

数字逻辑实验报告

实验 3

学 期	2022-2023 学年第 1 学期		实验日期	2022/11/4	
学 院	信息学部		专 业	计算机科学与技术（实验班）	
班 级	210710	学 号	21071003	姓 名	高立扬
组 号	43	学 号	21071004	姓 名	石昊阳

评 阅 内 容

任务一	任务二	总结	格式	成 绩

题 目

实验 3：计数器电路设计

一、实验目的

通过本实验,进一步了解典型时序逻辑电路的功能和特点;掌握计数器电路的基本分析方法和设计方法;掌握使用硬件描述语言设计计数器的方法。

1. 通过计数器、分频器等时序电路的设计与测试，掌握计数器电路的基本分析方法和设计方法。
2. 学会使用硬件描述语言编写计数器电路，并利用计数器电路解决实际问题。

二、任务一设计与实现

1. 要求

- (1)用硬件描述语言设计一个与 74LS163 功能一致的计数器。
- (2)清零和置数均为同步方式。
- (3)采用结构化描述方式将计数器 IP 和数码管 IP 相连，将计数结果显示在数码管上。

2. 设计思路

任务一要求设计一个具有清零、置数、加一、保持、进位功能的计数器，并将其用结构化描述方式将其与数码管相连。通过思考发现，用 if 语句很容易将上述计数器功能实现。再用结构化描述方式将其之前写完的数码管相连。



图 1.模 10 计数器设计思路

数字逻辑实验报告

实验 3

3. 详细设计

表 1.计数器功能表

输入					输出					说明
CLRn	LDN	ENP	ENT	CLK	Q _D	Q _C	Q _B	Q _A	RCO	
0	x	x	x	↑	0	0	0	0	0	置零
1	0	x	x	↑	D	C	B	A	*	置数
1	1	1	1	↑	计数				*	加 1
1	1	0	x	x	Q _D	Q _C	Q _B	Q _A	*	保持
1	1	x	0	x	Q _D	Q _C	Q _B	Q _A	0	保持

```
1 module counter(clrn, clk, enp, ent, ldn, data_in, q_out, rco);
2   input clrn, clk, ent, enp, ldn;
3   input [3:0] data_in;
4   output [3:0] q_out;
5   output rco;
6   reg [3:0] q_out;
7   always@(posedge clk)
8   begin
9     if(~clrn)
10      q_out <= 0;
11    else if(!ldn)
12      q_out <= data_in;
13    else if(enp && ent==1)
14      q_out <= q_out + 1;
15    else
16      q_out <= q_out;
17  end
18  assign rco = (q_out==4'b1111 && ent) ? 1:0;
19 endmodule
```

计数器

```
1 module led (out,in,n_en,sel);
2   output [6:0] out;
3   input [3:0] in;
4   input n_en;
5   output sel = 1'b0;
6   reg [6:0] out;
7   always @ (in or n_en)
8   begin
9     if(!n_en)
10      case(in)
11        4'd0:out = 7'b111_1110;
12        4'd1:out = 7'b011_0000;
13        4'd2:out = 7'b110_1101;
14        4'd3:out = 7'b111_1001;
15        4'd4:out = 7'b011_0011;
16        4'd5:out = 7'b101_1011;
17        4'd6:out = 7'b101_1111;
18        4'd7:out = 7'b111_0000;
19        4'd8:out = 7'b111_1111;
20        4'd9:out = 7'b111_1011;
21        default:out = 7'b0000_000;
22      endcase
23    else out = 7'b0000000;
24  end
25 endmodule
```

数码管

```
1 module counter_top(clrn, clk, enp, ent, ldn, data_in, rco ,a,b,c,d,e,f,g,sel,n_en);
2   input clrn, clk, ent, enp, ldn, n_en;
3   input [3:0] data_in;
4   output a,b,c,d,e,f,g, rco;
5   output sel;
6   wire [3:0] y;
7   counter(.data_in(data_in), .clk(clk),.clrn(clrn), .ent(ent), .enp(enp),.ldn(ldn), .q_out(y),.rco(rco));
8   led(.in(y),.out({a,b,c,d,e,f,g}),.sel(sel),.n_en(n_en));
9 endmodule
```

