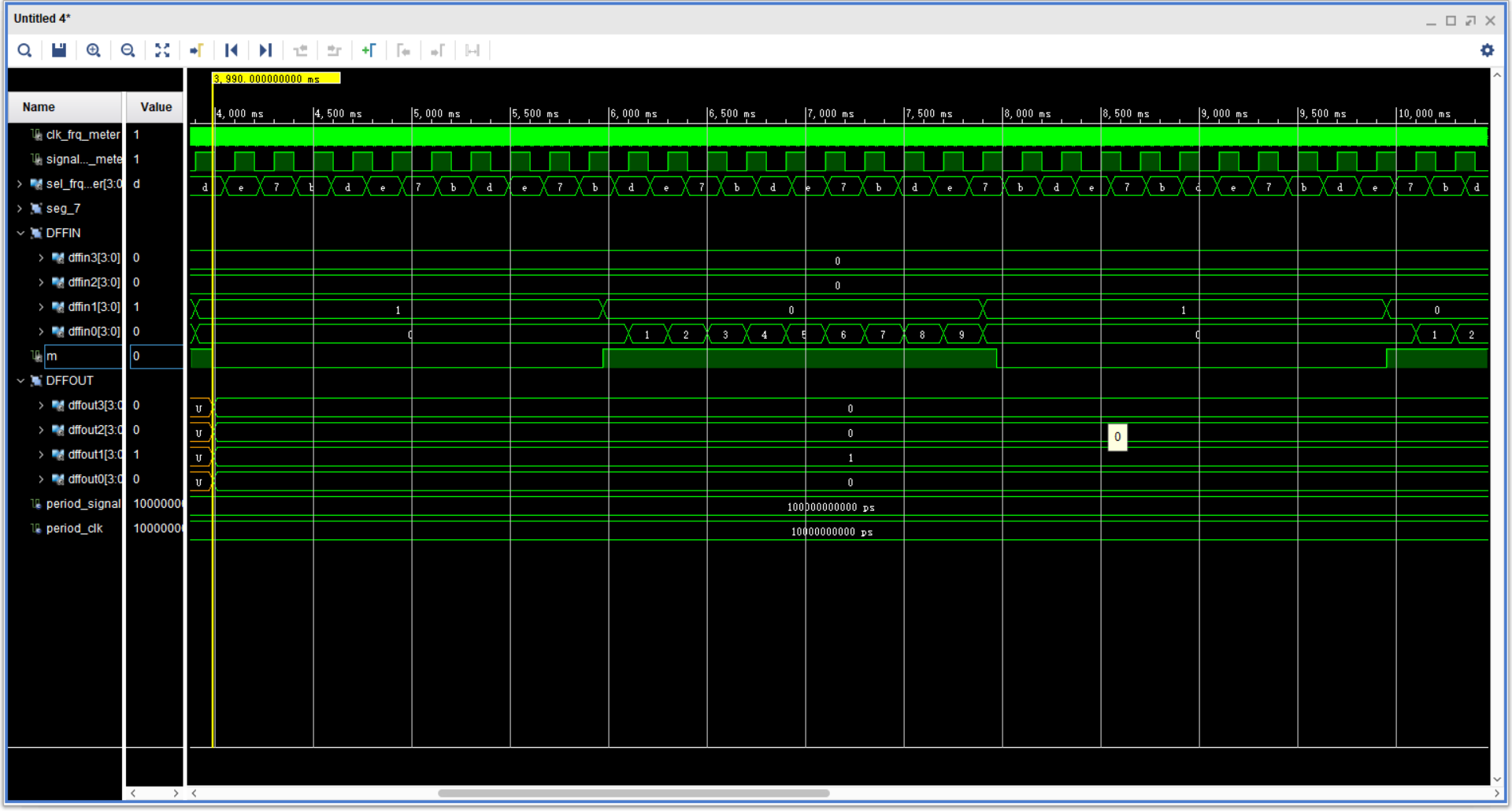
* 1. 为加快仿真进程，将测试输入时钟和测试信号的频率设定为上面算出来的值



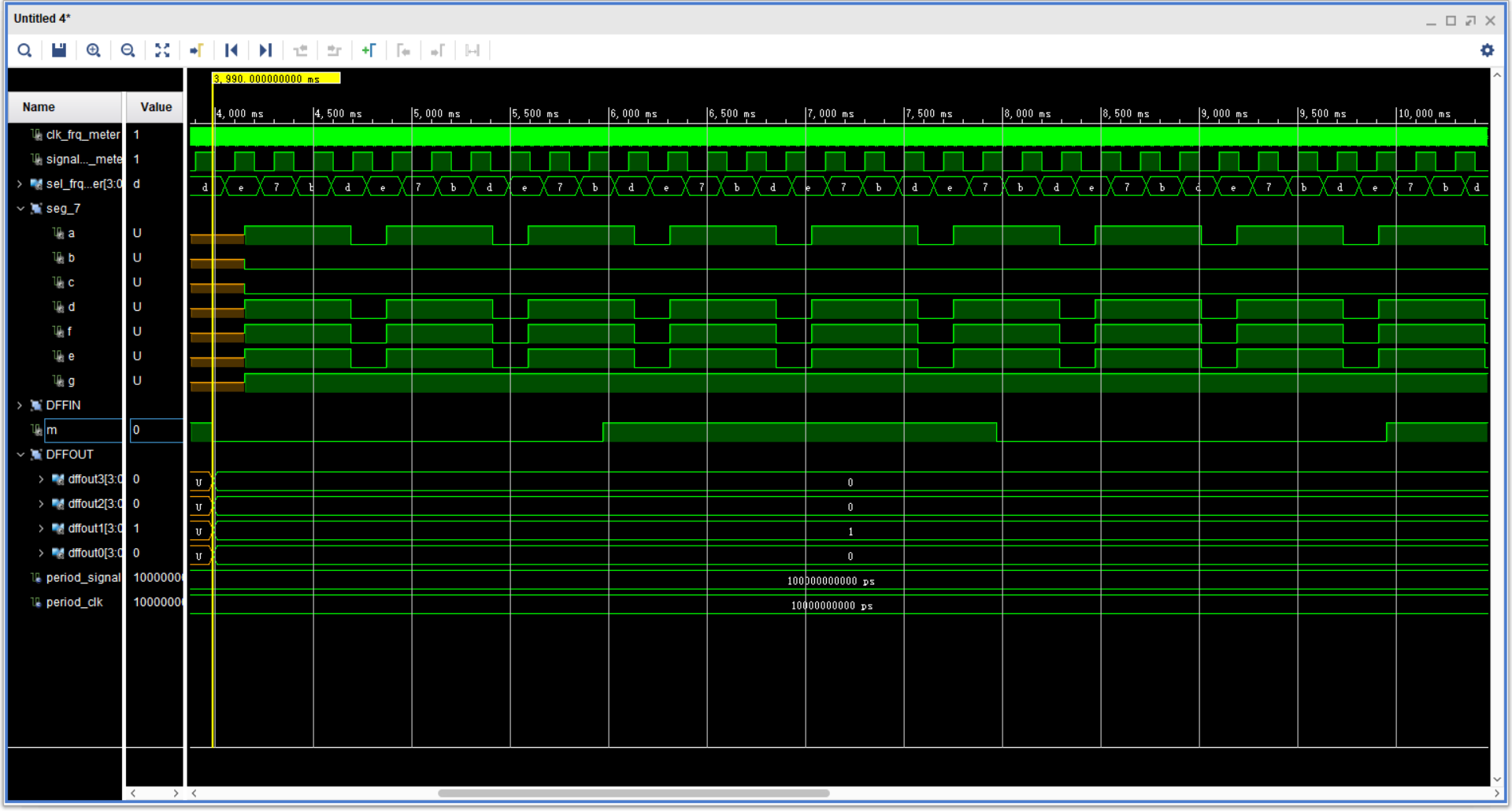
仿真的输入待测信号周期为200ms，故频率为5Hz；仿真时钟信号周期为20ms，频率为50Hz。分频器1的分频系数为200，故分频得到的驱动计数器工作的时钟信号频率为0.25Hz，周期为4s，对应于仿真图中的m信号。

计数器只在时钟的高电平期间计数工作，而m信号的一个周期内高电平的时间为2s，仿真输入信号的频率是5Hz，所以计数器的结果应该是2s\*5Hz=10，与仿真图中的dffin1，dffin0的结果‘1’‘0’对应。

所得结果在低电平期间由D触发器向数码管驱动传输。如下图，DFFOUT信号组在m信号变为低电平时更新为DFFIN最后的值，并保持。



结果由数码管驱动控制数码管刷新显示，所使用的数码管是共阳极的，故刷新时的位选信号应依次为“0111”，“1011”，“1101”和“1110”，对应的十六进制数为“7”，“b”，“d”和“e”，与信号组sel\_frq\_meter对应。



对应的译码信号由“abcdefg”输出，经查照，译码信号与刷新的位匹配。

小结：仿真的结果与预期计算结果一致，内部信号以及各个输出信号均正确。

## 实验结果

* 1. 管脚分配

set\_property PACKAGE\_PIN W4 [get\_ports {sel\_frq\_meter[3]}]

set\_property PACKAGE\_PIN V4 [get\_ports {sel\_frq\_meter[2]}]

set\_property PACKAGE\_PIN U4 [get\_ports {sel\_frq\_meter[1]}]

set\_property PACKAGE\_PIN U2 [get\_ports {sel\_frq\_meter[0]}]

set\_property PACKAGE\_PIN W7 [get\_ports a]

set\_property PACKAGE\_PIN W6 [get\_ports b]

set\_property PACKAGE\_PIN U8 [get\_ports c]

set\_property PACKAGE\_PIN V8 [get\_ports d]

set\_property PACKAGE\_PIN U5 [get\_ports e]

set\_property PACKAGE\_PIN V5 [get\_ports f]

set\_property PACKAGE\_PIN U7 [get\_ports g]

set\_property IOSTANDARD LVCMOS33 [get\_ports {sel\_frq\_meter[1]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {sel\_frq\_meter[0]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports a]

set\_property IOSTANDARD LVCMOS33 [get\_ports b]

set\_property IOSTANDARD LVCMOS33 [get\_ports c]

set\_property IOSTANDARD LVCMOS33 [get\_ports {sel\_frq\_meter[3]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports d]

set\_property IOSTANDARD LVCMOS33 [get\_ports e]

set\_property IOSTANDARD LVCMOS33 [get\_ports {sel\_frq\_meter[2]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports f]

set\_property IOSTANDARD LVCMOS33 [get\_ports g]

set\_property IOSTANDARD LVCMOS33 [get\_ports clk\_frq\_meter]

