



南京理工大学
NANJING UNIVERSITY OF SCIENCE & TECHNOLOGY

电子线路设计实验报告

---直接数字频率合成器 (DDS)

学校： 南京理工大学

学院： 电子工程与光电技术学院

专业： 通信工程

姓名： 樊楚仪

学号： 9151160D0102

指导老师： 蒋立平

目录

摘要.....	2
ABSTRACT	3
一 实验要求说明	3
1 设计内容.....	3
2 实验目的.....	3
3 实验要求.....	4
(1) 设计基本要求	4
(2) 设计提高部分	4
二 工作原理.....	4
1 基本框图.....	4
图 1	5
图 2	5
2 设计思想.....	5
3 DDS 工作流程.....	6
三 各子模块设计原理	6
(一) 基本设计.....	6
1 分频电路	6
2 频率与相位字产生	10
3 相位累加器	11
4 波形存储器	13
5 波形选择电路	15
6 测频电路	15
7 译码显示电路	17
(二) 附加功能.....	19
1 AM 调制	19
2 一片 1/4 ROM 实现 sin 波形存储.....	22
总电路图.....	24
四、程序下载、仿真与调试	25
4.1 调试.....	25
4.2 仿真.....	25
4.3 管脚分配.....	25
五、实验中遇到的问题及解决办法	26
六、实验感想	27

摘要

本实验设计一个直接数字合成器（DDS）。用 quartus II 设计、仿真、调试，最后下载到 SmartSOPC 实验系统上验证功能。

用户可根据需要选择合适的频率字输出 $0-500000\text{Hz}$ ($1/2f_c$) 任意频率的正弦，余弦及其他波形。频率字相位字采用在数码管上显示计数的方式输入，能输入到千位，且具有移相功能。具有测频和显示功能（显示 6 位十进制数），并具有频率相位字和测得频率的显示切换电路。除要求正弦波之外同时能输出余弦波，方波，三角波，锯齿波等波形，且采用地址码（逻辑电路）实现，整个电路只用了一片 $1/4\text{ROM}$ 。并在原电路的基础上将无符号 ROM 变为有符号 ROM 后进行 AM 调试。

关键词： DDS 频率字 相位字 $1/4\text{ROM}$ 测频显示 移相

ABSTRACT

The experiment is to design a direct digital frequency synthesizer(dds). QuartusII is used to design, stimulate and debug. Finally the project is downloaded to SmartSOPC to test the function.

User can insert frequency word to output sine, cosine and other signal of specific frequency from 0 to 500000hz ($1/2 f_c$). Frequency word and phase word are inserted by counting displaying in the digital tubes and thousands can be inserted. It has the phase shift function. It has the frequency measurement (up to million) and display function and the frequency word and frequency measurement display switch circuit. The reset and enablement function is available. It can output pulse, triangular wave, sawtooth wave without ROM. The whole circuit use only a piece of $1/4\text{ROM}$. And on the basis of the original circuit, unsigned ROM is changed to signed ROM for debugging.

Keywords: DDS frequency word phase word $1/4\text{ROM}$

frequency measuring and display phase shift

一 实验要求说明

1 设计内容

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（DDS）。

2 实验目的

学习和掌握 Quartus II 软件的设计、仿真等方法和思想。

3 实验要求

（1）设计基本要求

- ①利用 QuartusII 软件和 SmartSOPC 实验箱实现 DDS 的设计；
- ②DDS 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 $f_c=1\text{MHz}$, 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弦波形；
- ⑥通过开关（实验箱上的 K_i ）输入 DDS 的频率和相位控制字，并能用示波器观察加以验证；

（2）设计提高部分

- ①通过按键（实验箱上的 S_i ）输入 DDS 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能够同时输出正余弦两路正交信号；
- ③在数码管上显示生成的波形频率；
- 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；
- ⑥基于 DDS 的 AM 调制器的设计；
- ⑦自己添加其他功能。

二 工作原理

1 基本框图

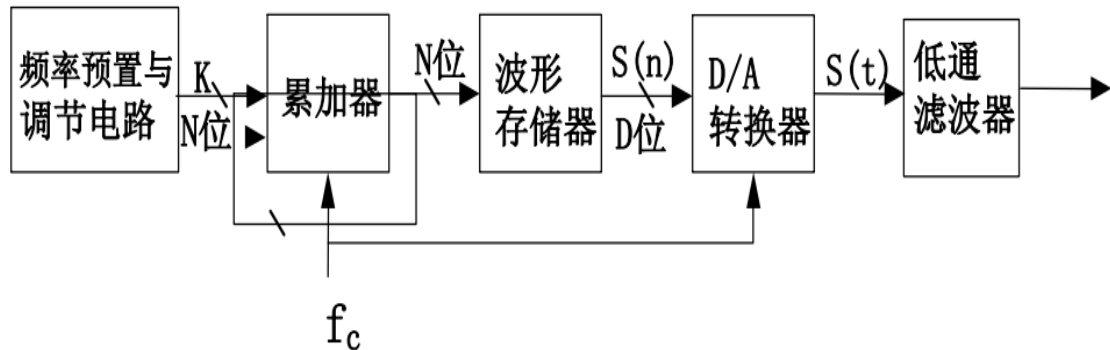
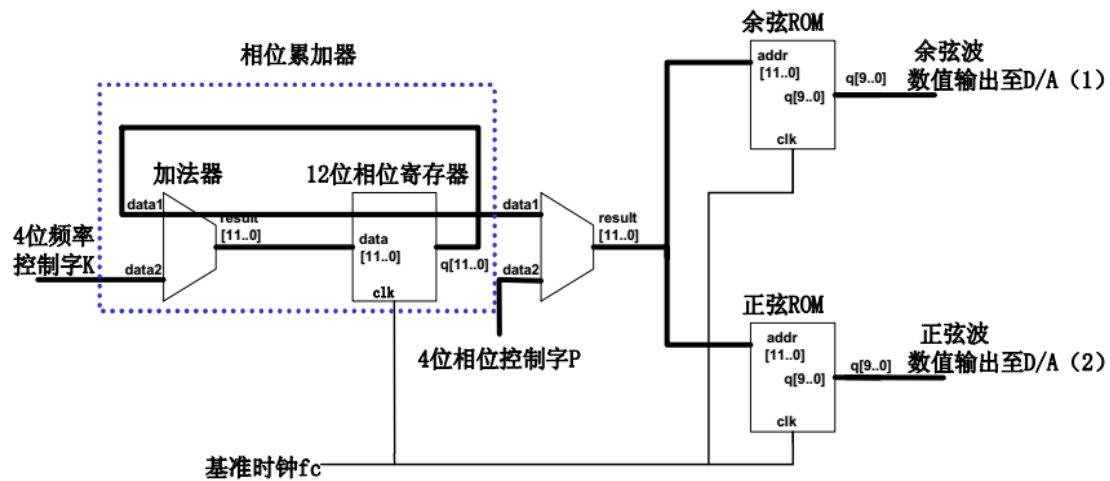


图 1



频率和相位均可控制的具有正弦和余弦输出的DDS
核心单元电路示意图

图 2

2 设计思想

DDS 的基本结构主要由相位累加器、相位调制器、正弦波数据表 (ROM)、D/A 转换器构成。相位累加器由 N 位加法器 N 位寄存器构成。每来一个时钟信号 $CLOCK$ ，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。由此，相位累加器在每一个时钟脉冲输入时，把频率控制

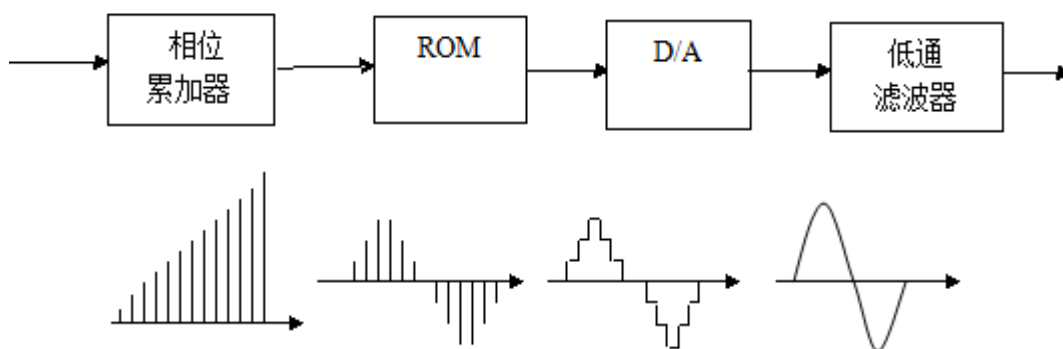
字累加以此，相位累加器输出的数据作为波形存储器的相位取样地址，这样就可把存储在波形存储器内的波形抽样值进行找表查出，完成相位到幅值的转换。