顶层文件

图 2-4.计数器和数码管的代码以及顶层文件的代码

4. 仿真验证

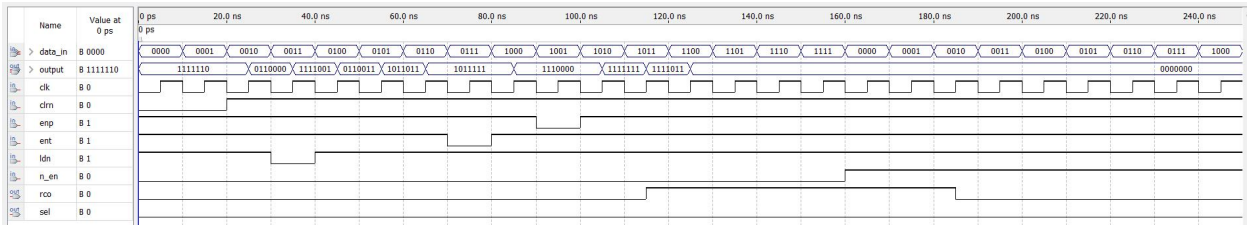


图 5.波形图

由图，一开始计数器因 clrn 为 0 所以无法计数，从 20ns 开始，clrn 置 1，开始计数。可见 35ns 时，写入权限 ldn 激活，clk 又是上升沿，因此 data 的 0011 被写入，数码管显示 3，并且正常计时。在 75ns 的时候 ENT 为 0，因此计数器锁定，85ns 的时候全为 1，clk 上升沿继续计数，90ns 的时候 ENP 为 0，计数器依然是锁定。计数为 9 的时候，rco 变为 1，数码管也按照预期做出了对应的反应。

5. 引脚分配

数字逻辑实验报告

实验 3

表 2.引脚分配

端口名称	输入端									
	使能端 n_en	时钟信号 clk	ENT	ENP	置零 CLRn	读取 LDN	data_in[3]	data_in[2]	data_in[1]	data_in[0]
引脚编号	A3	AB15	E5	C3	AB17	AB18	N18	M20	AA15	V13
平台端口	SW16	F1	SW12	SW11	SW9	SW10	SW1	SW2	SW3	SW4

表 2 续表

输出端								
LED							选位	进位
output[6]	output[5]	output[4]	output[3]	output[2]	output[1]	output[0]	sel	RCO
AA20	W20	R21	P21	N21	N20	M21	V16	-
LA	LB	LC	LD	LE	LF	LG	DS8	-

Node Name	Direction	Location
out a	Output	PIN_AA20
out b	Output	PIN_W20
out c	Output	PIN_R21
in clk	Input	PIN_AB15
in clrn	Input	PIN_AB17
out d	Output	PIN_P21
in data_in[3]	Input	PIN_N18
in data_in[2]	Input	PIN_M20
in data_in[1]	Input	PIN_AA15
in data_in[0]	Input	PIN_V13
out e	Output	PIN_N21
in enp	Input	PIN_C3
in ent	Input	PIN_E5
out f	Output	PIN_N20
out q	Output	PIN_M21
in ldn	Input	PIN_AB18
in n_en	Input	PIN_A3
out rco	Output	
out sel	Output	PIN_V16
<<new node>>		

图 6.引脚锁定

6. 实验现象

按一次按钮，LED 显示的数字就+1，因为没有消除按钮防抖，因此按一次按钮，LED 的数字不一定加 1，甚至还出现了加 9 的偶然现象。但是通过 n_e，clrn，ldn 的功能和波形图，可以验证逻辑电路的设计是没问题的。因为 LED 模块没有考虑 10-15 的 LED 显示，计数器加到 9 之后没有写归零代码，因此计数器加到 9 之后，还要按下 6 次才能到 1。

三、任务二设计与实现

1. 要求

(1) 基准频率为 50MHz。

数字逻辑实验报告

实验 3

- (2)要求从四个输出端分别输出频率 1Hz、10Hz、100Hz 和 1KHz。
- (3)将设计好的分频器 1hz 频率与计数器电路的时钟信号相连,要求采用结构化描述龙式:其它频率接至未用到的 LED 灯。