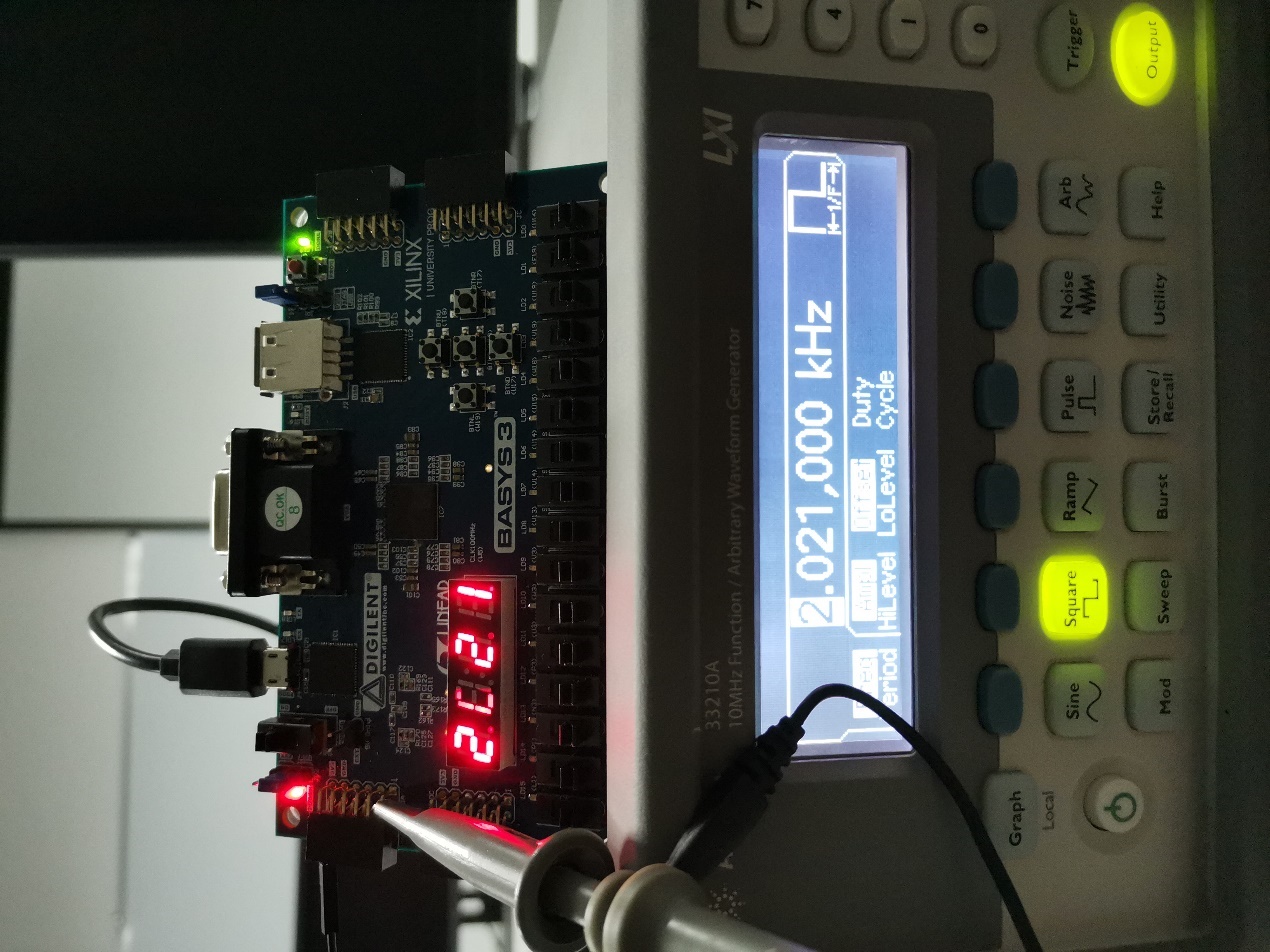
set\_property PACKAGE\_PIN W5 [get\_ports clk\_frq\_meter]

set\_property IOSTANDARD LVCMOS33 [get\_ports signal\_frq\_meter]

set\_property PACKAGE\_PIN J1 [get\_ports signal\_frq\_meter]

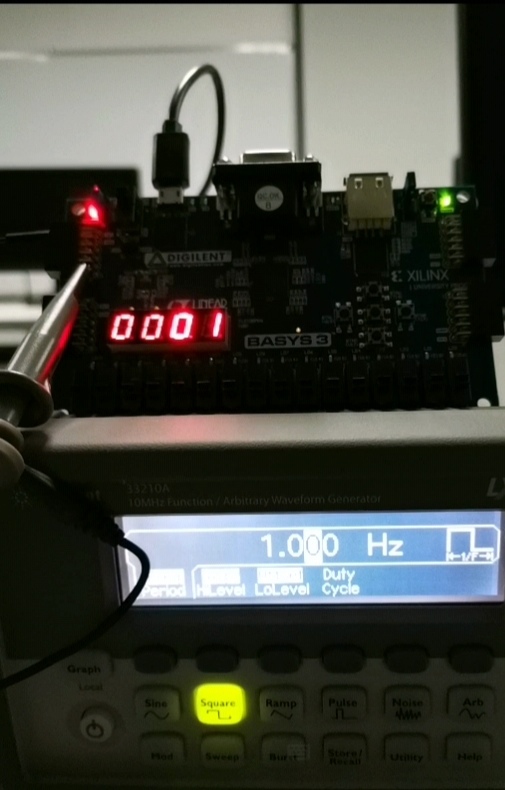
set\_property CLOCK\_DEDICATED\_ROUTE FALSE [get\_nets signal\_frq\_meter\_IBUF]

