

实验: 计数器

学号:2016011086

姓名:田林龙

一.实验目的

- 掌握简单时序逻辑电路的设计方法
- 了解任意进制计数器的设计方法

二.实验原理

同步计数器原理

- 输入时钟信号,通过记录输入时钟脉冲的数目达到计数的目的
- 通过判断前位是否都为1来决定该位是否需要翻转

异步计数器原理

- 利用低一位的输出反相作为高一位的输入,达到加法的目的
- 利用低一位的输出作为高一位的输入,达到减法的目的

三.实验设计

同步计数器

计数模块

- 通过verilog的行为级设计,当累加数目达到 $1111_{(2)}$ 时,计数转为 $0000_{(2)}$,否则执行+1操作,减法同理

BCD显示模块

- 根据计数信号控制7段数码管,通过条件语句达到对应显示的目的

注:实验中,通过ud信号的切换,达到改变计数器功能的目的(加法与减法)

异步计数器

触发器模块

- 当时钟信号上升沿产生,将输入信号q进行反转

计数器模块

- 由四个触发器模块组成,其中高位的输入端与低位的输出端反相相连
- 四个触发器的四个输出端构成4bit的计数信号

BCD模块

- 将4bit信号转化为7段数码管的显示信号

四.实验代码

1.同步计数器(加减法综合)

```
1  module cnt_dec1(clk,rst,LED0,ud);
2  input  clk,rst,ud;
3  output [6:0] LED0;
4  wire[3:0] cnt;
5  counter U1(.clk(clk),.cnt(cnt),.rst(rst),.ud(ud));
6  BCD7 B(.din(cnt),.dout(LED0));
7  endmodule
8
9  module counter(clk,cnt,rst,ud);
10 input  clk,rst,ud;
11 output [3:0] cnt;
12 reg[3:0] cnt;
13
14 always @(posedge clk or negedge rst) begin
15     if(~rst)
16         cnt <= 4'b0000;
17     else begin
18         if (ud) begin
19             if(cnt == 4'b1111) cnt <= 4'b0000;
20             else cnt <= cnt + 1;
21         end
22         else begin
23             if(cnt == 4'b0000) cnt <= 4'b1111;
24             else cnt <= cnt - 1;
25         end
26     end
27 end
28 endmodule
29
30 module BCD7(din,dout);
31 input  [3:0] din;
32 output [6:0] dout;
33
34 assign  dout=(din==4'h0)?7'b1000000:
35           (din==4'h1)?7'b1111001:
36           (din==4'h2)?7'b0100100:
37           (din==4'h3)?7'b0110000:
38           (din==4'h4)?7'b0011001:
39           (din==4'h5)?7'b0010010:
40           (din==4'h6)?7'b0000010;
```