2. 设计思路

任务要求设计一个拥有一输入四输出的分频器,并将其 1hz 输出端与计数器电路的时钟信号相连。通过思考发现可以用四个 always 语句实现。

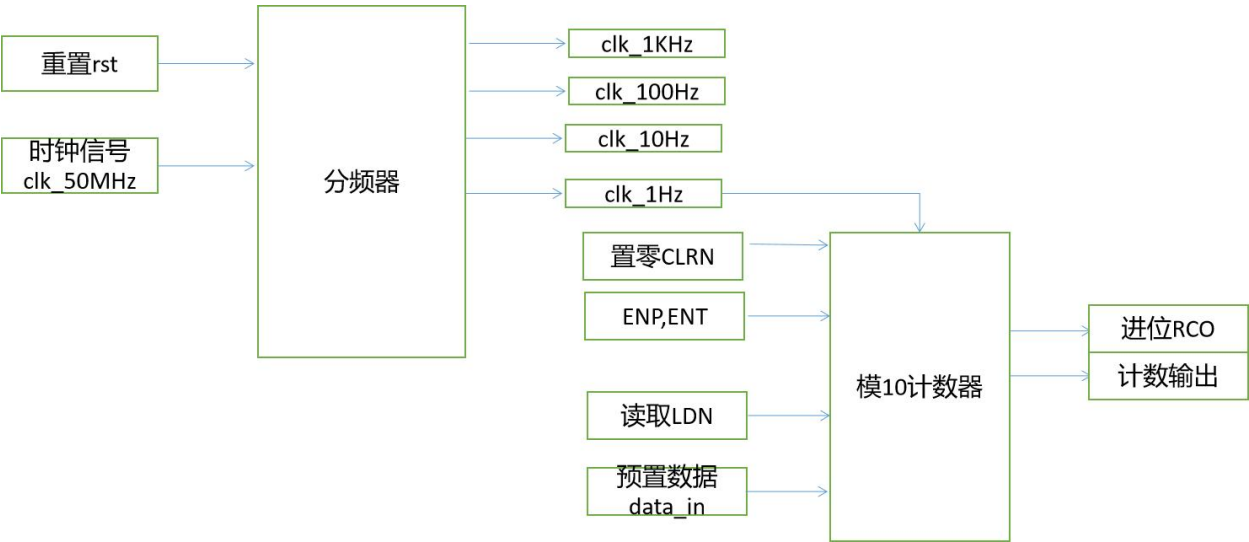


图 7.分频器电路设计思路

3. 详细设计

```
1 module frequency_divider(clk_50mhz,rst,clk_1khz,clk_100hz,clk_10hz,clk_1hz);
2   input clk_50mhz,rst;
3   output clk_1khz,clk_100hz,clk_10hz,clk_1hz;
4   reg clk_1khz,clk_100hz,clk_10hz,clk_1hz;
5   reg [31:0]cnt1,cnt2,cnt3,cnt4;
6   parameter A=50000;
7   parameter B=500000;
8   parameter C=5000000;
9   parameter D=50000000;
10  //parameter A = 4, B = 10, C = 10, D =10;
11  always@(posedge clk_50mhz)
12  begin
13    if(!rst)
14    begin
15      cnt1<=1'b0;
16      clk_1khz<=1'b0;
17    end
18    else
19      if(cnt1<A/2-1)
20        cnt1<=cnt1+1'b1;
21      else
22      begin
23        cnt1<=1'b0;
24        clk_1khz<=~clk_1khz;
25      end
26    end
27    always@(posedge clk_50mhz)
28    begin
29      if(!rst)
30      begin
31        cnt2<=1'b0;
32        clk_100hz<=1'b0;
33      end
34      else
35      begin
36        cnt2<=cnt2+1'b1;
37        if(cnt2<B/2-1)
38          cnt2<=cnt2+1'b1;
39        else
40        begin
41          cnt2<=1'b0;
42          clk_100hz<=~clk_100hz;
43        end
44      end
45      always@(posedge clk_50mhz)
46      begin
47        if(!rst)
48        begin
49          cnt3<=1'b0;
50          clk_10hz<=1'b0;
51        end
52        else
53        begin
54          cnt3<=cnt3+1'b1;
55          if(cnt3<C/2-1)
56            cnt3<=cnt3+1'b1;
57          else
58          begin
59            cnt3<=1'b0;
60            clk_10hz<=~clk_10hz;
61          end
62        end
63        always@(posedge clk_50mhz)
64        begin
65          if(!rst)
66          begin
67            cnt4<=1'b0;
68            clk_1hz<=1'b0;
69          end
70          else
71          begin
72            cnt4<=cnt4+1'b1;
73            if(cnt4<D/2-1)
74              cnt4<=cnt4+1'b1;
75            else
76            begin
77              cnt4<=1'b0;
78              clk_1hz<=~clk_1hz;
79            end
80          end
81        end
82      end
83    end
84  endmodule
```

图 8-10.分频器和顶层文件的代码

4. 仿真验证

数字逻辑实验报告

实验 3

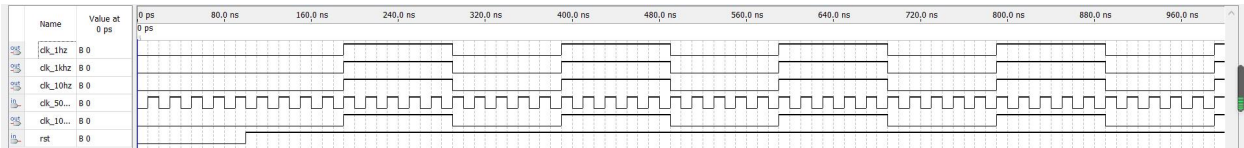


图 11.波形图仿真

为了方便观察代码逻辑的正确与否，因此频率均设置为 10 倍缩放。由图可见，四个 clk 输出，频率确实被缩放了 10 倍，因此验证了代码的准确性。把 parameter 注释掉，重新写为五千万，五百万，五十万，五万，即可按照预期分频了。其中 1Hz 的信号接到计数器的 clk 输入口上。