* 1. 实验结果

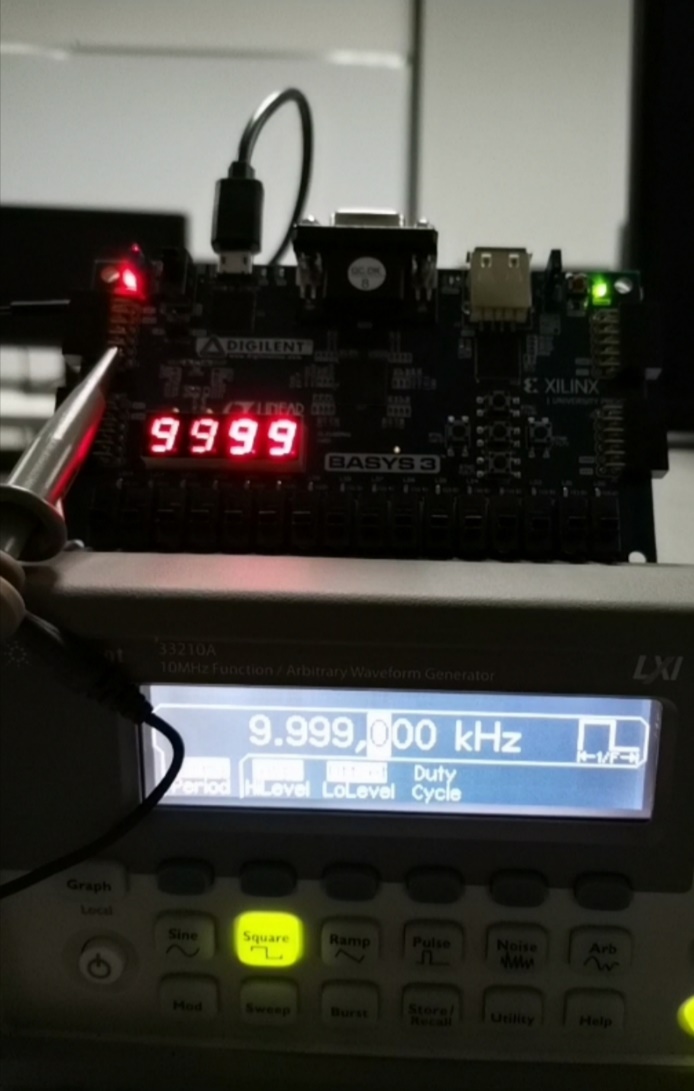


将信号源调节为Hilevel = +3.3V，LoLevel = +00mV，Freq = 2.021kHz，将探头钩到BASYS3板的J1引脚上，地线共地，按下信号源的output按钮产生信号，过大约1到2秒中，在数码管上读出频率的度数。旋转旋钮调节信号源的Freq，测试数码的各个位的显示是否正确。

可以看到数码管的显示有一位不全，是因为在拍照的时候刚好刷新到这一位。这也体现数码管的刷新率还不够高，导致显示闪烁感比较强。



测试频率计的工作的下限，将信号源调制Freq = 1.000Hz，在数码管上读书为1Hz。



测试频率计工作的上限，将信号源调制Freq = 9.999kHz，在数码管上读数为9999Hz。

## 实验总结

* 1. 遇到的几个问题和解决：1）总体仿真不成功，2）分频器仿真不成功，3）仿真时变量赋初值问题，4）数码管刷新问题

1）在进行总体仿真实验时，发现结果都处于U（不定状态）。为了弄明白哪个步骤出了问题，所以将原件在仿真文件中“展开”调试，最后发现是分频器没有正常运行。所以当整体出了问题时，可以在仿真中把整体拆分到原件层面去检查。

2）分频器在使用std\_logic类型进行输出时，如果使用的“计数翻转”形式的，那么将有可能得到U结果，原因是初状态未知。并且必须要指定端口类型位buffer。经过查阅资料，将分频器重新设计为“计数赋值”形式，问题解决。

3）仿真时的变量赋的初值在实际的综合中并不能使用。如果实际综合中要用初值可以采用generic语句进行赋值。

4）最后发现，在该实验所设计的数字系统中，数码管的刷新率对结果有影响，经过在BASYS3板上的实际实验调试，最后把控制刷新的分频器2的分频系数定在905000，即数码管刷新的频率是大约30Hz时，得到正确的结果。

* 1. 总结

本次实验利用BASYS3板和vivado软件，运用所学到的数字电路的知识，设计了一个数字频率计，测频对象是脉冲信号，范围是1~9999Hz，精度达到1Hz。

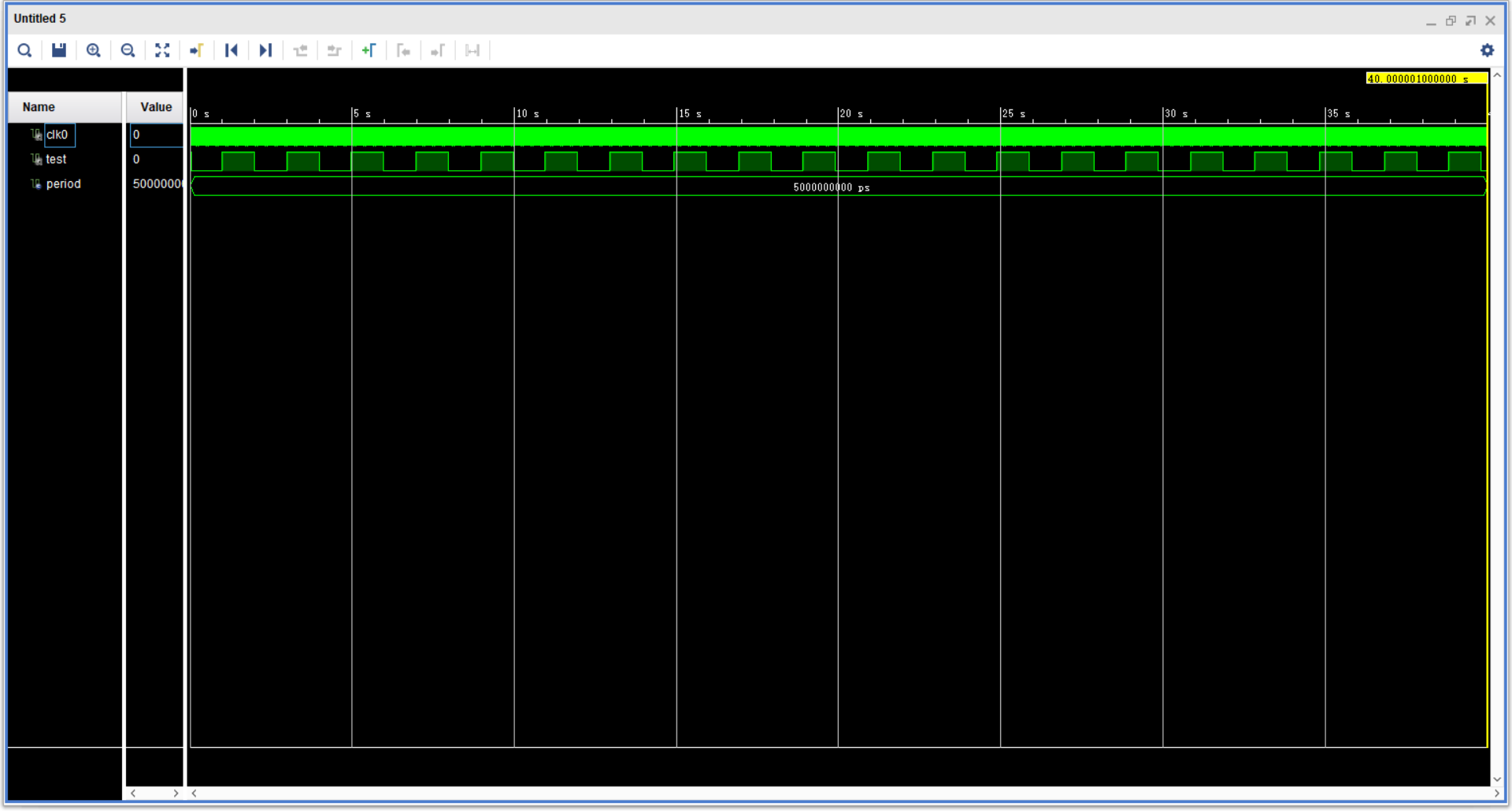
设计的思路是自顶向下，在进行设计的编程设计之前，先手工画了原件之间的连接草图，确定了各个部分的功能，以及其之间输入输出的关系。然后在分开对每个部分进行编程、仿真和调试。最后将各个部分结合起来，达到预期的功能。

