**数字系统（课程）设计**

**实 验 报 告**

|  |  |
| --- | --- |
| **课程名称：** | **数字系统设计** |
| **学生姓名：** | **江泽群** |
| **学生学号：** | **201530371299** |
| **学生专业：** | **电子科学与技术** |
| **开课学期：** | **5** |
| **实验成绩：** |  |

# **Quartus软件使用以及VHDL程序结构**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **地 点：** | 31 号 楼 | 312 房； | **实验台号：** | 36 |
| **实验日期与时间：** | 2015年10月20日 | | **评 分：** |  |
| **预习检查纪录：** |  | | **实验教师：** |  |
| **批改意见：** |  | | | |

预习内容：

1. Quartus创建工程的基本步骤是：

1、启动 Quartus II 软件；

2、使用 New Project Wizard(File 菜单)或 quartus\_map 可执行文件建立新工程。建立新工程时，指定工程工作目录，分配工程名称，指定顶层设计实体的名称。 在用户图形界面点击 File 菜单，在 File 下拉菜单中点击 New Project Wizard，点击 New Project Wizard 图 错误!文档中没有指定样式的文字；

在指定路径之前，可以先新建一个工程文件夹，工程的顶层设计实体名应与工程名一致，然后点击 next；3、出现添加文件的对话框。当然，在这里可以选择不添加文件而直接点击 next 进入下一步，因为在 Quartus II 用户图形界面中也可以通过点击 assignment-> setting-> files 来 添加您所需要的文件；

4、出现器件系列设置对话框。对于不需要下载到开发板而只想软件仿真的设计，在这里可以不用指定特定的器件，让软件自动选择适当的器件，但对于需下载到开发板来实现的设计，就 应该在这里选择与开发板上对应的芯片。然后点击 Next 进入下一步；

5、出现EDA工具设置对话框。Quartus II 软件本身包含了一套完整的开发流程，从设计输入、分析与综合、 功能仿真、适配、时序分析、时序仿真，到下载都可以在 Quartus II 中完成， 对于没有特别的要求可以不调用其他的 EDA 软件，在这里直接选择 Next 进入下 一步；

6、出现新建工程摘要界面。至此，一个新工程已经建立好，点击 Finish 可以进入到此工程的编辑环境。

1. Quartus下仿真的基本步骤是：

1、首先建立波形文件，点击 File ->New->Other Files，在出现的对话框中双击 Vector Waveform File 使进入了波形文件编辑环境。

2、右键点击左边空白处，选择 Insert Node or Bus„，添加用以仿真的输入输出信号。

3、在出现的 Insert Node or Bus 对话框中点击 Node Finder 后，即会出现一 个为仿真添加输入输出信号的界面，在界面的 Filter 中选择 Post-Complication 在点击 List，那么工程编译后所有内外部的信号都会被列出来，这里所指的外 部信号即系统的所有输入输出信号，而内部信号则是系统内的一些寄存器的输入 输出信号，不同进程间用于通信的信号等，一般只选择输入输出信号即可，如果 需要更详细的分析系统，则通常也将与寄存器相关的一些信号添加到仿真波形文件中。选择需添加到波形文件的信号后点击ok；

