



电子电路综合设计

直接数字频率合成器（DDFS）

作者 : 许晓明 学号 : 9161040G0734

学院 : 电子工程与光电技术学院

专业 : 电子信息工程

班级 : 9161042103

题目 : 电子电路综合设计:

直接数字频率合成器（DDFS）

指导者 : 谭雪琴

2018 年 12 月

摘要

直接数字频率合成器是从相位概念出发直接合成所需要波形的一种新的频率合成技术。本实验用quartus II设计了一个直接数字频率合成器（DDFS）。通过仿真、调试，最后下载到SmartSOPC实验系统上验证功能。

本报告分析了 DDFS 设计要求说明，整体电路工作原理的方案论证，并说明了各子模块设计原理以及后期工作中的调试、仿真、编程下载的过程。

报告中的DDFS直接数字频率合成器，可根据需要选择合适的频率字输出正弦，余弦，三角波，方波及锯齿波。频率字和相位字采用在数码管上显示计数的方式输入，计数范围为0-39（为4位测频可显示的最大值），具有移相功能。具有测频和显示功能（显示4位十进制数），能在数码管上显示当前频率以供和示波器测得的频率比较。利用正弦函数波形对称性特点，能明显节省ROM空间，提高波形精度。并在原电路的基础上进行了AM调制。

设计过程中，充分使用了VHDL语言灵活、简便的特点，大多数模块均使用VHDL语言设计，大大降低了电路设计的工作量，提高了模块的准确性。

关键词：QuartusII DDFS 测频 VHDL 仿真

Abstract

The direct digital frequency synthesizer is a new frequency synthesis technique that directly synthesizes the required waveforms from the phase concept. In this experiment, a direct digital frequency synthesizer (DDFS) was designed using quartus II. Through simulation, debugging, and finally downloaded to the verification function on the SmartSOPC experimental system.

This report analyzes the DDFS design requirements, the overall circuit working principle, and explains the design principles of each sub-module and the debugging, simulation, and programming download process in the later work.

The DDFS direct digital frequency synthesizer in the report can select the appropriate frequency word to output sine, cosine, triangle, square and sawtooth waves as needed. The frequency word and the phase word are input by displaying the count on the digital tube, and the counting range is 0-39 (which is the maximum value that can be displayed by the 4-bit frequency measurement), and has a phase shifting function. With frequency measurement and display function (displaying 4-digit decimal number), the current frequency can be displayed on the digital tube for comparison with the frequency measured by the oscilloscope. By using the symmetry characteristic of the sinusoidal function waveform, the ROM space can be obviously saved and the waveform precision can be improved. And AM modulation is performed on the basis of the original circuit.

In the design process, the VHDL language is fully utilized to be flexible and simple. Most modules are designed using VHDL language, which greatly reduces the workload of circuit design and improves the accuracy of the module.

key words: QuartusII, DDFS, Frequency measurement, VHDL, simulation

目录

1	设计要求说明	1
1.1	设计基本要求	1
1.2	设计提高部分要求	1
2	方案论证（整体电路的工作原理）	2
2.1	频率预置与调节电路	2
2.2	累加器	2
2.3	波形存储器	3
2.4	D/A转换器	3
2.5	低通滤波器	3
3	各子模块设计原理	5
3.1	脉冲发生电路	5
3.1.1	VHDL分频原理	5
3.1.2	分频模块	5
3.2	频率字、相位字产生电路	7
3.2.1	电路原理	7
3.2.2	模40模块	7
3.3	BCD码转二进制电路	8
3.3.1	电路作用	8
3.3.2	BCD码转二进制模块	8
3.4	累加器	10
3.4.1	加法器模块	10
3.4.2	寄存器模块	11
3.5	波形存储器与波形选择电路	12
3.5.1	波形文件的生成	12
3.5.2	波形选择模块	14
3.6	消颤电路	14
3.6.1	消颤电路原理	14

3.6.2 消颤模块	14
3.7 测频电路	14
3.7.1 测频原理	14
3.7.2 测频模块	14
3.8 译码显示电路	18
3.8.1 芯片搭建的译码显示电路原理	18
3.8.2 译码显示模块	18
3.9 节省ROM	18
3.9.1 取反电路	18
3.9.2 节省rom模块	21
3.10 AM调制	21
3.10.1 AM调制原理	21
3.10.2 AM模块	22
4 直接数字频率合成器总电路	24
5 调试、仿真、编程下载	24
6 结论	25
7 实验感想	27
7.1 实验过程中遇到的问题及解决问题的方法	27
7.2 实验的收获与感受	27
7.3 期望及要求	28
参考文献	29

1 设计要求说明

设计一个频率及相位均可控制的具有正弦和余弦输出的直接数字频率合成器 (Direct Digital Frequency Synthesizer 简称DDS或DDFS)。

1.1 设计基本要求

时钟的基本设计具体要求如下：

- 1、 利用QuartusII软件和SmartSOPC实验箱实现DDFS的设计；
- 2、 DDFS中的波形存储器模块用Altera公司的CycloneIII系列 FPGA芯片中的RAM实现， RAM结构配置成 $2^{12} \times 10$ 类型；
- 3、 具体参数要求：频率控制字K取4位；基准频率 $f_c = 1MHz$,由实验板上的系统时钟分频得到；
- 4、 系统具有使能功能；
- 5、 利用实验箱上的D/A转换器件将ROM输出的数字信号转换为模拟信号，能够通过示波器观察到正、余弦两路波形；
- 6、 通过开关（实验箱上的 K_i ）输入DDFS的频率和相位控制字，并能用示波器观察加以验证；

1.2 设计提高部分要求

完成基本设计要求后，可选做提高性要求。实验中我做的功能有：

- 1、 通过按键（实验箱上的Si）输入DDFS的频率和相位控制字，以扩大频率控制和相位控制的范围；(注意：按键后有消颤电路)
- 2、 在数码管上显示生成的波形频率；
- 3、 设计能输出多种波形（三角波、锯齿波、方波等）的多功能波形发生器；
- 4、 充分考虑ROM结构及正弦函数的特点，进行合理的配置，提高计算精度；
- 5、 基于DDFS的AM调制器的设计；

2 方案论证（整体电路的工作原理）

DDFS的组成大致如图1。

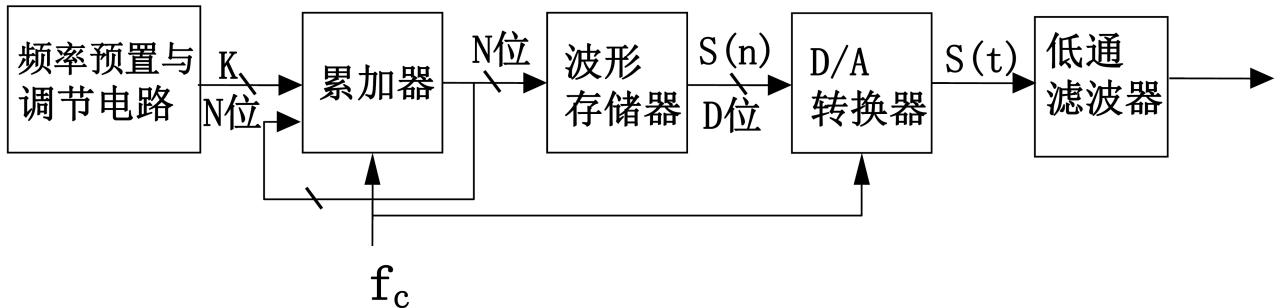


图 1

2.1 频率预置与调节电路

DDFS的输出频率为

$$f_0 = \frac{f_c K}{2^N}$$

f_0 为输出频率， f_c 为时钟频率。当 $K = 1$ 时，DDFS输出最低频率（也即频率分辨率）为 $\frac{f_c}{2^N}$ ；由Nyquist采样定理，DDFS的最大输出频率为 $\frac{f_c}{2}$ ，也就是说 K 的最大值为 2^{N-1} 。