5. 引脚分配

表 3.引脚分配

端口名称	输入端									
	重置 rst	时钟信号 clk_50mh z	data_in[0]	data_in[1]	data_in[2]	data_in[3]	ENT	ENP	置零 CLR N	读取 LDN
引脚编号	R18	T1	V13	AA15	M20	N18	E5	C3	AB17	AB18
平台端口	F10	T1	SW4	SW3	SW2	SW1	SW1 2	SW1 1	SW9	SW1 0

表 3 续表

输出端							
RCO	clk_1Khz	clk_10hz	clk_100hz	计数输出 q_out[3]	计数输出 q_out[2]	计数输出 q_out[1]	计数输出 q_out[0]
-	U12	V12	V15	W15	Y17	R16	T17
-	LED1	LED2	LED3	LED5	LED6	LED7	LED8

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair
clk_1khz	Output	PIN_U12	4	B4_N1	PIN_U12	2.5 V (default)		8mA (default)	2 (default)	
clk_10hz	Output	PIN_V12	4	B4_N1	PIN_V12	2.5 V (default)		8mA (default)	2 (default)	
clk_50mhz	Input	PIN_T1	2	B2_N0	PIN_T1	2.5 V (default)		8mA (default)		
clk_100hz	Output	PIN_V15	4	B4_N0	PIN_V15	2.5 V (default)		8mA (default)	2 (default)	
clr_n	Input	PIN_AB17	4	B4_N0	PIN_AB17	2.5 V (default)		8mA (default)		
data_in[3]	Input	PIN_N18	5	B5_N0	PIN_N18	2.5 V (default)		8mA (default)		
data_in[2]	Input	PIN_M20	5	B5_N0	PIN_M20	2.5 V (default)		8mA (default)		
data_in[1]	Input	PIN_AA15	4	B4_N1	PIN_AA15	2.5 V (default)		8mA (default)		
data_in[0]	Input	PIN_V13	4	B4_N1	PIN_V13	2.5 V (default)		8mA (default)		
enp	Input	PIN_C3	8	B8_N1	PIN_C3	2.5 V (default)		8mA (default)		
ent	Input	PIN_E5	8	B8_N1	PIN_E5	2.5 V (default)		8mA (default)		
ldn	Input	PIN_AB18	4	B4_N0	PIN_AB18	2.5 V (default)		8mA (default)		
n_en	Input	PIN_A3	8	B8_N1	PIN_A3	2.5 V (default)		8mA (default)		
q_out[3]	Output	PIN_W15	4	B4_N0	PIN_W15	2.5 V (default)		8mA (default)	2 (default)	
q_out[2]	Output	PIN_Y17	4	B4_N0	PIN_Y17	2.5 V (default)		8mA (default)	2 (default)	
q_out[1]	Output	PIN_R16	4	B4_N0	PIN_R16	2.5 V (default)		8mA (default)	2 (default)	
q_out[0]	Output	PIN_T17	5	B5_N1	PIN_T17	2.5 V (default)		8mA (default)	2 (default)	
rco	Output				PIN_T15	2.5 V (default)		8mA (default)	2 (default)	
rst	Input	PIN_R18	5	B5_N0	PIN_R18	2.5 V (default)		8mA (default)		

图 12.引脚锁定

6. 实验现象

本任务，我们将 10Hz,100Hz,1kHz 的信号分别接在了 LED1~3 上，计数器的输出从高位到低位接到了 LED5~8 上。实验开始之后，10Hz 的 LED 开始闪动，1 秒一闪，而另两个 LED 则纹丝不动。观测 LED5~8，它们也 1 秒一变闪动规律，而且通过观察，是按照 0000~1111 按顺序变化，再次表明了电路设计无误。

四、扩展实验

数字逻辑实验报告

实验 3

1.设计思路

我们的想法是，设计一个可以自选频率的分频器，然后通过三个计数器配合，实现模 $10 \times N$ 计数，再通过分别输入三个七段数码管，实现计时器的功能。但是实验进行到引脚分配的时候，发现七段数码管无法分别传入数据，因此我们原本的实验设计（图 14）被迫换为了图 13 所示实验。我们在之后的实验中，会汲取更多经验和思路，来实现我们的想法。

本实验的分频器基于任务二修改而成，由于时间问题，我们的自选频率是一个伪自选：输入三位二进制数，分频器会根据 8 个数字，选择设定好的对应的八个不同的频率。因为高频信号不便于在实验台观察，因此我们只写了两种频率的频选。