由于相位累加器为 N 位，相当于把正弦信号在相位上的精度定为 N 位，所以分辨率为 $\frac{1}{2^N}$ 。若系统时钟频率为 f_c ，频率控制字为 K ，则输出频率

$f_{out} = K \times f_c / 2^N$ 当系统输入时钟频率 f_c 不变时，输出信号的频率由频率控制字 K 所决定。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。

选取 ROM 的地址时，可以间隔选项，相位寄存器输出的位数 D 一般取 10-16 位，这种截取方法称为截断式用法，以减少 ROM 的容量。 D 太大会导致 ROM 容量的成倍上升，而输出精度受 D/A 位数的限制未有很大改善。

3 DDS 工作流程



三 各子模块设计原理

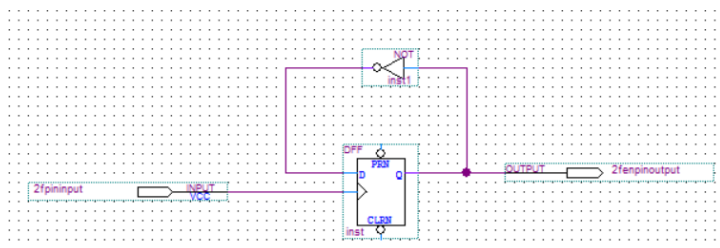
（一）基本设计

1 分频电路

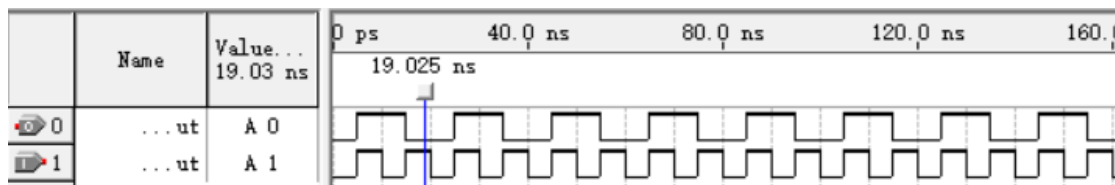
1.1 设计原理

设计的电路需要 $clk=1MHz$ ，数码管动态显示频率 $1KHz$ ，频率字输入计数时钟 $2Hz$ ，测频电路 $0.5Hz$ 。实验系统的输入时钟为 $48MHz$ ，要产生上述脉冲就需要分频出上述所需频率。可以通过不同的分频电路和语言实现分频。