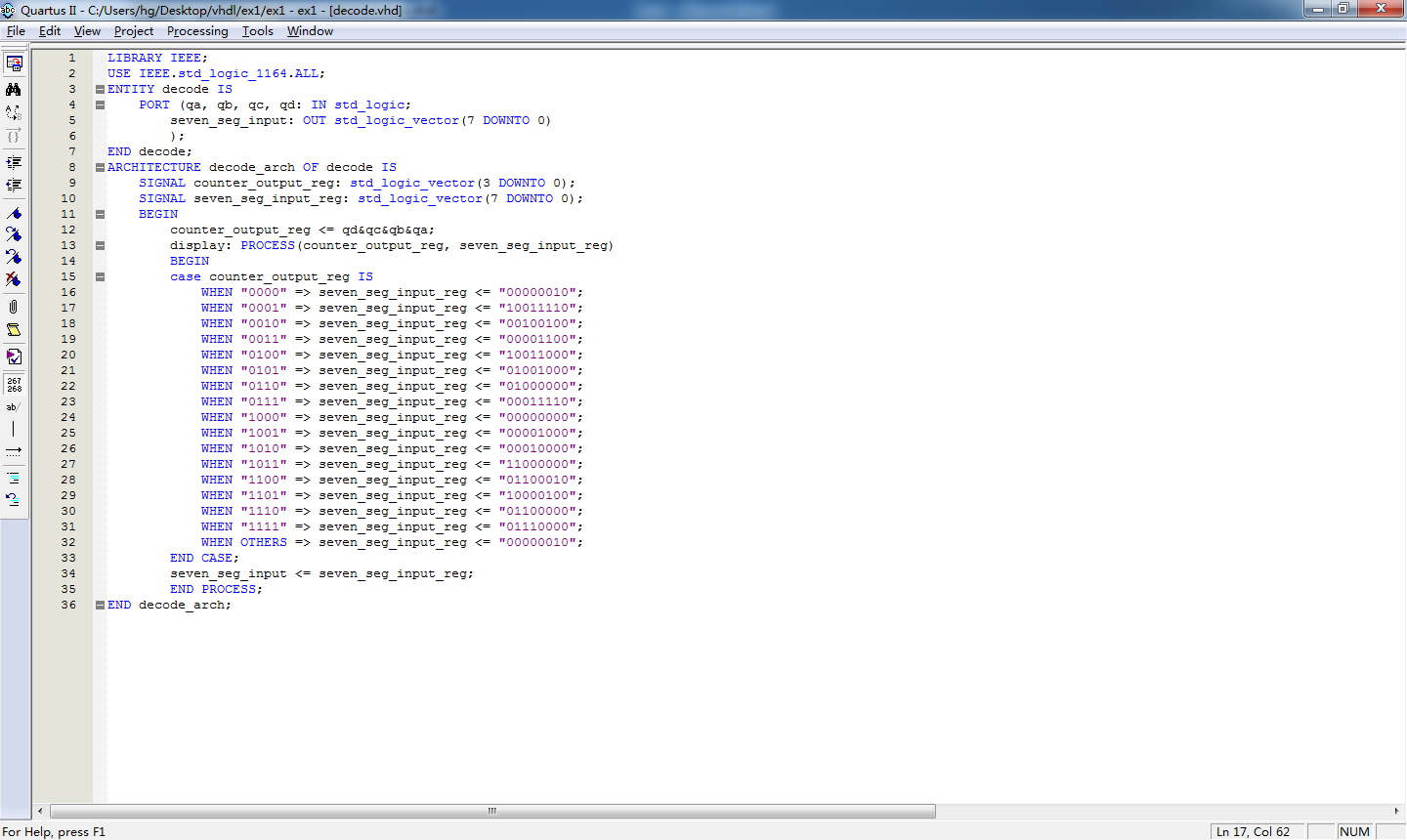
4、为波形文件中每个输入信号编辑激励信号；

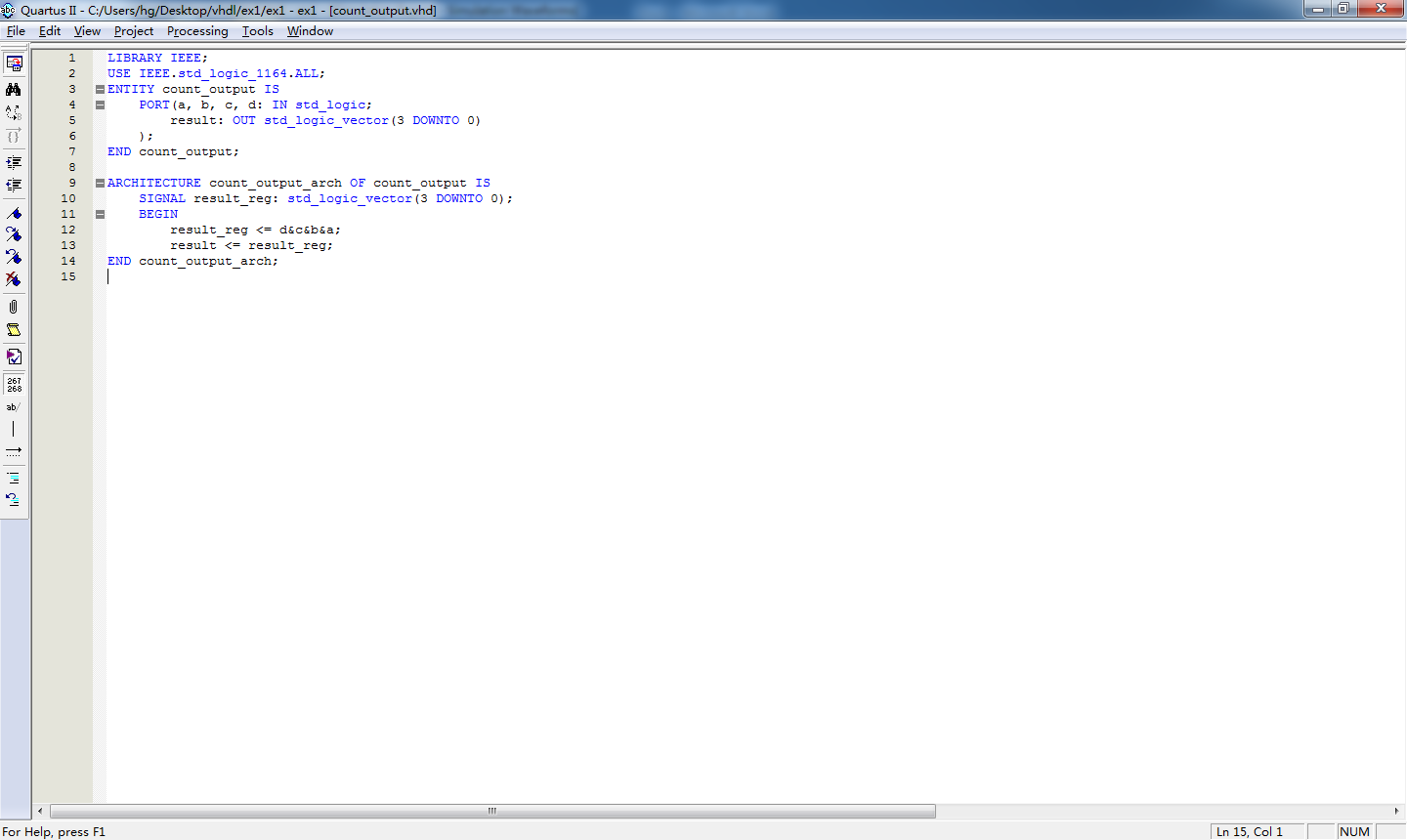
5、编辑好输入波形文件，保存后便可用于仿真。

1. VHDL从结构上分为两个部分是：实体和结构体。

