

# 电子技术实验 2 实验报告

学号：2206113602

班级：信息 005

姓名：王靳朝

编号：

## 6 计数器设计与应用

### 一 实验内容

模 16 计数器的实现

### 二 实验原理



#### 1. 异步计数器的电路分析

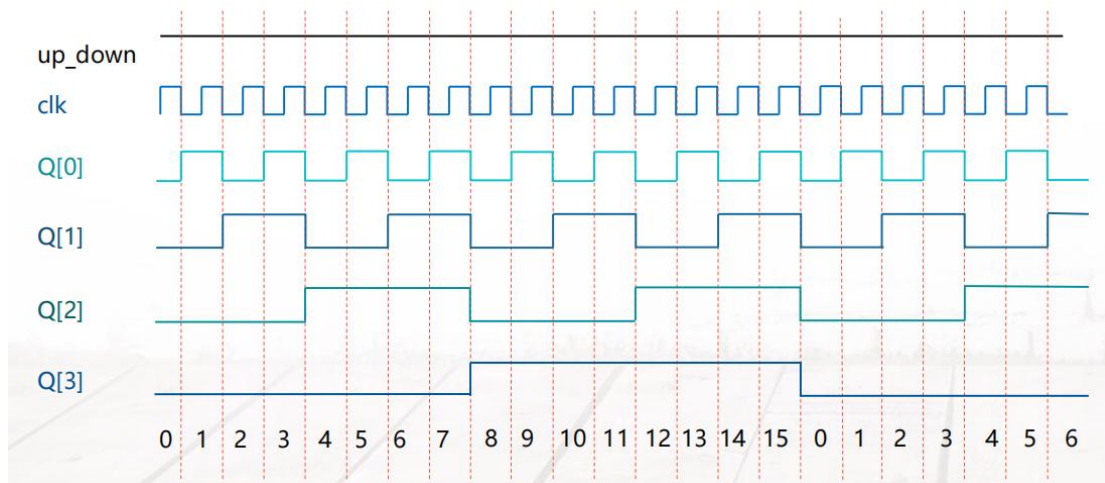
异步模 16 计数器由 4 个 D 触发器级联实现。每个 D 触发器的输出端取非之后进入输入端，同时每一个 D 触发器的输出端和 up/down 信号异或作为下一级的时钟信号。二进制的模 16 异步计数器分别由上一级的输出 Q 串行输入得到。

#### 2. 同步计数器的电路设计

同步模 16 计数器同样由 4 个 D 触发器实现，四个 D 触发器具有相同的时钟信号输入，因此时同步计数器。在每一个 D 触发器都具有计数功能，同时本级输入由上一级输出、上上一级的与门信号异或而成。

#### 3. 计数器实现分频器的原理分析

原理分析，4 位二进制模 16 计数器具有 4 位输出，分别为  $Q[0]$ ,  $Q[1]$ ,  $Q[2]$ ,  $Q[3]$ 。其中  $Q[0]$  在时钟信号的上升沿来临时进行高低电平的翻转， $Q[1]$  在  $Q[0]$  的上升沿来临时进行高低电平的翻转， $Q[2]$  在  $Q[1]$  的上升沿来临时进行高低电平的翻转， $Q[3]$  在  $Q[2]$  的上升沿来临时进行高低电平的翻转。总体来看，当 clk 信号周期确定时， $Q[0]$ ,  $Q[1]$ ,  $Q[2]$ ,  $Q[3]$  分别为时钟信号 clk 周期的 2、4、8、16 倍，从而分频实现频率的降低。



### 三 实验步骤

#### 1. 包括创建工程

整个工程包含三个子模块，最后需要将三个子模块级联实现最终功能。三个子模块均选择使用 verilog 语言实现。工程文件夹中包含三个.v 文件，与之对应的波形仿真文件，以及编译知州形成的符号文件。

#### 2. 计数器设计实现、分频器设计实现

分频器的实现代码：

```

1 module lab6_1
2   (
3     input clk_in,
4     output reg[23:0] count,
5     output reg clk_out
6   );
7
8   always @(posedge clk_in) begin
9     if(count==24'd12000000) begin
10      count<=24'd0;
11    end
12    else begin
13      count<=count+24'd1;
14    end
15  end
16
17  always @(posedge clk_in) begin
18    if(count==24'd0) begin
19      clk_out<= ~clk_out;
20    end
21    else begin
22      clk_out<=clk_out;
23    end
24  end
25 endmodule
26