因此，只要 N 足够大，DDFS可以得到很细的频率间隔。要改变DDFS的输出频率，只要改变频率控制字 K 即可。

2.2 累加器

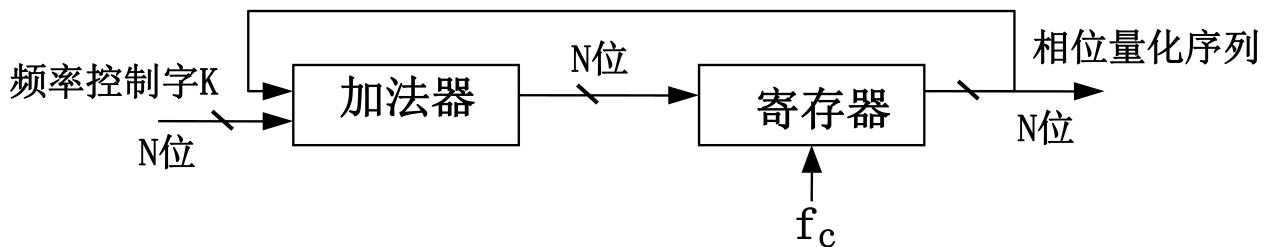


图 2

如图2，累加器由N位加法器与N位寄存器级联构成。每收到一个时钟脉冲 f_C ，加法器便将频率控制字 K 与寄存器输出的累加相位数据相加，再把相加

后的结果送至寄存器的数据输入端。寄存器将加法器在上一个时钟作用后所产生的相位数据反馈到加法器的输入端；使加法器在下一个时钟作用下继续与频率控制字进行相加。这样，相位累加器在时钟的作用下，进行相位累加。

当相位累加器累加满量时就会产生一次溢出，完成一个周期性的动作，这个周期应为

$$T_o = \frac{2^N}{K} T_c$$

2.3 波形存储器



图 3

如图3，将累加器输出的数据作为波形存储器的地址，进行波形的相位-幅值转换，可在给定的时间上确定输出波形的幅值。

N位的寻址ROM相当于把周期的波形信号离散成具有 2^N 个样值的序列，若波形ROM有D位数据位，则 2^N 个样值的幅值以D位二进制数值存在ROM中，按照地址的不同可以输出相应相位的波形信号的幅值。

2.4 D/A转换器

如图4，D/A转换器的作用是把已经合成的周期波的数字量转换成模拟量。波形幅度量化序列S(n)经D/A转换后变成了包络为周期波的阶梯波S(t)，S(t)的周期为

$$T = \frac{2^N}{K} T_C$$

2.5 低通滤波器

如图5，低通滤波器的作用是滤除生成的阶梯形正弦波中的高频成分，将其变成光滑的正弦波。

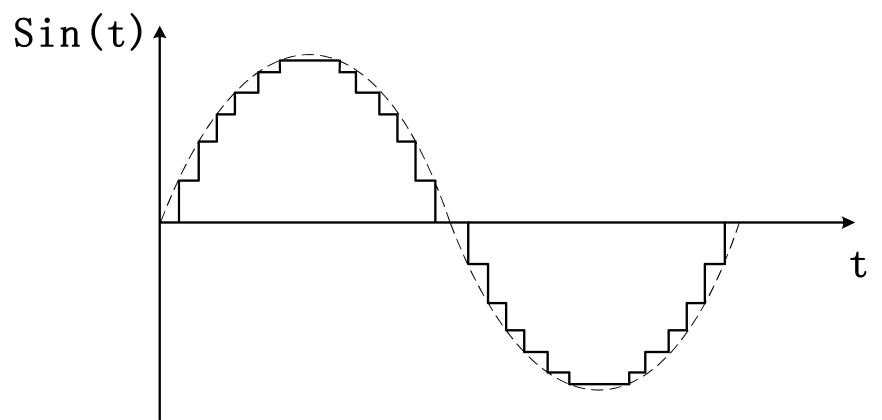


图 4

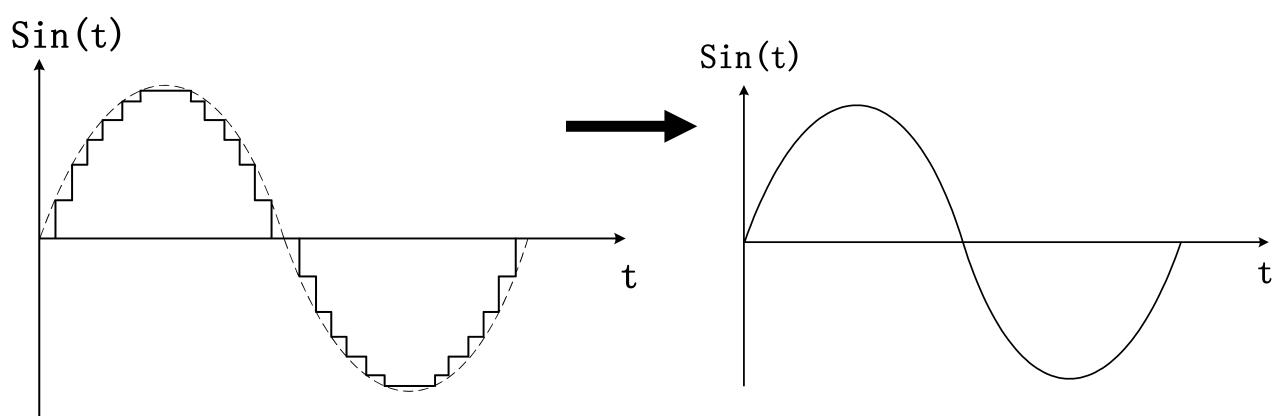


图 5

3 各子模块设计原理

3.1 脉冲发生电路

通过分频，将实验箱提供的48MHz的高频信号通过分频器变成所需的1Hz、0.5Hz和1KHz的信号，0.5Hz用于测频电路，1Hz用于计数，2Hz用于消颤，1KHz用于译码显示电路。

3.1.1 VHDL分频原理

对时钟信号进行上升沿检测，并记录检测的数（从0开始计数）。如果是N分频，每当计数到 $\frac{N}{2} - 1$ 时将信号翻转一次，将信号输出。

3.1.2 分频模块

相应的VHDL代码如下，由于要将48MHz分频出1Hz需要的截止时间过长，而分频原理其实是一致的，因此只要1MHz的分频结果正确，就可以认为分频无误，其仿真波形见图6，封装图见图7。

```
1 LIBRARY IEEE;
2 USE IEEE.STD_LOGIC_1164.ALL;
3 ENTITY fenpin2 IS
4 PORT( clk :IN STD_LOGIC;
5 mhz:buffer STD_LOGIC;
6 hz1:buffer STD_LOGIC;
7 hz0_5:buffer STD_LOGIC;
8 hz2:buffer STD_LOGIC;
9 khz:buffer STD_LOGIC);
10 END fenpin2;
11 ARCHITECTURE beh OF fenpin2 IS
12 SIGNAL count:integer range 0 to 24;
13 SIGNAL count1:integer range 0 to 24000000;
14 SIGNAL count2:integer range 0 to 48000000;
15 SIGNAL count3:integer range 0 to 12000000;
```

```

16 SIGNAL count4:integer range 0 to 24000;
17 BEGIN
18 PROCESS(clk)
19 BEGIN
20 IF (clk = '1')THEN
21 count<=count+1;
22 count1<=count1+1;
23 IF (count=23)THEN
24 count<=0;
25 mhz<=NOT mhz ;
26 END IF ;
27 IF (count1=23999999)THEN
28 count1<=0;
29 hz1<=NOT hz1 ;
30 END IF ;
31 IF (count2=47999999)THEN
32 count2<=0;
33 hz0_5<=NOT hz0_5 ;
34 END IF ;
35 IF (count3=11999999)THEN
36 count3<=0;
37 hz2<=NOT hz2 ;
38 END IF ;
39 IF (count3=11999)THEN
40 count4<=0;
41 khz<=NOT khz ;
42 END IF ;
43 END IF ;
44 END PROCESS;
45 END beh ;

```

3.2 频率字、相位字产生电路

3.2.1 电路原理

频率预置与调节电路的主要作用是实现频率控制量的输入，不变量K被称为相位增量，也叫频率控制字。DDFS的输出频率表达式为

$$f_{out} = K \frac{f_c}{2^N}$$

当K=1时，输出最低频率为 $\frac{f_c}{2^N}$ ，而DDFS的最高输出频率由Nyquist采样定理决定，即 $f_{out} = \frac{f_c}{2}$ 。由于后续还需要测频并用4位显示，而4位可显示的最大频率对应的K值为40，于是我设计频率控制字K为一个模40的计数器。

相位控制字和频率控制字在原理上没有区别，因此用同一个模块。

3.2.2 模40模块

相应的VHDL代码如下，其仿真波形见图8，封装图见图9。

```
1 library ieee;
2 use ieee.std_logic_1164.all;
3 use ieee.std_logic_unsigned.all;
4 ENTITY mmo40 IS
5 PORT
6   ( en    :IN std_logic;
7     clear:IN std_logic;
8     clk    :IN std_logic;
9     cout   :OUT std_logic;
10    qh    :buffer std_logic_vector(3 downto 0);
11    ql    :buffer std_logic_vector(3 downto 0)
12  );
13 END mmo40;
14 ARCHITECTURE behave OF mmo40 IS
15 BEGIN
16 cout <='1'when(qh="0011"and ql="1001"and en='0')else '0';
17
```

```

18  PROCESS( clk , clear )
19    BEGIN
20      IF ( clear = '1' )THEN
21          qh<= " 0 0 0 0 ";
22          ql<= " 0 0 0 0 ";
23      ELSIF ( clk 'EVENT AND clk = '1' )THEN
24          if ( en = '0' )then
25              if (( ql=9 and qh=3) or ql=9)
26                  then
27                      ql<= " 0 0 0 0 ";
28                      if ( qh=3)then
29                          qh<= " 0 0 0 0 ";
30                          else
31                              qh<=qh+1;
32                          end if ;
33                      else
34                          ql<=ql+1;
35                      end if ;
36                  end if ;
37      END IF ;
38  END PROCESS;
END behave ;

