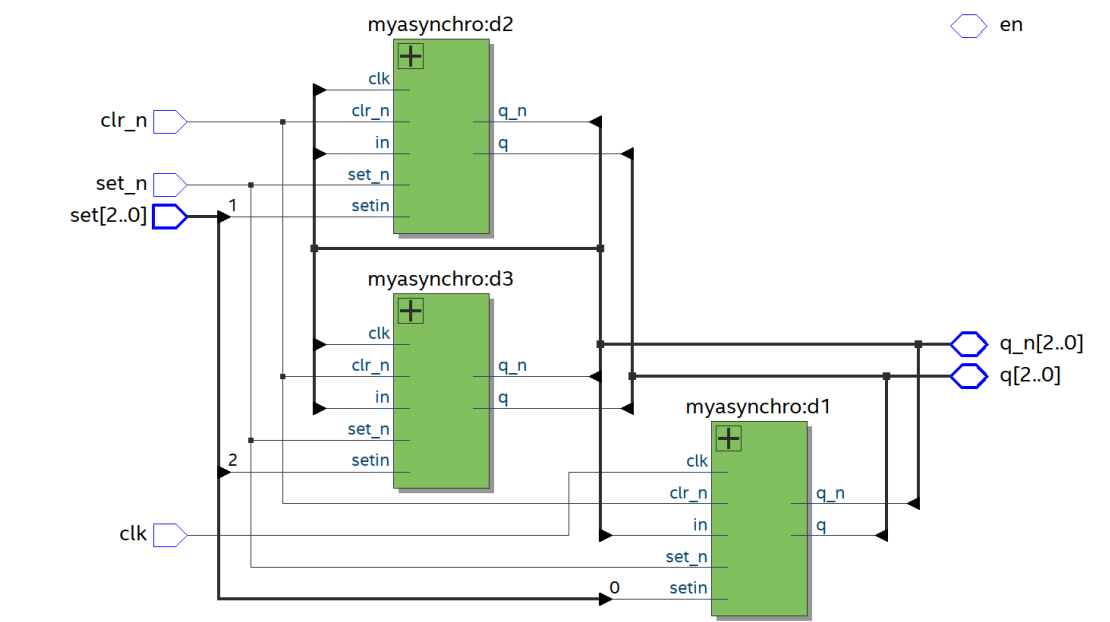


实验五 计数器和时钟

实验报告

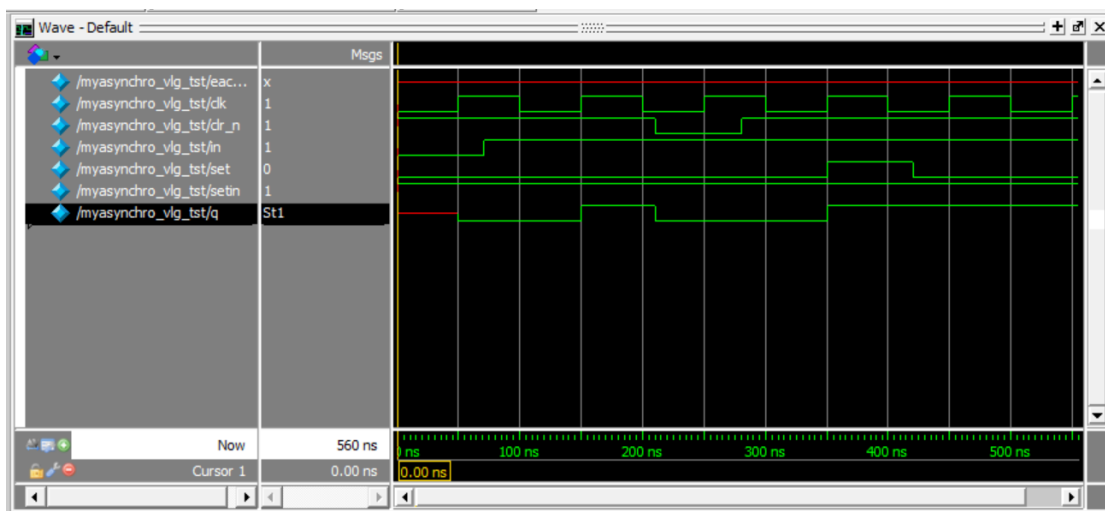
181860085 汤昊

数字电路与数字系统 2 班
邮箱: 1174639585@qq.com
2019.10.3



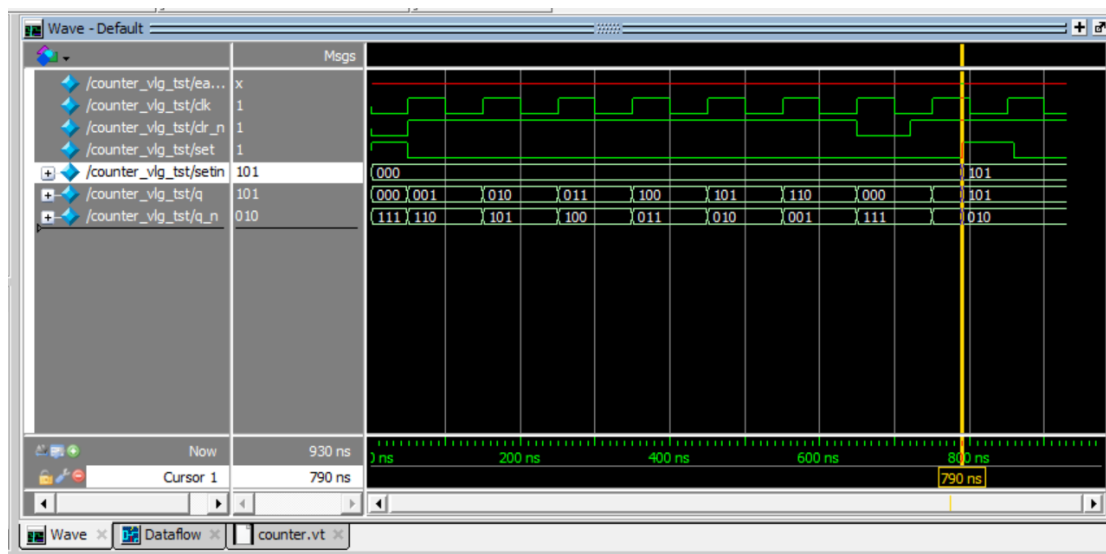
(4) 仿真模拟

- 带清零和置数端的 D 触发器测试



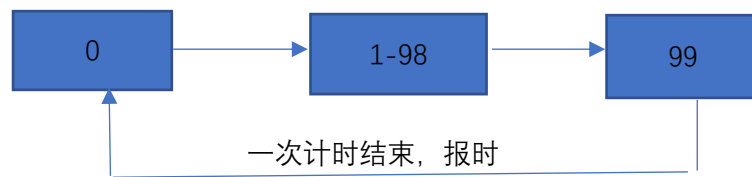
Clr_n 有效时立即清零，set 有效时立即转为 setin 的值

- 用上述三个 D 触发器实现的行波计数器测试



能够正常循环计数 000~111，且在清零端时有效时清零（图中 110—000），在置数端有效时置为输入(图中 000—101)

五．定时器



(1) 分频器

输入 50MHz 的时钟，输出 1Hz，周期为 1s 的信号，

设计：时钟每秒变化 50M 次，那么产生 25M 的上升沿，用一个变量每次上升沿计数，达到 25M 输出一次信号。由于 $2^{24}=16777216$ ， $2^{25}=33554432$ ， $2^{24}<25M<2^{25}$ ，因此 count_clk 的位宽应该设置为 25 位