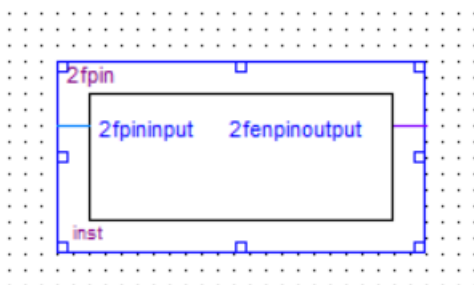
1.1.2 二分频电路



1.1.2 二分频电路

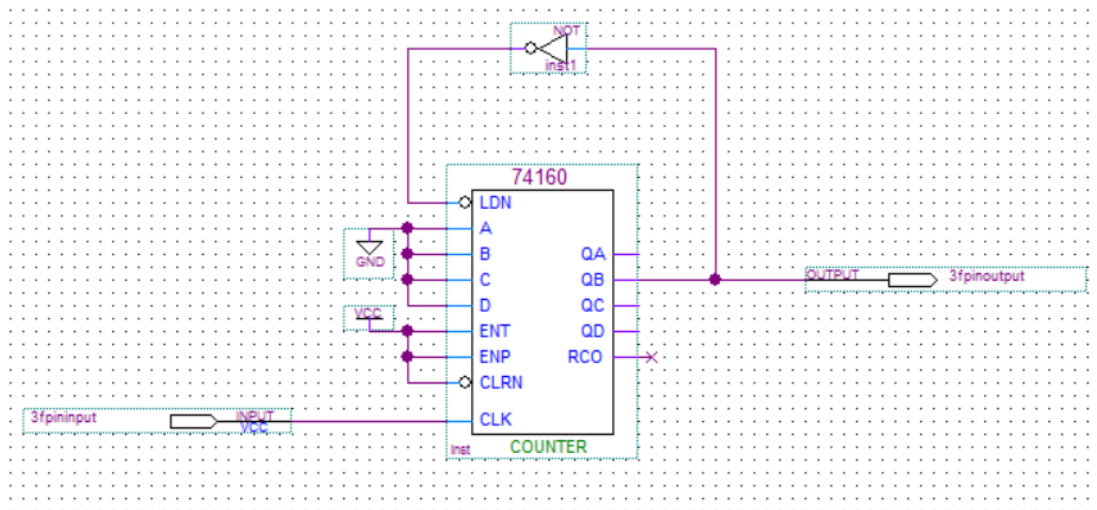


1.1.2 二分频仿真

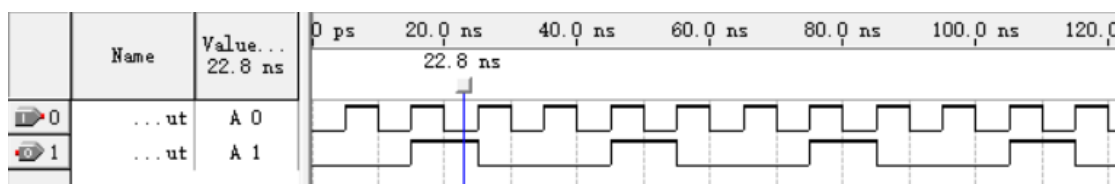


1.1.2 二分频封装

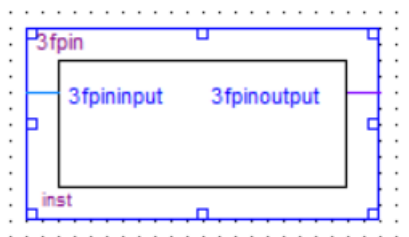
1.1.3 三分频电路



1.1.3 三分频电路

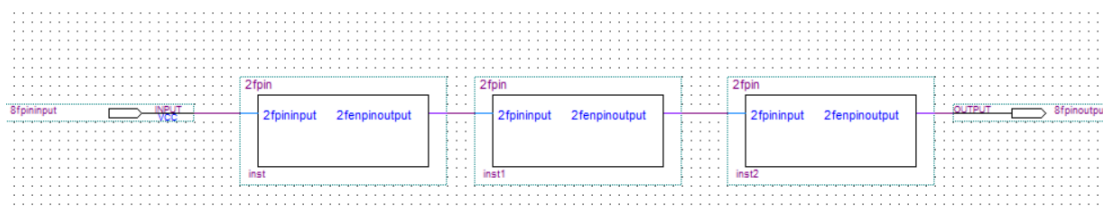


1.1.3 三分频仿真

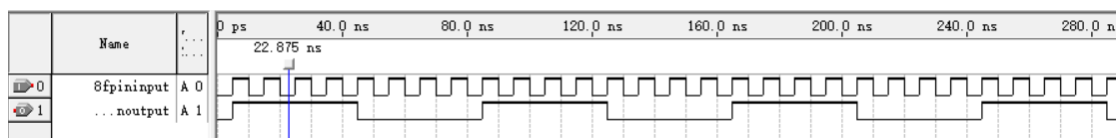


1.1.3 三分频封装

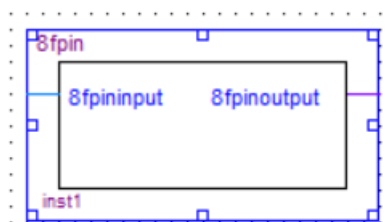
1.1.4 八分频电路



1.1.4 八分频仿真

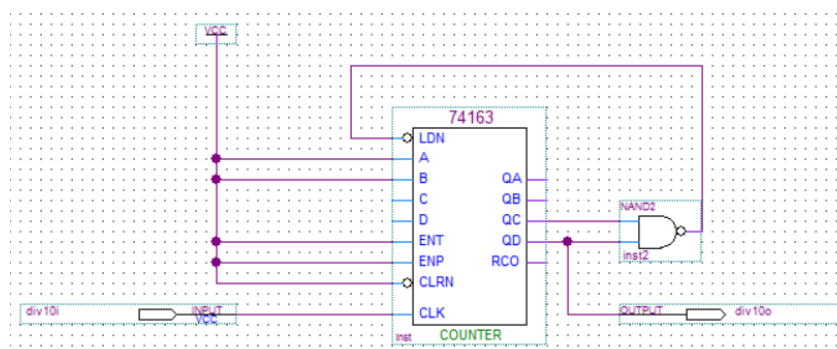


1.1.4 八分频仿真

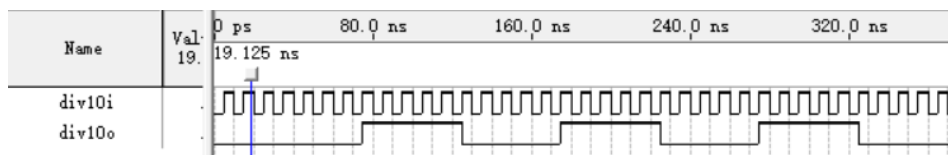


1.1.4 八分频封装

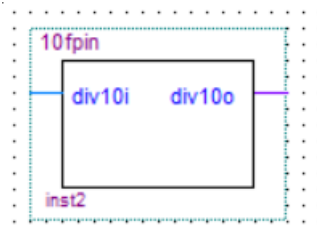
1.1.5 十分频电路



1.1.5 十分频电路

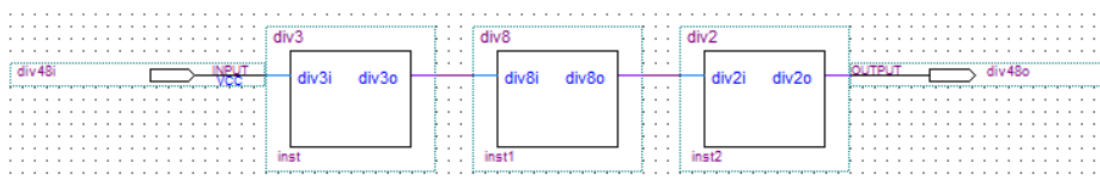


1.1.5 十分频仿真

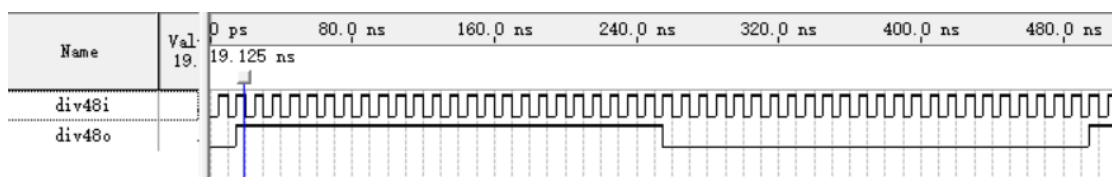


1.1.5 十分频封装

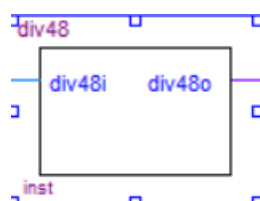
1.1.6 48 分频电路



1.1.6 24 分频电路

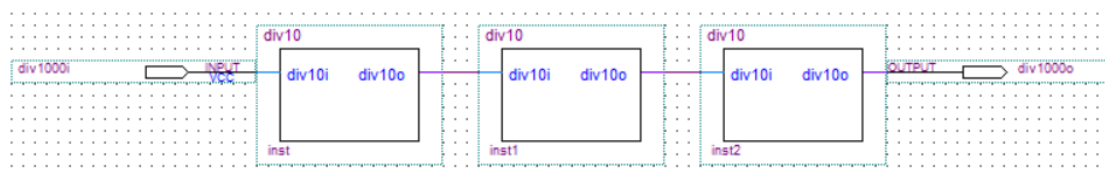


1.1.6 24 分频电路仿真

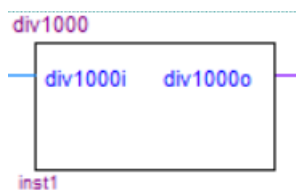


1.1.6 24 分频电路封装

1.1.7 48 分频电路

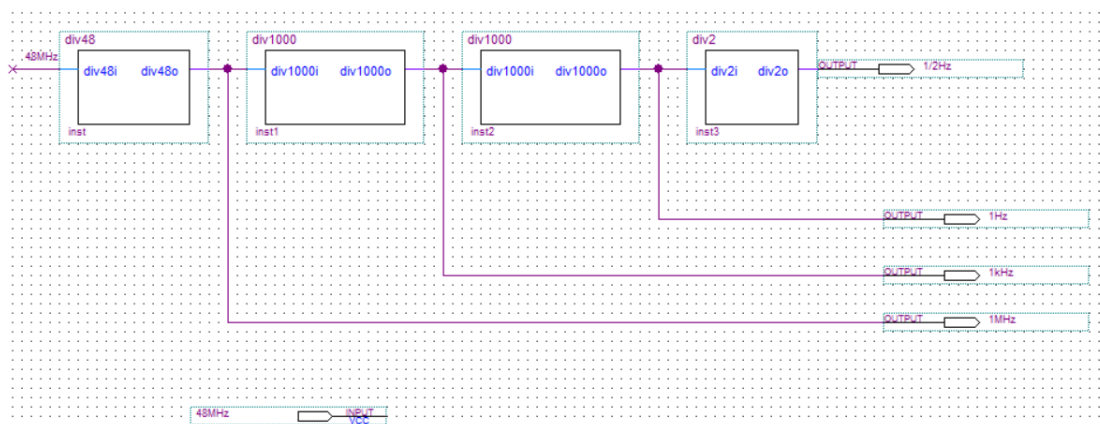


1.1.7 1000 分频电路

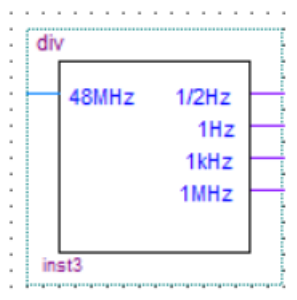


1.1.7 1000 分频电路封装

1.1.8 总分频电路



1.1.8 总分频电路

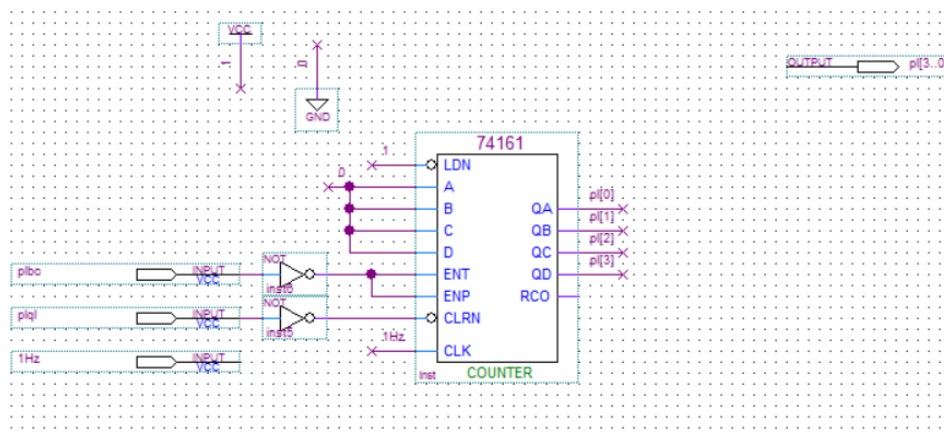


1.1.8 总分频电路封装

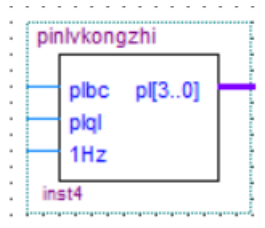
2 频率与相位字产生

2.1 频率控制字产生

频率控制字通过一个 74161 计数器产生,通过开关控制他保持或者计数或者清零,把 1Hz 的脉冲信号输入到 CLK,让 74161 以 1Hz 的频率自动计数,计数范围为 0-15,这个计数的数作为输出连接到下一级。



