



南京理工大学
NANJING UNIVERSITY OF SCIENCE & TECHNOLOGY

电子线路课程设计

——DDS直接频率合成器

作 者： 施尘玥 学 号： 9151040G0507

学 院： 电光学院

专 业： 通信工程

题 目： 电子线路课程设计

——DDS 直接频率合成器

指导者： 蒋立平

2017 年 12 月

摘要

直接数字频率合成技术是一项基于全数字技术,从相位概念出发直接合成所需波形的一种新型的频率合成技术。它很容易实现频率、相位和幅度的数控调制,广泛应用于通讯领域。

本报告分析了 DDS 设计要求说明,整体电路工作原理的方案论证,还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。利用 FPGA 中的 ROM 模块,成功输出显示正弦波形,在此基础上双路输出两路正余弦波形,并能改变频率键字、相位键字得到所需的输出波形,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。另外,该系统还能生成三角波、矩形脉冲、方波等特殊波形。利用波形对称性特点,能明显节省 ROM 空间,提高波形精度。设计过程中,充分使用 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。

关键字: 频率合成器 QUARTUS II VHDL FPGA

Abstract

Direct digital frequency synthesis technology is a new frequency synthesis technology based on the whole digital technology, which directly synthesizes the desired waveform from the phase concept. It is easy to realize the frequency, phase and amplitude of numerical control modulation, widely used in the communication field.

This report analyzed the DDS design requirements, the overall working principle of the circuit scheme comparison, also illustrates the principle of each module design and simulation, programming and debugging work of the late download process. Use a ROM module in FPGA, successful output sine wave, on the basis of the double output two road is the cosine wave, and can change the frequency of key words to get the desired output waveform, phase key word, can display the current frequency on digital tube for and oscilloscope measured frequency comparison. In addition, the system can generate special waveforms such as triangular wave, rectangular pulse and square wave. Using the characteristic of waveform symmetry, it can obviously save ROM space and improve the waveform precision. In the design process, the full use of VHDL language is flexible and convenient. Most modules are designed with VHDL language, which greatly reduces the workload of the circuit design and improves the accuracy of the module.

Key words: frequency synthesizer QUARTUS II VHDL FPGA

目录

一、 实验内容及设计要求.....	5
1.实验内容简介.....	5
2.设计要求.....	5
二、 方案论证.....	5
1. DDS 基本概念.....	5
2. DDS 的组成及工作原理.....	6
三、 子模块设计原理.....	8
1. 脉冲发生电路.....	8
1.1 分频实现原理.....	8
1.2 VHDL 代码	8
1.3 封装与仿真波形.....	9
2.频率预置电路.....	10
2.1 电路原理.....	10
2.2 VHDL 代码	10
2.3 封装与仿真波形.....	11
3.BCD 转二进制电路	12
3.1 电路原理.....	12
3.2 VHDL 代码	12
3.3 封装.....	12
4.累加器.....	12
4.1 累加器的原理.....	13
4.2 VHDL 代码	13
4.3 累加器的封装图.....	14
5.相位调节模块.....	14
5.1 相位调节原理.....	14
5.2 相位调节电路图.....	15

6.波形选择电路.....	15
6.1 正弦波存储器（ROM）的原理	15
6.2 由量化公式生成 mif 文件	16
6.3 波形选择电路的封装图.....	17
7.测频电路.....	17
7.1 电路原理.....	18
7.2 测频电路原理图.....	18
7.3 封装图.....	19
8.译码显示电路.....	19
8.1 模 8 模块.....	19
8.2 八选一模块.....	20
8.3 三八译码器模块.....	21
8.4 七段译码器模块模块	22
9.基础部分总电路图.....	24
四、 提高部分设计.....	25
1.通用寄存器设计.....	25
2. 节省 ROM	26
3.AM 调制电路	28
4.消颤电路.....	29
5.显示复用电路.....	29
6.多种波形显示.....	30
五、 总电路图.....	32
六、 调试仿真和编程下载.....	33
七、 结论.....	34
1.实验结果.....	34
2.实验感悟.....	36
八、 参考文献.....	38

一、实验内容及设计要求

1.实验内容简介

设计一个频率及相位均可控制的具有正弦输出的直接数字频率合成器(DDS)；基于直接数字频率合成器完成一些相关电路设计。

2.设计要求

- (1) 设计双路输出、两路相位差可调正弦信号发生器；
- (2) 设计测频电路，验证信号发生器输出信号的正确性；（带显示）
- (3) 结合高频电子线路课程，设计数字式振幅调制电路（AM）
- (4) 结合高频电子线路课程，设计数字式频率调制电路（FM）

二、方案论证

1. DDS 基本概念

● 频率合成技术

利用频率合成的方法，使某一（或多个）基准频率，通过一定的变换与处理后，形成一系列等间隔的离散频率。

● 频率合成技术应用

短波通信：要求通信机能在 2--30MHz 频段内，提供以 100Hz 为间隔的 28 万个频率点；

移动通信：要求通信机能在 150、400、900、1800MHz 频率附近提供上百个频率点；

其他各种电子应用领域，如：电子对抗、现代电子测量仪器等。

● 频率合成常用的方法

➤ 直接式频率合成器

利用单个或多个不同频率的晶体振荡器作为基准信号源，经过倍频、

分频、混频等途径直接产生许多离散频率的输出信号。

➤ 间接式频率合成器（锁相频率合成器）

将相位反馈理论和锁相技术应用于频率合成领域。指合成器的输出信号不是直接从参考源经过变换而得，而是由锁相环的压控振荡器间接产生所需要的频率输出。

➤ 直接数字频率合成器（DDS）

它是目前最为典型、应用最广泛的一种频率合成技术。

● 直接数字频率合成技术

➤ 概念

直接数字频率合成器（Direct Digital Synthesizer）是从相位概念出发直接合成所需波形的一种频率合成技术。

➤ 组成

基于全数字技术。由参考时钟、相位累加器、只读存储器、数模转换器和滤波器组成的频率合成器，输出频率高达几百 MHz。

➤ 特点

工作频率范围很宽；极高的频率分辨力；极短的频率转换时间；任意波形输出能力；数字调制性能好。

2. DDS 的组成及工作原理

DDS 的组成如下图所示：

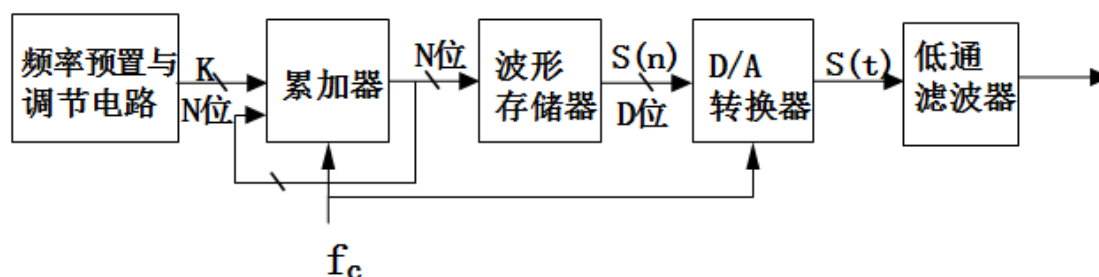


图 1 DDS 的组成结构

● 频率预置与调节电路

DDS 方程::
$$f_0 = \frac{f_c K}{2^N}$$

f_0 为输出频率, f_c 为时钟频率。当 $K=1$ 时, DDS 输出最低频率 (也即频率分辨率) 为 $\frac{f_c}{2^N}$, 而 DDS 的最大输出频率由 Nyquist 采样定理决定, 即 $f_c/2$, 也就是说 K 的最大值为 2^N-1 。因此, 只要 N 足够大, DDS 可以得到很细的频率间隔。要改变 DDS 的输出频率, 只要改变频率控制字 K 即可。

● 累加器

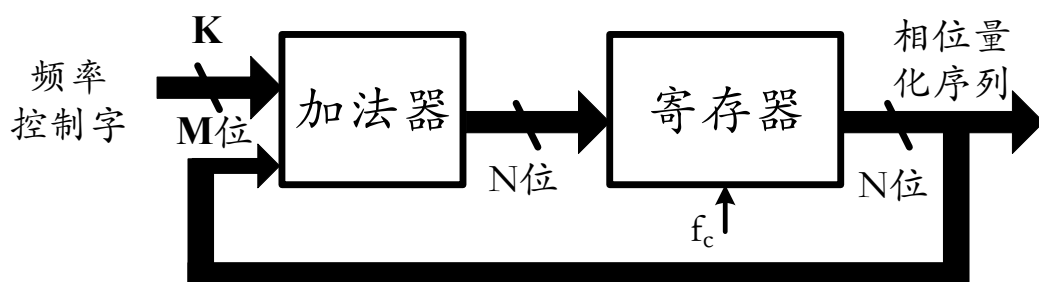


图 2 累加器框图

相位累加器由 N 位加法器与 N 位寄存器级联构成。每来一个时钟脉冲 f_c , 加法器将频率控制字 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; 以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