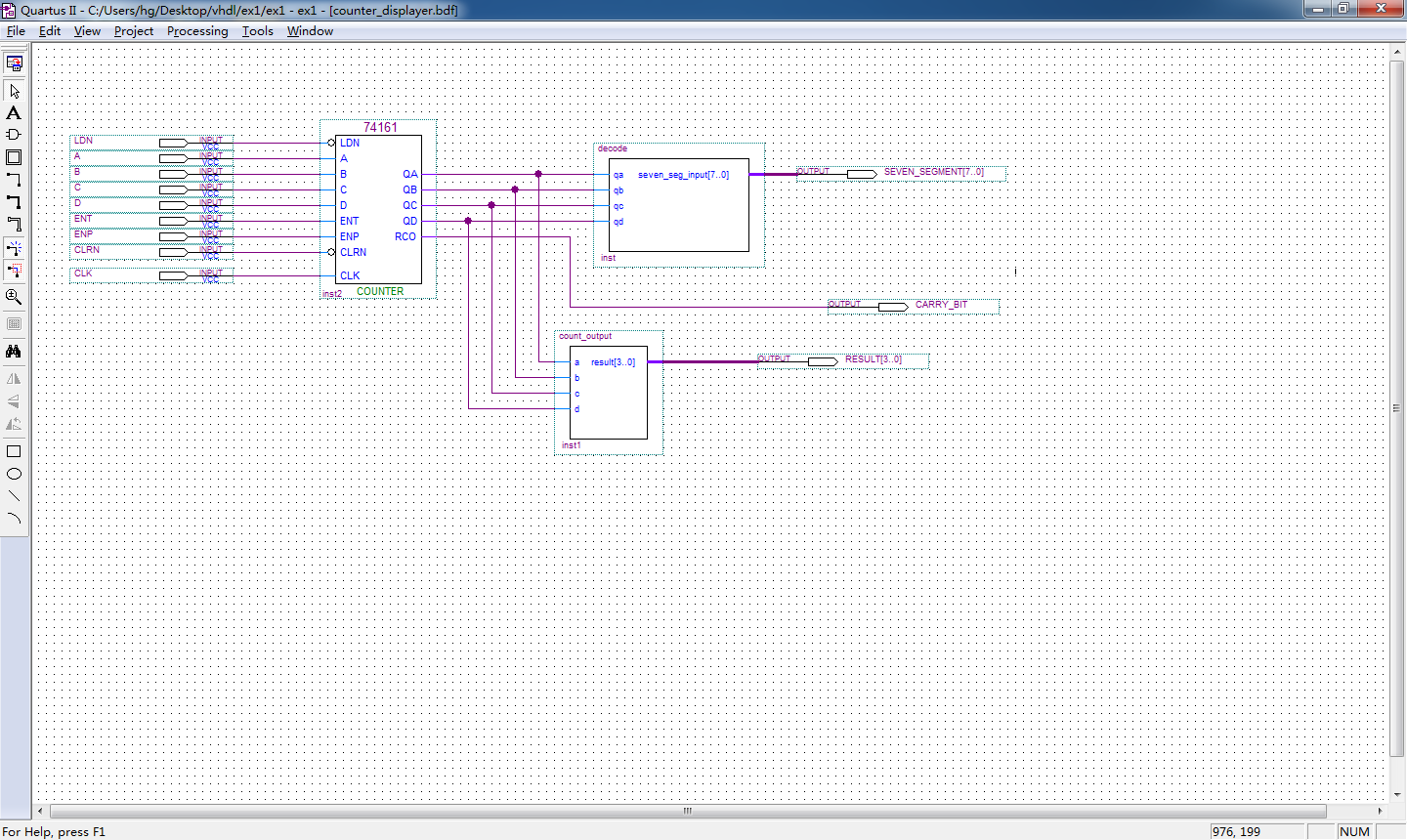
各自的作用是：实体是VHDL程序的基本单元，用于说明设计系统的外部接口信息，相当于提供了一个设计单元的公共信息。对于一个已经确定的系统，实体的描述是唯一的；结构体用于描述相应实体的行为、功能或电路结构。