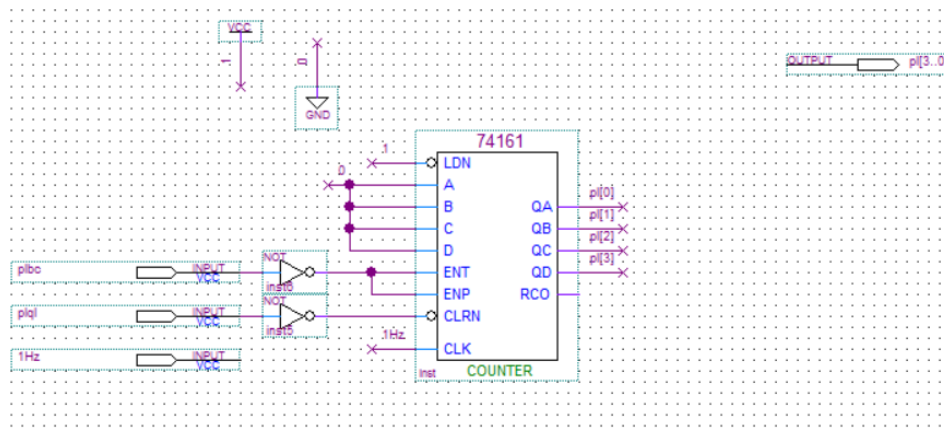
2.1 频率控制字产生电路



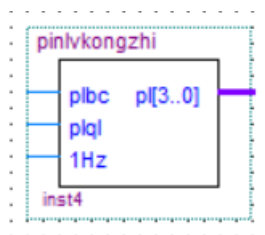
2.1 频率控制字产生电路封装

2.2 相位控制字产生

相位控制字和频率控制字产生的电路一样，所以我采用的是同一个电路。通过一个 74161 计数器产生，通过开关控制他保持或者计数或者清零，把 1Hz 的脉冲信号输入到 CLK，让 74161 以 1Hz 的频率自动计数，计数范围为 0-15，这个计数的数作为输出连接到下一级。



2.1 相位控制字产生电路



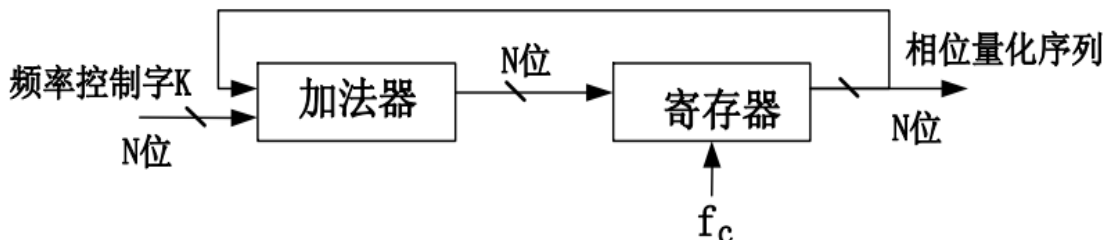
2.1 相位控制字产生电路封装

3 相位累加器

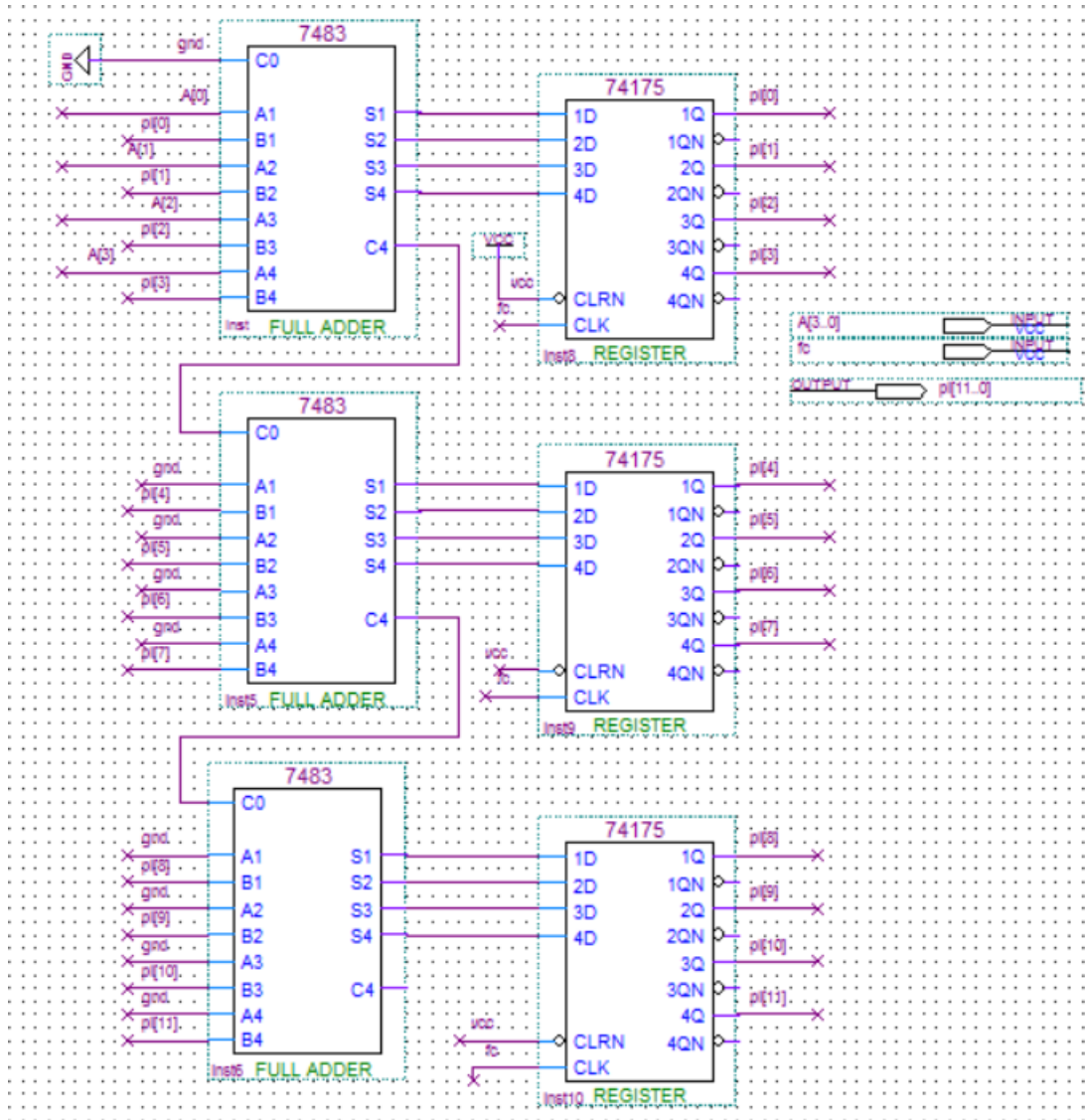
3.1 设计原理

相位累加器由 12 位加法器与 12 位寄存器级联构成。每来一个时钟脉冲，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。寄存器将加法器的上一个时钟作用后所产生的相位数据反馈至加法器的输入端，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。