(2) 计时功能

由于计数范围是 0-99，所以用两个寄存器分别记录时间的个位和百位，用分频器输出的信号作为时钟，每次上升沿进行循环加操作。

(3) 开始，暂停和清零功能

开始键高电平有效，有效时计数，无效时数码管无显示

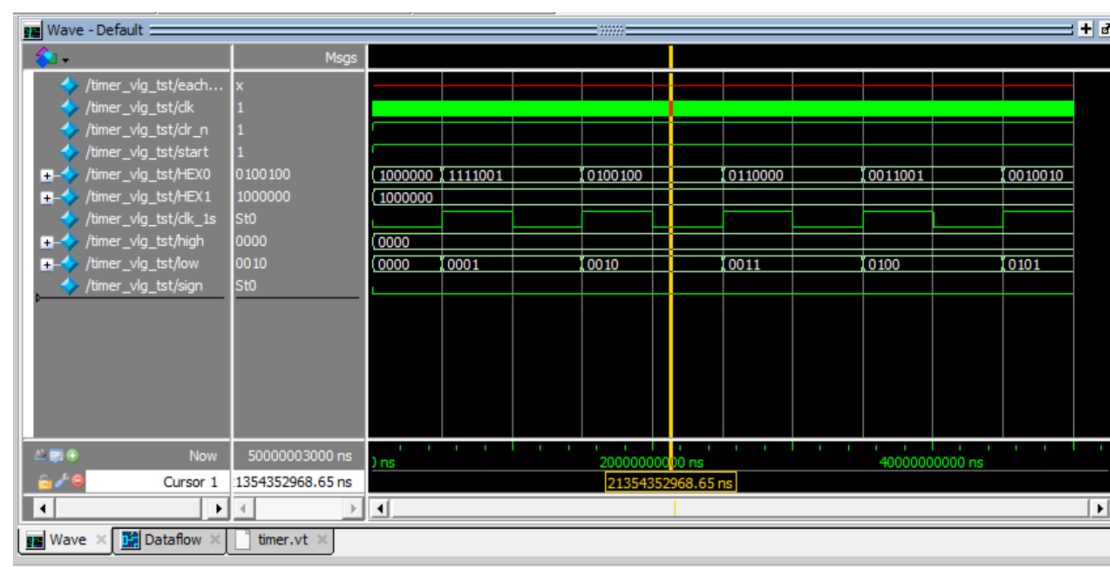
暂停键低电平有效，有效时时钟保持在当前状态

清零键低电平有效，有效时时钟计数归 0

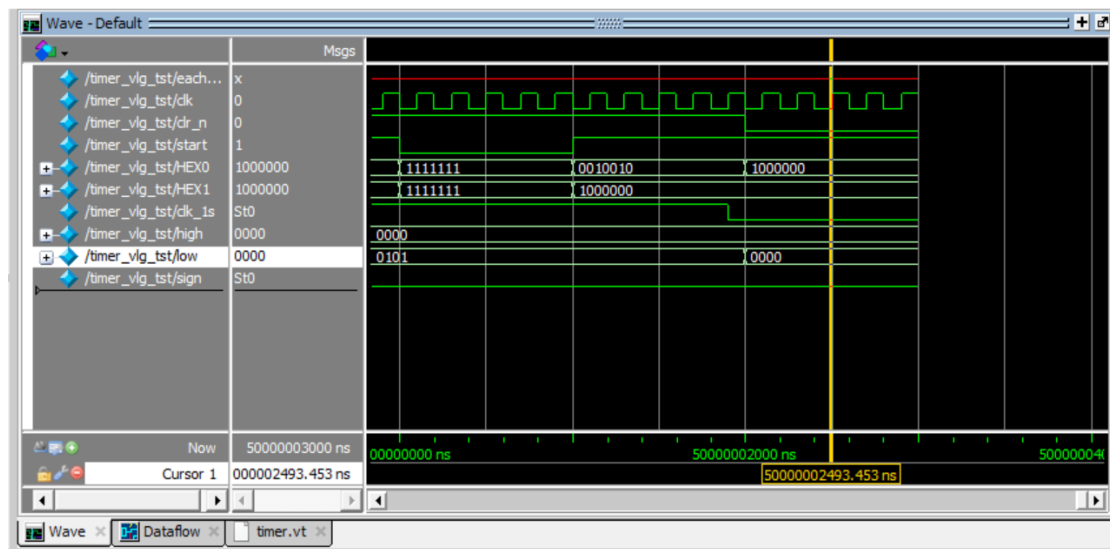
(4) 计时结束指示

用一个变量指示，当达到 99 要+1 时，变量为 1，其余时间都赋值为 0，这样指示的效果可以维持到下一次高电平。

(5) 仿真模拟



仿真模拟了五个上升沿，可以看到在每个 clk_1s 上升沿，low 代表的个位数字递增，HEX0 的编码也相应变化



在 start 无效时代表计数暂停，此时内部计时暂停，数码管全一不再显示，而在清零端 clr_n 有效时，high 和 low 存储的值都清零，数码管也为 0 的编码