1. 实验代码

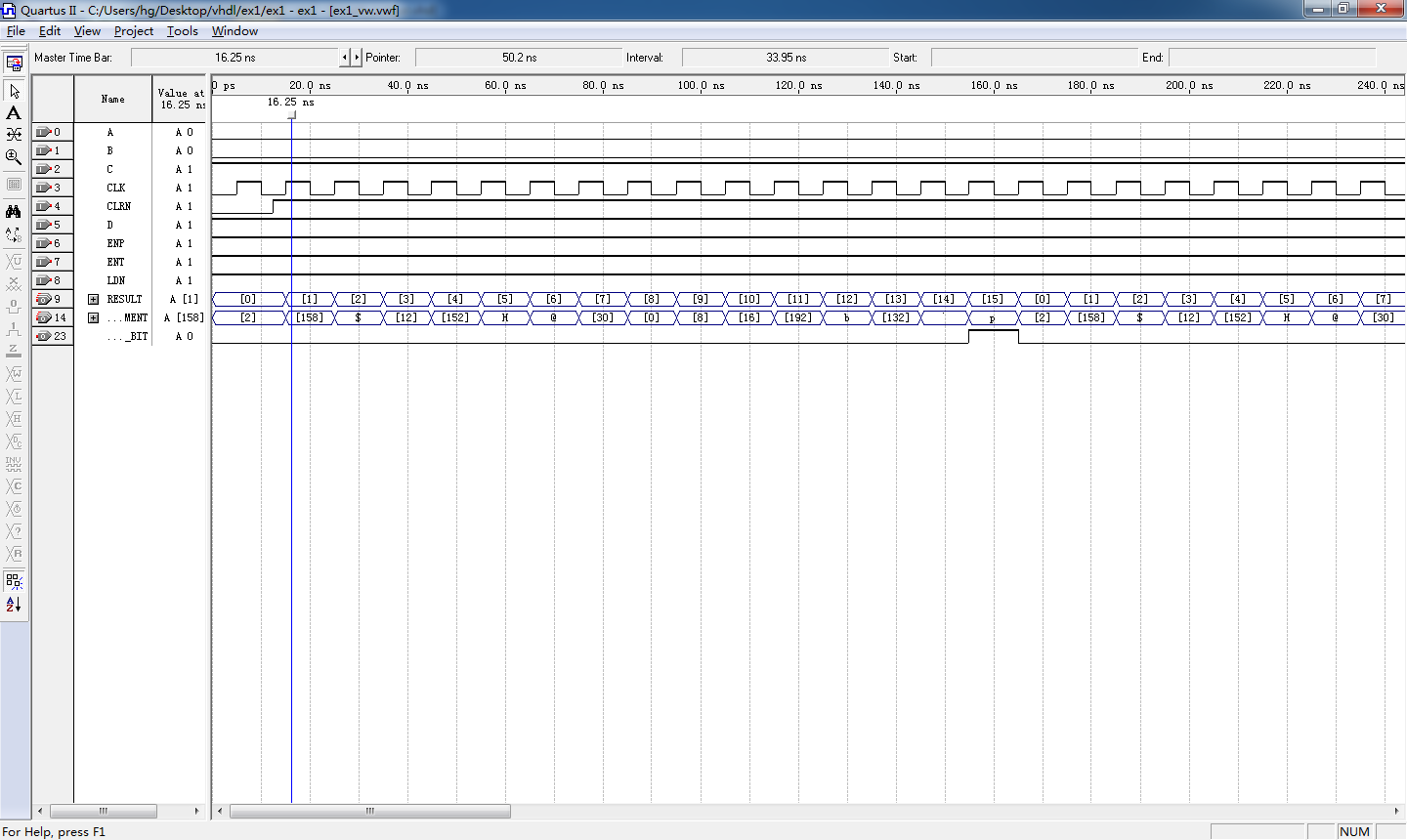




1. 实验电路图



1. 实验仿真结果



# 实验二 基本电路模块设计（组合、时序）

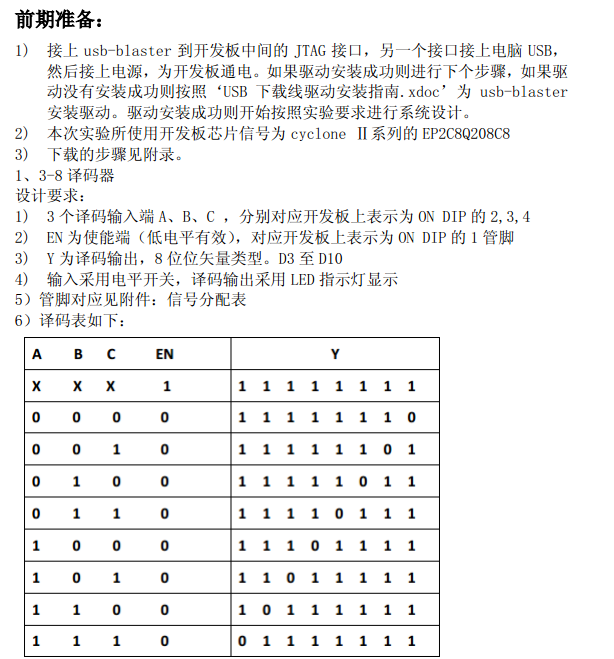
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **地 点：** | 31号 楼 | 312 房； | **实验台号：** | 42 |
| **实验日期与时间：** | 2017年11月3日 | | **评 分：** |  |
| **预习检查纪录：** |  | | **批改教师：** |  |

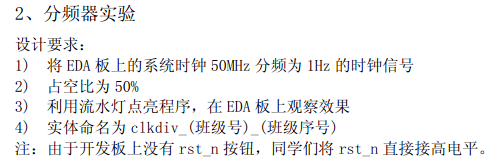
报告内容：

一，实验要求：

用 VHDL 语言设计组合逻辑、时序逻辑电路模块。

1. 实验内容：





1. 实验设计原理；

分频器的设计原理是将50M 分频的任务分成一次50分频和两次1000分频，通过级联的方式实现50M 分频。最后通过流水灯输出最终信号。

3-8译码器是负责将三位编码输出为独热编码，实验中输入信号a, b, c分别设定为20ns, 40ns, 80ns，占空比均为50%的周期方波信号， 就能看到输入信号在所有编码情况下对应的输出信号。

