

数字电路高层次综合设计 第十一周作业

范云潜 18373486

微电子学院 184111 班

日期：2020 年 11 月 20 日

目录

1 N 位通用移位寄存器	1
2 手动模 M 计数器	2
3 分频器	2
4 简易电子时钟	3
5 7 段管循环显示	3

List of Figures

1 移位寄存器结果	2
2 取模器结果	2
3 分频器结果	3
4 数字时钟结果	3
5 循环显示结果	4

本次作业很明显的体现了代码复用的原则，在之后开始一周的作业时，应先对复用性进行一定的分析。本次报告中将模块的定义与端口通过伪代码形式给出。

1 N 位通用移位寄存器

顶层为 `./01shift/shift.v`，对本实验进行模块划分：

- 数据移位寄存器：`shift (clk_key, data_init, ctrl, data_out)`
- 数据显示七段管：`dec (data, HEX)`

需要注意，Quartus 对时钟与复位的要求是，触发类型相同，不能出现电平触发和沿触发混合的形式。对 `ctrl` 的控制通过 `case` 进行选择即可。

图 1: 移位寄存器结果

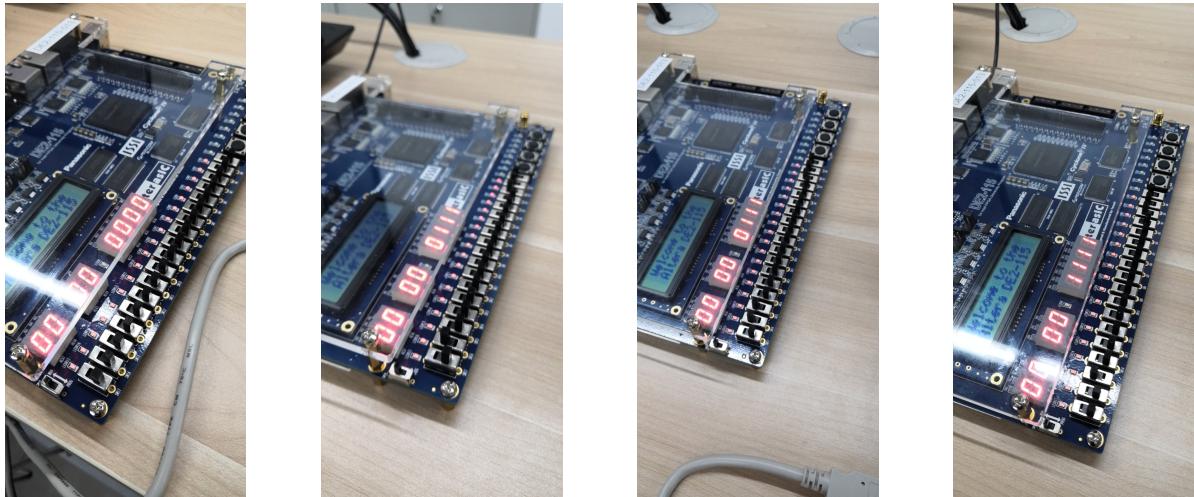
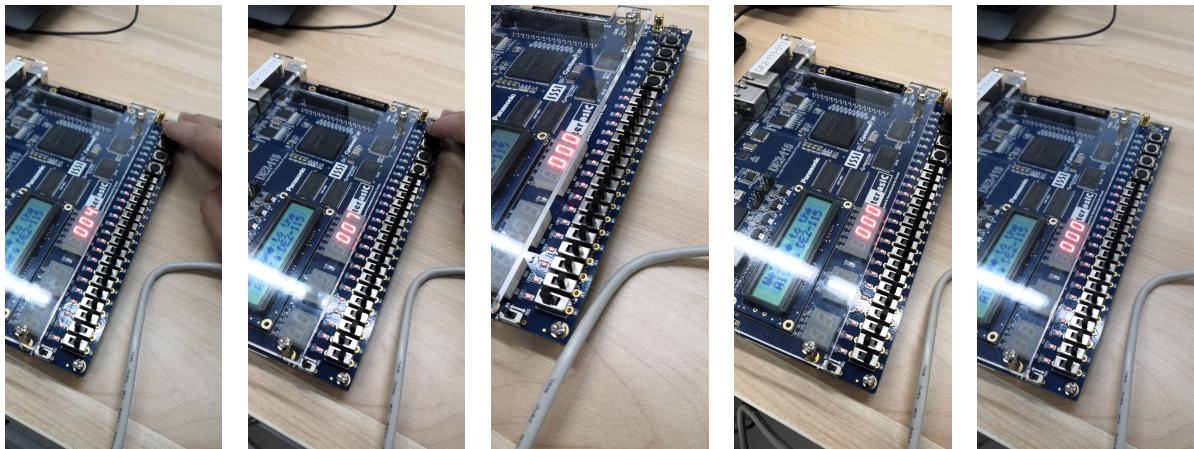