```

3.3 BCD码转二进制电路

3.3.1 电路作用

由于BCD码无法直接用乘法器或加法器进行运算，因此需将BCD转二进制，使频率字、相位字得出的BCD码转换为二进制。

3.3.2 BCD码转二进制模块

相应的VHDL代码如下，封装图见图10。

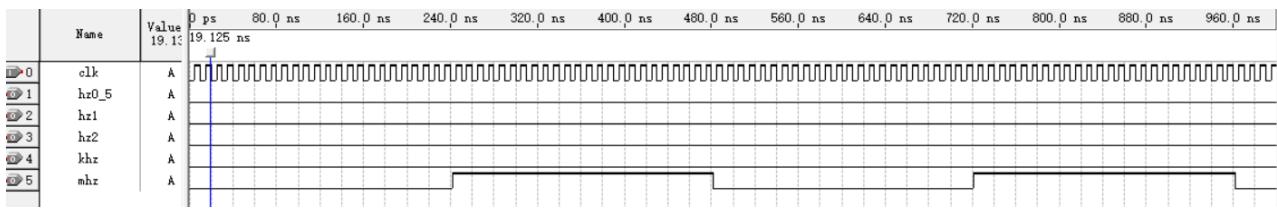


图 6

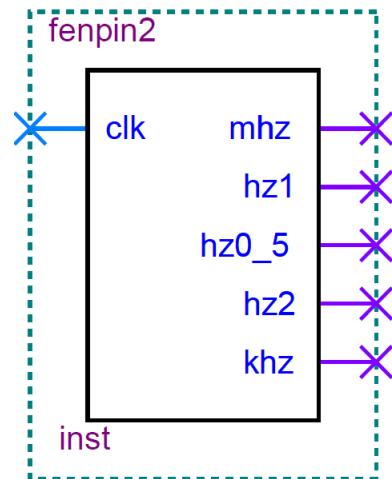


图 7

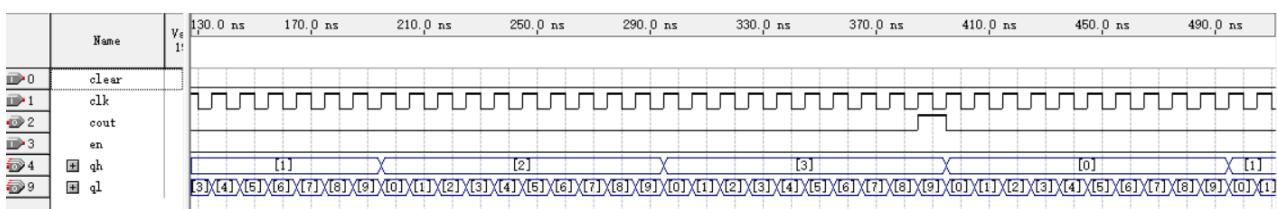


图 8

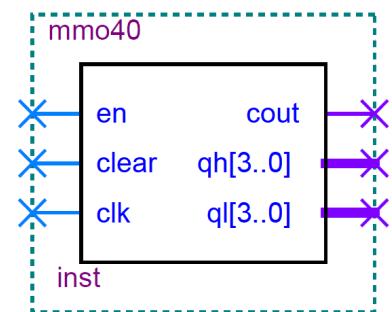


图 9

```

1 entity BCD is
2 port(d0,d1:in integer range 0 to 9;
3 binary:out integer range 0 to 63);
4 end BCD;
5 architecture one of BCD is
6 begin
7 binary<=d0+10*d1;
8 end one;

```

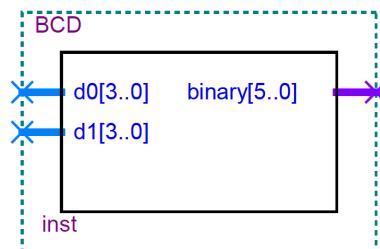


图 10

3.4 累加器

累加器由N位加法器N位寄存器构成。

如图2，其作用是，每接收一个时钟clk，加法器就将频率控制字与累加寄存器输出的累加相位数据相加，相加的结果又反馈送至累加寄存器的数据输入端，以使加法器在下一个时钟脉冲的作用下继续与频率控制字相加。这样，相位累加器在时钟作用下，不断对频率控制字进行线性相位累加。

3.4.1 加法器模块

相应的VHDL代码如下，其仿真波形见图11，封装图见图12。

```

1 LIBRARY ieee;
2 USE ieee.std_logic_1164.ALL;
3 USE ieee.std_logic_unsigned.ALL;
4
5 ENTITY adder12 IS

```

```

6 PORT(A:IN std_logic_vector(11 DOWNTO 0);
7     K:IN std_logic_vector(5 DOWNTO 0);
8     S:OUT std_logic_vector(11 DOWNTO 0));
9
10 END adder12;
11
12 ARCHITECTURE done OF adder12 IS
13 signal sum:std_logic_vector(11 downto 0);
14 BEGIN
15 sum<=K+A;
16 s<=sum;
17 END done;

```

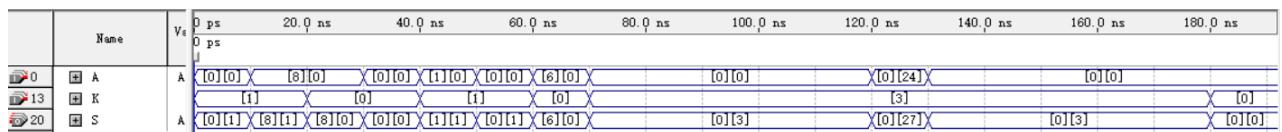


图 11

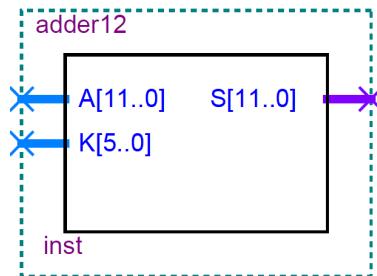


图 12

3.4.2 寄存器模块

相应的VHDL代码如下，其封装图见图13。

```

1 LIBRARY ieee;
2 USE ieee.std_logic_1164.ALL;
3 USE ieee.std_logic_unsigned.ALL;

```

```

4
5 ENTITY register_12 IS
6 PORT(D:IN std_logic_vector(11 DOWNTO 0);
7      CLK:IN std_logic;
8      Q:OUT std_logic_vector(11 DOWNTO 0));
9
10 END register_12;
11
12 ARCHITECTURE register_12_arch OF register_12 IS
13 BEGIN
14   PROCESS(CLK)
15   BEGIN
16     IF (CLK'EVENT AND CLK='1') THEN
17       Q<=D;
18     END IF;
19   END PROCESS;
20 END register_12_arch;

```

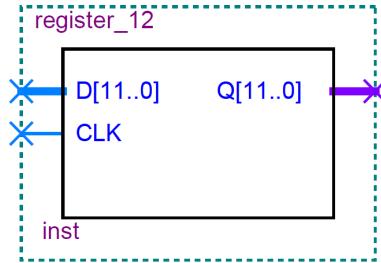


图 13

3.5 波形存储器与波形选择电路

3.5.1 波形文件的生成

利用EXCEL的函数，可产生正弦函数值，计算要产生4096个数据所需要的角频率和幅值，再利用ROUND函数取整，代码为

$$= ROUND((SIN(2 * PI())/4096 * (\$A1)) + 1) * 511, 0)$$

，就可以产生离散的数据，部分数据见表1，将EXCEL的数据复制到txt文档，替换表格间空格，再更改文件名，可以得到mif文件。

同理，可得到其他波形的mif文件。

利用lpm_rom进行波形封装，得到正余弦、方波、锯齿波、三角波的ROM封装文件，类似图14。

表 1

0	:	511	;
1	:	512	;
2	:	513	;
3	:	513	;
4	:	514	;
5	:	515	;
6	:	516	;
7	:	516	;
8	:	517	;
9	:	518	;
10	:	519	;
11	:	520	;
12	:	520	;
13	:	521	;
14	:	522	;
15	:	523	;
16	:	524	;
17	:	524	;
18	:	525	;
19	:	526	;
20	:	527	;
21	:	527	;
22	:	528	;
23	:	529	;
24	:	530	;