1. 实验过程记录（流程图或者实验逻辑思路过程）

1k 分频器流程图如图4-1所示（50分频与之类似）：

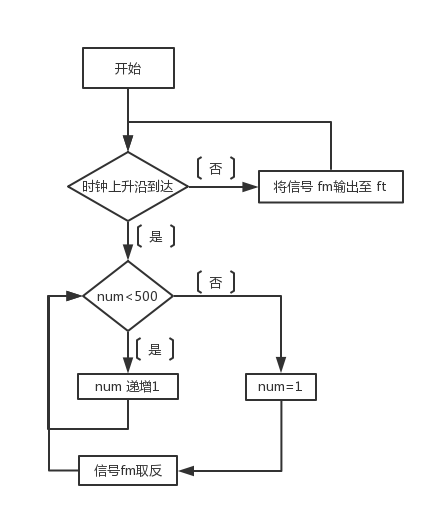


图4-1

流水灯流程图如图4-2所示：

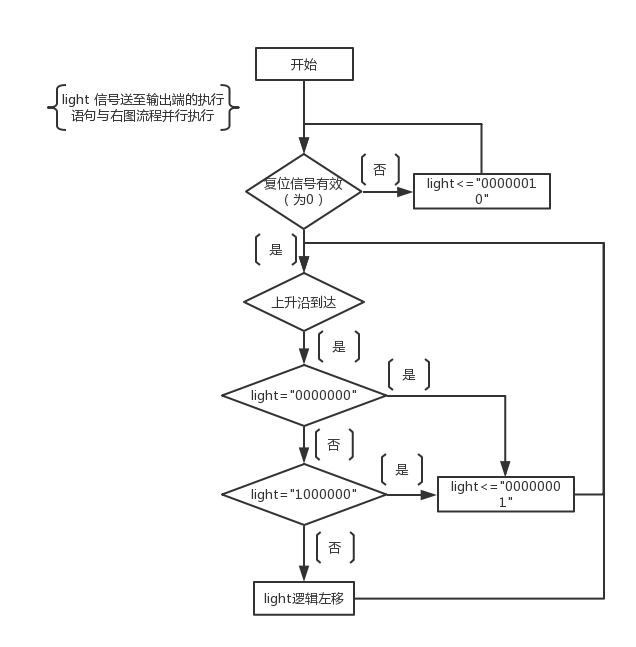


图4-2

3-8译码器流程图如图4-3所示：

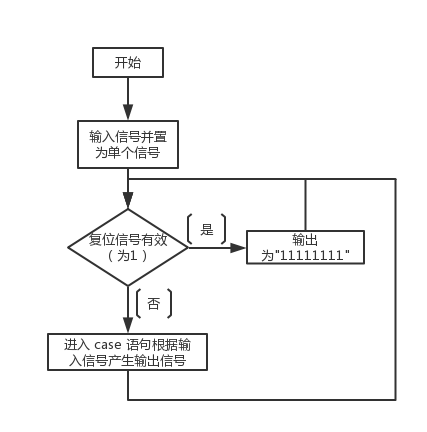


图4-3

1. 实验结果和分析

分频器图形输入文件如图5-1所示：

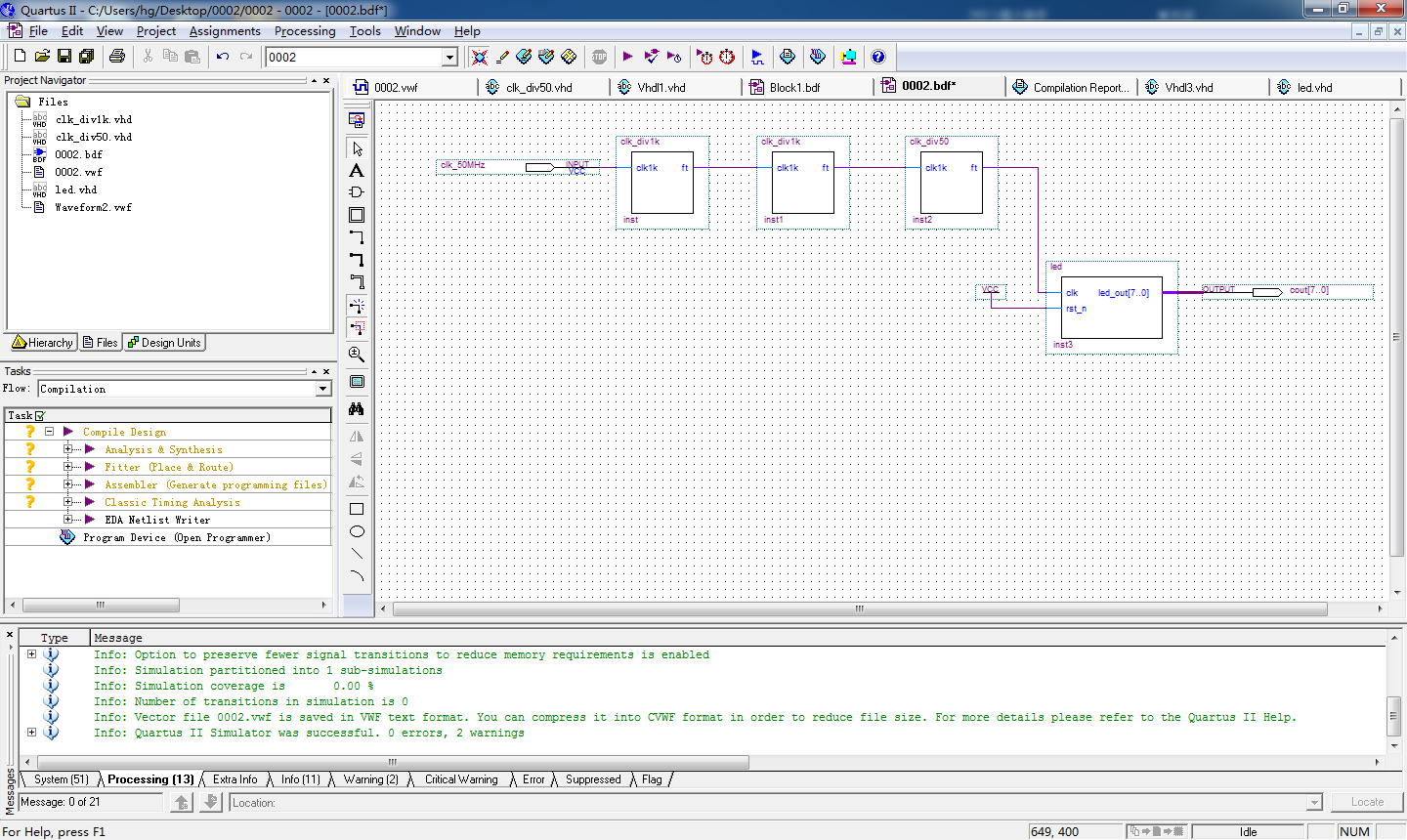


图5-1

分频器仿真波形图如图5-2所示：

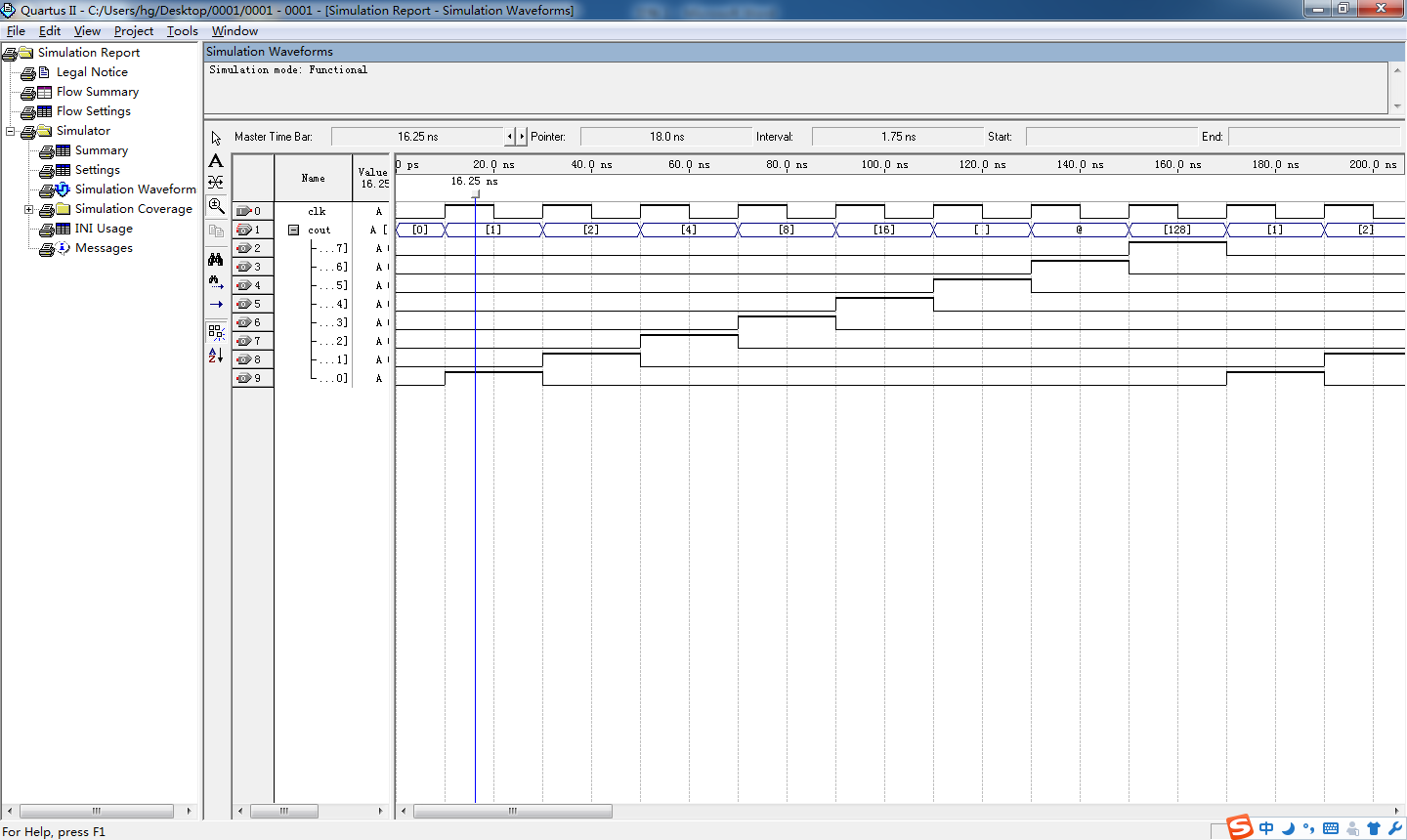


图5-2

3-8译码器仿真波形图如图5-3所示：

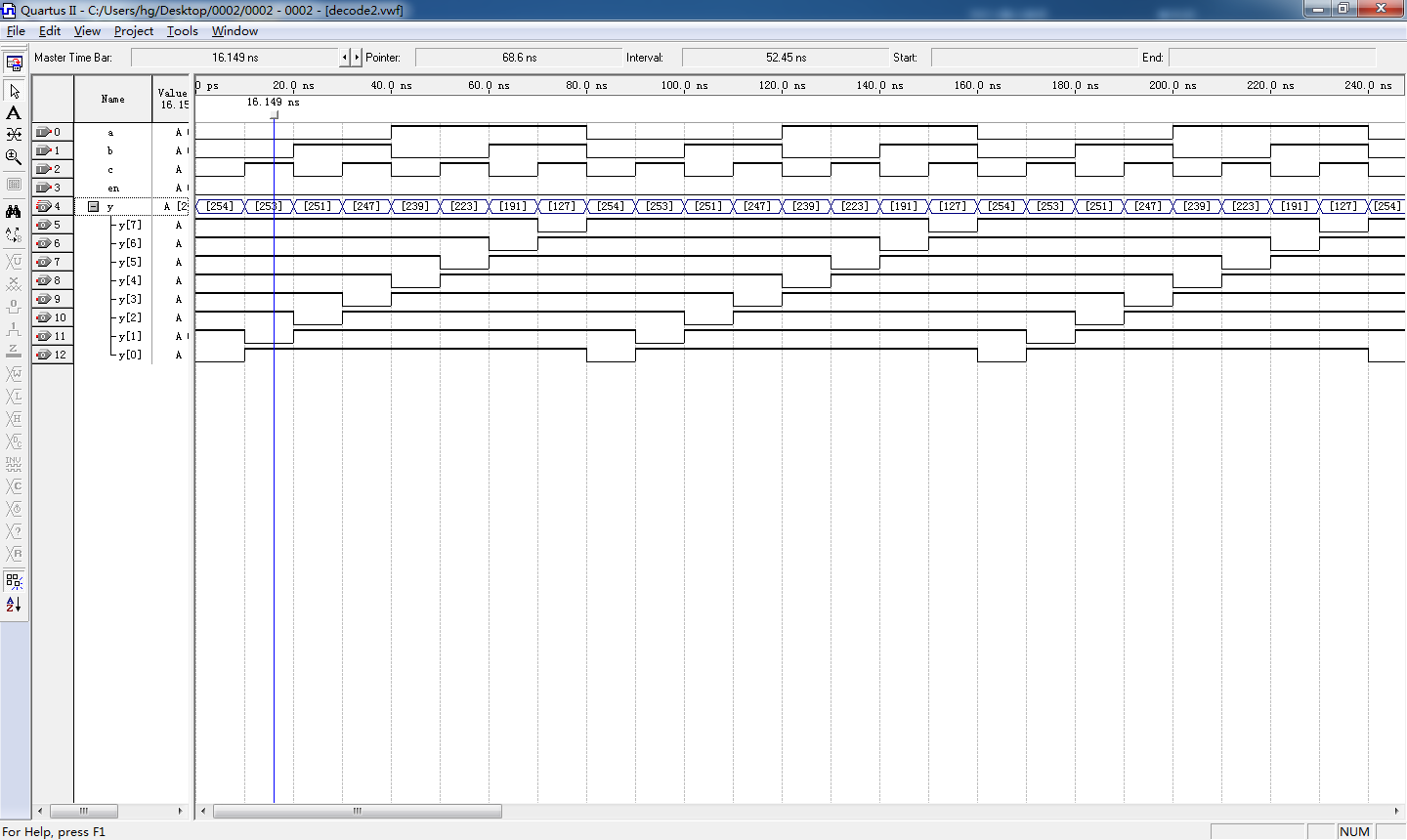


图5-3

实验结果符合要求。

六，附录（打印源代码页）

1kHz分频器实验代码如下：

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_arith.all;

use ieee.std\_logic\_unsigned.all;

entity clk\_div1k is

port (clk1k: in std\_logic;

ft: out std\_logic);

end clk\_div1k;

architecture a of clk\_div1k is

signal fm: std\_logic;

begin

process(clk1k)

variable num: interger range 0 to 1000;

begin

if clk1k’event and clk1k = ‘1’ then

if num < 500 then

num : = num + 1;

else

num := 1;

fm <= not fm;

end if;

end if;

ft <= fm;

end process;

end a;

输出流水灯实验代码如下：