在实验的过程中，遇到了许多问题，自己上网去查了资料，动手做了很多的实验，现在回过头来看收获很多。在之前学习的过程中，不明白为什么要进行仿真程序的编写和运行，后来在实验的过程中恍然大悟，因为仿真程序可以跳过综合步骤，快速地完成对设计思路的验证，而综合步骤对于一个比较大的项目来说，耗费的时间是巨大的。在最后数码管显示的关键步骤，非常感谢得到了老师的帮助，让我得以实现。总的来说，FPGA这门课远不止课堂上的内容，想要学会，学得扎实，还要自己亲自去做很多实验，过程可能是枯燥的，但是得到结果的那一瞬间的喜悦，让付出都是值得的。

本次设计的数字频率计还处于比较粗糙的阶段。在数码管刷新的部分的分频系数是单纯靠人工实验得出来的，并不知道其中的原理，缺乏理论依据。另外，数码管显示在实验中有不稳定的情况发生，刷新率较低，频率数值的更新速度依赖于计数器的工作周期的，为1秒，还比较慢，这些都是有待提高的地方。在实验中发现implementation design中还有有关于设备的结温以及功耗的考虑，这些都还是工作中未包含的方面。想要到工程的实际运用还有一段距离，FPGA课程的学习也不止于此。