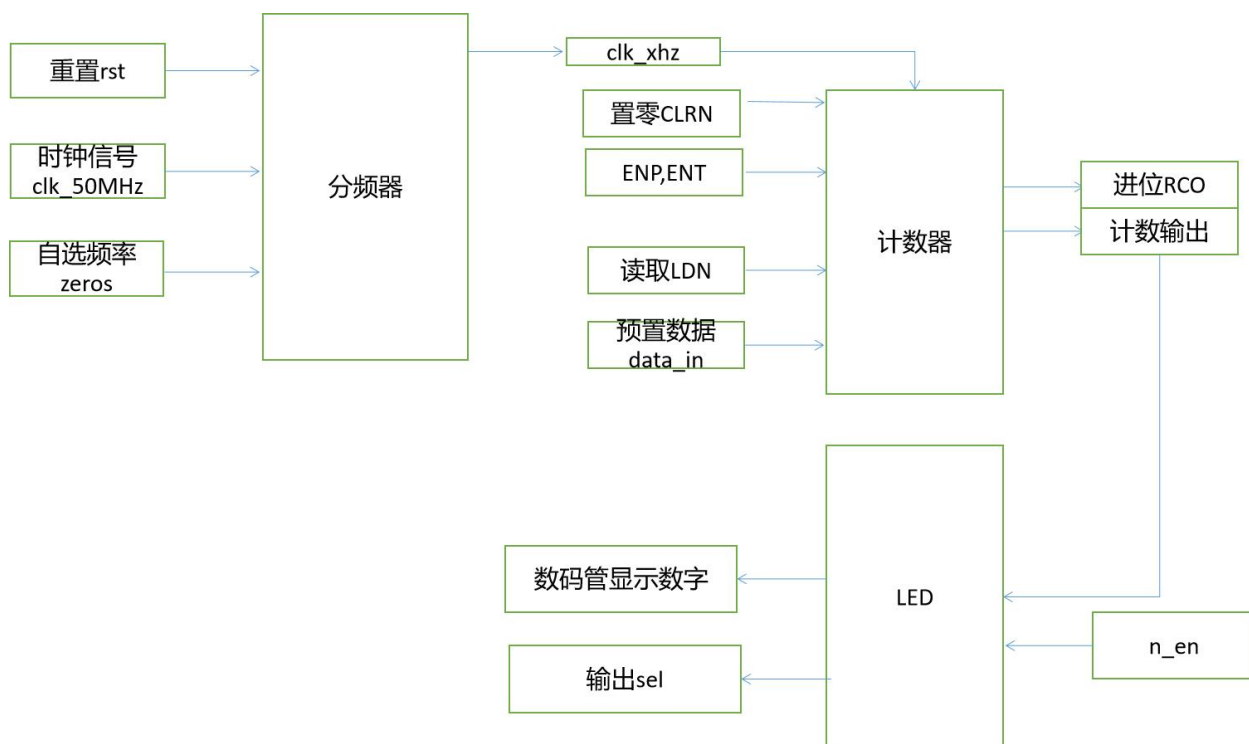


图 13.扩展实验设计思路

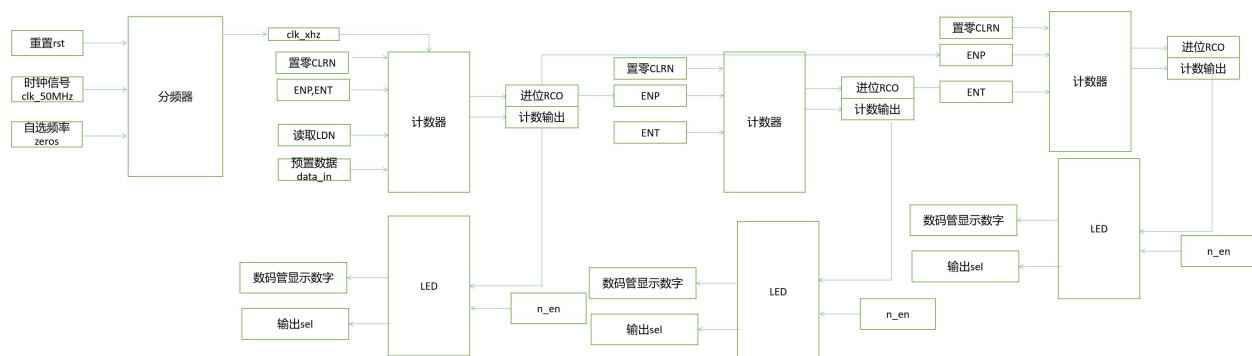


图 14.扩展实验原本的设计思路

2. 详细设计

数字逻辑实验报告

实验 3

```
1 module frequency(clk_50mhz,rst,clk_hz, zeros);
2 input clk_50mhz,rst;
3 input [2:0] zeros;
4 output clk_hz;
5 reg clk_hz;
6 reg [31:0]cnt1;
7
8 always@(posedge clk_50mhz)
9   case(zeros)
10     3'b000:
11       begin
12         if(!rst)
13           begin
14             cnt1<=1'b0;
15             clk_hz<=1'b0;
16           end
17         else
18           if(cnt1< 50000000/2-1)
19             cnt1<=cnt1+1'b1;|
20           else
21             begin
22               cnt1<=1'b0;
23               clk_hz<=~clk_hz;
24             end
25           end
26     3'b001:
27       begin
28         if(!rst)
29           begin
30             cnt1<=1'b0;
31             clk_hz<=1'b0;
32           end
33         else
34           if(cnt1< 2500000/2-1)
35             cnt1<=cnt1+1'b1;
36           else
37             begin
38               cnt1<=1'b0;
39               clk_hz<=~clk_hz;
40             end
41           end
42     default:
43       begin
44         if(!rst)
45           begin
46             cnt1<=1'b0;
47             clk_hz<=1'b0;
48           end
49         else
50           if(cnt1< 50000000/2-1)
51             cnt1<=cnt1+1'b1;
52           else
53             begin
54               cnt1<=1'b0;
55               clk_hz<=~clk_hz;
56             end
57           end
58       endcase
59   endmodule
```

```
1 module expand(clk_50mhz,rst,zeros, clrn, emp, ent, ldn,data_in,rco, q_o, out1,n_en,sel1);
2 input clk_50mhz,rst;
3 input [2:0] zeros;
4 wire clk_xhz;
5
6 input clrn, emp, ent, ldn;
7 input [3:0] data_in;
8 wire [3:0] q_o1, q_o2, q_o3;
9 output rco;
10
11 input n_en;
12 output sel1;
13 output [6:0] out1;
14
15 output q_o;
16
17
18
19 frequency(clk_50mhz(clk_50mhz), .rst(rst), .zeros(zeros), .clk_hz(clk_xhz));
20 counter c1(.clk(clk_xhz),.clrn(clrn), .ldn(ldn),.emp(emp), .ent(ent), .data_in(data_in), .rco(rco), .q_out(q_o1));
21 //counter c2(.clk(clk_xhz),.clrn(clrn), .emp(i), .ent(ent),.data_in(data_in),.rco(rco2),.q_out(q_o2));
22 //counter c3(.clk(clk_xhz),.clrn(clrn), .emp(i), .ent(i),.data_in(data_in), .rco(rco3),.q_out(q_o3));
23 led 11(sel1), .in(q_o1), .out(out1), .n_en(n_en));
24 //led 12(sel2), .in(q_o2), .out(out2), .n_en(n_en));
25 //led 13(sel3), .in(q_o3), .out(out3), .n_en(n_en));
26 assign q_o = clk_xhz;
27 endmodule
```