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_arith.all;

use ieee.std\_logic\_unsigned.all;

entity led is

port (clk: in std\_logic)

entity led is

port (clk: in std\_logic;

rst\_n: in std\_logic;

led\_out: out std\_logic\_vector(7 downto 0));

end led;

architecture behav of led is

signal light: std\_logic\_vector(7 downto 0);

begin

process (clk, rst\_n)

begin

if (rst\_n = ‘0’) then

light <= “00000010”;

elsif (clk’event and LIBRARY IEEE;

USE IEEE.std\_logic\_1164.ALL;

USE IEEE.std\_logic\_arith.ALL;

USE IEEE.std\_logic\_unsigned.ALL;

ENTITY led IS

PORT (clk:IN std\_logic;

rst\_n:IN std\_logic;

led\_out:OUT std\_logic\_vector(7 DOWNTO 0));

END led;

ARCHITECTURE behav OF led IS

SIGNAL light:std\_logic\_vector(7 DOWNTO 0);

BEGIN

PROCESS (clk, rst\_n)

BEGIN

if (rst\_n='0')THEN

light<="00000010";

ELSIF(clk'event and clk='1')THEN

IF (light="00000000")THEN

light<="00000001";

ELSE

IF(light ="10000000")THEN

light<="00000001";

ELSE

light<=light(6 DOWNTO 0)&'0';