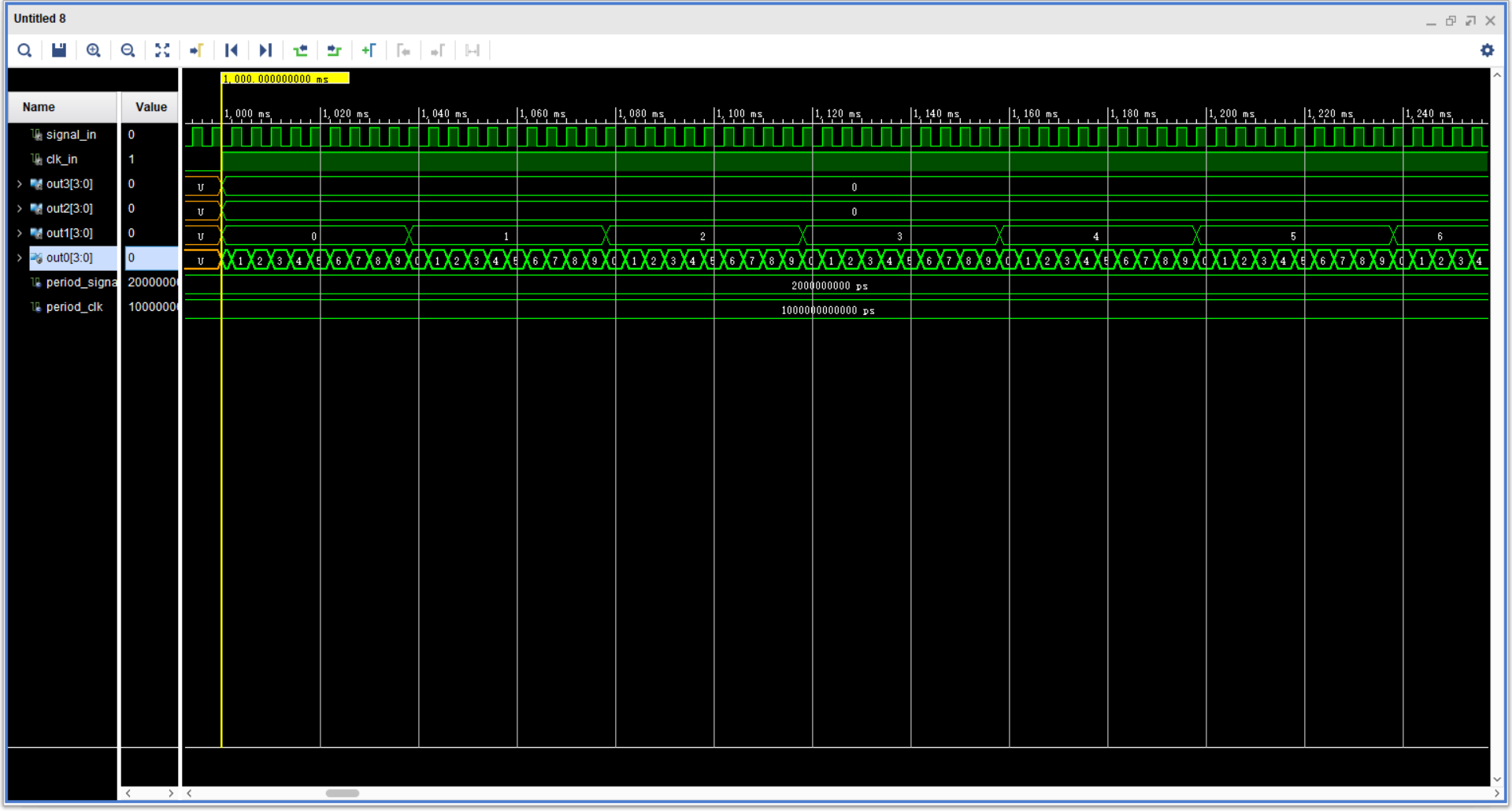
## 附录

* 1. 分频器仿真

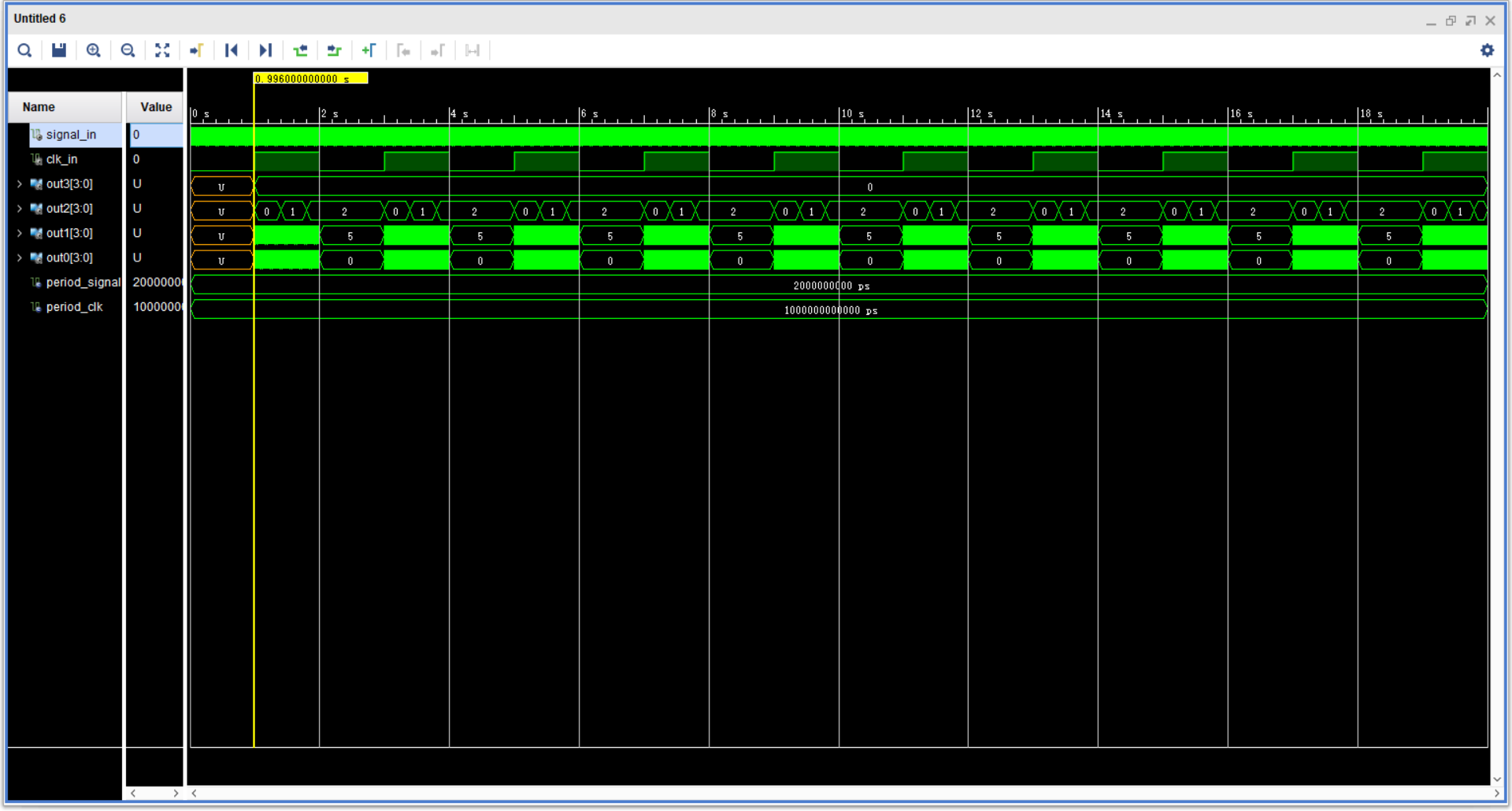


Clk0为输入的时钟信号，在分频器分频系数n = 200时，test为分频后的时钟信号。