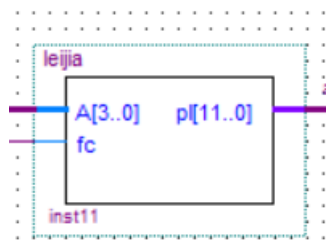
3.2 原理图



3.3 12 位加法器以及寄存器



3.3 12 位加法器以及寄存器电路



3.3 12 位加法器以及寄存器电路封装

4 波形存储器

4.1 设计原理

波形存储器是进行波形相位——幅值的转换。ROM 的 N 位地址把 $0^\circ \sim 360^\circ$ 的正弦角度离散成具有 2^N 个样值的序列；ROM 的 D 位数据位把 2^N 个样值的幅值

量化为 D 位二进制数据。其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 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。

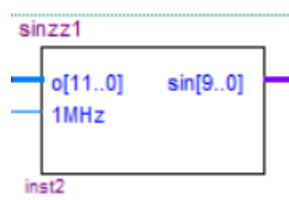


4.2 LPM_ROM 的设置

利用 excel 表格生成 ROM 的初始化数据，在 quartus II 中新建 mif 文件，将 excel 中的数据拷贝到 mif 文件中，保存 mif 文件。在元器库中选取 LPM-ROM，设置 ROM 的信息，数据宽度为 10bits，数据个数为 4096；设置 ROM 为寄存器输出，不需要时钟和异步清零信号；指定 ROM 的初始化数据来源，选择刚刚所生成的 mif 文件。然后点击“finish”即可生成 ROM 单元了。

4.2.1 sin 波形存储器

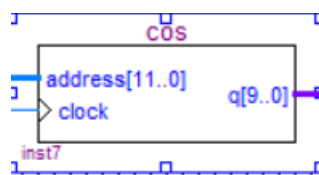
正弦波形存储器 Lpm-romsin，ROM 的 $N=12$ 位地址：把 0—4095 的正弦角度离散成有 $2^N=4096$ 个样值的序列。ROM 的 $D=10$ 位数据位，则 $2^N=4096$ 个样值的幅值量化为 $D=10$ 位二进制数据。生成 ROM 数据有两种方法，一是采取直接输入方式，二是采取语言编程得到所需数据。由于本实验所需数据量较大，为了方便，我们采取利用 Excel 软件，生成*.xls 应用程序，然后在 quartusII 中生成*.mif 文件，ROM 通过*.mif 文件可以把 mif 文件中的数据读出来，再经过 D/A 转换器，低通滤波器，就可以在示波器显示平滑的波形。



4.2.1 正弦波封装图

4.2.2 cos 波形存储器

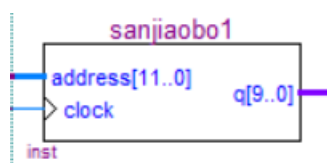
原理同生成 sin 波形存储器。



4.2.2 余弦波封装图

4.2.3 三角波形存储器

原理同生成 sin 波形存储器。



4.2.3 三角波封装图

4.2.4 锯齿波形存储器

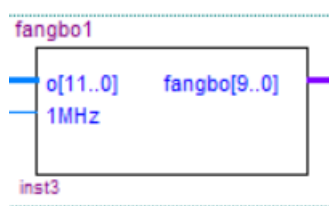
原理同生成 sin 波形存储器。



4.2.4 锯齿波封装图

4.2.5 方波形存储器

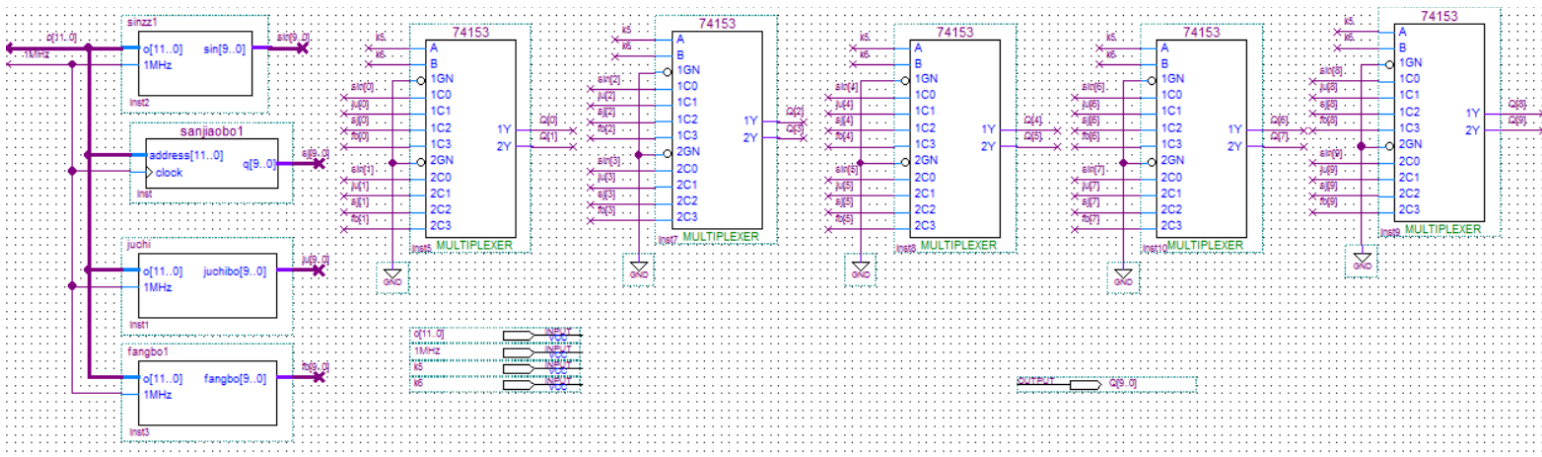
原理同生成 sin 波形存储器。



4.2.5 方波封装图

5 波形选择电路

由于在示波器上只能一次性输出两路波形，而其中的一路必须为余弦波，所以这时正弦波、方波、锯齿波和三角波的输出便是一个问题。所以，我采用了一



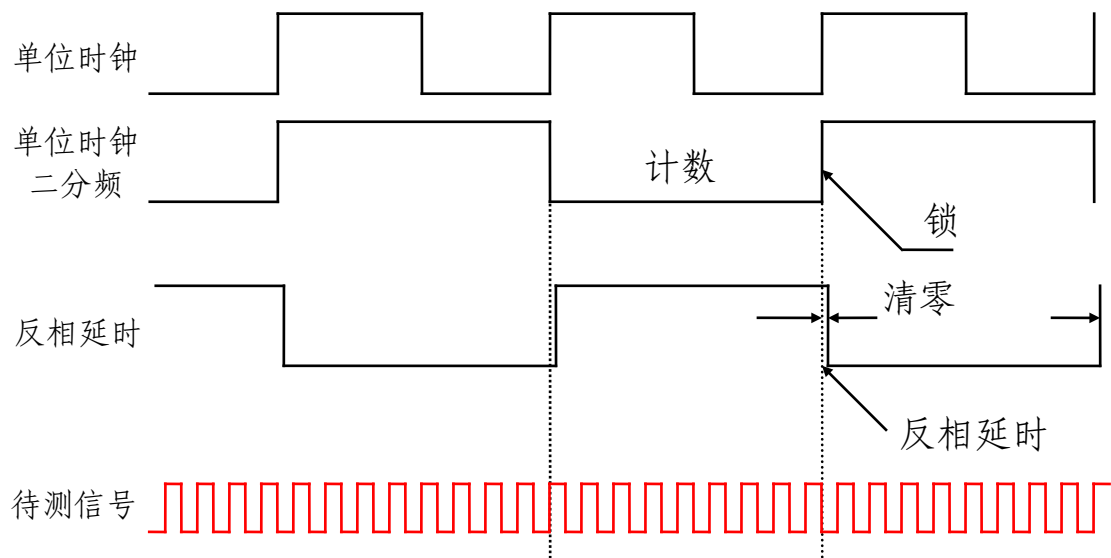
个模 4 计数器，k5、k6 是波形计数开关，从而实现了两个开关选择四种波形，计到 00 时输出正弦波，01 时输出三角波，10 时输出锯齿波，11 时输出方波。