END IF;

END IF;

END IF;

END PROCESS;

led\_out<=light;

END behav;clk = ‘1’) then

if (light = “0000000”) then

light <= “00000001”;

else

if (light = “10000000”) then

light <= “00000001”;

light <= light(6 downto 0) & ‘0’;

end if;

end if;

end if;

end process;

led\_out <= light;

end behav;

3-8译码器代码如下：

LIBRARY IEEE;

USE IEEE.std\_logic\_1164.ALL;

ENTITY decode IS

PORT (a, b, c, en : IN std\_logic;

y : OUT std\_logic\_vector(7 DOWNTO 0));

END decode;

ARCHITECTURE behav OF decode IS

SIGNAL input : std\_logic\_vector(2 DOWNTO 0);

BEGIN

input <= a & b & c;

PROCESS(input, en)

BEGIN

IF (en = '0') THEN

CASE input IS

WHEN "000" => y <= "11111110";

WHEN "001" => y <= "11111101";

WHEN "010" => y <= "11111011";

WHEN "011" => y <= "11110111";

WHEN "100" => y <= "11101111";

WHEN "101" => y <= "11011111";

WHEN "110" => y <= "10111111";

WHEN "111" => y <= "01111111";

WHEN OTHERS => y <= "XXXXXXXX";

END CASE;

ELSE

y <= "11111111";

END IF;

END PROCESS;

END behav;