3.5.2 波形选择模块

通过ipm_mux元件构成波形选择电路，原理电路见图15，封装图见图16。

3.6 消颤电路

由于开关的接通和关闭有可能产生毛刺，为了消除开关带来的竞争冒险，需要引入消颤模块。

3.6.1 消颤电路原理

通过一个D触发器对开关信号进行延时，从而得到稳定可靠的开关信号。

3.6.2 消颤模块

原理电路见图17，封装图见图18。

3.7 测频电路

3.7.1 测频原理

测频就是计算1秒钟内脉冲的个数。我们利用计数器和锁存器实现这一功能。由于累加器以频率控制字K为间隔，从0-4095计数，当累加器满量时就会产生一次溢出，完成一次周期性的动作，这个周期也就是DDFS信号的一个频率周期，因此，将累加器的最高位作为测频电路技术器的脉冲。

如图19，将0.5Hz脉冲送入锁存器的时钟端，0.5Hz反相延时后的脉冲送入计数器的清零端。这样就使计数器在2s的脉冲周期内，1s内清零，1s内计数。由于锁存器的脉冲和计数器的脉冲是反相的，且有一定的延时，所以当锁存器有效脉冲来到时，计数器是清零状态，锁存器就锁存前1s内计数器的计数信号。这样就完成了1s内的脉冲计数，再将锁存器的输出送入译码显示电路，就可以在数码管上显示波形频率了。