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

● 波形存储器

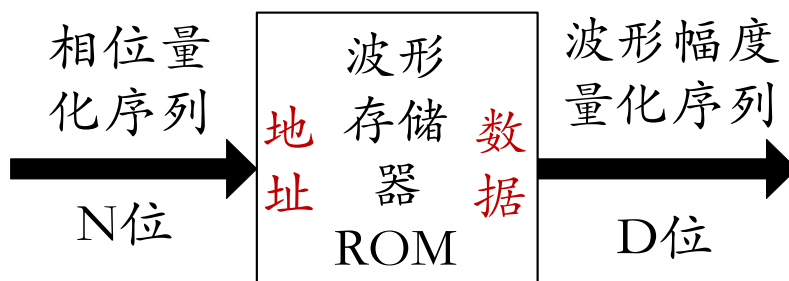


图 3 波形存储器框图

用相位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, 即可在给定的时间上确定输出的波形的幅值。 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。

● D/A 转换器

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 $T = \frac{2^N}{K} T_C$ 。需要注意的是，频率合成器对 D/A 转换器的分辨率有一定的要求，D/A 转换器的分辨率越高，合成的周期波 $S(t)$ 台阶数就越多，输出的波形的精度也就越高。

● 滤波器

滤波器的作用是将阶梯的模拟波形中的高频分量滤除，得到连续的有用波形。

三、子模块设计原理

1. 脉冲发生电路

分频的基本设计思想是将实验箱提供的 48MHz 的高频信号通过分频器变成所需的 1Hz、0.5Hz、2Hz 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。

1.1 分频实现原理

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。

1.2 VHDL 代码

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY fenpin IS
PORT (clk:IN STD_LOGIC;
```

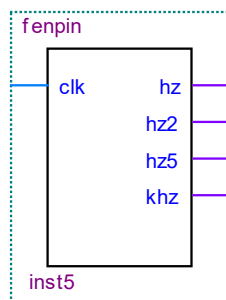
```

hz:buffer STD_LOGIC;
hz2:buffer STD_LOGIC;
hz5:buffer STD_LOGIC;
khz:buffer STD_LOGIC);
END fenpin;
ARCHITECTURE beh OF fenpin IS
SIGNAL count:integer range 0 to 24000000;
SIGNAL count1:integer range 0 to 48000000;
SIGNAL count2:integer range 0 to 24000000;
SIGNAL count3:integer range 0 to 24000000;
BEGIN
PROCESS (clk)
BEGIN
IF (clk='1') THEN
count<=count+1;
count1<=count1+1;
count2<=count2+1;
count3<=count3+1;
IF (count=23999999) THEN
count<=0;
hz<=NOT hz;
END IF;
IF (count1=47999999) THEN
count1<=0;
hz5<=NOT hz5;
END IF;
IF (count2=23999) THEN
count2<=0;
khz<=NOT khz;
END IF;
IF (count3=11999999) THEN
count3<=0;
hz2<=NOT hz2;
END IF;
END IF;
END PROCESS;
END beh;

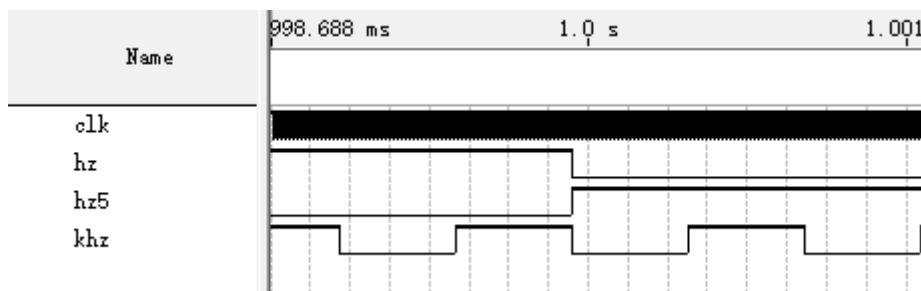
```

1.3 封装与仿真波形

将分频模块封装成如下所示：



分频模块仿真波形如下所示：



2.频率预置电路

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。DDS 的输出频率表达式为 $f_{out} = K \times f_c / 2^N$ 。当 $K=1$ 时，输出最低频率为 $f_c / 2^N$ ；而 DDS 的最高输出频率由 Nyquist 采样定理决定，即 $f_{out} = f_c / 2$ ，即 $K = 2^{N-1}$ ，此时 K 为最大值。我设计频率控制字 K 范围为 0-64。

2.2 VHDL 代码

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
ENTITY mo64 IS
PORT
( en   :IN  std_logic;
  clear:IN  std_logic;
  clk  :IN  std_logic;
  cout :out std_logic;
  qh   :buffer std_logic_vector(3 downto 0);
```

```

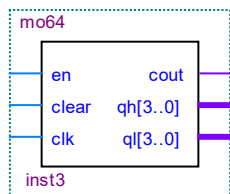
    q1  :buffer std_logic_vector(3 downto 0)
    );
END mo64;
ARCHITECTURE behave OF mo64 IS
BEGIN
    cout<='1'when(qh="0110"and q1="0011"and en='1')else'0';

    PROCESS(clk,clear)
    BEGIN
        IF(clear='0')THEN
            qh<="0000";
            q1<="0000";
        ELSIF(clk'EVENT AND clk='1')THEN
            if(en='1')then
                if((q1=3 and qh=6) or q1=9) then
                    q1<="0000";
                    if(qh=6)then
                        qh<="0000";
                    else
                        qh<=qh+1;
                    end if;
                else
                    q1<=q1+1;
                end if;
            end if;
        END IF;
    END PROCESS;
END behave;

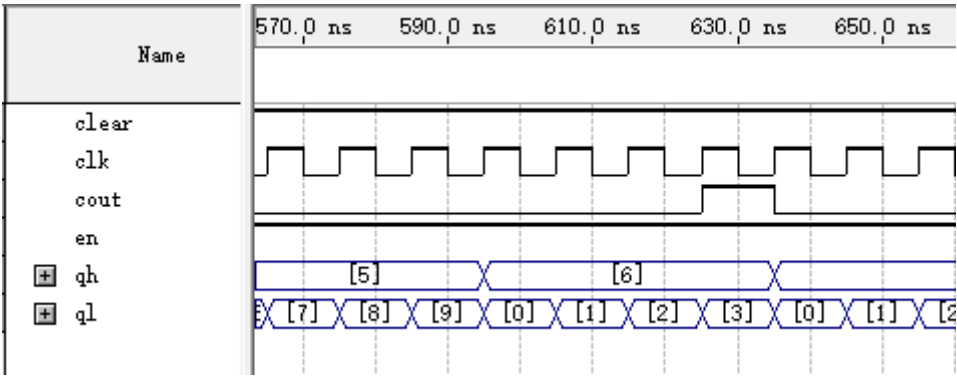
```

2.3 封装与仿真波形

将频率预置模块封装成如下所示：



频率预置模块仿真波形如下所示：



3.BCD 转二进制电路

3.1 电路原理

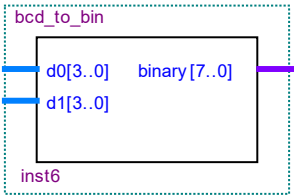
由于 BCD 码无法用乘法器或加法器进行运算，因此需设计 BCD 转二进制电路，使频率预置得出的 BCD 码转换为二进制。

3.2 VHDL 代码

```
entity bcd_to_bin is
port(d0,d1:in integer range 0 to 9;
binary:out integer range 0 to 128);
end bcd_to_bin;
architecture rel of bcd_to_bin is
begin
binary<=d0+10*d1;
end rel;
```

3.3 封装

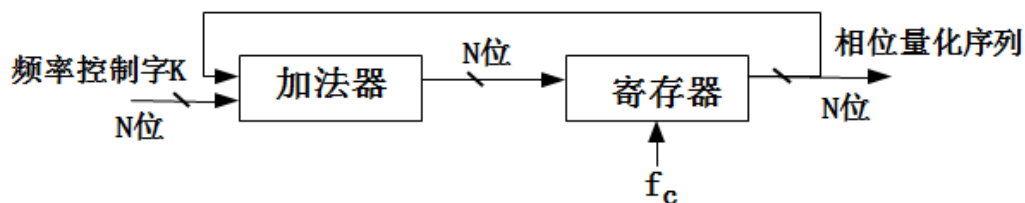
将 BCD 转二进制电路封装成如下所示：



4.累加器

4.1 累加器的原理

累加器由 N 位加法器 N 位寄存器构成，如下图所示。



其作用是，每来一个时钟 clk ，加法器就将频率控制字 K 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。