**（实验报告作品相片粘贴页）**

（工程软件比如电路图截图、仿真结果或者实验平台运行效果照片）

分频器程序下载至开发板实物图如图7-1和7-2所示：

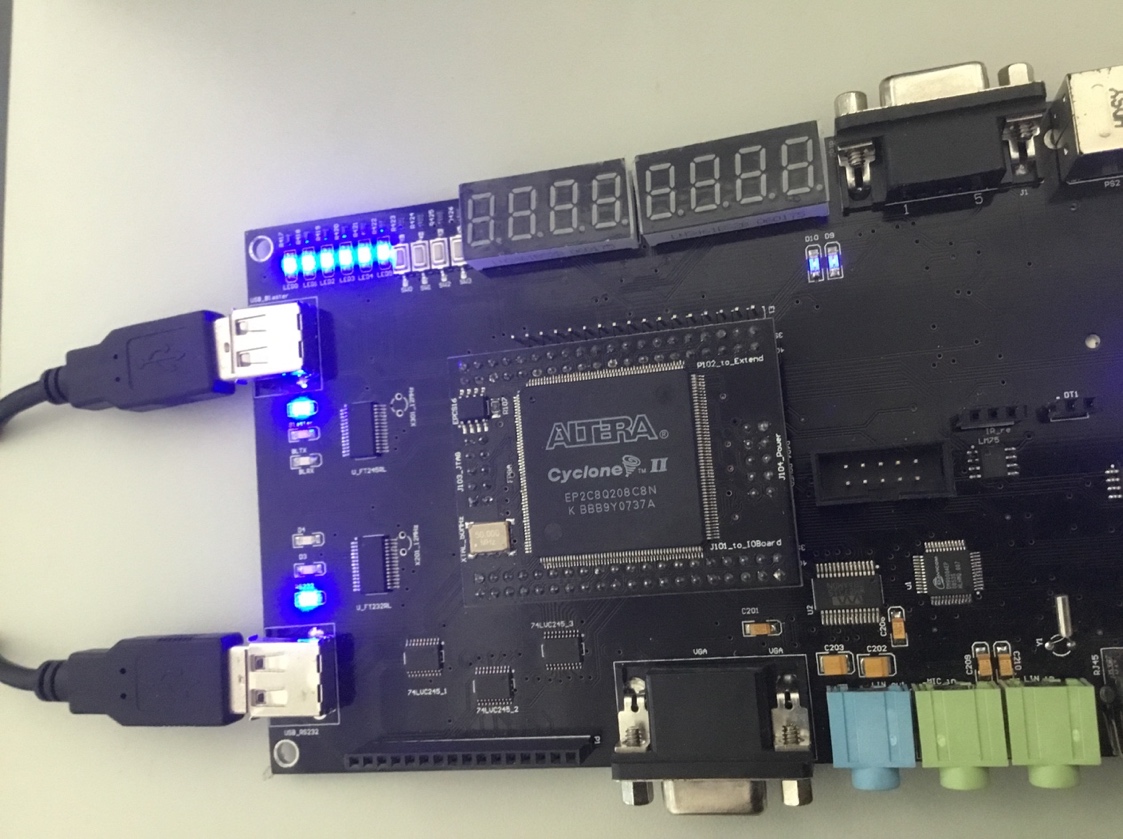


图7-1

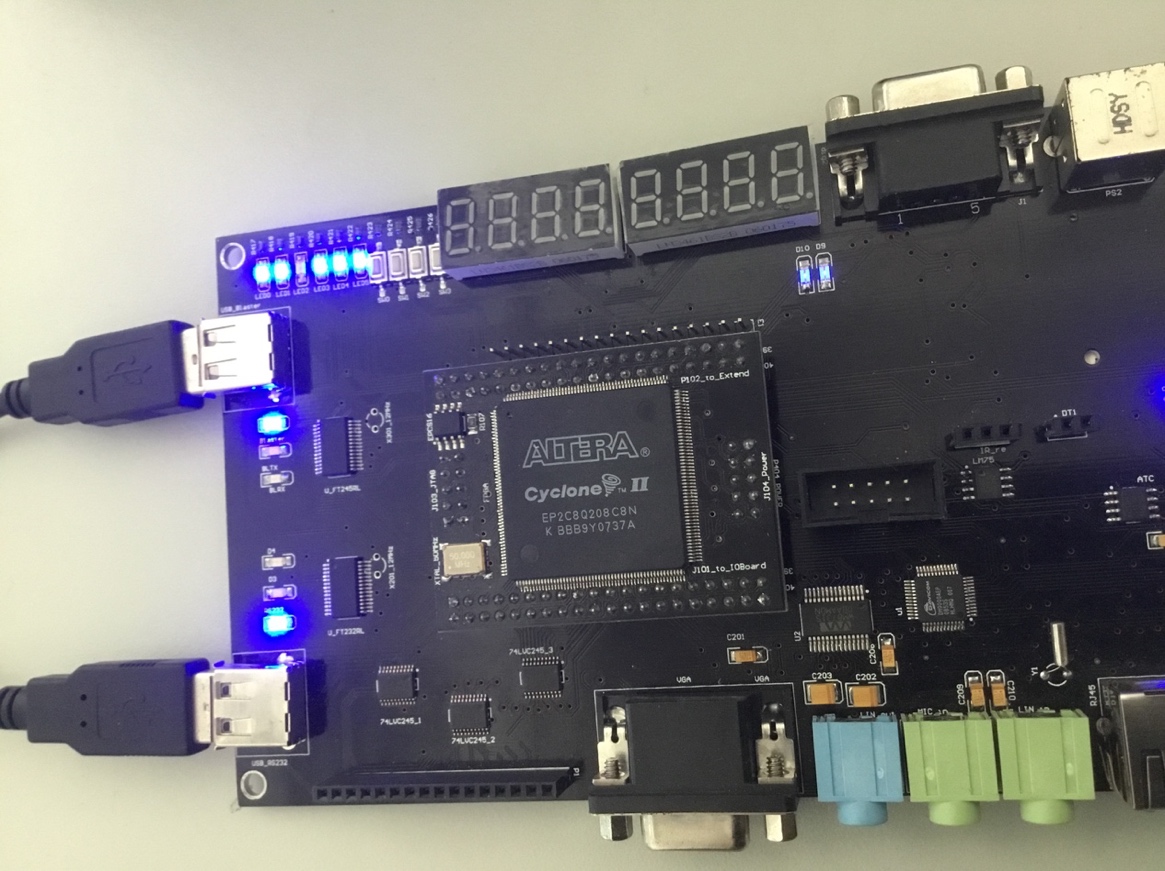


图7-2