sign 的测试由于需要到 99-0 时体现，耗时较长，直接在开发板上验证，实际效果为 99 变成 0 时灯亮，持续到变为 1 时。

六．电子时钟



(1) 模块划分

分为分频器 (divider): 利用硬件原有的时钟提供 1s 的时钟输入

时钟 (digitclock): 实现计时，秒表，设置时间等功能

显示 (display): 实现时间的数码管显示和闹钟的指示

(1) 模块设计

- 分频器: 与上述计时器相同
- 时钟: 用 stopwatch 和 settime 代表时钟的三种状态
 - Stopwatch 和 settime 都为 0: 正常计时，每个上升沿用 if-else 语句对时间状态进行判断，得出+1s 的结果
 - Stopwatch 为 1: 秒表状态，进行 0-59 的循环计数，只显示秒数
 - Settime 为 1: 设置时间，考虑到开发板的 switch 开关只有 10 个，使用两位的 setselect 指示要设置的时间类型，其他 8 位为时间输入
 - Setselect==2'b00: 将秒数修改为输入
 - Setselect==2'b01: 将分钟修改为输入
 - Setselect==2'b10: 将小时修改为输入
 - Setselect==2'b11: 暂停

- 显示：用 case 语句将输入的时间转换为相应的数码管编码，当输入符合程序设置的某个值时，闹钟指示灯亮

(2) 仿真模拟

测试方法：通过设置时间信号将时钟定位到某些特殊时刻，检查是否能正常计数

```
clk=0;settime=0;stopwatch=0;#2500000000;//正常计时
```

```
set_l=4'b1001;
```

```
set_h=4'b0101;//设置输入为 59
```

```
settime=1;
```

```
setselect=2'b00;#2000000000;//设置秒
```

```
settime=0;#1000000000;
```

```
settime=1;
```

```
setselect=2'b01;#1500000000;//设置分
```

```
settime=0;#2000000000;
```

```
settime=1;
```

```
set_l=4'b0011;
```

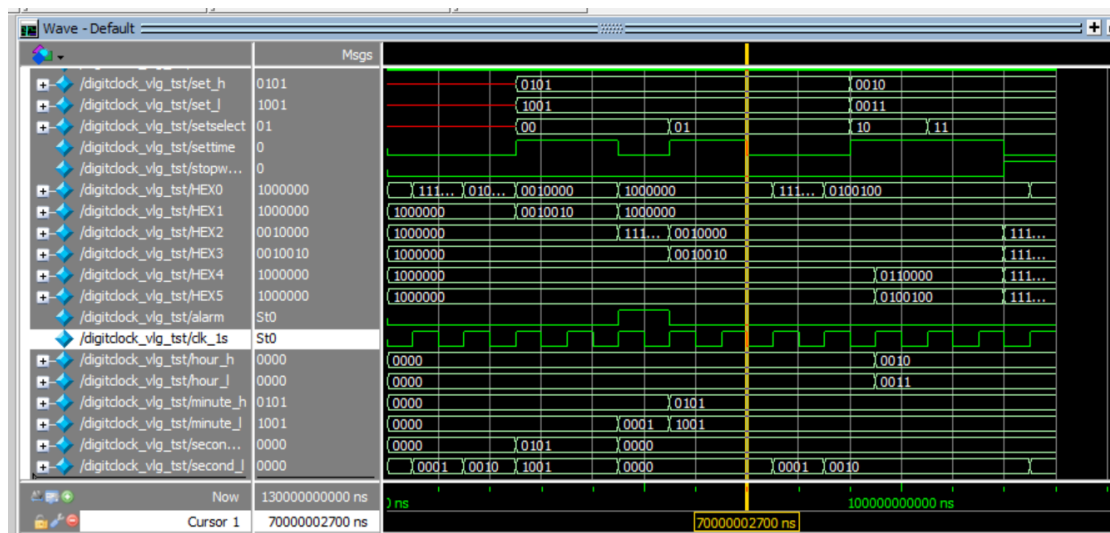
```
set_h=4'b0010;
```

```
setselect=2'b10;#1500000000;//设置小时
```

```
setselect=2'b11;#1500000000;//暂停
```

```
settime=0;stopwatch=1;#1000000000;//测试秒表
```

```
$stop;
```



开始正常计时，秒从 0 到 2，之后 settime 有效，输入为 59，秒数置为 59，settime 置 0，恢复计时，下一个上升沿产生进位，秒数为 0，分钟为 1，接着 settime 有效，分钟被置为 59，之后输入改为 23，小时置为 23，最后状态切换为秒表，只有秒数变化

七．实验反思与收获

- Always 不可同时监测一个变量的上升沿和下降沿
- 可以对 reg 型变量赋初值，不可以对 wire 型变量赋初值
- 模块实例化的参数错误在对单个文件编译时无法发现