4.2 VHDL 代码

```

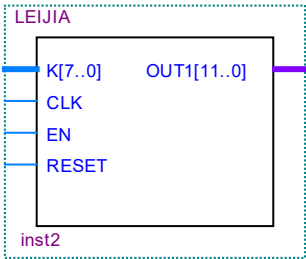
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY LEIJIA IS
PORT (K:IN STD_LOGIC_VECTOR(7 DOWNTO 0);
CLK:IN STD_LOGIC;
EN:IN STD_LOGIC;
RESET:IN STD_LOGIC;
OUT1:OUT STD_LOGIC_VECTOR(11 DOWNTO 0));
END ENTITY LEIJIA;
ARCHITECTURE ART OF LEIJIA IS
SIGNAL TEMP:STD_LOGIC_VECTOR(11 DOWNTO 0);
BEGIN
PROCESS (CLK,EN,RESET,TEMP) IS
BEGIN
IF RESET='0' THEN
TEMP<="000000000000";
ELSE
IF CLK'EVENT AND CLK='1' THEN
IF EN='1' THEN
TEMP<=TEMP+K;
END IF;
END IF;

```

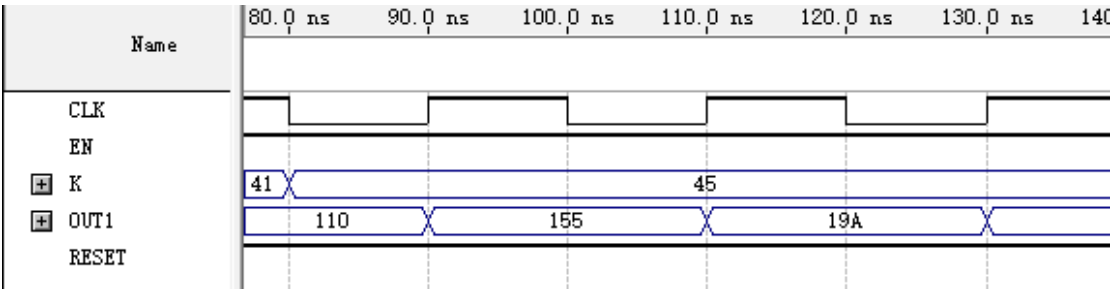
```
END IF;  
OUT1<=TEMP;  
END PROCESS;  
    END ARCHITECTURE ART;
```

4.3 累加器的封装图

累加器模块封装成如下所示：



累加器模块仿真波形如下所示：

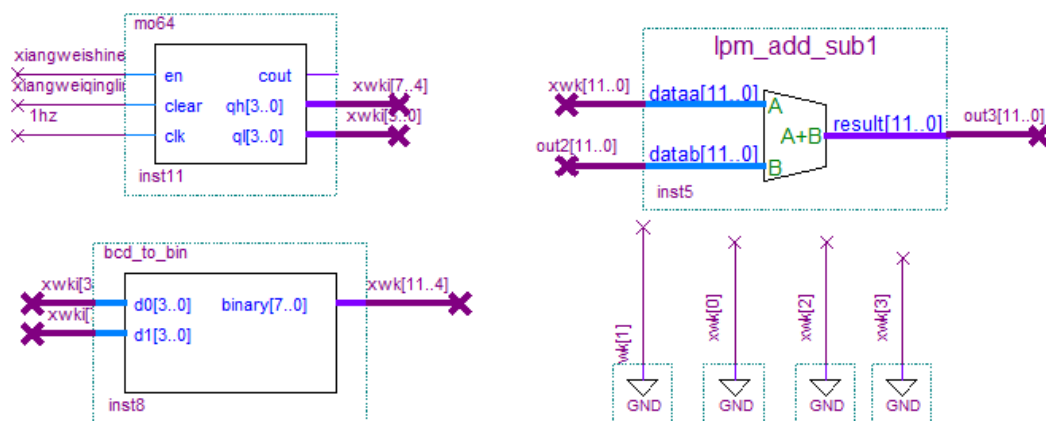


5.相位调节模块

5.1 相位调节原理

相位控制模块实际上是用一个 12 位的加法器将之前累加器的输出结果与相位控制字相加，从而构成相位控制模块。我设计的相位控制字范围是 0-64，其中清零与保持端分别由开关控制，以便得到所需相位。设计使相位控制字高八位与累加器结果相加，使相位变化更明显。

5.2 相位调节电路图



6.波形选择电路

波形存储器（ROM）的原理图如下图所示：



波形存储器（ROM）的相位取样地址来自于相位累加器输出的数据这样就可把存储在波形存储器内的波形抽样值（二进制编码）经查找表查出，完成相位到幅值转换。同时，波形存储器中还可存放不同种类波形的地址，例如正弦波、余弦波、方波、矩形波、锯齿波、三角波等，这些波均可通过后面的 D/A 转换器及低通滤波器将数字信号转化为模拟信号从而进行连续信号的输出与恢复。

6.1 正弦波存储器（ROM）的原理

正弦波形存储器 \sin_rom ，N（12）位的寻址 ROM 相当于把 $0 \sim 2\pi$ 的正弦信号离散成具有 2^N 个样值的序列，波形 ROM 有 D（10）位数据位，所以设置 2^N 个样值的 \sin 值以 D 位二进制数值固化在 ROM 中，这里设置 D=10，所以 ROM 中的数据范围应该从 0 到 1023，但是正弦值只从 -1 到 1，所以要对其进行量化，公式如下所示：

$$\begin{cases} \text{存储数据} = \text{round}(511 \times \sin(k) + 512) \\ k = n \times 2\pi / 4096 \end{cases}$$

其中, n 为存储地址, 范围是从 0 到 4095。

6.2 由量化公式生成 mif 文件

在 6.1 中我们已经求出了各种波所对应的量化公式, 但光有量化公式还不行, 我们还要把根据量化公式计算出的存储数值放到存储地址当中去, 这个地址文件就是 mif 文件, 进而再根据 mif 文件生成波形存储器封装图。我这以正弦波为例, 来说明其操作步骤。

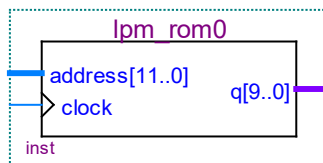
我们利用 Matlab 产生 sin 的采样数据, 相关的 Matlab 程序如下所示:

```
clear all;
close all;
clc;
index=linspace(0,2*pi,2^12+1);
cos_val=fix((2^9-1)*sin(index)+0.5);
fid=fopen('C:\Users\ThinkPad\Desktop\DDS\sin1.mif','w');
fprintf(fid,'WIDTH=10;\n');
fprintf(fid,'DEPTH=4096;\n'); fprintf(fid,'ADDRESS_RADIX=UNS;\n');
fprintf(fid,'DATA_RADIX=DEC;\n');
fprintf(fid,'CONTENT?BEGIN\n');
for j=1:4096
    i=j-1;
    fprintf(fid,'%3d',i);
    fprintf(fid,':');
    fprintf(fid,'%3d',cos_val(j));
    fprintf(fid,';\n');
end
fprintf(fid,'END;\n');
```

此时, matlab 将自动生成 mif 文件。

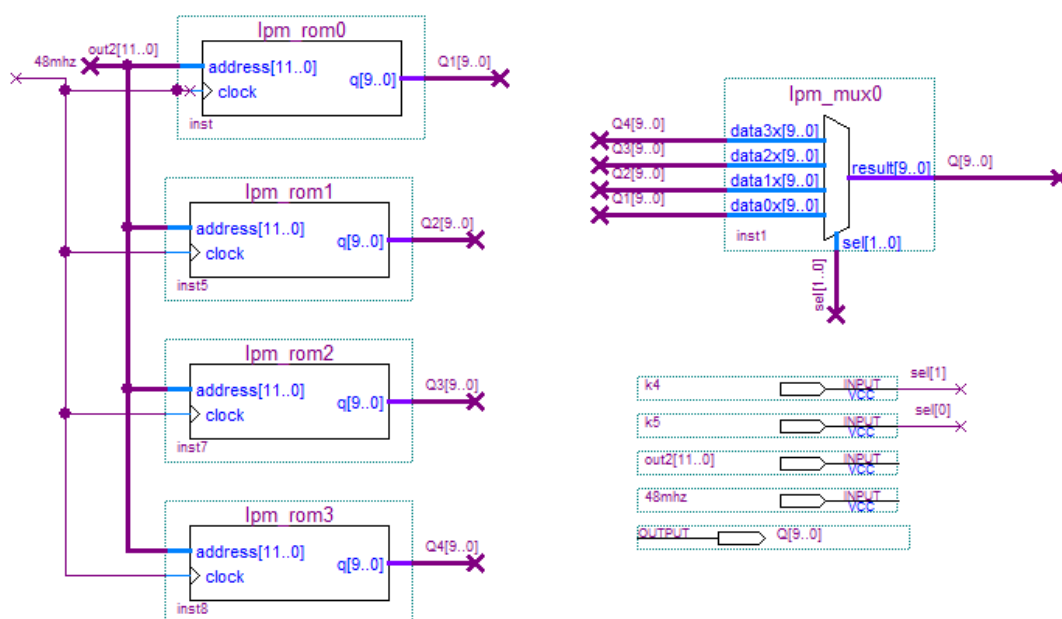
将每个波形的波形数据存储到*.mif 文件中后需要建立其对应的 ROM 封装电路, 以正弦波为例, 具体步骤如下所示: 首先要新建 Block Diagram/Schematic File 文件, 双击空白区, 在 name 条中填写 lpm_rom, 点击 OK; 接着在输出文件类型中选择 VHDL, 同时填写对应 ROM 文件的文件名, 分别设置为 10bit 和 4096words, 勾选 'q' output port, 在 File name 中选择对应*.mif 文件路径, 点击