3.7.2 测频模块

测频电路的原理电路见图20，封装图见图21。

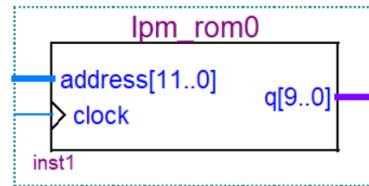


图 14

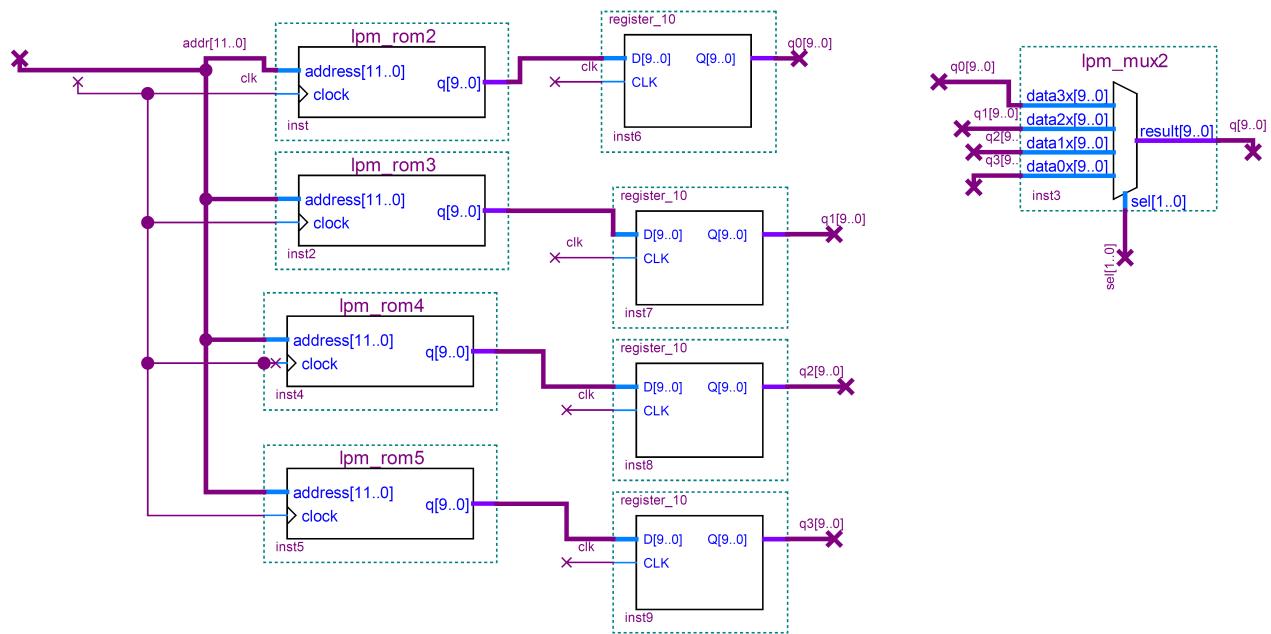


图 15

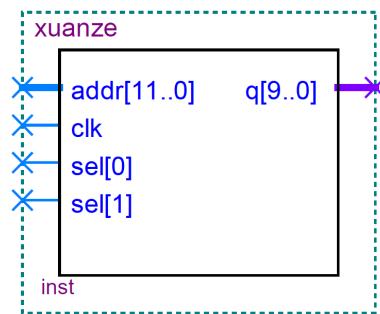


图 16

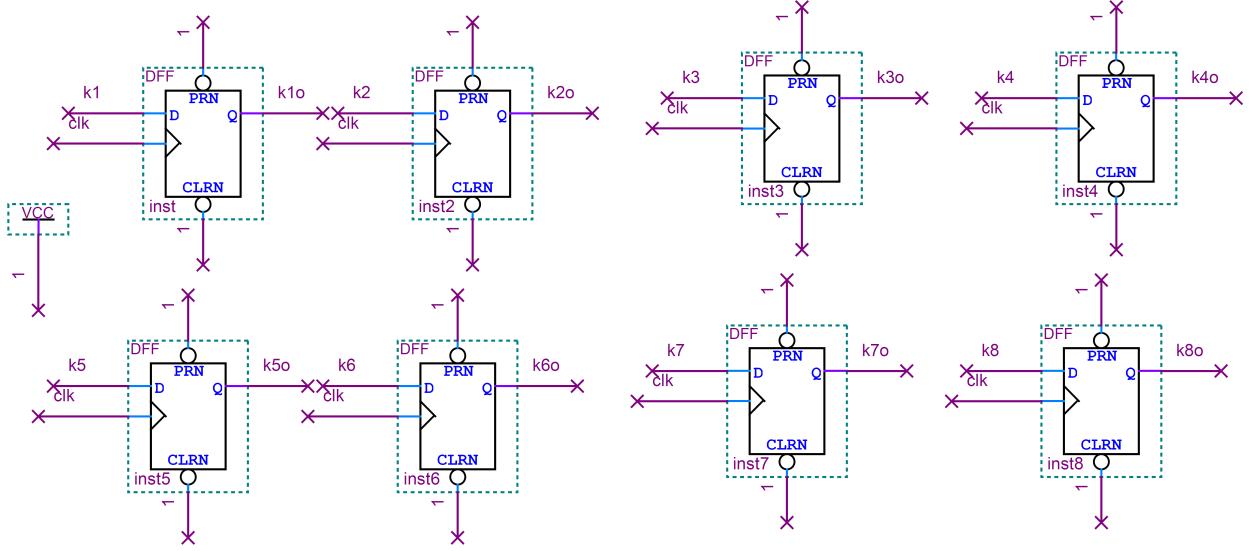


图 17

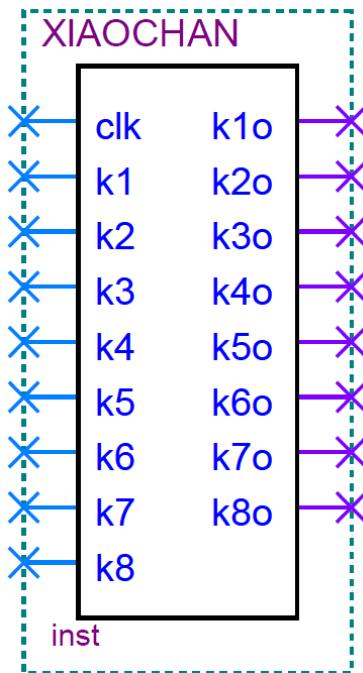


图 18

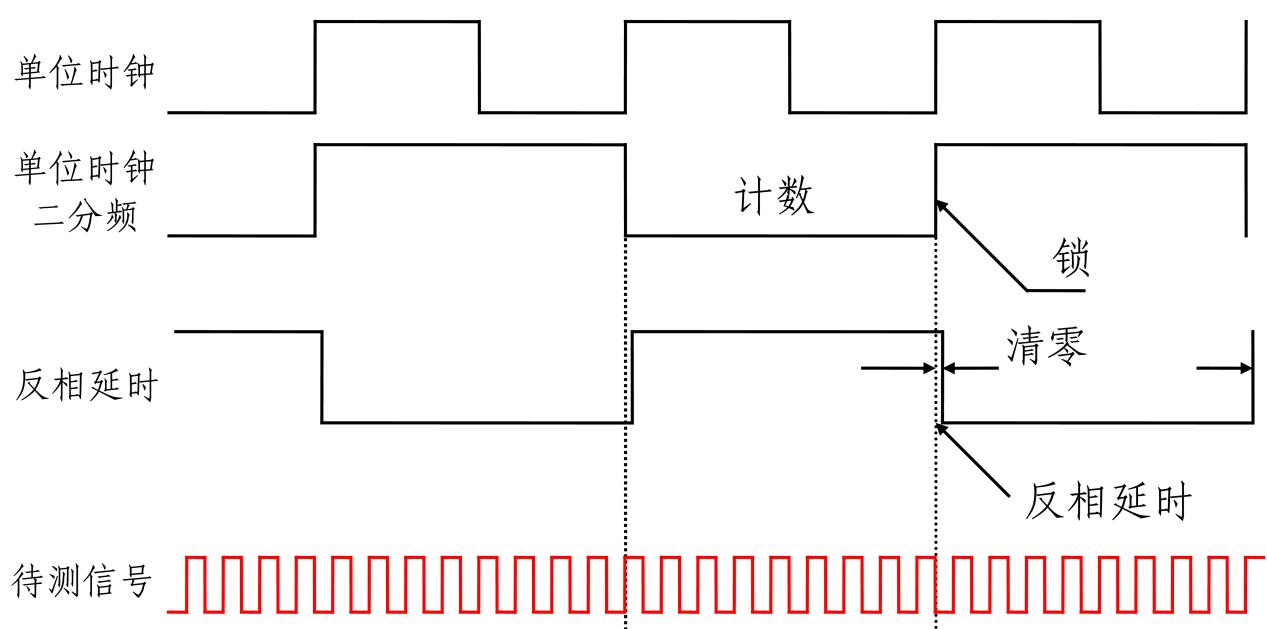


图 19

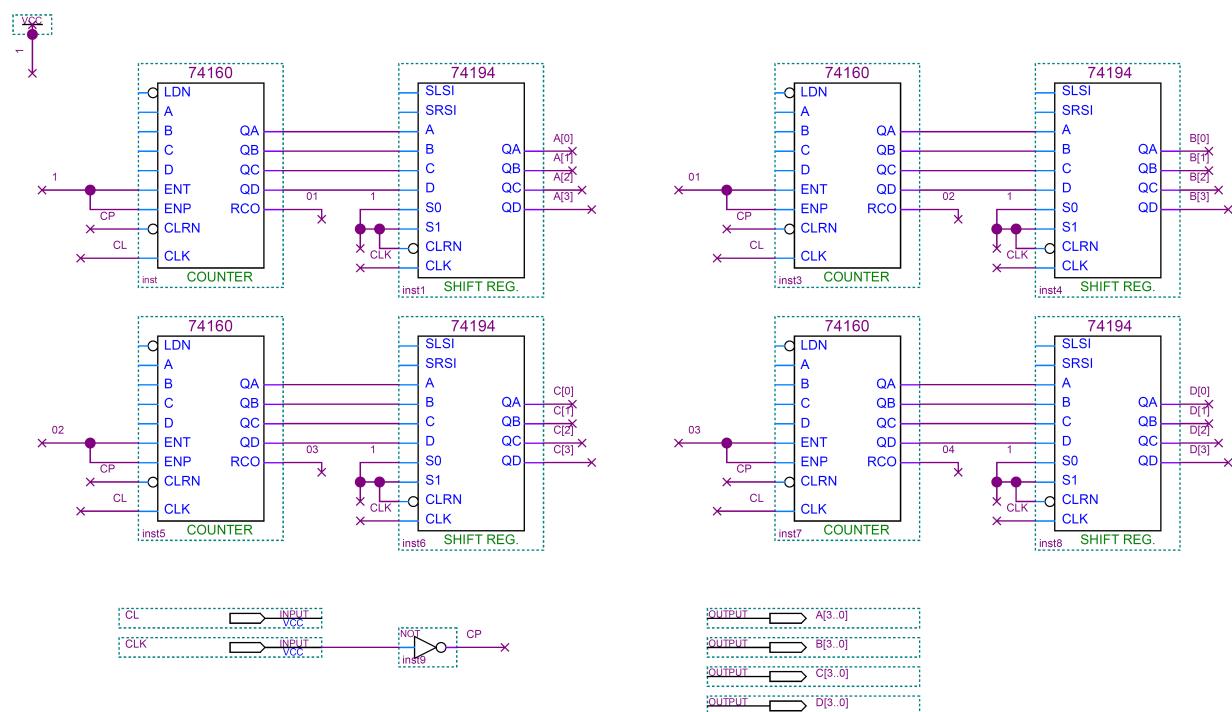


图 20

3.8 译码显示电路

这一部分在EDA(II)中我是采用代码来完成的，但与同学交流后发现，使用原理图来搭建反而思路更清晰，因此在这里我尝试重新用原理图来完成译码显示部分。

3.8.1 芯片搭建的译码显示电路原理

显示译码电路用于数码管的动态显示，该实验需要8个显示管参与显示（2位频率控制字、2位相位控制字、4位测频显示），因此采用计数器74161设计模8循环计数；其输出作为4片74151的控制端，又作为3—8译码器74138的控制端。

模8计数器的输出端a[0]、a[1]、a[2]连接到四片74151地址端，通过74151同时选取对应位进行输出，组成计时器某一位的BCD编码并接入显示译码器7447。

与此同时，根据接入端的a[0]、a[1]、a[2]，74138译码器也从8个数码管的使能端选择对应位有效，从而在对应数码管上显现一个有效数据。

扫描的频率为1KHz,根据人眼的视觉暂留现象，会使人感觉到8个数码管在同时显示。

3.8.2 译码显示模块

译码显示电路的原理电路见图22，封装图见图23。

3.9 节省ROM

在波形存储器中，所用的ROM有12位的地址线和10位的数据线，ROM中共有4096个数值。

但是正弦函数具有对称性，可以用 $\frac{1}{4}$ 个周期的波形就能完整地表现出完整周期的波形：第二个 $\frac{1}{4}$ 周期与第一个 $\frac{1}{4}$ 周期波形是相位相反的关系，第三个 $\frac{1}{4}$ 周期与第一个 $\frac{1}{4}$ 周期波形是函数值相反的关系，第四个 $\frac{1}{4}$ 周期与第一个 $\frac{1}{4}$ 周期波形则是相位、函数值同时反相的关系。其关系如图24所示。