图 15-17.扩展实验代码

3. 引脚分配

表 4.引脚分配

端口名称	输入端									
	重置 rst	时钟信号 clk_50mh z	data_in[0]	data_in[1]	data_in[2]	data_in[3]	ENT	ENP	置零 CLR N	读取 LDN
引脚编号	F7	T1	V13	AA15	M20	N18	E5	C3	AB17	AB18
平台端口	SW15	T1	SW4	SW3	SW2	SW1	SW12	SW11	SW9	SW10
端口名称	n_en	zeros[0]	zeros[1]	zeros[2]						
引脚编号	A3	F8	E7	C8						
平台端口	SW16	SW8	SW7	SW6						

数字逻辑实验报告

表 4 续表

输出端									
RCO	q_o	sel1	q_out[6]	q_out[5]	q_out[4]	q_out[3]	q_out[2]	q_out[1]	q_out[0]
W15	U12	AB20	AA20	W20	R21	P21	N21	N20	M21
LED 5	LED1	DS1	LA	LB	LC	LD	LE	LF	LG

Node Name	Direction	Location	I/O Bank	VREF Group	Filter Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair
clk_50mhz	Input	PIN_T1	2	B2_N0	PIN_T1	2.5 V (default)		8mA (default)		
clrn	Input	PIN_AB17	4	B4_N0	PIN_AB17	2.5 V (default)		8mA (default)		
data_in[3]	Input	PIN_N18	5	B5_N0	PIN_N18	2.5 V (default)		8mA (default)		
data_in[2]	Input	PIN_M20	5	B5_N0	PIN_M20	2.5 V (default)		8mA (default)		
data_in[1]	Input	PIN_AA15	4	B4_N1	PIN_AA15	2.5 V (default)		8mA (default)		
data_in[0]	Input	PIN_V13	4	B4_N1	PIN_V13	2.5 V (default)		8mA (default)		
enp	Input	PIN_C3	8	B8_N1	PIN_C3	2.5 V (default)		8mA (default)		
ent	Input	PIN_E5	8	B8_N1	PIN_E5	2.5 V (default)		8mA (default)		
ldn	Input	PIN_AB18	4	B4_N0	PIN_AB18	2.5 V (default)		8mA (default)		
n_en	Input	PIN_A3	8	B8_N1	PIN_A3	2.5 V (default)		8mA (default)		
out1[6]	Output	PIN_AA20	4	B4_N0	PIN_AA20	2.5 V (default)		8mA (default)	2 (default)	
out1[5]	Output	PIN_W20	5	B5_N1	PIN_W20	2.5 V (default)		8mA (default)	2 (default)	
out1[4]	Output	PIN_R21	5	B5_N0	PIN_R21	2.5 V (default)		8mA (default)	2 (default)	
out1[3]	Output	PIN_P21	5	B5_N0	PIN_P21	2.5 V (default)		8mA (default)	2 (default)	
out1[2]	Output	PIN_N21	5	B5_N0	PIN_N21	2.5 V (default)		8mA (default)	2 (default)	
out1[1]	Output	PIN_N20	5	B5_N0	PIN_N20	2.5 V (default)		8mA (default)	2 (default)	
out1[0]	Output	PIN_M21	5	B5_N0	PIN_M21	2.5 V (default)		8mA (default)	2 (default)	
q_o	Output	PIN_U12	4	B4_N1	PIN_U12	2.5 V (default)		8mA (default)	2 (default)	
rco	Output	PIN_W15	4	B4_N0	PIN_M5	2.5 V (default)		8mA (default)	2 (default)	
rst	Input	PIN_F7	8	B8_N1	PIN_F7	2.5 V (default)		8mA (default)		
sel1	Output	PIN_AB20	4	B4_N0	PIN_AB20	2.5 V (default)		8mA (default)	2 (default)	
zeros[2]	Input	PIN_C8	8	B8_N0	PIN_C8	2.5 V (default)		8mA (default)		
zeros[1]	Input	PIN_E7	8	B8_N1	PIN_E7	2.5 V (default)		8mA (default)		
zeros[0]	Input	PIN_F8	8	B8_N1	PIN_F8	2.5 V (default)		8mA (default)		