图 5 波形选择电路

6 测频电路

6.1 设计原理

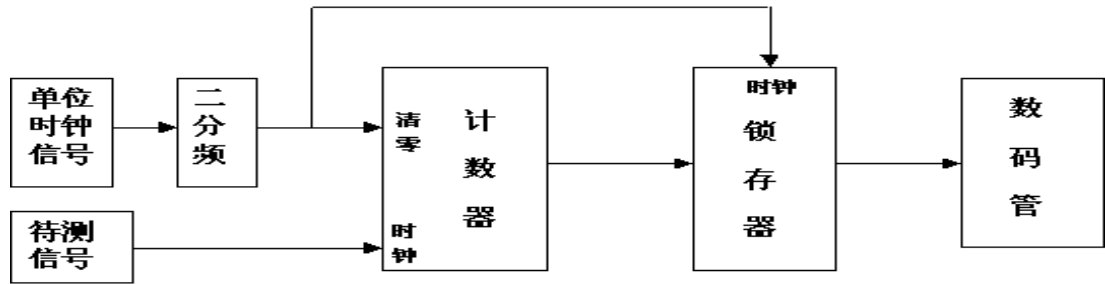
测频就是计算 1 秒钟内脉冲的个数。我们利用计数器和锁存器实现这一功能。由于累加器以频率控制字 K 为间隔,当累加器满量时就会产生一次溢出,完成一次周期性的动作,这个周期也就是 DDS 信号的一个频率周期,所以将累加器的最高位作为测频电路技术器的脉冲。



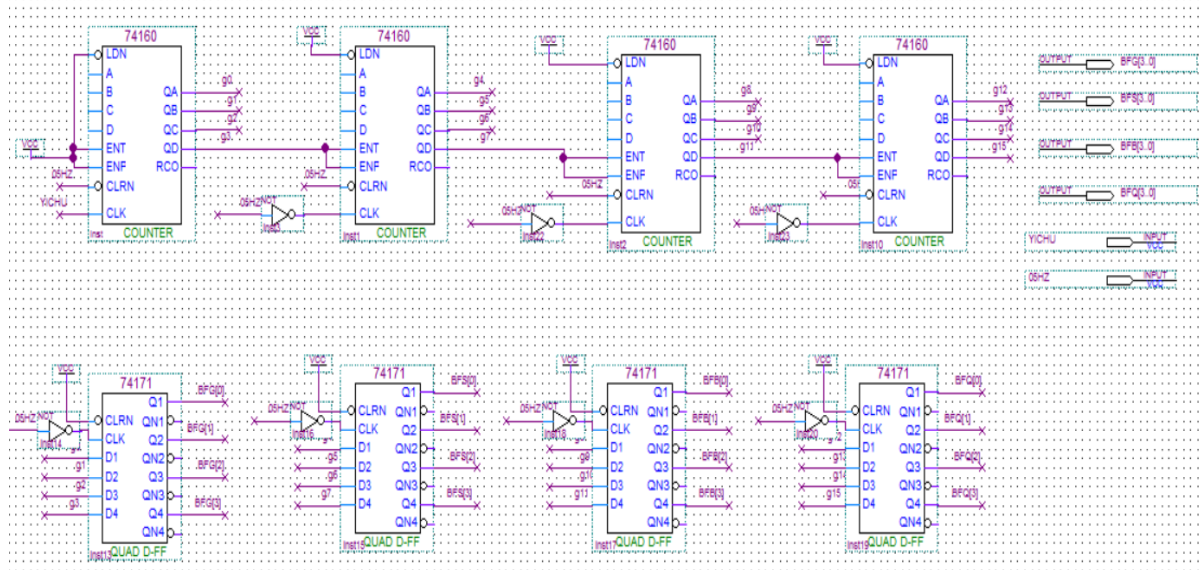
将 0.5Hz 脉冲送入锁存器的时钟端,0.5Hz 反相延时后的脉冲送入计数器的清零端。这样就使计数器在 2s 的脉冲周期内,1s 内清零,1s 内计数。由于锁存器

