2020-9-28

秦嘉余

191220088

1348288404@qq.com

数字电路实验五 计数器和时钟

目录

[1.实验目的 1](#_Toc53603280)

[2.实验原理 1](#_Toc53603281)

[3.实验环境/器材 1](#_Toc53603282)

[4. 基础实验 2](#_Toc53603283)

[1.设计思路 2](#_Toc53603284)

[2. 实验代码 2](#_Toc53603285)

[3.RTL视图 3](#_Toc53603286)

[4.引脚分配图 3](#_Toc53603287)

[5.实验结果 3](#_Toc53603288)

[5. 拓展实验 4](#_Toc53603289)

[1.设计思路 4](#_Toc53603290)

[2.实验代码 4](#_Toc53603291)

[3.RTL视图 6](#_Toc53603292)

[4.引脚分配图 7](#_Toc53603293)

[5.实验结果 7](#_Toc53603294)

[6.实验实验中遇到的问题及解决办法 7](#_Toc53603295)

1.实验目的

本次实验的目的是利用开发板上的时钟完成一个计数器与多功能时钟，同时复习时序电路的特点，熟悉verilog语言的多模块写法

2.实验原理

加法计数器和减法计数器都是数字电路中非常常见的原件，由多个D触发器组成，而在verilog中，这种器件很好实现，只需要写出逻辑代码，大部分任务都由编译器完成。同时，开发板上带有时钟信号，利用这个是时钟作为我们的初始时钟信号即可完成很多功能。开发板上的时钟器件为50MHz的时钟，所以我们需要使用分频器变为我们需要的时钟频率的时钟信号。有了我们需要的时钟信号，即可不断拓展完成我们的实验。

3.实验环境/器材

本次实验的环境为Quartus17.1版本

本次实验的器材为DE10 Standard开发板

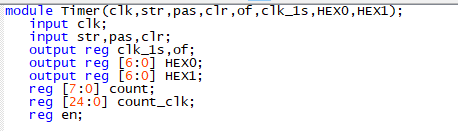
4. 基础实验

## 1.设计思路

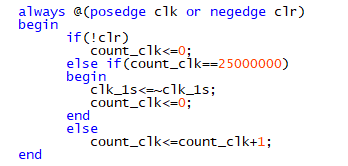
我们的开发板上的时钟信号为50MHz的时钟，所以我们需要使用分频器来将该时钟信号变为1Hz的时钟信号。Always语句来检测clk时钟的信号，不断在计数count上加一作为秒数，当最大为99时变为0，将count显示在2个7段显示管上。除此之外，设置一个使能端en，由str，pas控制，完成计数器的开始与暂停，最后在clk的always语句中加入clr为敏感信号，来为coun清零

## 2. 实验代码

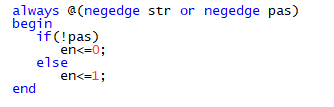
模块名为Timer，包含输入clk，str，pas，clr分别为时钟信号，开始，暂停，清零，输出有clk\_1s，of，HEX0，HEX1分别为输出的1s的时钟信号，计数到99的标志，两个七段显示管。同时还有中间变量count，count\_clk，en分别为计数，50MHz的计数，暂停开始使能端



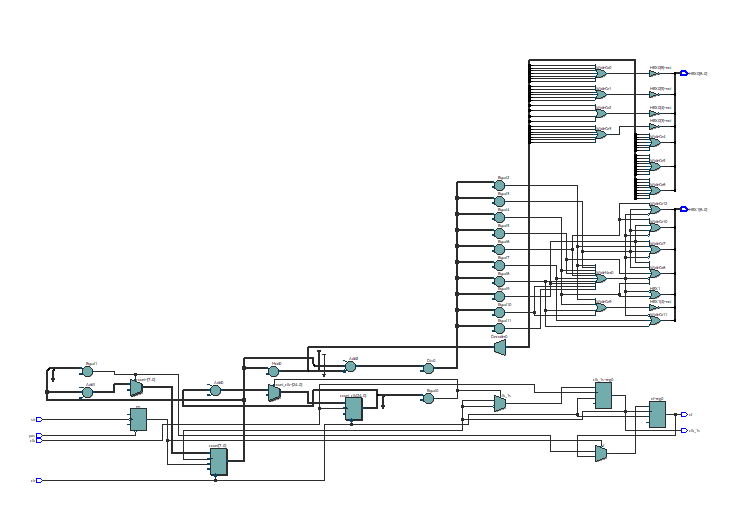
clr与clk都会改变count\_clk由于verilog 的限制，必须在同一always语句中



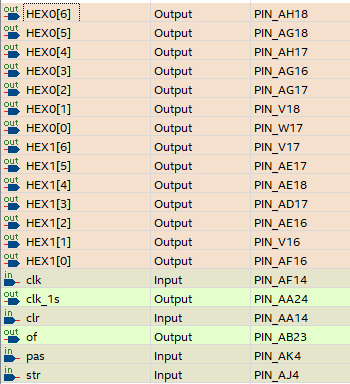
pas与str会改变en的大小



## 3.RTL视图



## 4.引脚分配图



## 5.实验结果

已在开发板上验收

5. 拓展实验

## 1.设计思路

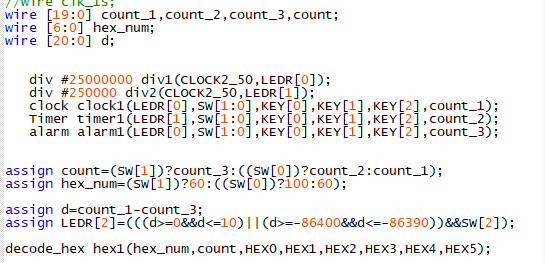
拓展试验的原理其实大致上与基础实验一致，都是先分频，利用这个时钟来拓展一系列操作，但在本实验中，需要拓展的部分较多，首先，数码显示管需要显示的内容有三种，分别为时钟，秒表与闹钟，这就需要用开关来切换，同时每个功能都是同时运行的的，互不干扰，只是显示哪一个的问题，但是调时或者开始暂停清零按钮却只在显示该功能时有效，所以为了完成这个复杂的功能，分模块是很有必要的。