图 2: 取模器结果



2 手动模 M 计数器

顶层为 `./02shift/mod.v`，对本实验进行模块划分：

- 计数器：`cnt (clk, reset, sum)`，对给定信号的上升沿进行计数，并且设定计数上限为 M 即可
- 使能信号：可以添加到上一模块，或者将上一模块的输入修改为 `clk & en`
- 显示模块：将计数器的 `sum` 输入到 BCD 编码器，将 BCD 编码器的输出接到七段管需要注意不能使用除法以及取模操作，消耗资源极大。

3 分频器

首先明确，晶振和需要的输出之间频率的数量级差距很大，因此无需过分在意奇偶分频的区别，通过计数上升沿进行计算的误差会很小。

顶层为 `./03clk/clk.v`，模块划分：

图 3: 分频器结果

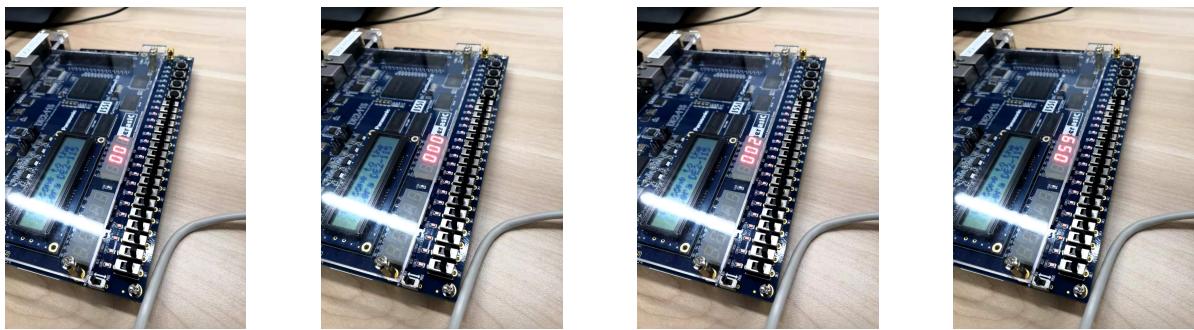
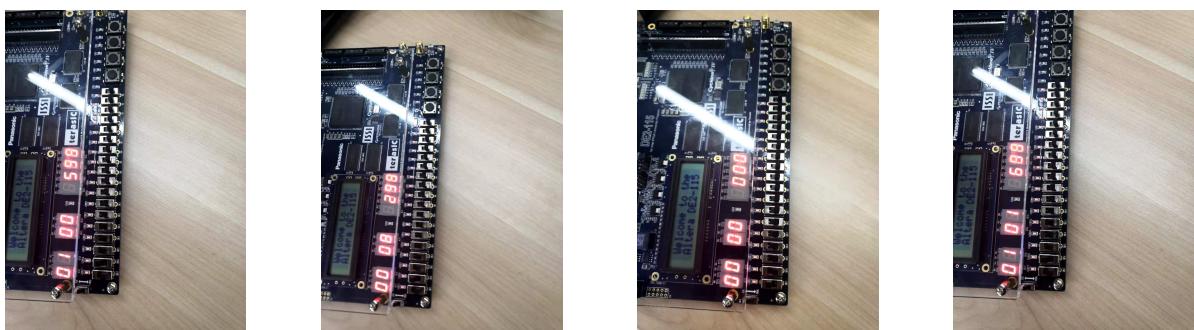


图 4: 数字时钟结果



- 分频器¹: `tik #(.BASE_FREQ(), .NEEDED_FREQ())(clk, reset, tik_clk)`
- 计数器: `cnt (clk, reset, sum)`, 对给定信号的上升沿进行计数, 并且设定计数上限为 M 即可, 对分频后的时钟信号进行计数
- 显示模块: 将计数器的 `sum` 输入到 BCD 编码器, 将 BCD 编码器的输出接到七段管

4 简易电子时钟

顶层为 `./04time/clk.v`, 模块划分:

- 分频器: `tik #(.BASE_FREQ(), .NEEDED_FREQ())(clk, reset, tik_clk)` 使用上一实验中的分频器即可, 获得毫秒周期的信号
- 时序逻辑计数: 对毫秒进行计数, 并且手动产生秒与分钟的进位
- BCD 编码器: 将之前实验中的 BCD 输入修改为 10 位, 来满足 0-1000 范围内的编码
- 显示模块: 将计数器的毫秒、秒、分钟输入到 BCD 编码器, 将 BCD 编码器的输出接到七段管