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。

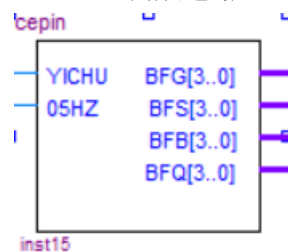
6.2 原理图



6.3 设计电路



6.3 测频电路

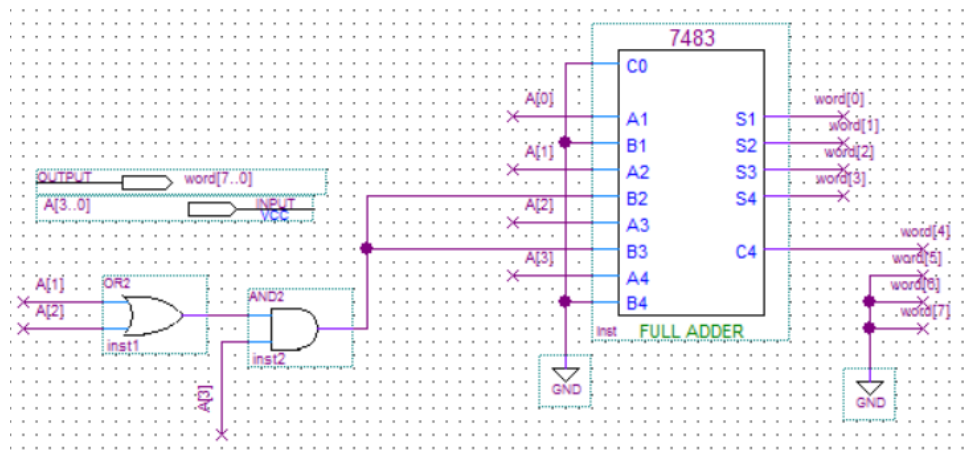


6.3 测频电路封装

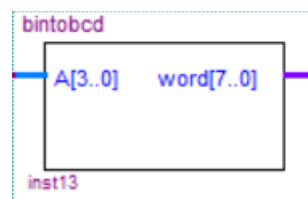
7 译码显示电路

7.1 BCD 码转换电路

由于频率字和控制字输出的是二进制码，所以要通过一个 BCD 码转换电路，把频率字和控制字转换成 BCD 码再进行输出。



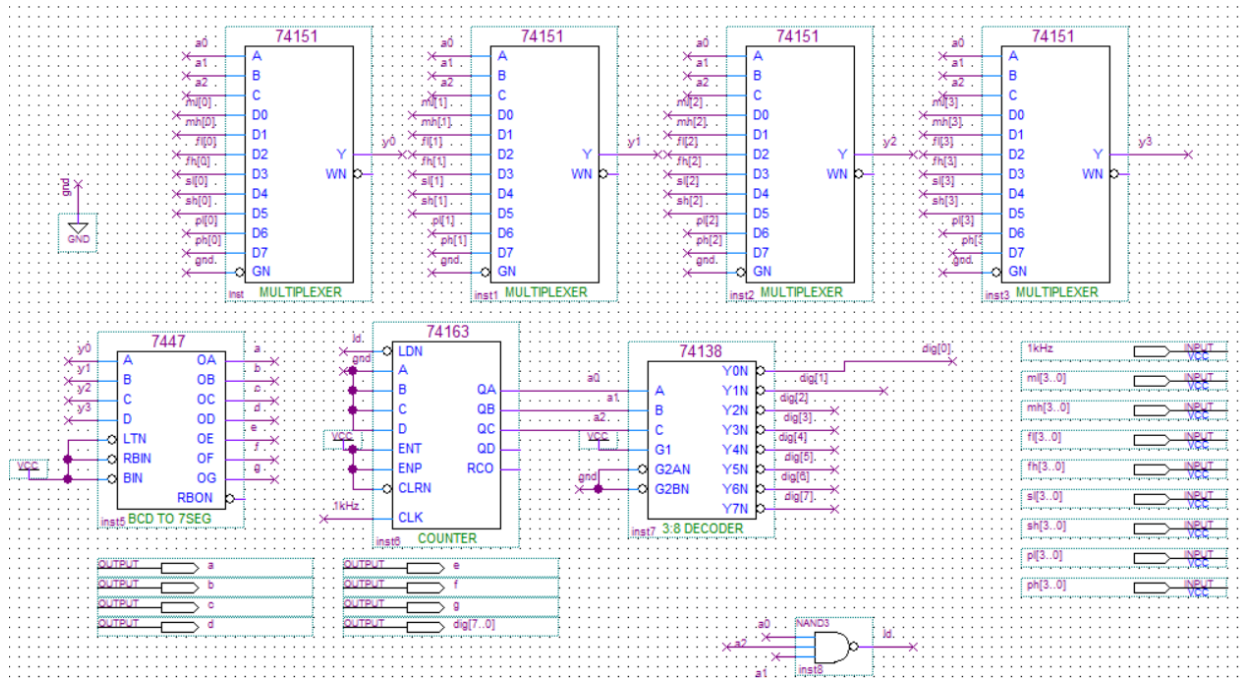
7.1 BCD 码转换电路



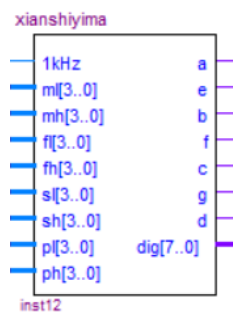
7.1 BCD 码转换电路封装

7.2 译码显示电路

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（时、分、秒各两位），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 a0、a1、a2 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 a0、a1、a2，74138 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。前 4 个显示测频得到的频率，后四个分别是频率控制字和相位控制字。



7.2 译码显示电路



7.2 译码显示电路封装

(二) 附加功能

1 AM 调制

1.1 设计原理

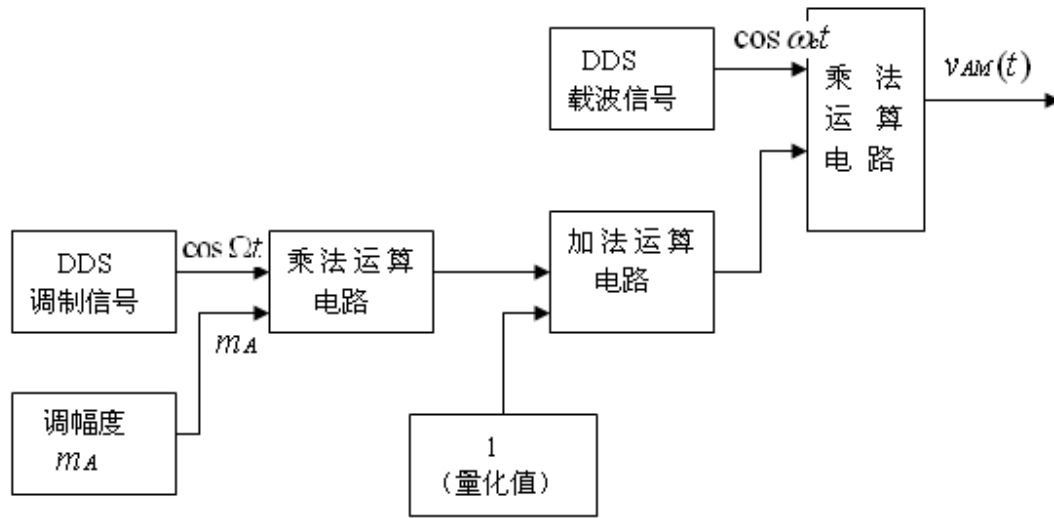
通信理论中将信号调制定义为调制信号对载波的幅度、频率和相位进行交换，AM 即标准调制信号，除了来自消息的基带信号外，还包含了直流信号，它是调制后输出信号既含载波分量又含边带分量的标准调制信号。

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： $m_A = V_{\Omega m} / V_{cm}$ 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。