3.9.1 取反电路

借助ipm_inv元件实现取反，电路原理见图25，封装图见图26。

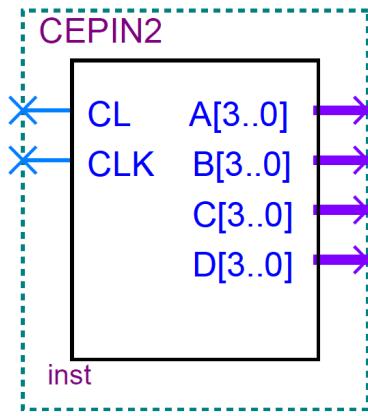


图 21

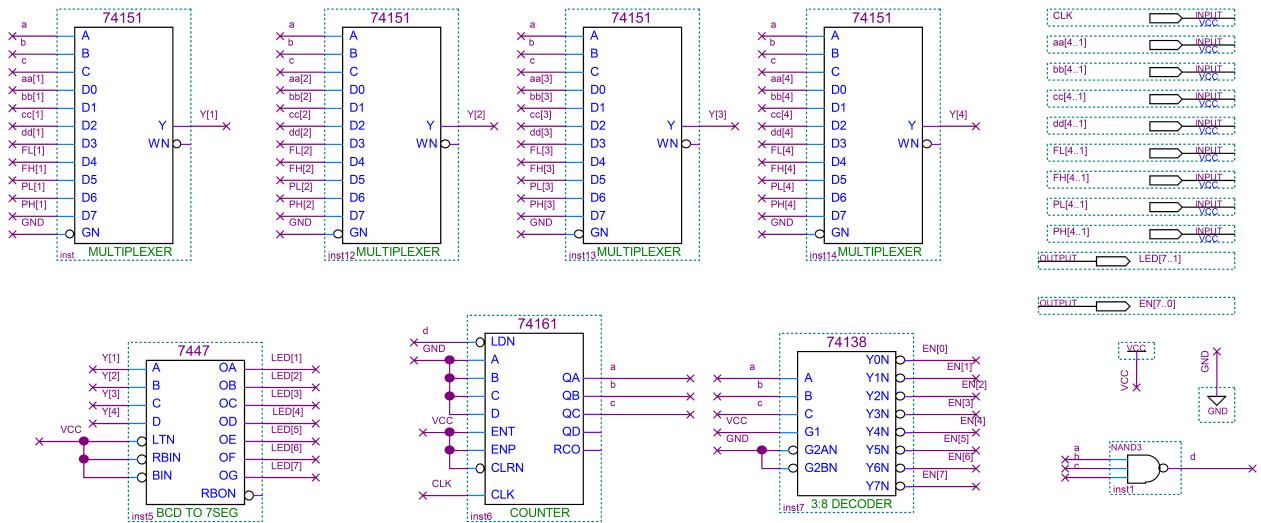


图 22

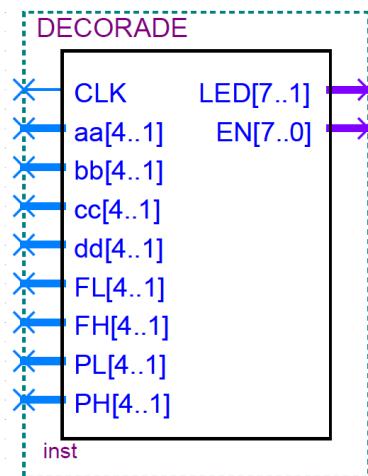


图 23

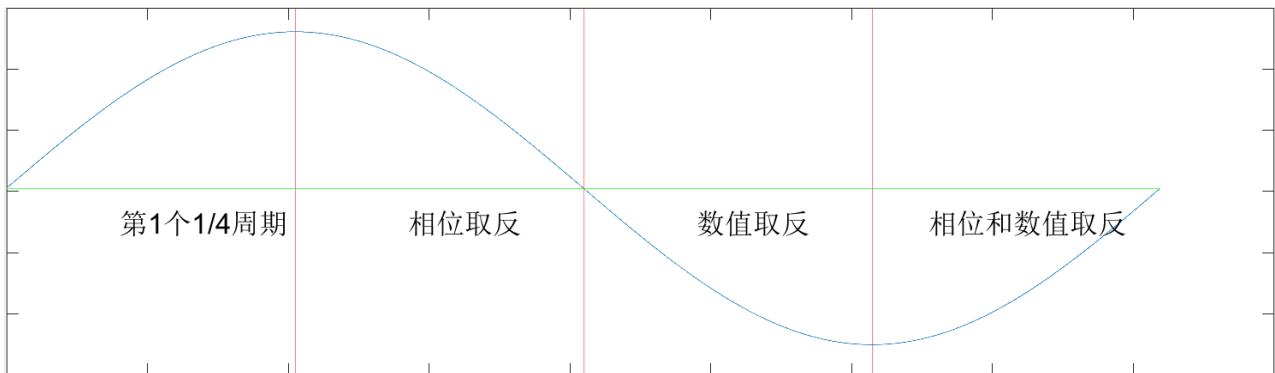


图 24

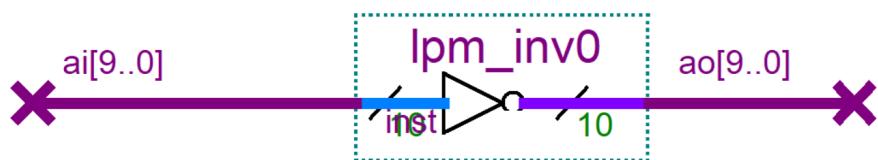


图 25

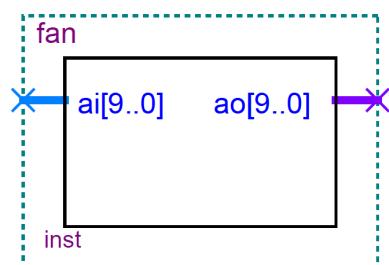


图 26

3.9.2 节省rom模块

运用取反电路实现地址取反与输出取反，rom模块的电路原理见图27，封装图见图28。

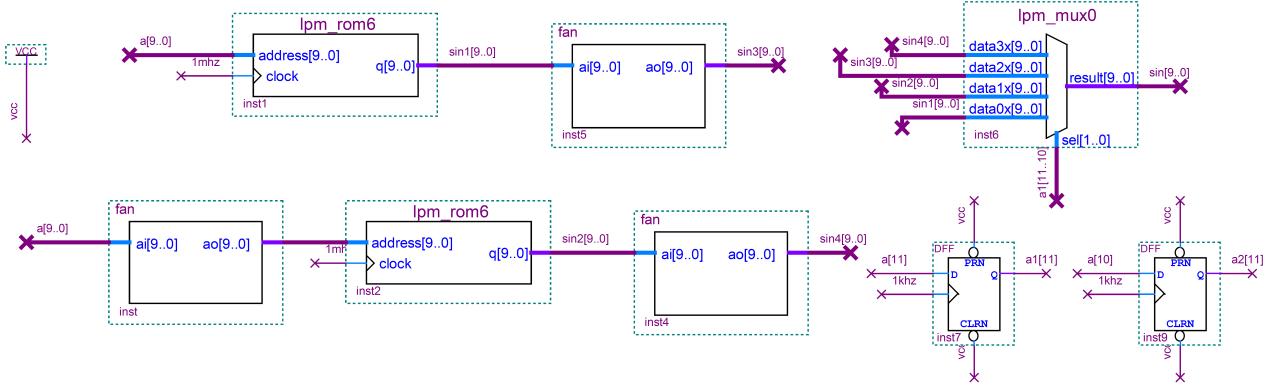


图 27

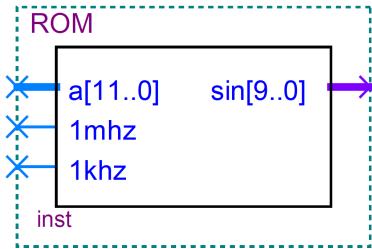


图 28

3.10 AM调制

信号调制是调制信号对载波的幅度、频率和相位进行变换。AM即标准调制信号，除了来自消息的基带信号外，还包含了直流信号，它是调制后输出信号既含载波分量又含边带分量的标准调幅信号。

3.10.1 AM调制原理

如图29，在标准幅度调制器（AM）中，设载波信号为： $v_c(t) = V_{cm} \cos \omega_c t$ ，调制信号为： $v_f(t) = V_{\Omega m} \cos \Omega t$ 。则标准调幅波信号为：

$$v_{AM}(t) = [V_{cm} + v_f(t)] \cos \omega_c t = (V_{cm} + V_{\Omega m} \cos \Omega t) \cos \omega_c t = V_{cm}(1 + m_A \cos \Omega t) \cos \omega_c t$$

其中, $m_A = \frac{V_{\Omega m}}{V_{cm}}$ 被称为调幅度, 是调幅信号的一个重要的参数, 一般小于1, 当 m_A 大于1时会出现过调幅。