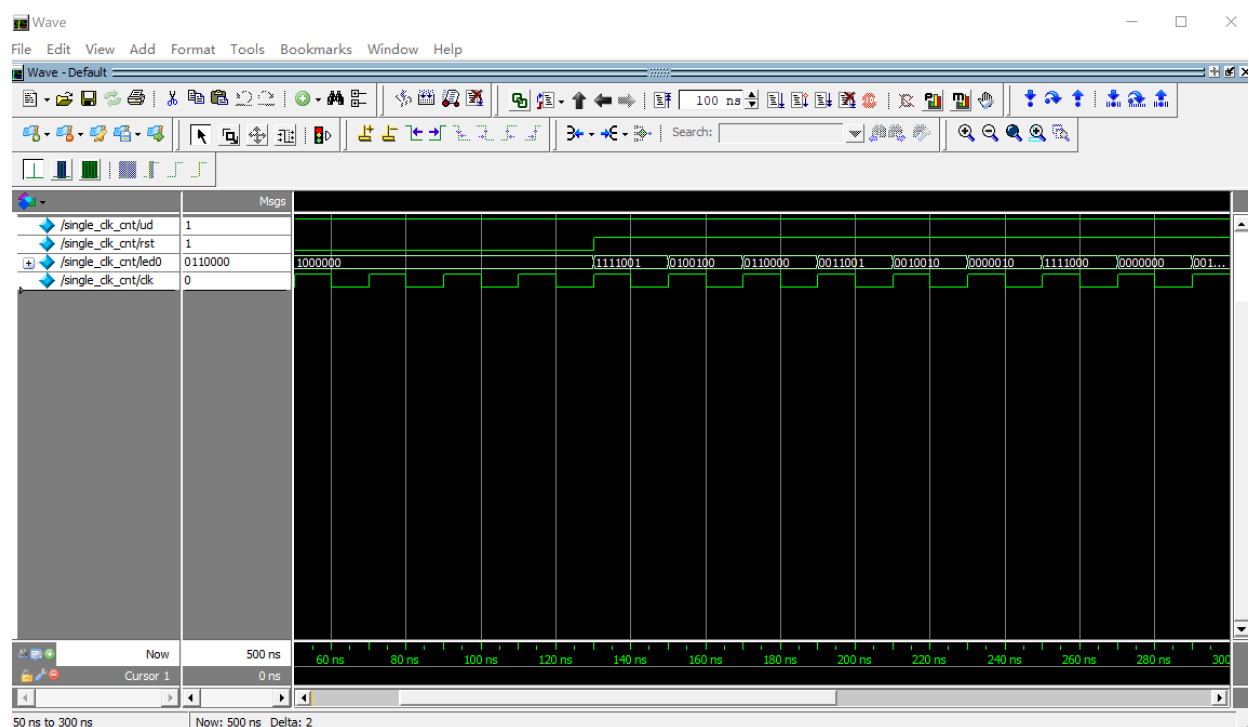
```

41         (din==4'h7)?7'b1111000:
42         (din==4'h8)?7'b0000000:
43         (din==4'h9)?7'b0010000:
44         (din==4'hA)?7'b0001000:
45         (din==4'hB)?7'b0000011:
46         (din==4'hC)?7'b1000110:
47         (din==4'hD)?7'b0100001:
48         (din==4'hE)?7'b0000110:
49         (din==4'hF)?7'b0001110:7'b0;
50     endmodule

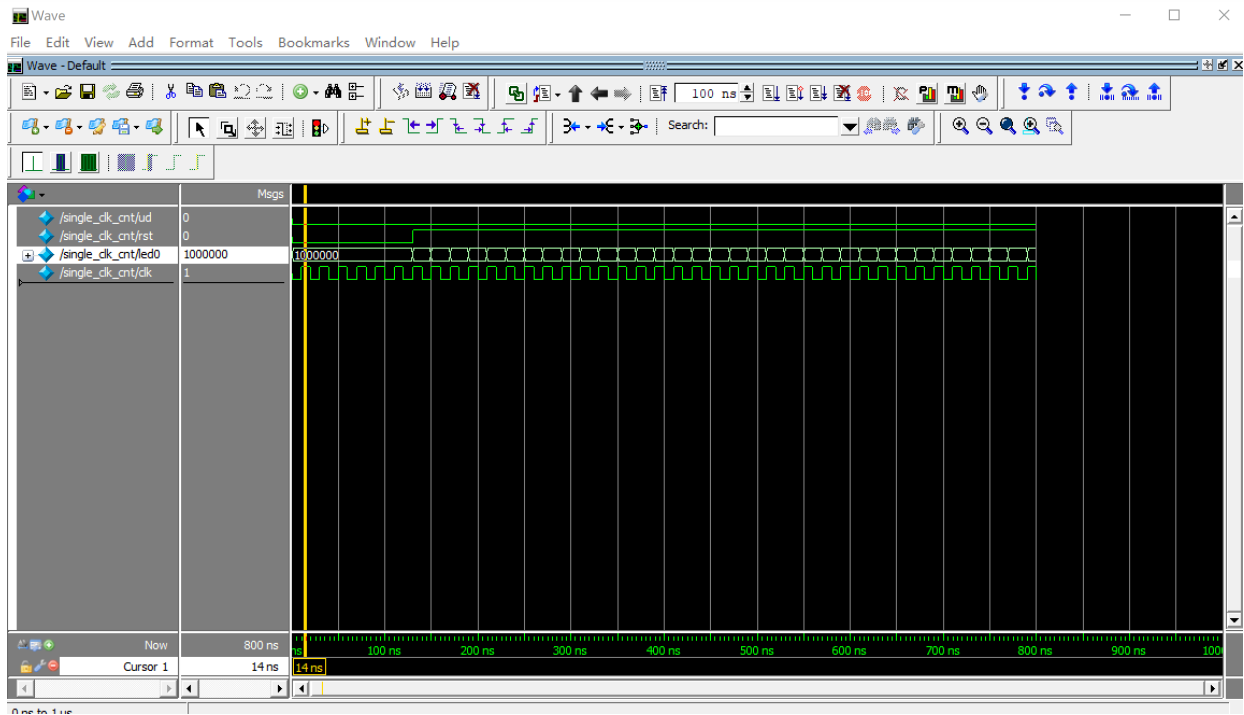
```

仿真结果

加法器



减法器



2.异步加法器

```

1  module cnt_dec2(clk,rst,LED0);
2  input clk;
3  input rst;
4  output [6:0] LED0;
5  wire [3:0] cnt;
6  counter U1(.clk_in(clk),.rst(rst),.cnt(cnt));
7  BCD7 B(.din(cnt),.dout(LED0));
8  endmodule
9
10 module counter(clk_in,rst,cnt);
11 input clk_in,rst;
12 output [3:0] cnt;
13 wire [3:0] cnt;
14 Dfff dff1(.clk(clk_in),.rst(rst),.q(cnt[0]));
15 Dfff dff2(.clk(~cnt[0]),.rst(rst),.q(cnt[1]));
16 Dfff dff3(.clk(~cnt[1]),.rst(rst),.q(cnt[2]));
17 Dfff dff4(.clk(~cnt[2]),.rst(rst),.q(cnt[3]));
18 endmodule
19
20 module Dfff(clk,rst,q);
21 input clk,rst;
22 output q;
23 reg q;
24 always @(negedge rst or posedge clk)
25 begin
26     if(~rst)
27         q <= 0;
28     else begin