* 1. 四位十进制计数器仿真

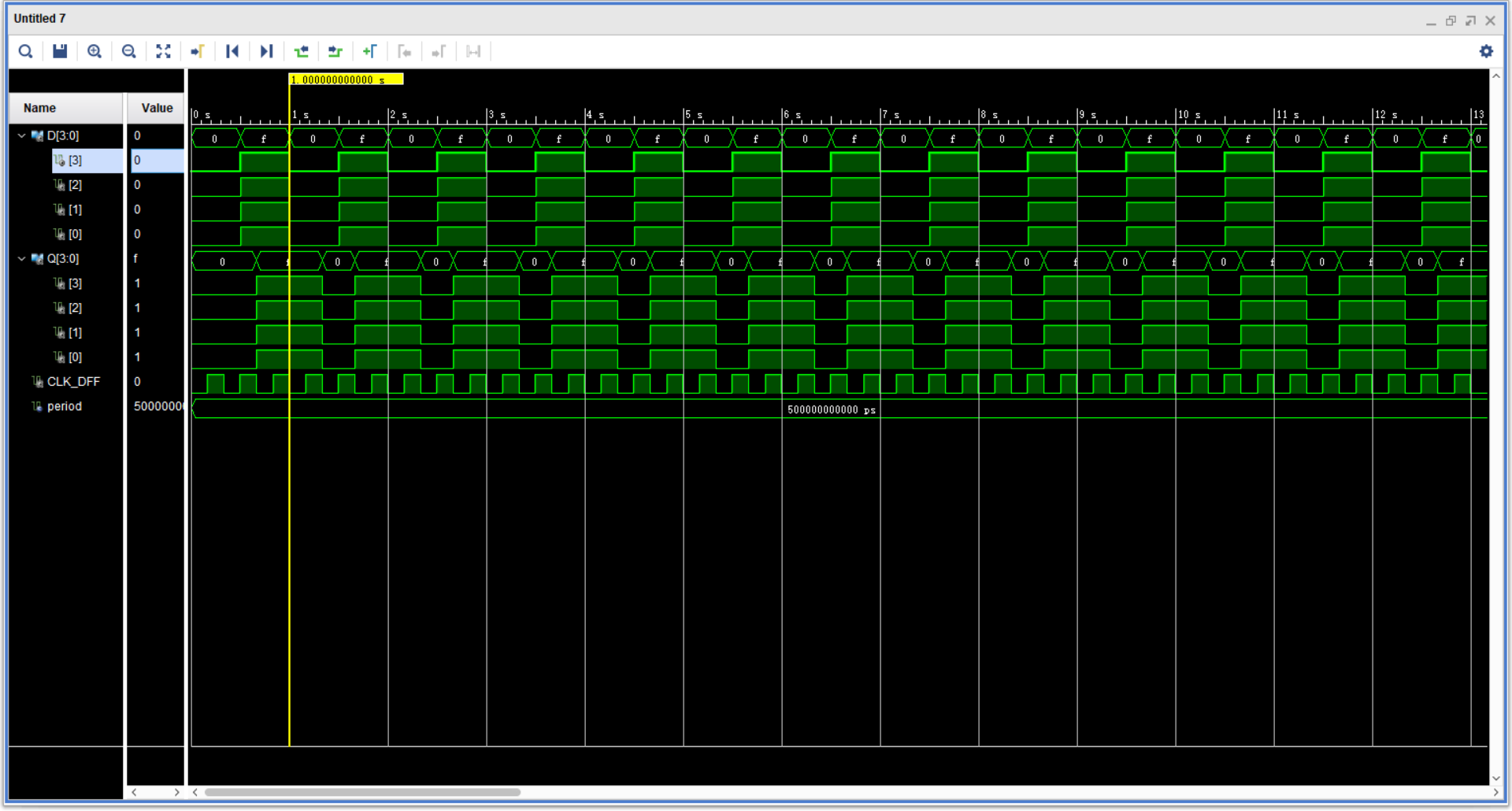


Clk\_in是输入的时钟信号，signal\_in是仿真待测信号。可以看到，当clk\_in是高电平时，计数器工作，当out0（最低位）到达“9”时，高一位计数器进“1”，然后依次类推，如下图。