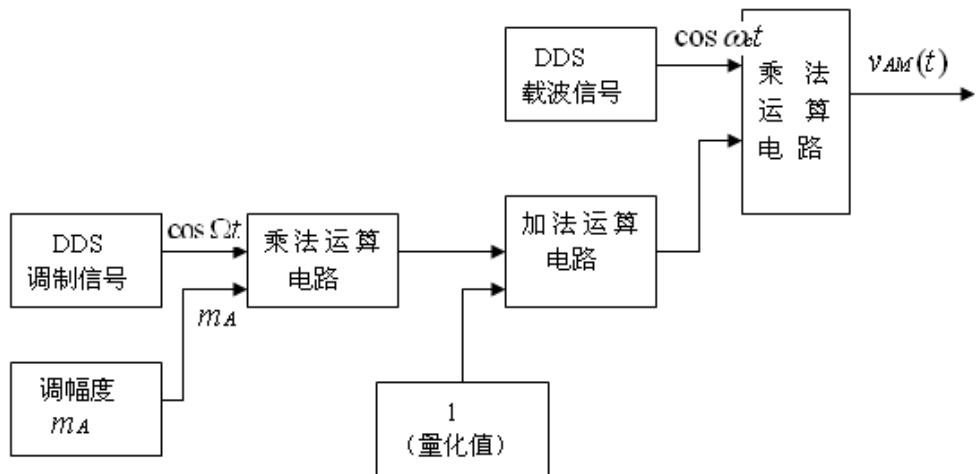


图 29

3.10.2 AM模块

AM调制电路原理见图30, 封装图见图31。

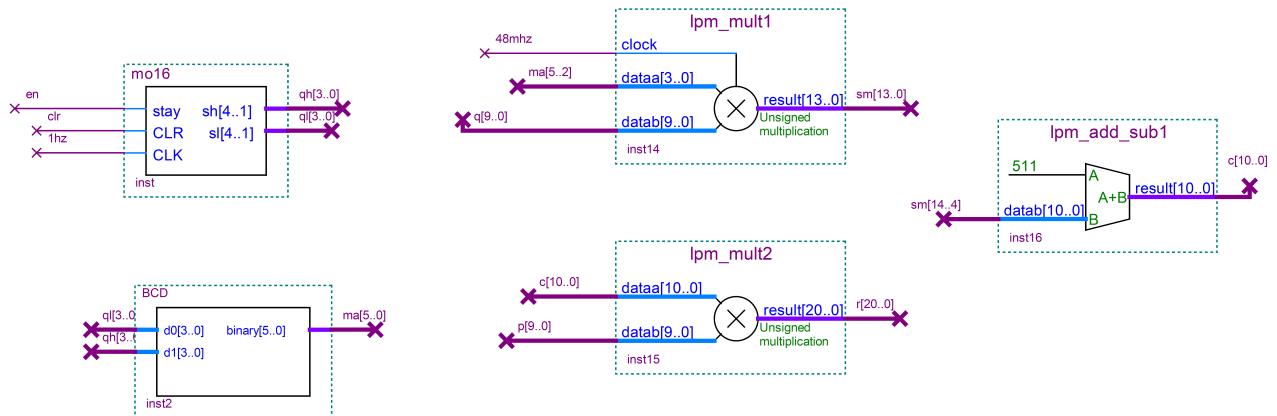


图 30

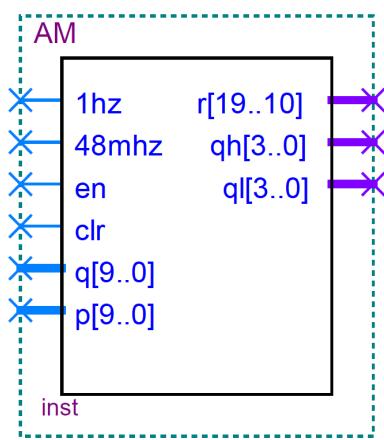


图 31

4 直接数字频率合成器总电路

将以上几个模块相连，借助 IPM_MUX 实现显示复用，通过逻辑门的组合实现开关复用，直接数字频率合成器总电路如图32。

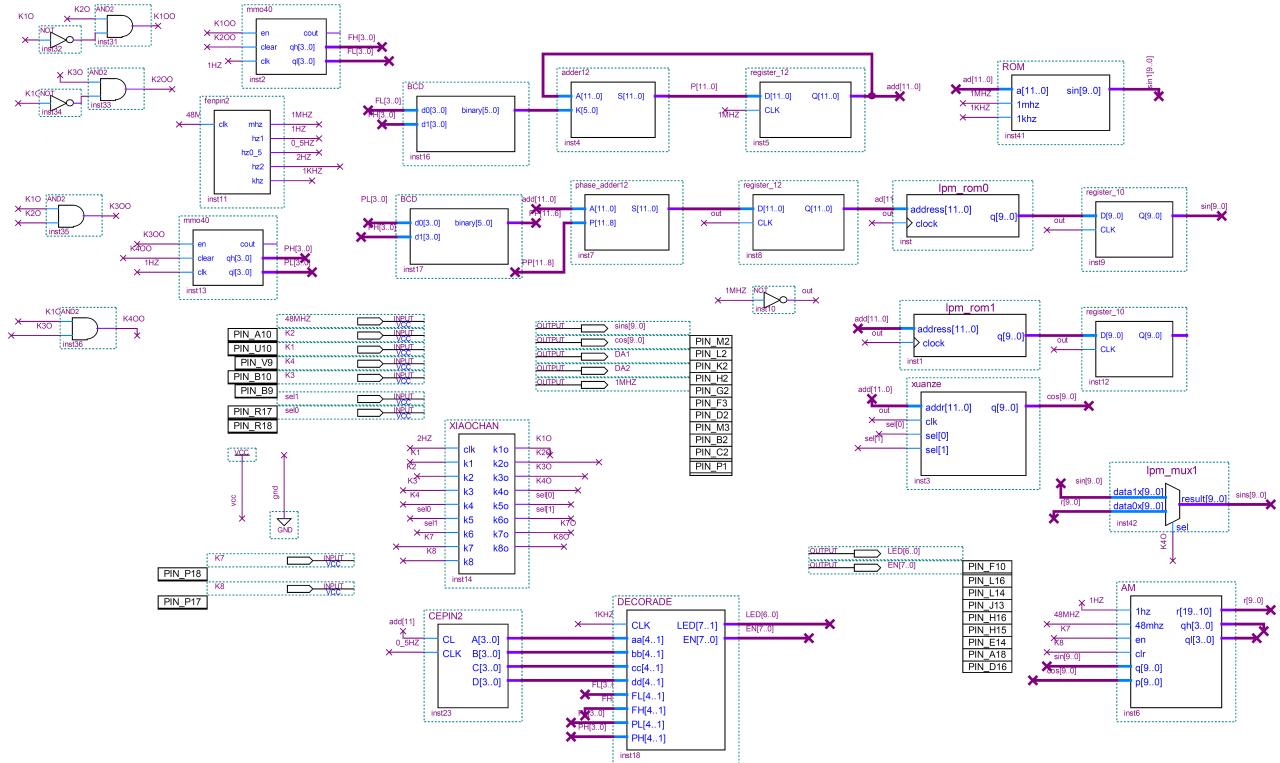


图 32

5 调试、仿真、编程下载

每个模块设计好之后都需要进行调试。调试可以帮助我们避免一些常见的小错误，如漏掉加电源VCC或接地GND等，或标注时不小心标注错误引脚不存在等。这样可以避免出现电路全搭完了但是结果出不来，不知道问题具体在哪的情况出现。这是最基本的检查，只能检查出一些语法运用的错误。

然后还要对其进行波形仿真验证，来检查电路设计的正确与否，直至调试的仿真图完全正确为止，仿真得到的波形图在各个模块的论证时已经给出。

波形仿真完成后，就要下载到实验箱上进行验收。下载方法比较繁琐，通常要设置一些程序的初始值、使能端等，但这是必须的一步，否则可能烧坏实

验箱上的某些原件，造成较大的损失。

将编译好的程序管脚分配，然后下载到实验系统中进行调试和验证。其中管脚分配如图33。

	Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved
1	1MHZ	Output	PIN_A2	8	B8_N0	2.5 V (default)	
2	48MHZ	Input	PIN_A10	7	B7_N0	2.5 V (default)	
3	cos[9]	Output	PIN_P2	2	B2_N0	2.5 V (default)	
4	cos[8]	Output	PIN_F7	8	B8_N0	2.5 V (default)	
5	cos[7]	Output	PIN_C1	1	B1_N0	2.5 V (default)	
6	cos[6]	Output	PIN_E1	1	B1_N0	2.5 V (default)	
7	cos[5]	Output	PIN_G1	1	B1_N0	2.5 V (default)	
8	cos[4]	Output	PIN_H1	1	B1_N0	2.5 V (default)	
9	cos[3]	Output	PIN_K1	2	B2_N0	2.5 V (default)	
10	cos[2]	Output	PIN_L1	2	B2_N0	2.5 V (default)	
11	cos[1]	Output	PIN_M1	2	B2_N0	2.5 V (default)	
12	cos[0]	Output	PIN_P1	2	B2_N0	2.5 V (default)	
13	DA1	Output	PIN_F6	8	B8_N0	2.5 V (default)	
14	DA2	Output	PIN_H6	1	B1_N0	2.5 V (default)	
15	EN[7]	Output	PIN_L16	5	B5_N0	2.5 V (default)	
16	EN[6]	Output	PIN_L14	5	B5_N0	2.5 V (default)	
17	EN[5]	Output	PIN_J13	6	B6_N0	2.5 V (default)	
18	EN[4]	Output	PIN_H16	6	B6_N0	2.5 V (default)	
19	EN[3]	Output	PIN_D16	7	B7_N0	2.5 V (default)	
20	EN[2]	Output	PIN_A18	7	B7_N0	2.5 V (default)	
21	EN[1]	Output	PIN_E14	7	B7_N0	2.5 V (default)	
22	EN[0]	Output	PIN_H15	6	B6_N0	2.5 V (default)	
23	K1	Input	PIN_V9	3	B3_N0	2.5 V (default)	
24	K2	Input	PIN_U10	4	B4_N0	2.5 V (default)	
25	K3	Input	PIN_B9	8	B8_N0	2.5 V (default)	
26	K4	Input	PIN_B10	7	B7_N0	2.5 V (default)	
27	K7	Input	PIN_P18	5	B5_N0	2.5 V (default)	
28	K8	Input	PIN_P17	5	B5_N0	2.5 V (default)	
29	LED[6]	Output	PIN_C14	7	B7_N0	2.5 V (default)	
30	LED[5]	Output	PIN_E13	7	B7_N0	2.5 V (default)	
31	LED[4]	Output	PIN_D12	7	B7_N0	2.5 V (default)	
32	LED[3]	Output	PIN_C12	7	B7_N0	2.5 V (default)	
33	LED[2]	Output	PIN_F12	7	B7_N0	2.5 V (default)	
34	LED[1]	Output	PIN_E11	7	B7_N0	2.5 V (default)	
35	LED[0]	Output	PIN_F10	7	B7_N0	2.5 V (default)	
36	sel0	Input	PIN_R18	5	B5_N0	2.5 V (default)	
37	sel1	Input	PIN_R17	5	B5_N0	2.5 V (default)	
38	sins[9]	Output	PIN_M2	2	B2_N0	2.5 V (default)	
39	sins[8]	Output	PIN_L2	2	B2_N0	2.5 V (default)	
40	sins[7]	Output	PIN_K2	2	B2_N0	2.5 V (default)	
41	sins[6]	Output	PIN_H2	1	B1_N0	2.5 V (default)	
42	sins[5]	Output	PIN_G2	1	B1_N0	2.5 V (default)	
43	sins[4]	Output	PIN_F3	1	B1_N0	2.5 V (default)	
44	sins[3]	Output	PIN_D2	1	B1_N0	2.5 V (default)	
45	sins[2]	Output	PIN_M3	2	B2_N0	2.5 V (default)	
46	sins[1]	Output	PIN_B2	1	B1_N0	2.5 V (default)	
47	sins[0]	Output	PIN_C2	1	B1_N0	2.5 V (default)	
48	<<new node>>						