Next, 结束创建, 可得正弦波 ROM 封装如下图:

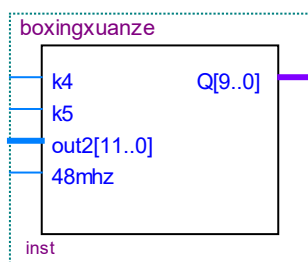


6.3 波形选择电路的封装图

以相同的方式可以建立余弦、锯齿波、方波的 ROM 封装, 并通过选择器构成波形选择电路, 电路图如下:

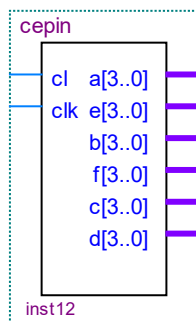


波形选择电路封装图如下所示:



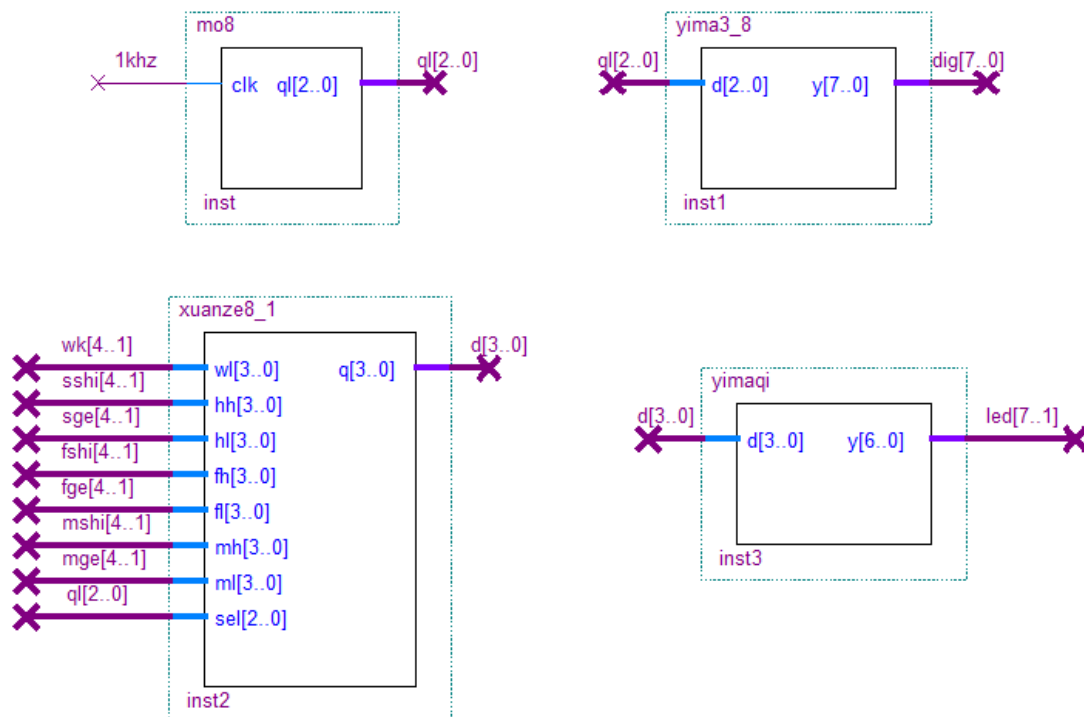
7.测频电路

7.3 封装图



8.译码显示电路

在数码管上的显示为动态扫描，每一个时间只能有一个数码管有效。当扫描的频率比较大时，就可以认为所有的数码管都是亮的。为了美观，使不需要的数码管显示为‘一’。显示模块由模 8 模块、八选一模块、七段译码器模块、三八译码器模块组成。总原理图如下图所示：

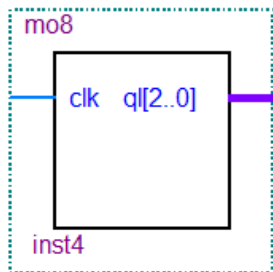


8.1 模 8 模块

模 8 计数器是为了只让一个数码管有效，且作为八选一选择器的输入端，选

择输出的数字是多少。

模 8 的原理和代码与计时电路中模 7 的原理类似，在此不进行赘述，仅给出封装图：



8.2 八选一模块

8.2.1 实现原理

八选一模块用来选择当前有效的数码管应该显示时钟的什么数字，星期、短横线、时十位、时个位、分十位、分个位、十位、秒个位，这八种状态。其中短横线为 1111 来代替，在后面的七段译码器模块中，将 1111 译为在数码管中只有 g 段亮，即为短横线。

8.2.2 八选一模块 VHDL 代码

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity xuanze8_1 is
port(wl,hh,h1,fh,f1,mh,m1:in std_logic_vector(3 downto 0);
sel:in std_logic_vector(2 downto 0);
q:out std_logic_vector(3 downto 0));
end xuanze8_1;
architecture xuanze of xuanze8_1 is
begin
process(sel,wl,hh,h1,fh,f1,mh,m1)
begin
case sel is
when "000"=>q<=wl;
when "001"=>q<="1111";
when "010"=>q<=hh;
when "011"=>q<=h1;
when "100"=>q<=fh;

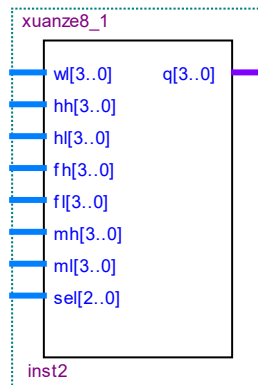
```

```

when "101" => q <= f1;
when "110" => q <= mh;
when "111" => q <= ml;
when others => q <= "0000";
end case;
end process;
end xuanze;

```

可得八选一模块封装图如下：



8.3 三八译码器模块

8.3.1 实现原理

该译码器将计数器输出的三位数据译码成八位数据，作为数据分配器使用，控制数码管 的是否有效。

8.3.2 三八译码器模块 VHDL 代码

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity yima3_8 is
port(d:in std_logic_vector(2 downto 0);
y:out std_logic_vector(7 downto 0));
end yima3_8;
architecture yima of yima3_8 is
begin
process(d)
begin
case d is
when "000" => y <= "01111111";
when "001" => y <= "10111111";
when "010" => y <= "11011111";

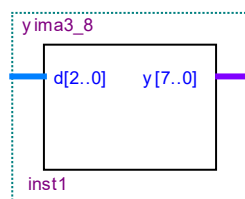
```

```

when "011" => y <= "11101111";
when "100" => y <= "11110111";
when "101" => y <= "11111011";
when "110" => y <= "11111101";
when "111" => y <= "11111110";
when others => y <= "11111111";
end case;
end process;
end yima;

```

可得三八译码器模块封装如下：



8.4 七段译码器模块模块

8.4.1 实现原理

该译码器用于将二进制的数字译码为数码管中 a、b、c、d、e、f、g 段的显示情况，能直观地显示出数字来。我们所用的数码管为共阳。

显示字型	g, f, e, d, c, b, a
0	1 0 0 0 0 0 0
1	1 1 1 1 0 0 1
2	0 1 0 0 1 0 0
3	0 1 1 0 0 0 0
4	0 0 1 0 0 1 0
5	0 0 1 0 0 1 0
6	0 0 0 0 0 1 0
7	1 1 1 1 0 0 0
8	0 0 0 0 0 0 0
9	0 0 1 0 0 0 0

8.4.2 七段译码器模块 VHDL 代码

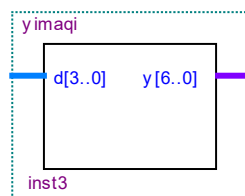
```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity yimaqi is
port(d:in std_logic_vector(3 downto 0);