图 18.引脚锁定

4.实验现象

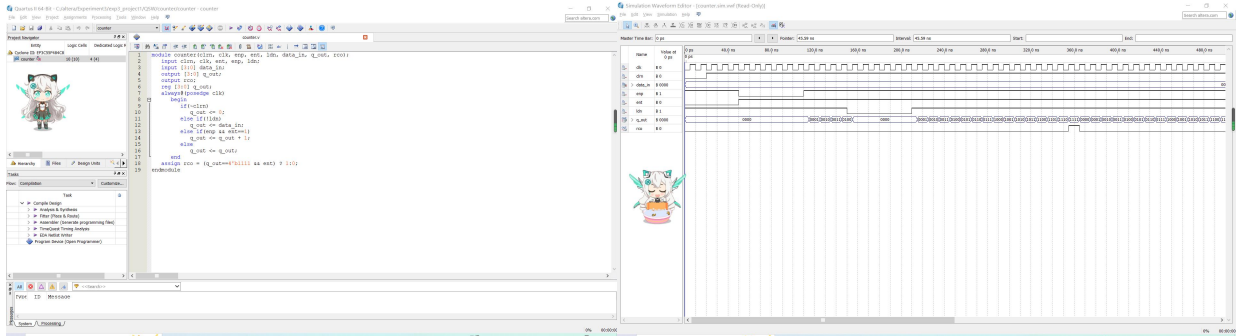
LED 灯的代码添加了 10~15 的 A-F 显示。实验开启的时候，默认为 1Hz，因此计数器一秒一变，当计数到 9 的时候，RCO 接的 LED 灯闪烁。改变频选的时候，七段数码管改变频率明显上升，RCO 小灯也如此。ENP,ENT，ldn 等功能无误。

五、总结

收获：

遇到的问题或现象

- ①实验一中，出现了按一次按钮，计数器自加若干次的现象，这是因为没有解决键盘抖动的问题。
- ②实验二中，由于不熟悉代码模块调用，导致我没有更改工程绑定的“.v 文件”，导致卡了很久。好在及时发现了问题，进行了改正，顺利完成实验
- ③实验二中，100Hz 和 1000Hz 的 LED 没有闪烁，这是因为闪烁频率太高，有可能人眼观测不到闪动，或者实验台为了保护电路，设置了闪动频率阈值。
- ④扩展实验中，我们本来想设计七段数码管计时器，但是引脚锁定的时候发现，八个七段数码管只能输入同一组数据，因此没有实现预期功能，也说明了我们相关方面的知识仍不足。