模块如下图所示：

其中1s频供时钟模块与闹钟模块使用，1ms模块供秒表使用；时钟，秒表，闹钟频产生的count给decode\_hex使用来输出

## 2.实验代码

顶层模块如下：



**其中：**

count\_1,count\_2,count\_3分别为三个功能产生的计数，count为最终计数，由assign语句来根据选择端来赋值

hex\_num为hex显示的低两位起端显示管的进位，因为时钟与闹钟最低位为60进位，而秒表为100进位

d为时钟时间与闹钟时间的差，用来使得闹钟触发灯亮显示一段时间

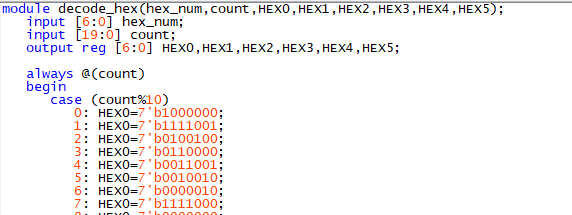
SW[2]决定闹钟是否开启

div模块的参数分别赋值为不同值产生两个不同的时钟信号1s与10ms给时钟闹钟与秒表使用

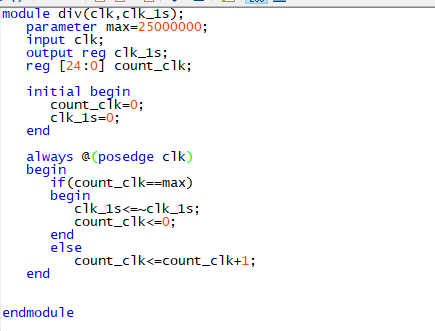
clock，timer，alarn模块传入时钟与选择端，控制端，给出计数输出counr\_x

其中SW[1:0]为0时显示时钟，1时显示秒表，2时显示闹钟，但这只是显示的功能，所有功能在选择端不为当前功能时仍然在运行，这符合多功能时钟的特性

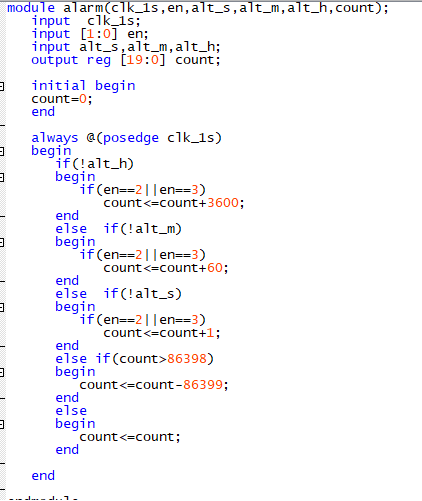
显示模块如下:



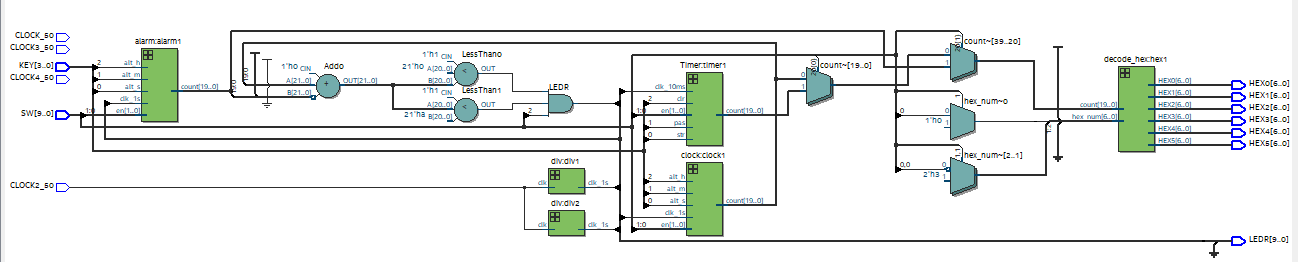
分频模块如下：



其余模块与计数器类似，只不过拓展不同的功能，由于篇幅限制，只展示alarm模块



## 3.RTL视图



每个模块内部的RTL视图由于篇幅不再展示

## 4.引脚分配图

本项目由build软件自动生成，则不再展示引脚分配图

## 5.实验结果

已在开发板上验收

6.实验实验中遇到的问题及解决办法

本次实验由于较为复杂，遇到了很多问题，大多为对verilog是一门硬件语言理解不足，许多在软件语言中显而易见的语句在verilog中不能成立，所以发生了很多语法问题，下列举几个：

Question1:

Always语句中不能存在 @(clk or negedge clr)这种含有上升下降沿和下降沿的形式

Question2：

多个always语句不能对同一个变量赋值

Question3：

多个变化的always中必须出现最后一个变化沿的判断if语句

这些都是与硬件息息相关的内容，例如多个always不能对一个变量赋值，这符合硬件电路中不能给一个输出多个接多条线的原则

所以在写verilog文件时，需要时时刻刻注意这是一门硬件语言，避免犯语法错误