```

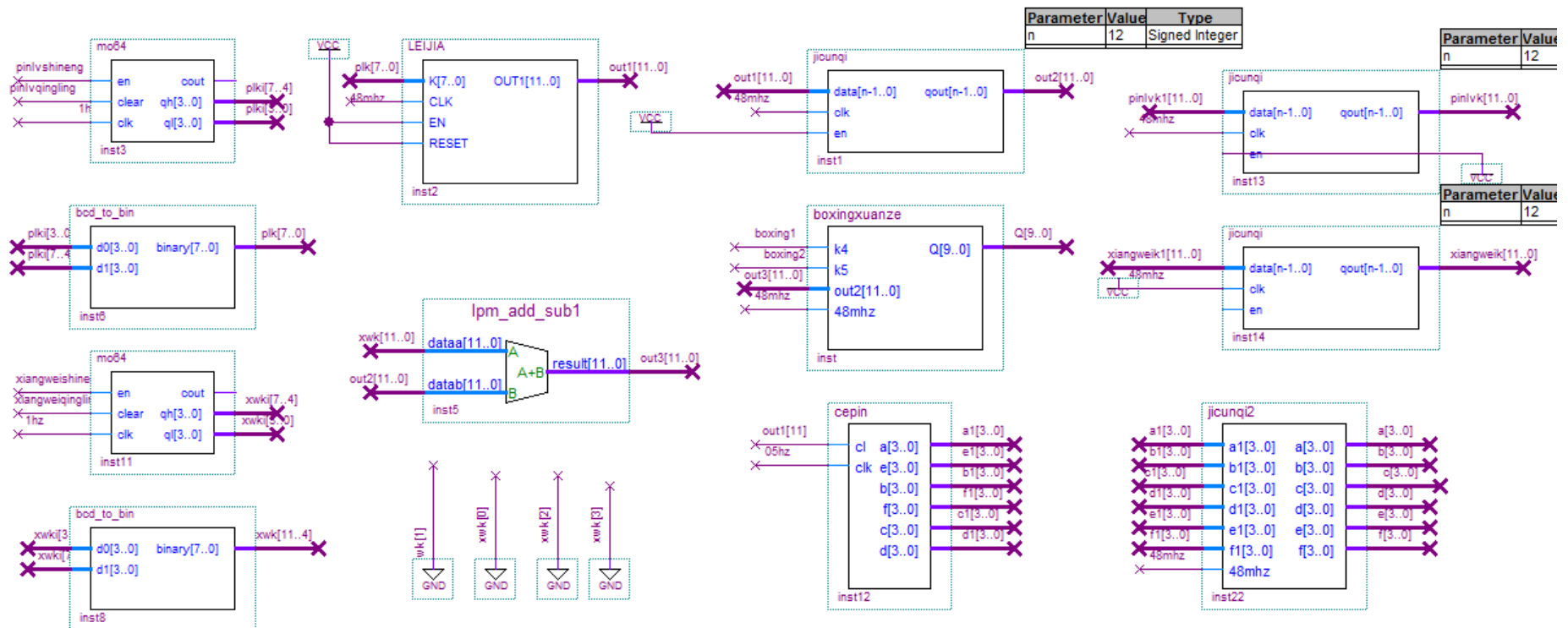
```
y:out std_logic_vector(6 downto 0));
end yimaqi;
architecture yima of yimaqi is
begin
process (d)
begin
case d is
when "0000" => y <= "1000000";
when "0001" => y <= "1111001";
when "0010" => y <= "0100100";
when "0011" => y <= "0110000";
when "0100" => y <= "0011001";
when "0101" => y <= "0010010";
when "0110" => y <= "0000010";
when "0111" => y <= "1111000";
when "1000" => y <= "0000000";
when "1001" => y <= "0010000";
when "1111" => y <= "0111111";
when others => y <= "1111111";
end case;
end process;
end yima;
```