数字逻辑实验报告

实验 3

图 1-2 任务一计数器代码和波形图

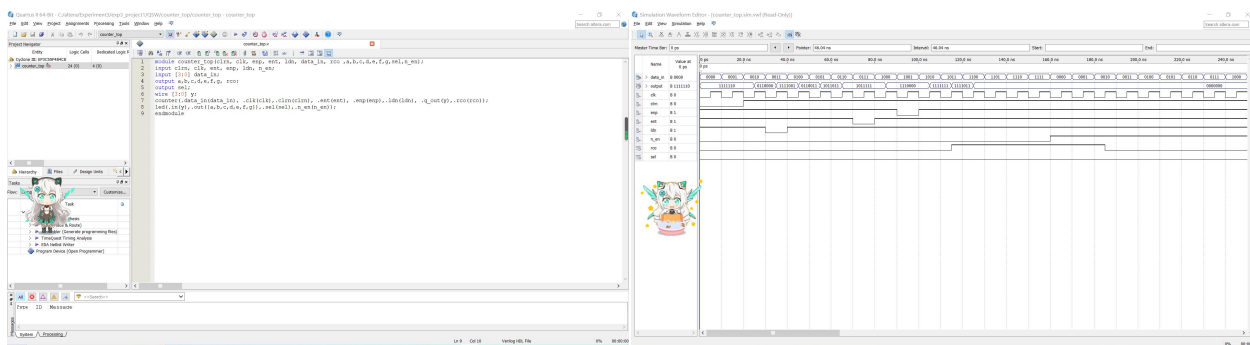


图 3-4.任务一顶层文件及波形图

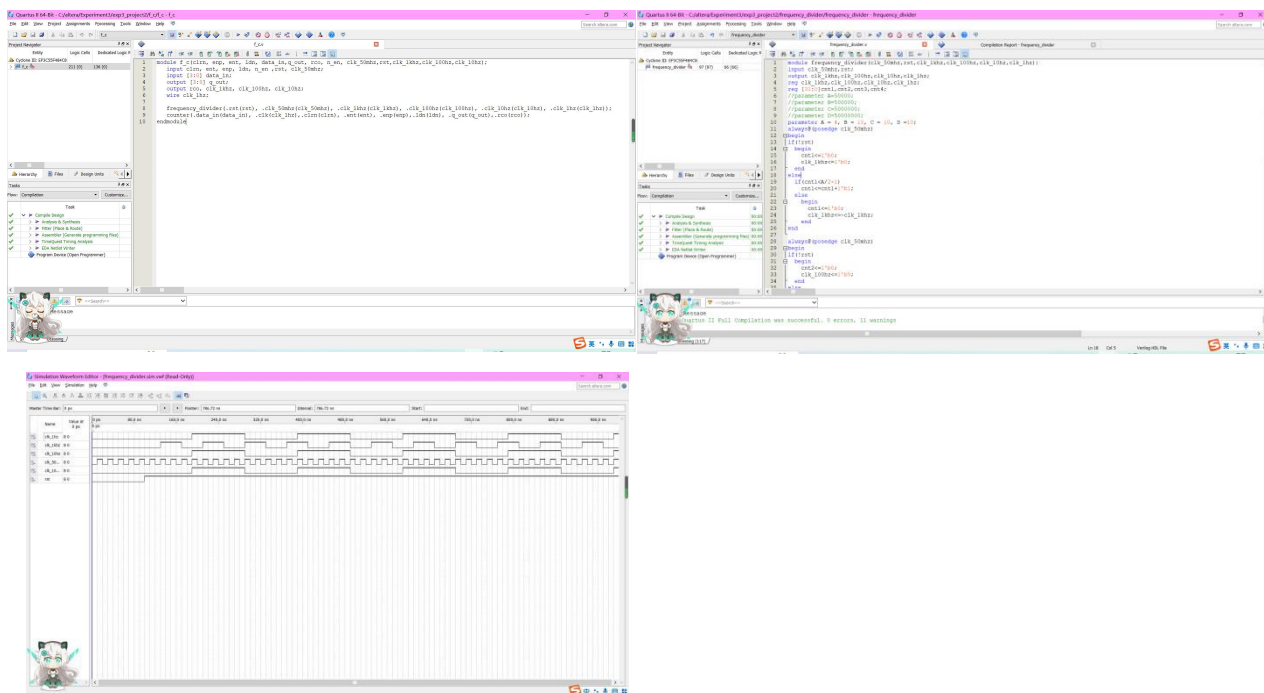


图 5-7 任务二分频器代码和波形图，以及顶层文件代码

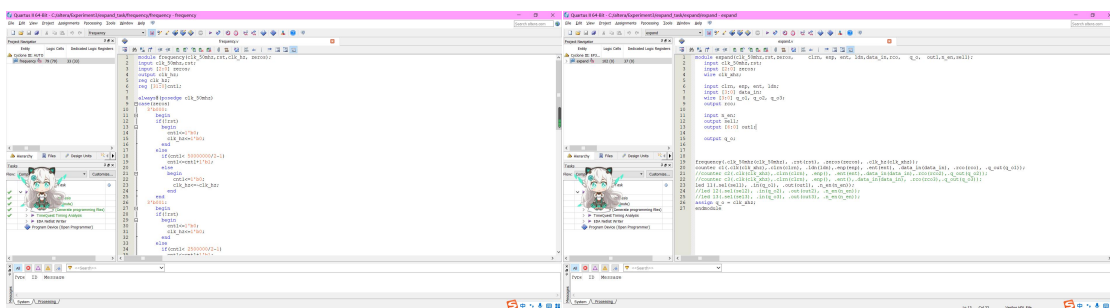


图 8-9.扩展任务频选分频器代码和顶层文件代码