```

分频器需要将芯片晶振产生的 24MHz 时钟信号转换为 1Hz 的时钟信号，计算得到输出为 24 位二进制数，代码实现用计数器实现分频器，特别的需要考虑当技术达到 12M 时要重新制 0。

四位二进制计数器的实现代码：

```

1 module lab6_2
2   (
3     input clk_in,
4     output reg[3:0] count
5   );
6   always @(posedge clk_in) begin
7     count<=count+4'd1;
8   end
9 endmodule

```

七段数码管实现代码:

```
1 module lab6_3
2   (
3     input [3:0] k,
4     output reg [6:0] seg7out
5   );
6
7   always @(k) begin
8     case(k)
9       4'b0000:seg7out=7'b0000001;
10      4'b0001:seg7out=7'b1001111;
11      4'b0010:seg7out=7'b0010010;
12      4'b0011:seg7out=7'b0000110;
13      4'b0100:seg7out=7'b1001100;
14      4'b0101:seg7out=7'b0100100;
15      4'b0110:seg7out=7'b0100000;
16      4'b0111:seg7out=7'b0001111;
17      4'b1000:seg7out=7'b0000000;
18      4'b1001:seg7out=7'b0000100;
19      4'b1010:seg7out=7'b0001000;
20      4'b1011:seg7out=7'b1100000;
21      4'b1100:seg7out=7'b0110001;
22      4'b1101:seg7out=7'b1000010;
23      4'b1110:seg7out=7'b0110000;
24      4'b1111:seg7out=7'b0111000;
25    endcase
26  end
27 endmodule
28
```

### 3. 自底向上构建完整工程设计

每一项子模块搭建完成、编译通过并且进行仿真验证之后，需要生成对应的符号文件，最后新建原理图将符号文件级联。注意当某一级的输出信号位总线但是下一级的输入信号位单个线路时，需要将总线信号拉出并重新命名。

### 4. 管脚分配

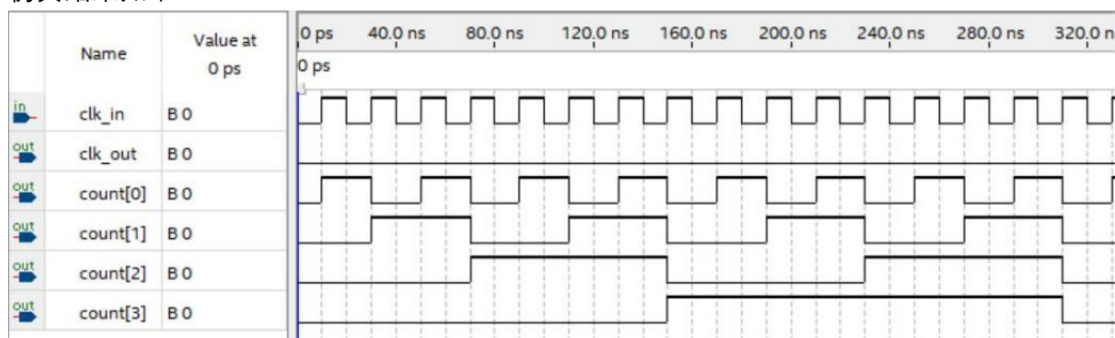
24MHz 时钟信号由 7pin 管脚产生，此外管脚分配与 7 段数码管相同。

### 5. 下载验证

将工程文件下载到芯片中，进行验证。

## 四 实验结果

仿真结果如下：



## 五 总结和思考题

### 1. D 触发器为基础的异步计数器的优点是什么？缺点是什么？

异步计数器由于没有统一的时钟信号，采用串行结构，每一级的信号向后传输会有延时，延时累计会导致输出的错误。优点是异步二进制加法计数器线路连接简单，可靠。

2. 为克服 1 中的缺点，D 触发器为基础的同步计数器的优点是什么？又带来了什么样的缺点？

同步计数器采用统一的时钟信号输入，每一级的输入信号延时相同，因此输出发生错误的概率较小。但是当 D 触发器的数量增大时，对时钟信号的带负载能力有要求。

3. 总结对比自顶向下和自底向上的数字电路设计方法

自底向上的电路设计是将各个子模块的功能制作和验证完成后，将不同模块进行组装达到最终目的。这种方法适用于规模较小的系统，逻辑架构比较简单明了。

自顶向下的电路设计与前者相反，先分析顶层模块，在分析构成顶层模块必要的底层，最后制作、验证。这种方法由于开始确定了顶层，因此逻辑较为清晰，各功能模块之间的关系也较为明确。它适用于比较大型的系统设计，适用于多人合作。