图 33

6 结论

运行后可实现的功能有：

- 1、开关全部置0时，频率控制字、相位控制字以1-39计数的方式输入，每秒自增，同时显示在数码管的左4位，前两位是频率控制字，后两位是相位

控制字。示波器双路输出，共用频率控制字，相位控制字只对第一路起作用。第一路为用节省ROM实现的正弦波；第二路可由用户设定，开关全部置零时为方波。测频模块自动测量频率，并显示在数码管的右四位。

- 2、 K_1, K_2, K_3 组合起来进行频率控制字、相位控制字的使能与清零。具体实现如下：
 - (1) K_1 是频率控制字/相位控制字选择开关： $K_1 = 0$ 时， K_2, K_3 负责频率控制字； $K_1 = 1$ 时， K_2, K_3 负责相位控制字；
 - (2) K_2 是使能开关： $K_2 = 1$ 时，频率控制字/相位控制字计数暂停； $K_2 = 0$ 时，频率控制字/相位控制字计数继续；
 - (3) K_3 是清零开关： $K_3 = 1$ 时，频率控制字/相位控制字计数清零； $K_3 = 0$ 时，频率控制字/相位控制字计数正常计数。
- 3、 K_5, K_6 是波形选择开关：当 K_5, K_6 对应的二进制码为00时，输出方波；为01时，输出锯齿波；为10时，输出三角波；为11时，输出余弦波；
- 4、 K_4 是AM 调制开关：当 $K_4 = 0$ 时，示波器第一路显示正弦波；当 $K_4 = 1$ 时，示波器第一路显示调幅AM波，同时 K_7, K_8 开始起作用；
- 5、 K_7 是AM调制模块的调幅度控制开关：当AM调制开启， $K_7 = 0$ 时，AM调幅波的调幅度以16为周期，逐渐增大，又回到最小（通过模16计数器实现）； $K_7 = 1$ 时，AM调幅波的调幅度保持不变；
- 6、 K_8 是AM调制模块的调幅度清零开关：当AM调制开启， $K_8 = 0$ 时，AM调幅波的调幅度正常计数； $K_8 = 1$ 时，AM调幅波的调幅度清零；
- 7、测频电路能准确、稳定工作，当调整频率控制字稳定后，数码管能显示当前波形的频率，与示波器上所测得的频率基本一致，只有很小的误差。
- 8、用节省ROM模块替换原ROM模块后，示波器上能检测到准确的正弦函数波形，相比较之前的ROM，改进后的ROM精度更高。

7 实验感想

7.1 实验过程中遇到的问题及解决问题的方法

实验过程中遇到的问题主要有：

- 1、在仿真总分频器时，由于采用了代码来实现，无法像上次一样去掉某一个分频来产生变化的波形了。我一度想用代码编各个分频器然后组合。经过同学的提醒明白了，因为各个频率的分频代码是一样的，只要保证分出1Mhz的分频没问题（这也是不要用长截止时间就可以仿真出来的），就可以认为分频没有问题。最终采用了这种方式验证。
- 2、在设计测频电路的时候，原本我是想用VHDL语言进行设计的，但设计的过程并不顺利，并且因此浪费了很多时间，最后只能选择用电路图的方法搭建测频电路。
- 3、在下载到实验箱运行时也出现了问题，再次出现了实验板上没有任何显示的情况。有了上次的经验，我一开始就检查管脚的分配，发现又是管脚分配有问题，改正这个错误后，就显示出来了。
- 4、做节省ROM时有毛刺出现，毛刺出现的原因是在输出 $\frac{\pi}{2} - \pi$ 的波形时，是从 $0 - \frac{\pi}{2}$ 的第一个点开始取反的，这个时候波形输出从最高点一下落到最低点导致出现了毛刺。通过查阅相关的书籍以及文献资料，我决定采用两个D触发器实现延时，从而避免了毛刺的出现。
- 5、在做提高功能AM调制时，也遇到了不少的问题。在一开始的时候，没有考虑到载波信号频率应该要比调制信号的频率大得多，所以调制出来的波形一直都不理想，后来经过向同学请教，找到了这个问题，并作了相应的修改，从而解决了这个问题。

7.2 实验的收获与感受

这是第二次使用QuartusII做电路设计，老师一开始只讲了一些基本原理以及需要注意的事项，并没有给具体的思路，需要我们另外查阅相关的资料进行设计。由于在前一周完成了多功能数字钟的设计实验，这一周再参加这个直

接数字频率合成器设计的实验明显感觉有所不同，设计过程中自己也不再像上一周那样那么的迷茫。同时，我对VHDL语言的使用也更得心应手，大大降低了设计的工作量。

完成这个实验，必须要对电路的总体框架有一个大概的把握，需要用到什么功能、设计什么模块，都是需要提前考虑的。

电路系统圆满完成了要求的基本功能，基础模块主要包括分频、控制字/相位字产生、进制转换、累加器、测频、显示译码等内容。

与第一个实验相比较，没有涉及到的就是累加器，需要额外设计，除了测频电路，其他模块相对来说是比较容易实现的。而显示译码模块部分，我在EDA(II)中用的是代码，但是与同学交流后发现，电路图搭建反而更直观，因此本实验用了电路图来实现。

在此基础上，我实现了更大范围的频率控制字与相位控制字、双路输出、节省ROM电路模块以及输出三角波、方波、锯齿波等多种波形、AM调制等拓展功能的设计。而在设计节省ROM电路模块的时候，发现正弦输出有毛刺的问题，在查阅资料后得到了解决的方案。

这次设计实验中，我意识到只要有了正确的设计思路，设计电路时考虑全面，遇到问题一步一步地解决，是可以实现想实现的功能的。

而在原理图搭建好之后下载到试验箱的过程中，一定要仔细分配管脚，要看清楚，不能填错，要和电路对应好。检查时要耐心，有条不紊地进行，不能急于求成。

最后，非常感谢老师和周围同学在本次实验中给我的帮助，让我受益匪浅。这次实验，我再次体验到了设计数字逻辑电路的不易与乐趣，大大激发了我动手实践的热情。希望有机会能多多尝试做做电路设计。相信这次实验也会给今后的学习和生活带来帮助。

7.3 期望及要求

这次实验的结果总体也还算圆满，但同样存在一些小小的遗憾。实验中设计测频电路的时候，我原本计划用VHDL语言进行设计，但设计的过程并不顺利，最后只能选择电路图的方法，希望有机会能研究出来。

这次实验，使我对于调制的理解更加深刻，有机会希望能完成其他形式的调制电路的设计。

参考文献

- [1] 数字逻辑电路与系统设计.蒋立平.北京:电子工业出版社,2008.7
- [2] VHDL数字电路设计教程.佩德罗尼.北京:电子工业出版社, 2013.1
- [3] 晏细兰,谢景明,熊茂华.基于FPGA和VHDL的高精度数字频率计研究与设计[J].计算机光盘软件与应用,2014,17(15):91-94.
- [4] 谢煜 ,黃为.一种基于VHDL语言数字频率计的设计与实现[J].电子设计应用,2003(07):25-27.