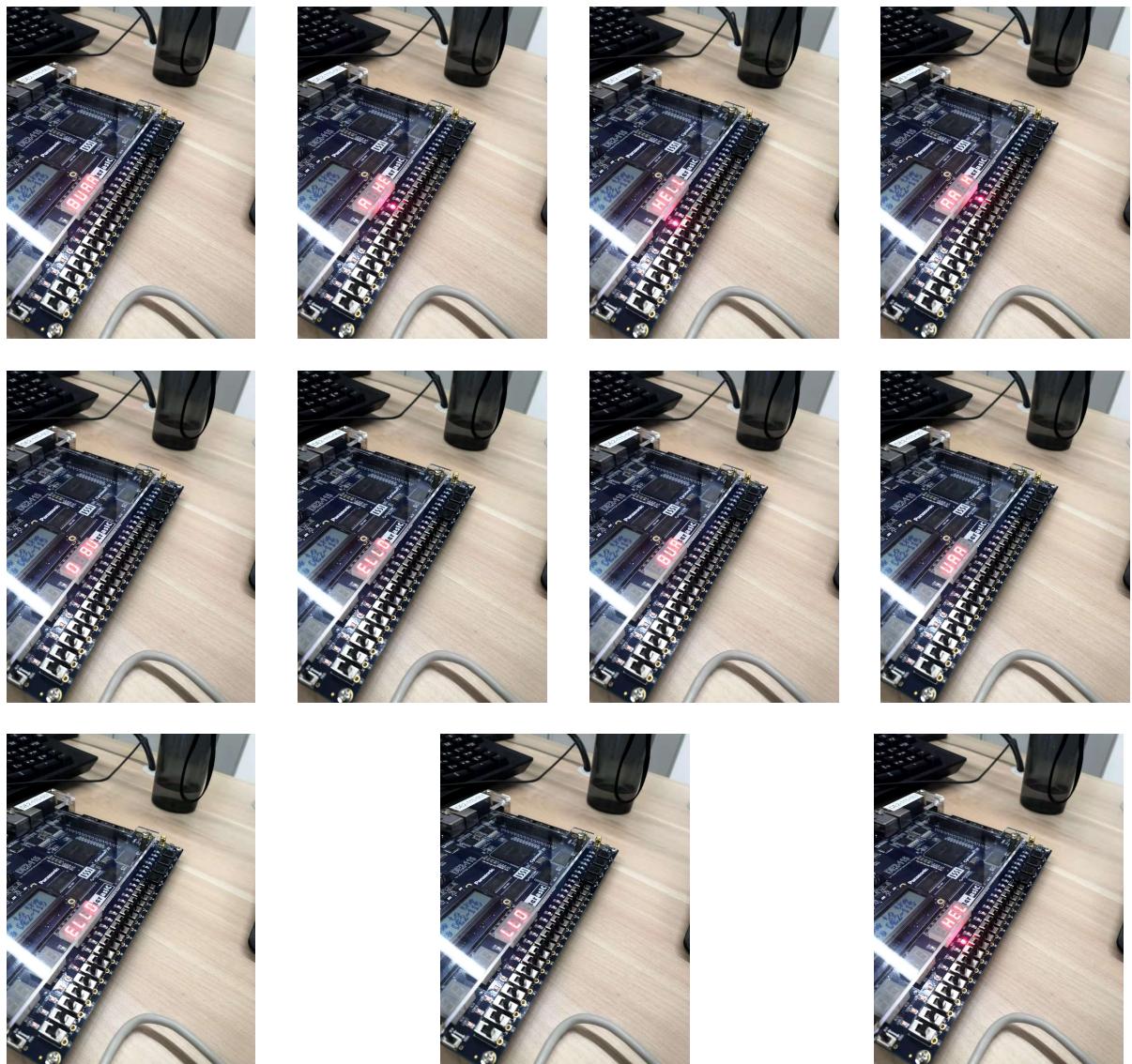
5 7 段管循环显示

顶层为 `./05buaa/show.v`, 模块划分:

- 循环移位寄存器: 设定以四位为单元的移位寄存器, 将长度与需要的初始值写入参数:

¹类似上一实验中的计数器, 但是输出变成一位的 `tik` 信号

图 5: 循环显示结果



```
shift #(.N(), .[4*N-1:0]INIT())(clk, reset, out)
```

- LED 显示: 通过组合逻辑写出
- 分频器: 产生一秒的时钟信号
- 显示模块: 需要定义新字符的段码

将字符串 'HELLO BUAA' 进行分析, 其中不存在 'HELUA' , 其他的可以从对应关系中导出:
0->U, 8->B, ' '->default , 恰好补全十进制七段管显示的 0xa-0xe 。