可得三八译码器模块封装如下：



9.基础部分总电路图

基础电路部分总电路图如下：



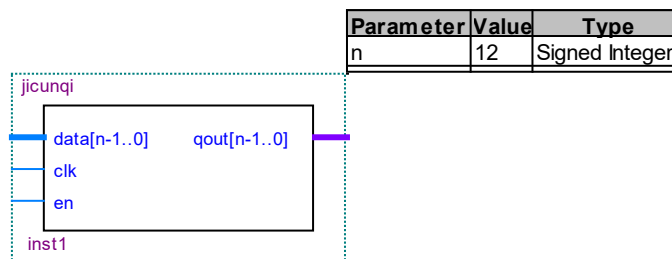
四、提高部分设计

1.通用寄存器设计

累加器电路后，还需一个 12 位的寄存器，时钟上升沿来到时更新寄存器当中的数据。为了便于拓展，我用 VHDL 语言设计了一个任意位通用寄存器，代码如下：

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
entity jicunqi is
generic( n:positive:=8);
port( data:in std_logic_vector(n-1 downto 0);
      clk,en: in std_logic;
      qout:out std_logic_vector(n-1 downto 0));
end jicunqi;
architecture r of jicunqi is
signal temp:std_logic_vector(n-1 downto 0);
begin
process(clk,en,temp)
begin
if en='1' then
if rising_edge(clk) then
temp<=data;
else temp<=temp;
end if;
end if;
end process;
qout<=temp;
end r;
```

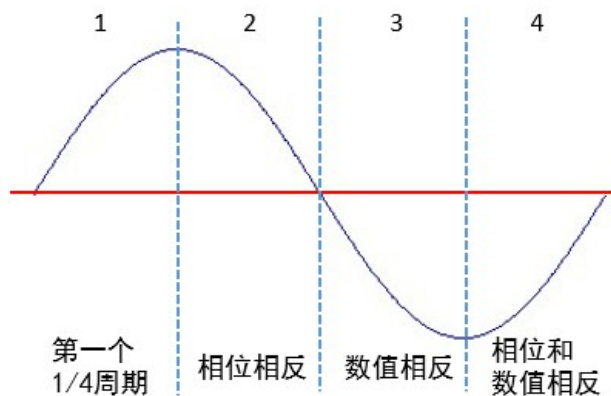
封装如下：



2. 节省 ROM

ROM 模块是本系统的核心部分之一,该模块中储存了所需要的函数信息,用 Matlab 生成储存函数信息的 .mif 文件,再用 Quartus II 中的 LPM ROM 模块便能轻松产生对应的 ROM 模块,该系统所用的 ROM 有 12 位的地址线和 10 位的数据线,ROM 中共有 4096 个数值。

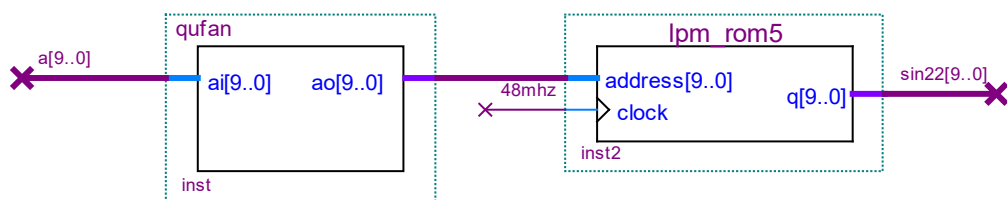
但是我们仔细观察会发现正弦函数具有对称性,用 1/4 个周期的波形就能完整地表现出完整周期的波形:第二个 1/4 周期与第一个 1/4 周期波形是相位相反的关系,第三个 1/4 周期与第一个 1/4 周期波形是函数值相反的关系,第四个 1/4 周期与第一个 1/4 周期波形则是相位、函数值同时反相的关系。其关系可以用图 3-8-1 来描述。



取反电路:



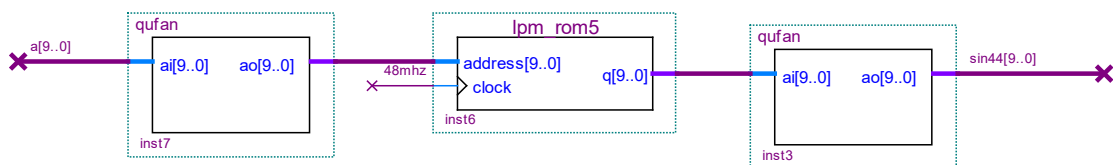
地址取反:



输出取反:

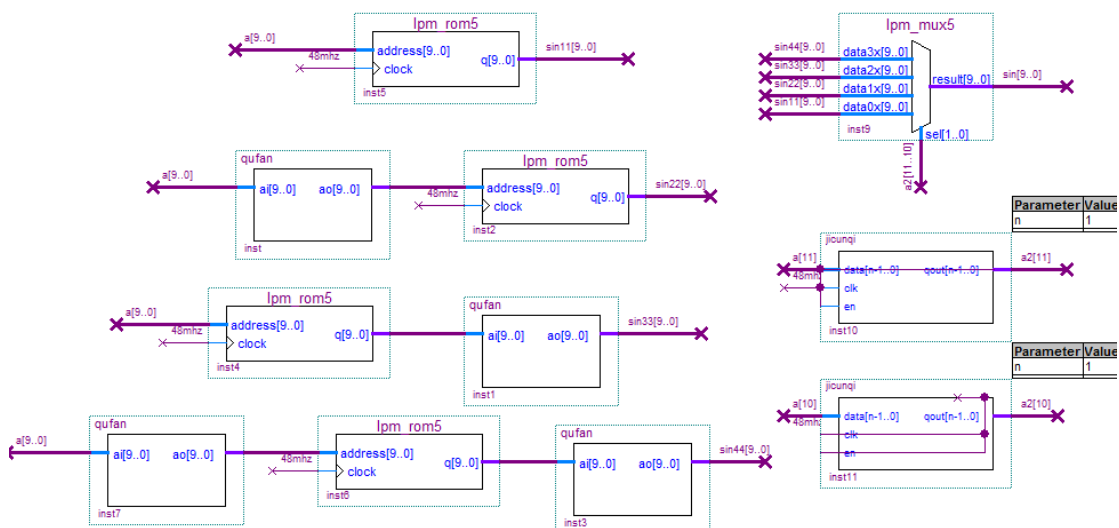


地址输出取反:

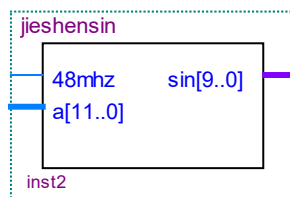


节省 ROM 电路:

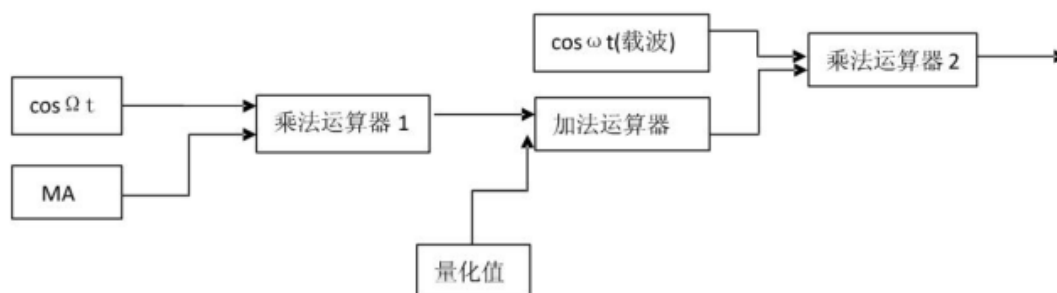
(为避免输出波形有毛刺, 所以设计了两个寄存器来消除毛刺)



封装后，如下图：

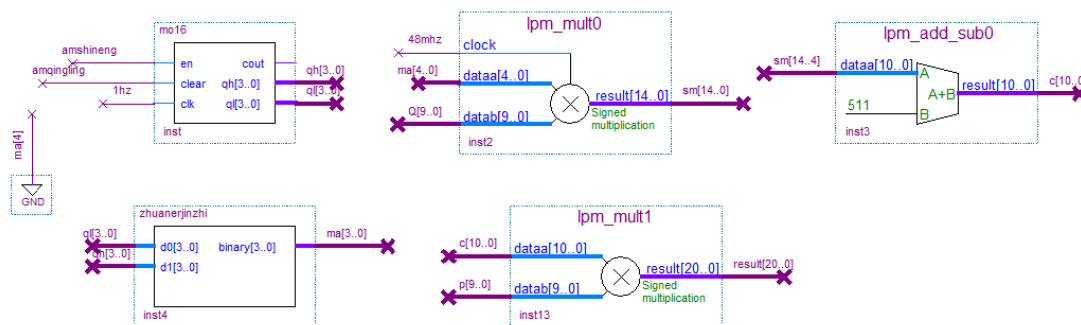


3. AM 调制电路

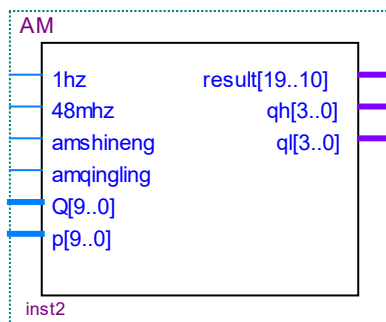


通信理论中将信号调制定义为调制信号对载波的幅度、频率和相位进行变换。AM 即标准调制信号，除了来自消息的基带信号外，还包含了直流信号，它是调制后输出信号既含载波分量又含边带分量的标准调幅信号。

AM 调制电路如下：

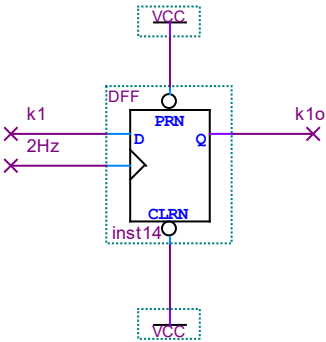


封装图为：

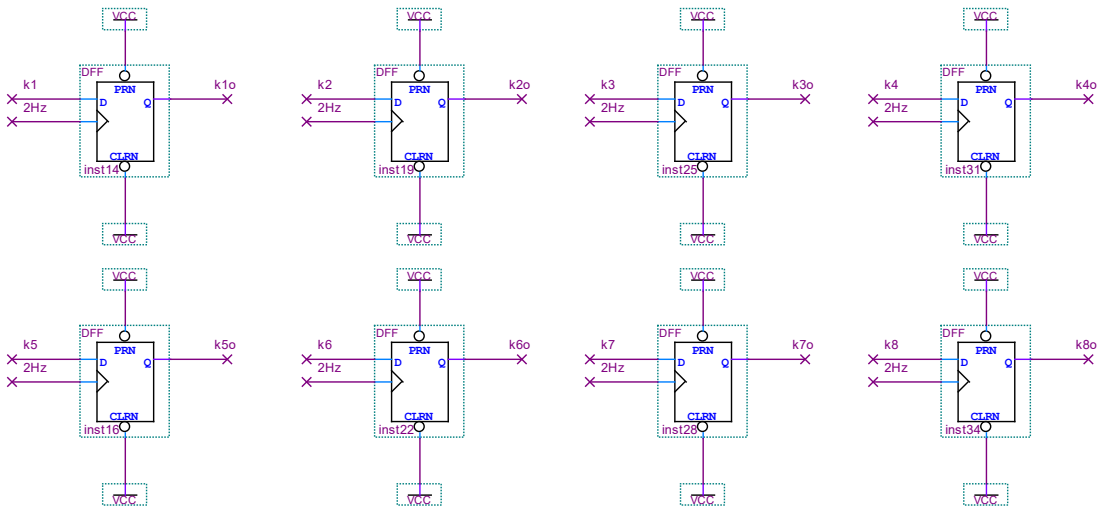


4.消颤电路

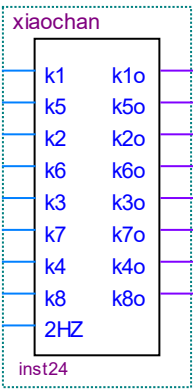
消颤电路图



消颤开关组



封装如下:

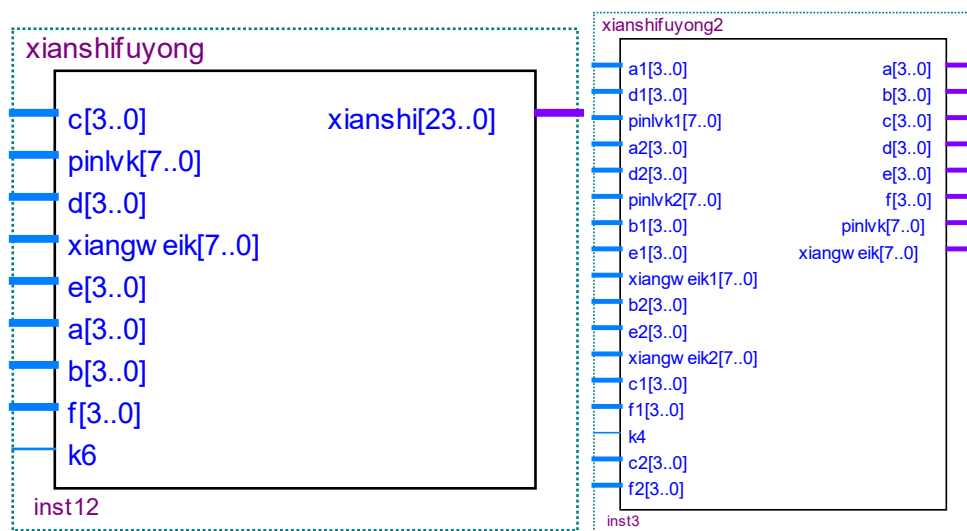


5.显示复用电路

为了使通道一和通道二可以分别调节,使通道一和通道二的频率和相位调节

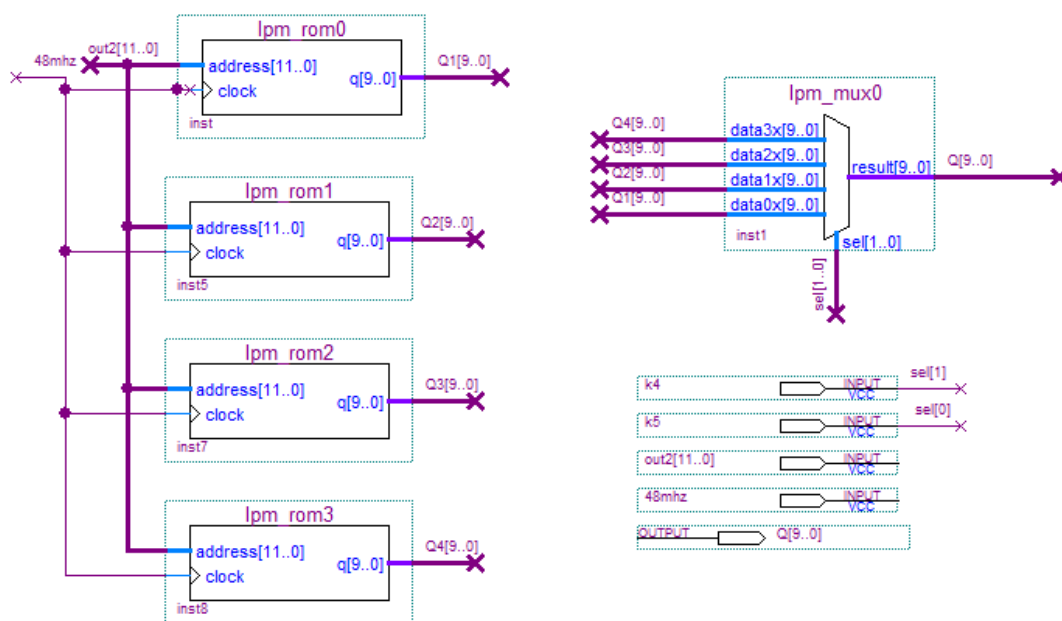
字可以分别控制，并可通过开关显示在数码管上，因此，设计显示复用电路，使数码管可以显示“AM 调节字-通道一相位控制字-通道一频率控制字”，“AM 调节字-通道二相位控制字-通道二频率控制字”，以及测频电路所测出的两个通道的频率四种显示。为此设计了两个显示复用模块。

两个复用模块封装如下：

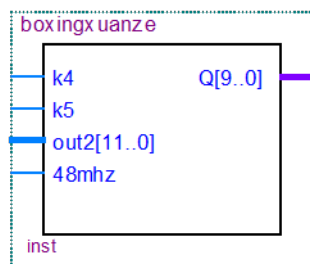


6.多种波形显示

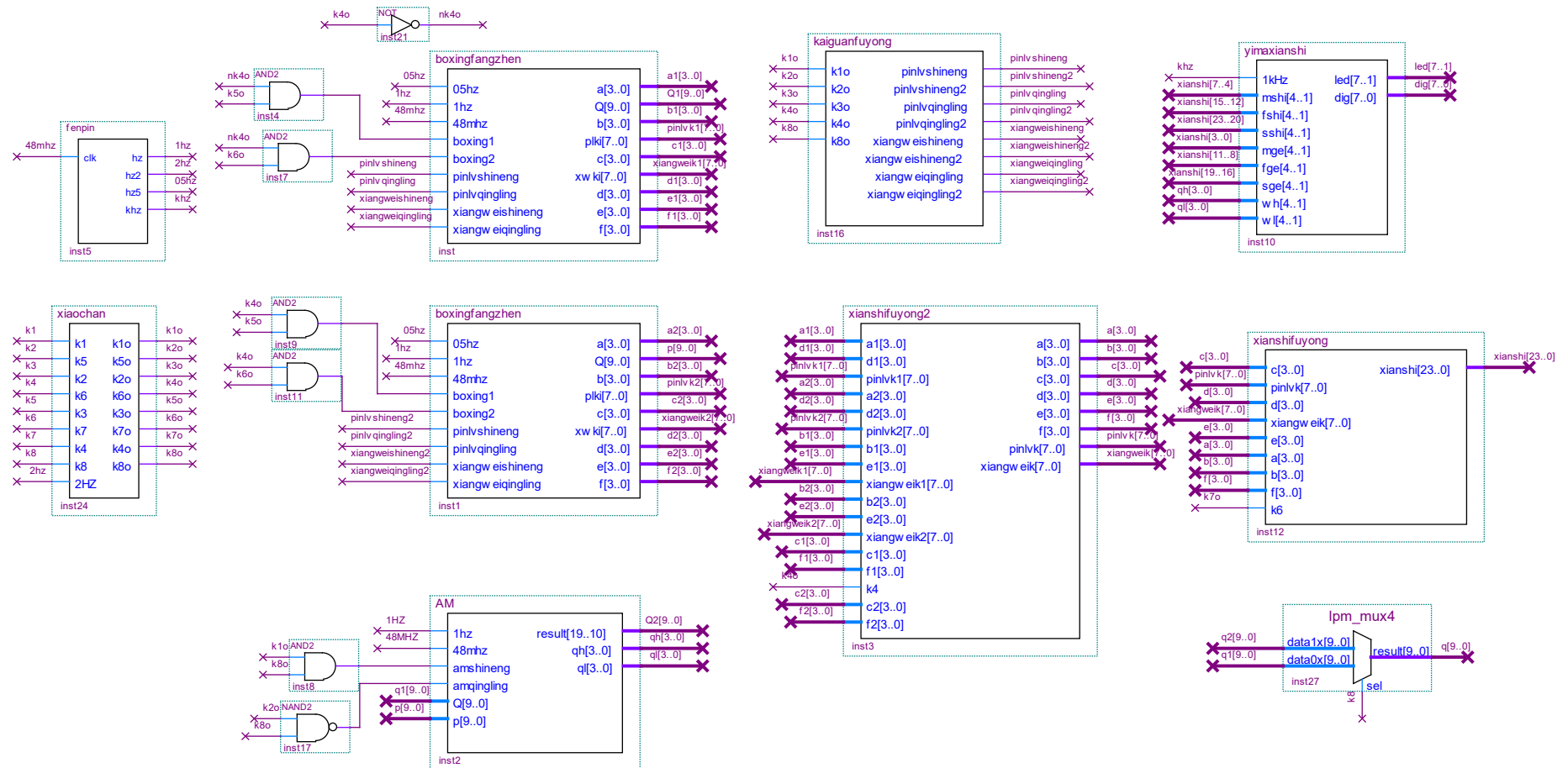
建立正弦波 ROM 后，以相同的方式可以建立余弦、锯齿波、方波的 ROM 封装，并通过选择器构成波形选择电路，电路图如下：



波形选择电路封装图如下所示：



五、总电路图





















六、 调试仿真和编程下载