```

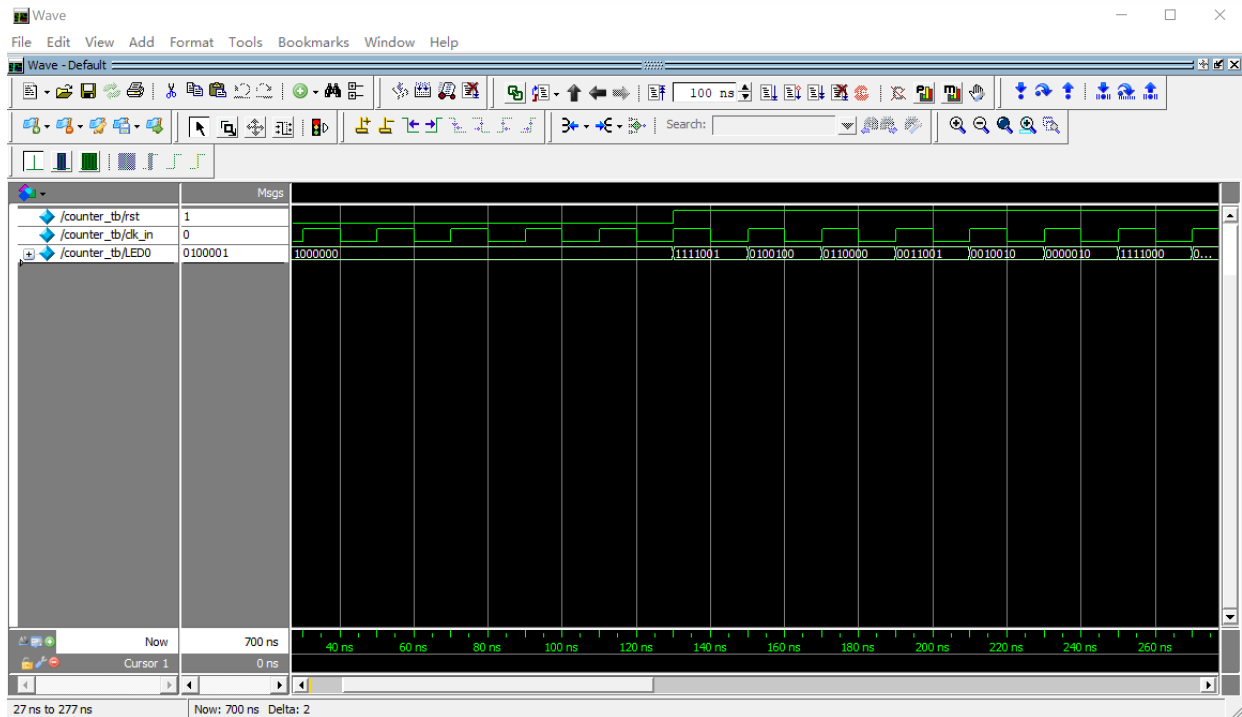
```

29         q <= ~q;
30     end
31 end
32 endmodule
33
34 module BCD7(
35     din,
36     dout
37 );
38 input    [3:0]    din;
39 output   [6:0]    dout;
40
41 assign   dout=(din==4'h0)?7'b1000000:
42             (din==4'h1)?7'b1111001:
43             (din==4'h2)?7'b0100100:
44             (din==4'h3)?7'b0110000:
45             (din==4'h4)?7'b0011001:
46             (din==4'h5)?7'b0010010:
47             (din==4'h6)?7'b0000010:
48             (din==4'h7)?7'b1111000:
49             (din==4'h8)?7'b0000000:
50             (din==4'h9)?7'b0010000:
51             (din==4'hA)?7'b0001000:
52             (din==4'hB)?7'b0000011:
53             (din==4'hC)?7'b1000110:
54             (din==4'hD)?7'b0100001:
55             (din==4'hE)?7'b0000110:
56             (din==4'hF)?7'b0001110:7'b0;
57 endmodule

```

仿真结果

加法器(异步)



- 上述三个仿真结果说明,随着时钟的变化,LDE灯正常完成的计数功能

五.实验小结

1. 实验中出现如下报错信息

```
1 Error (12004): Port "rst" does not exist in primitive "dff" of instance "dff"
```

经过分析,认识到是模块名dff与内置出现重复,从而导致错误,将"dff"改为"Ddff"便顺利通过

2. output变量可以是net,reg,当模块中量要被赋值时,需要声明为reg变量,但是外部必须连接net类型因而在测试文件中,要注意输出变量的类型可以是net