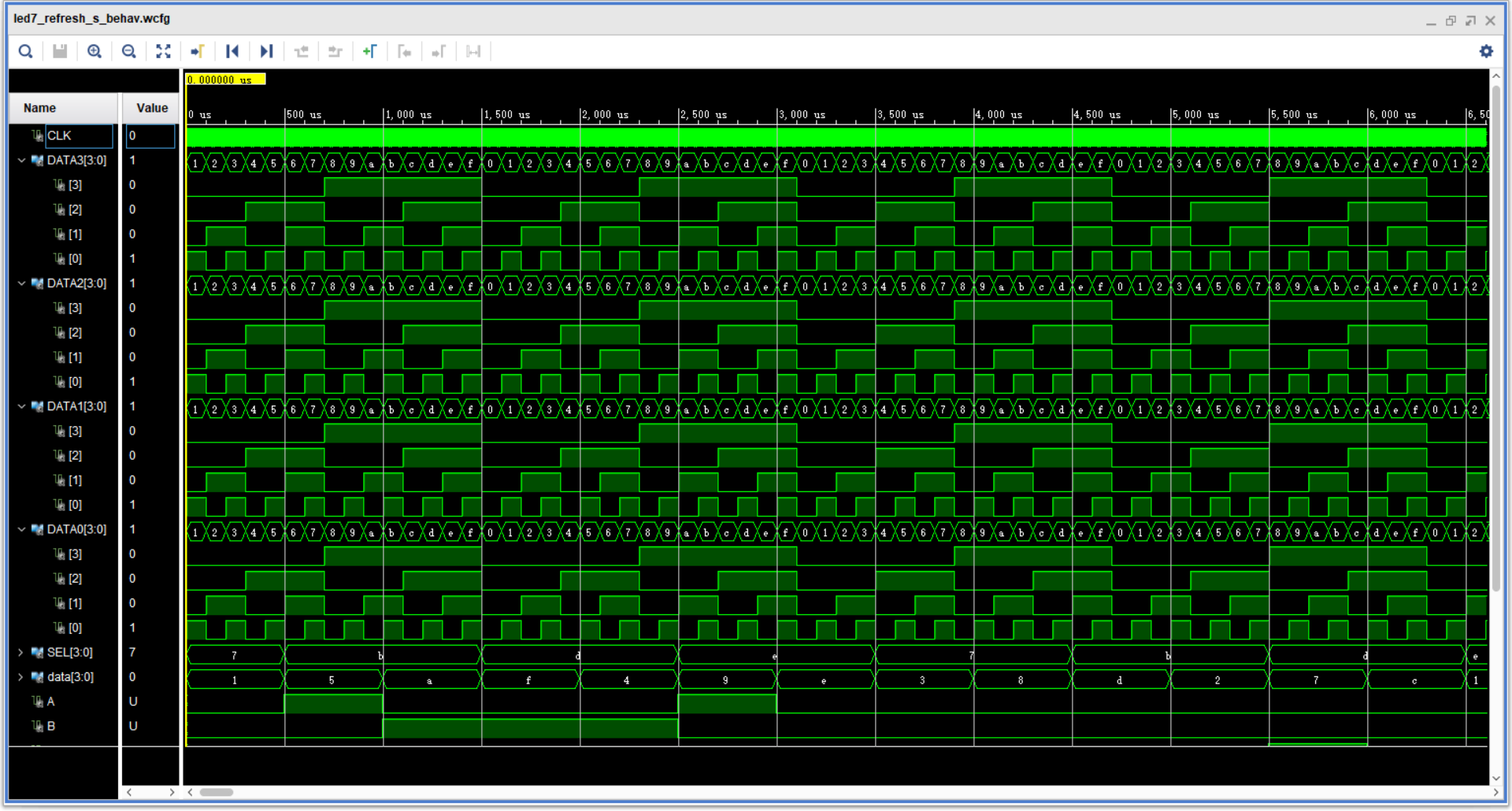
当时钟低电平时，计数器停止工作，并将结果保留至下一个时钟信号的上升沿到来时。

* 1. D触发器仿真

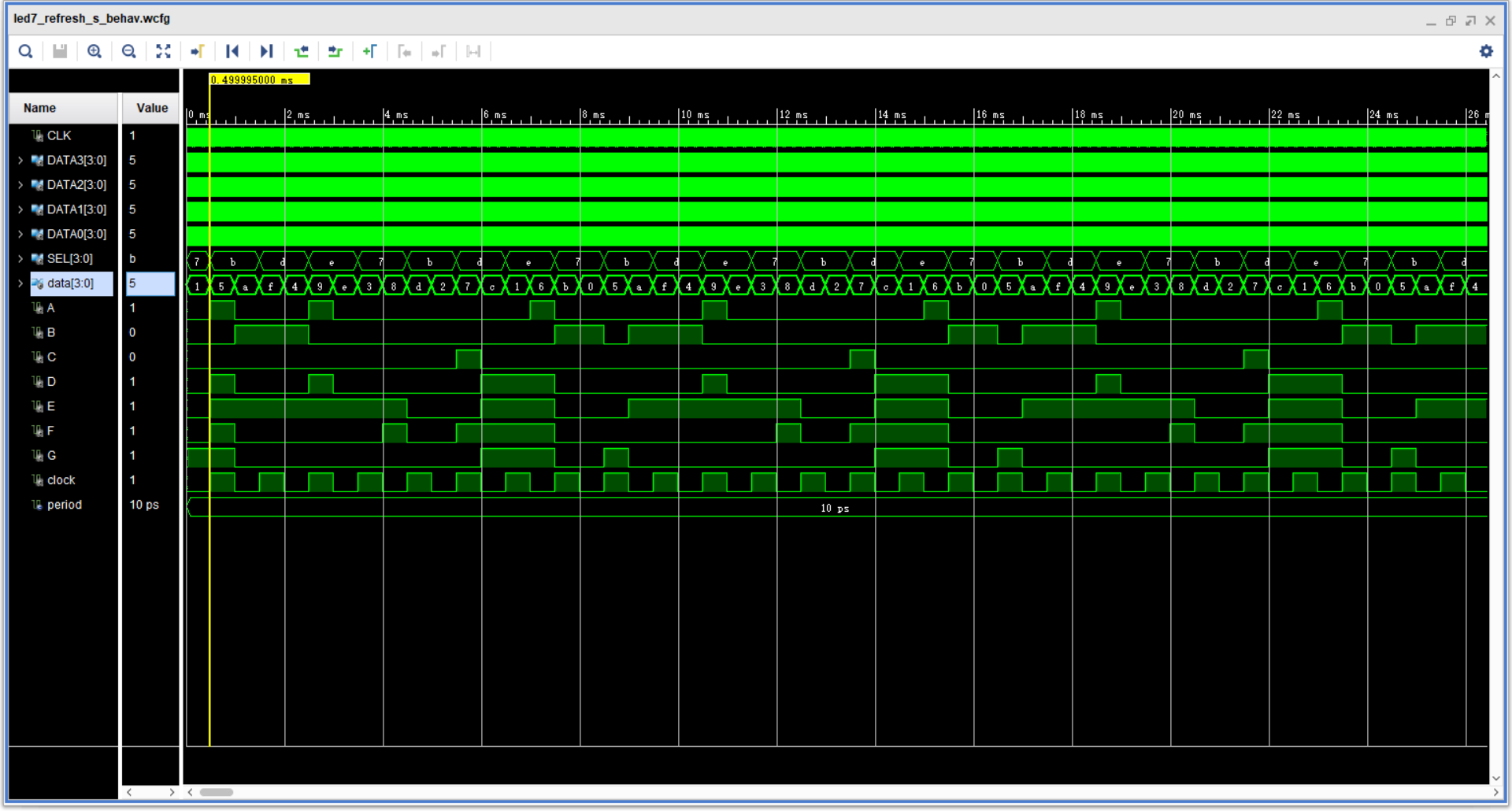


触发器一共有四位，设置每隔0.5s输入端的D就由“0000”翻转为“1111”，然后隔0.5s在翻转到“0000”。触发器时时钟下降沿触发，可以看到，当时钟信号DFF\_CLK到来时，触发器的输出Q等于时钟下降沿到来前一刻的D的值。

* 1. 数码管驱动程序仿真



设置四位数码管的每一位都从“0000”等间隔增加到“1111”，对应图中的DATA3，DATA2，DATA1和DATA0。然后由输入的时钟信号CLK驱动更新SEL（数码管位选信号）和译码信号“A”~“G”。



可以看到对应的数码管的译码信号（“1”~“9”）以及位选信号“0111”，“1011”，“1101”和“1110”对应地进行翻转。