在 EDA 实验中,电路设计是第一步也是最为关键的一步,电路设计完成后,还要对已完成的电路图保存后进行编译,检查有无出错的地方,比如线路定义出错,引脚未定义等软件能检查出来的错误,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。将程序下载至 FPGA 芯片之前,还要进行管脚的分配,管脚分配图如下:

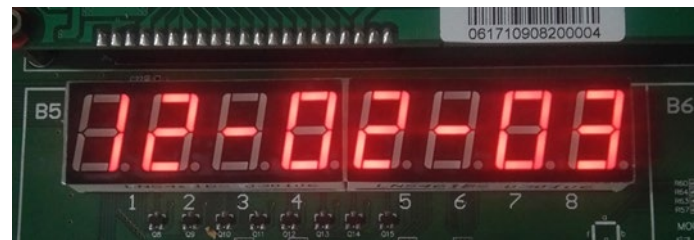
	Node Name /	Direction	Location	I/O Bank	VREF Group	I/O Standard
1	1hz	Unknown	PIN_M5	2	B2_N0	2.5 V (default)
2	48mhz	Input	PIN_A10	7	B7_N0	2.5 V (default)
3	clk	Unknown	PIN_F17	6	B6_N0	2.5 V (default)
4	dig[0]	Output	PIN_H15	6	B6_N0	2.5 V (default)
5	dig[1]	Output	PIN_E14	7	B7_N0	2.5 V (default)
6	dig[2]	Output	PIN_A18	7	B7_N0	2.5 V (default)
7	dig[3]	Output	PIN_D16	7	B7_N0	2.5 V (default)
8	dig[4]	Output	PIN_L14	5	B5_N0	2.5 V (default)
9	dig[5]	Output	PIN_L16	5	B5_N0	2.5 V (default)
10	dig[6]	Output	PIN_H16	6	B6_N0	2.5 V (default)
11	dig[7]	Output	PIN_J13	6	B6_N0	2.5 V (default)
12	hz2	Unknown	PIN_R4	2	B2_N0	2.5 V (default)
13	k1	Input	PIN_V9	3	B3_N0	2.5 V (default)
14	k2	Input	PIN_U10	4	B4_N0	2.5 V (default)
15	k3	Input	PIN_B9	8	B8_N0	2.5 V (default)
16	k4	Input	PIN_B10	7	B7_N0	2.5 V (default)
17	k5	Input	PIN_R18	5	B5_N0	2.5 V (default)
18	k6	Input	PIN_R17	5	B5_N0	2.5 V (default)
19	k7	Input	PIN_P18	5	B5_N0	2.5 V (default)
20	k8	Input	PIN_P17	5	B5_N0	2.5 V (default)
21	led[1]	Output	PIN_F10	7	B7_N0	2.5 V (default)
22	led[2]	Output	PIN_E11	7	B7_N0	2.5 V (default)
23	led[3]	Output	PIN_F12	7	B7_N0	2.5 V (default)
24	led[4]	Output	PIN_C12	7	B7_N0	2.5 V (default)
25	led[5]	Output	PIN_D12	7	B7_N0	2.5 V (default)
26	led[6]	Output	PIN_E13	7	B7_N0	2.5 V (default)
27	led[7]	Output	PIN_C14	7	B7_N0	2.5 V (default)
28	p[0]	Output	PIN_C2	1	B1_N0	2.5 V (default)
29	p[1]	Output	PIN_B2	1	B1_N0	2.5 V (default)
30	p[2]	Output	PIN_M3	2	B2_N0	2.5 V (default)
31	p[3]	Output	PIN_D2	1	B1_N0	2.5 V (default)
32	p[4]	Output	PIN_F3	1	B1_N0	2.5 V (default)

33		p[5]	Output	PIN_G2	1	B1_N0	2.5 V (default)
34		p[6]	Output	PIN_H2	1	B1_N0	2.5 V (default)
35		p[7]	Output	PIN_K2	2	B2_N0	2.5 V (default)
36		p[8]	Output	PIN_L2	2	B2_N0	2.5 V (default)
37		p[9]	Output	PIN_M2	2	B2_N0	2.5 V (default)
38		Q[0]	Output	PIN_P1	2	B2_N0	2.5 V (default)
39		Q[1]	Output	PIN_M1	2	B2_N0	2.5 V (default)
40		Q[2]	Output	PIN_L1	2	B2_N0	2.5 V (default)
41		Q[3]	Output	PIN_K1	2	B2_N0	2.5 V (default)
42		Q[4]	Output	PIN_H1	1	B1_N0	2.5 V (default)
43		Q[5]	Output	PIN_G1	1	B1_N0	2.5 V (default)
44		Q[6]	Output	PIN_E1	1	B1_N0	2.5 V (default)
45		Q[7]	Output	PIN_C1	1	B1_N0	2.5 V (default)
46		Q[8]	Output	PIN_F7	8	B8_N0	2.5 V (default)
47		Q[9]	Output	PIN_P2	2	B2_N0	2.5 V (default)
48		tongdao1	Output	PIN_F6	8	B8_N0	2.5 V (default)
49		tongdao2	Output	PIN_H6	1	B1_N0	2.5 V (default)
50		yinjiao	Output	PIN_A2	8	B8_N0	2.5 V (default)

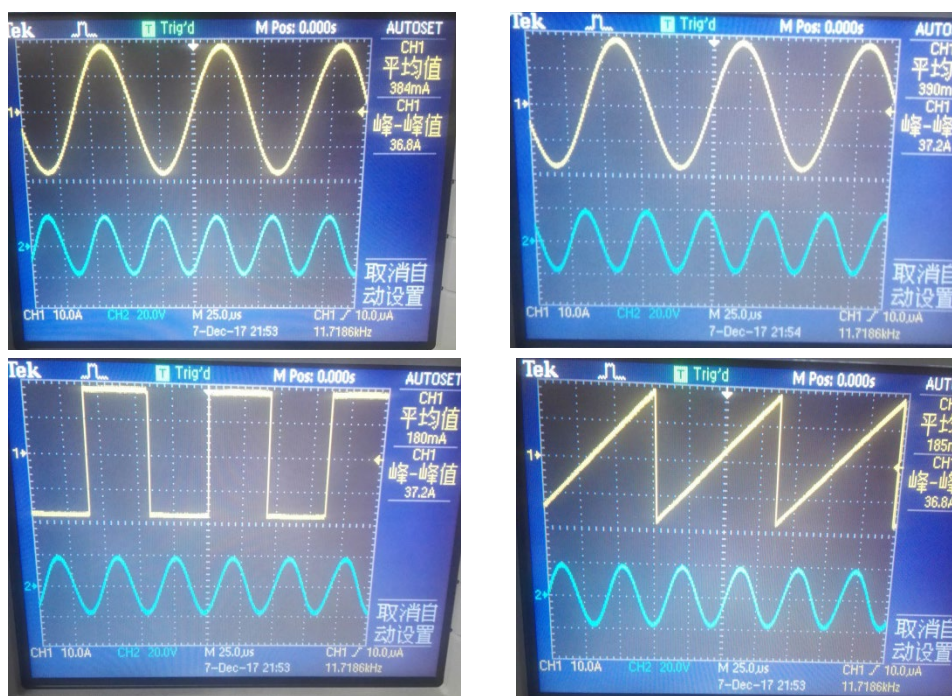
七、 结论

1.实验结果

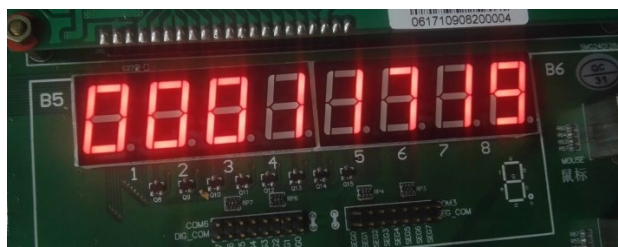
- 1. 在默认情况下，用示波器检测，示波器显示为两条直线；
- 2. 拨动使能开关 K1，改变通道一频率字，出现正弦波；
- 3. 拨动清零开关 K2，通道一输出清零，通道一显示为一条直线；
- 4. 拨动 K3，可由调节频率控制字转换为调节相位控制字。
- 5. 拨动 K4，可转换通道一和通道二，K4=1 为通道二，此时可使用 K1、K2、K3 对通道二进行频率控制字和相位控制字的调节。并且数码管的显示也随着 K4 的拨动而转变为显示通道二的频率控制字和相位控制字。见下图：12 为 AM 控制字，02 为相位控制字，03 为频率控制字。



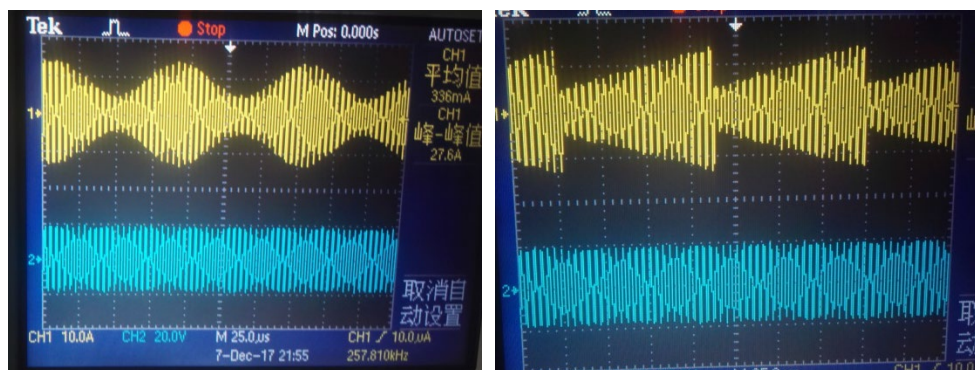
- 6. 拨动波形选择开关 K5、K6，当 K5K6 对应的二进制码为 00 时，为正弦输出，为 01 时，为余弦输出；为 10 时，为方波输出；为 11 时，为锯齿波输出；



7. K7 为显示复用开关, K1=1 时, 数码管由显示“AM 控制字-频率控制字-相位控制字”转换为测频电路测出的频率, 并可通过拨动 K4 分别测得通道一和通道二各自的频率。



8. K8 为 AM 控制开关, K8 等于 1 时, 示波器通道二显示 AM 调制后的波形, 并可通过 K1、K2、K3 控制 AM 控制字的调节。也可通过 K5、K6 对不同波形进行调制。





9. 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。
10. 用改进型（提高精度）的正弦函数 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，放大波形比较发现，相比较之前的 ROM，改进后的 ROM 精度更高，在显示上不容易出现锯齿或阶梯形状的波形。

2.实验感悟

在前一周完成了多功能数字钟的设计实验之后，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到熟悉了很多，整个设计过程中自己也不再像上一周那样那么的迷茫和紧张，而正得益于前一周的实验，自己对整个设计流程、方法都很熟悉，对 VHDL 语言也使用的得心应手，整个系统中，用 VHDL 语言设计的电路模块超过了 70%，这不仅大大降低了设计的工作量，也使得系统的便于拓展修改，便于理解。

该电路系统圆满完成了要求的基本功能，在此基础上也实现了更大范围的频率控制字、数码管上显示波形频率、双路输出正余弦波形、节省 ROM 电路模块以及输出三角波、方波、锯齿波等多种波形等拓展功能的设计。但是，实验中还是有一点问题，在设计测频电路的时候，我原本是想用 VHDL 语言进行设计，但设计的过程并不顺利并且因此浪费了很多时间，最后只能选择电路图的方法，希望在空闲的时候我能把他研究出来。设计节省 ROM 电路模块的时候，还发现正弦输出有毛刺的问题，在更改了示波器 CLK 接口后得到了解决。

在 DDS 实验设计中，我采用了分模块设计的思路，对于频率控制电路、相位累加器、寄存器、相位控制字电路、测频电路分别设计，最后将各模块连接起来。由于自己有一定的 VHDL 基础，我编写了一个通用寄存器的模块，这个模块功能十分强大，让我能够根据需求改变 N 值使用，大大方便了设计过程。

这次设计实验也更让我学习到，在原理图搭建好之后分配管脚，重新编译，下载到试验箱的过程中出现问题时，要逐个对各个模块进行检查、调试，分配管脚时要看清楚，不能填错，要和电路对应好。检查时要耐心，尤其是输入输出较多的电路图，更要看清每一部分。

最后，我要感谢学长，和善的态度和认真的教导，让我受益匪浅。

八、参考文献

- [1]蒋立平, 姜萍, 数字逻辑电路与系统设计, 北京: 电子工业出版社, 2008
- [2]南京理工大学电子技术中心, EDA 设计实验指导书
- [3]侯伯亨, VHDL 硬件描述语言与数字逻辑电路设计