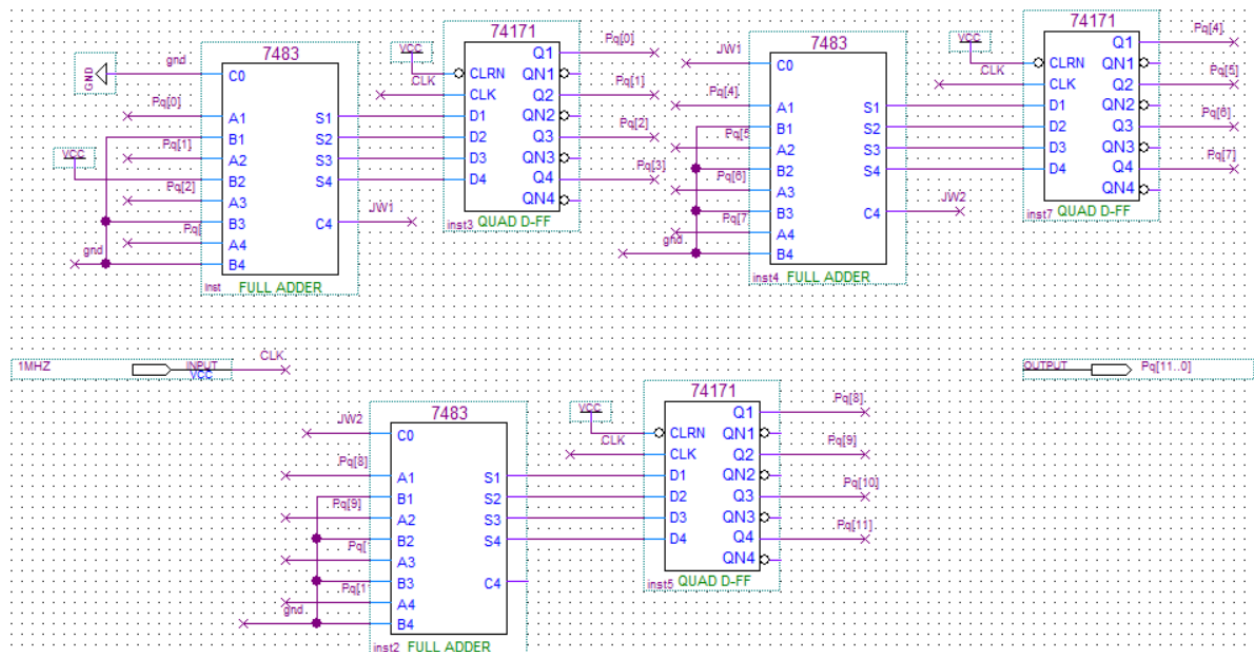


1.2 用于产生调幅信号的 ROM

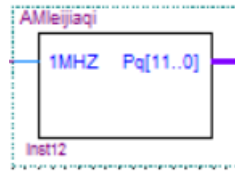
在 AM 调制中，ROM 必须是有符号数，所以各种波形的 ROM 必须重新做。

1.3 调制信号发生器

产生调制信号的电路与基本功能的电路类似，主要目的是产生频率控制字和相位控制字可调的不同载波信号。



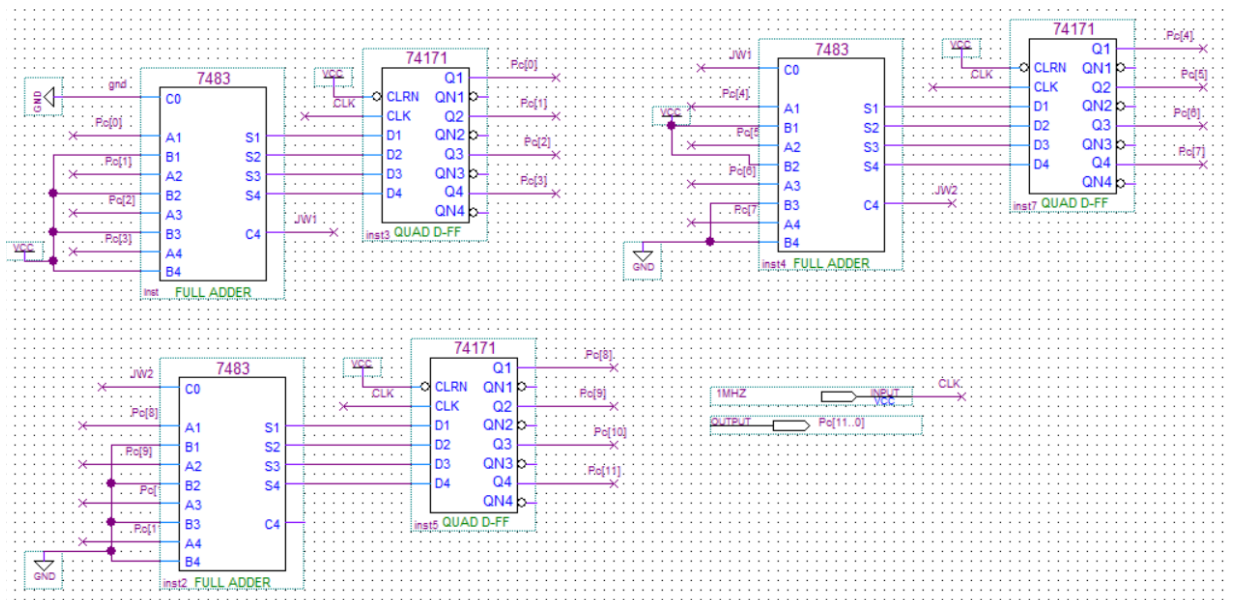
1.3 调制信号产生电路



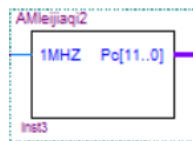
1.3 调制信号产生电路封装

1.4 载波信号发生器

载波信号的频率一般远远高于调制信号。这里，我使用了余弦函数作为载波函数，其输出波形的频率主要与频率控制字有关，可以通过调节频率控制字来达到目的。需要注意的是，载波信号的频率一定要比较大。

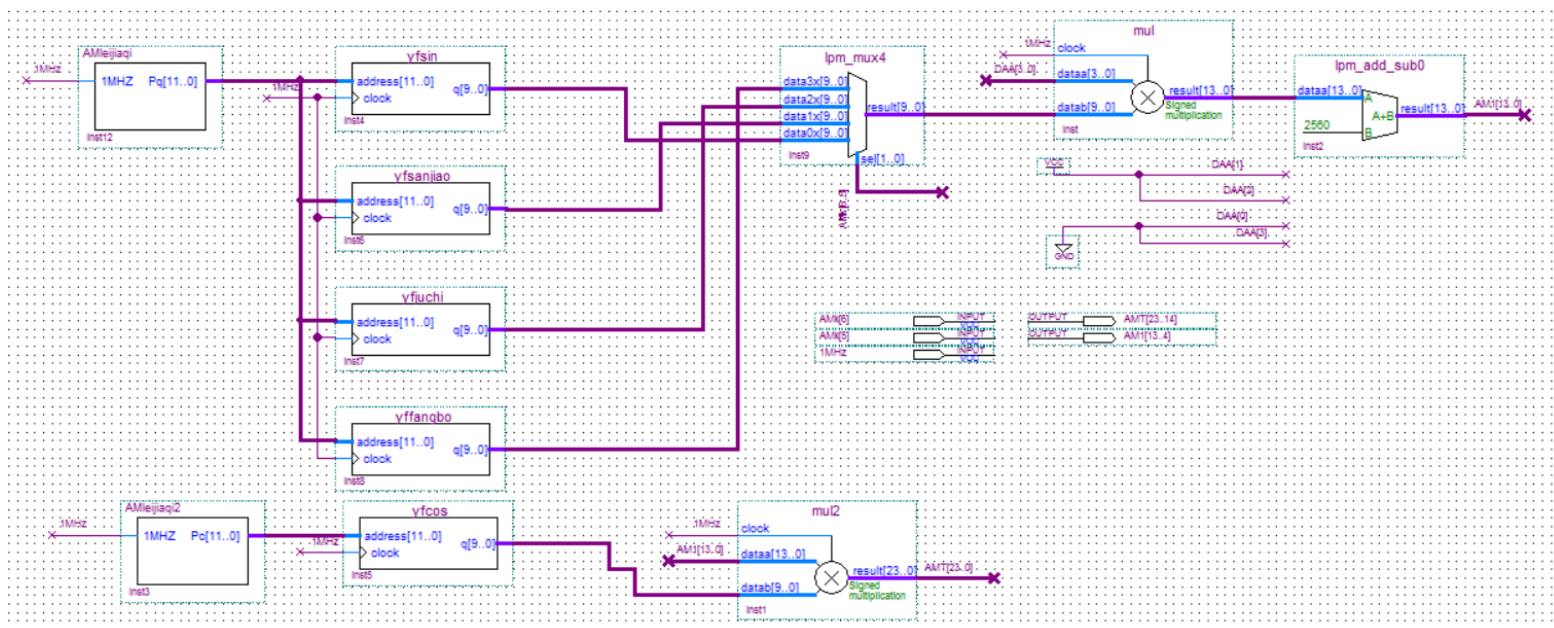


1.4 载波信号产生电路

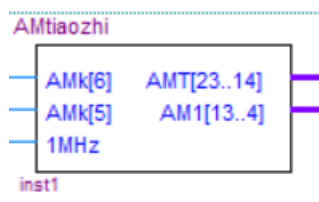


1.4 载波信号产生电路封装

1.5 AM 调制电路



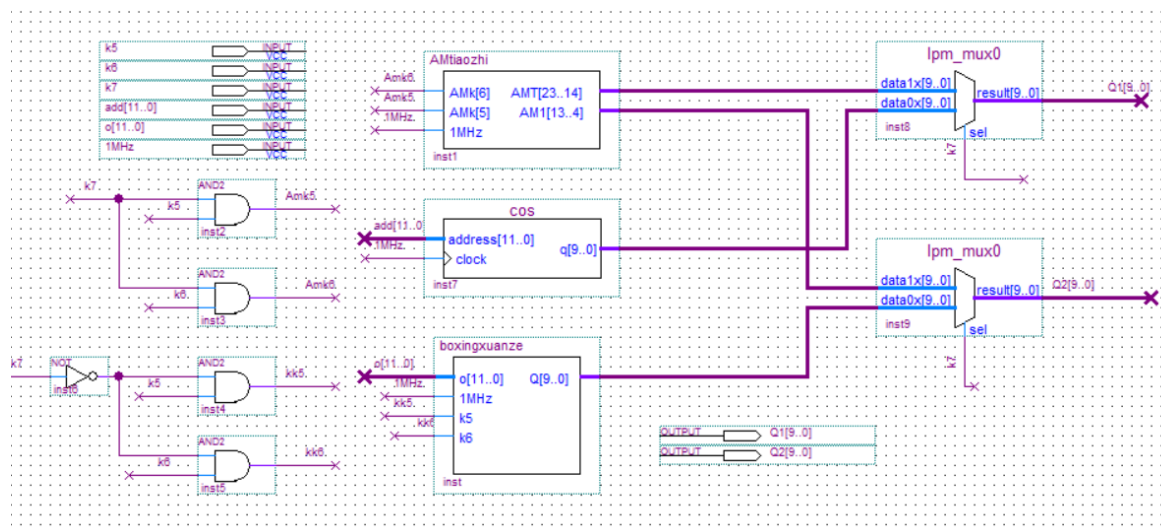
1.5 AM 调制电路



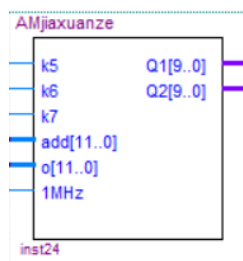
1.5 AM 调制电路封装

1.6 选择工作模式电路

通过一个 k7 来控制 AM 调制和正常工作状态。k7 为 1 的时候，选通的 AMK5 和 AMK6，这个时候是 AM 调制工作方式，k7 为 0 的时候，选通的事 kk5 和 kk6，这个时候正常工作方式。再通过两个二选一数据选择器来选通哪个数据作为输出，输出到示波器上，这个数据选择器也是通过开关 k7 来控制。



1.6 选择工作模式电路



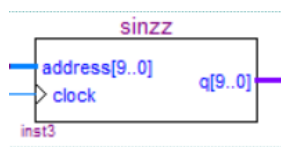
1.5 选择工作模式电路封装

2 一片 1/4 ROM 实现 sin 波形存储

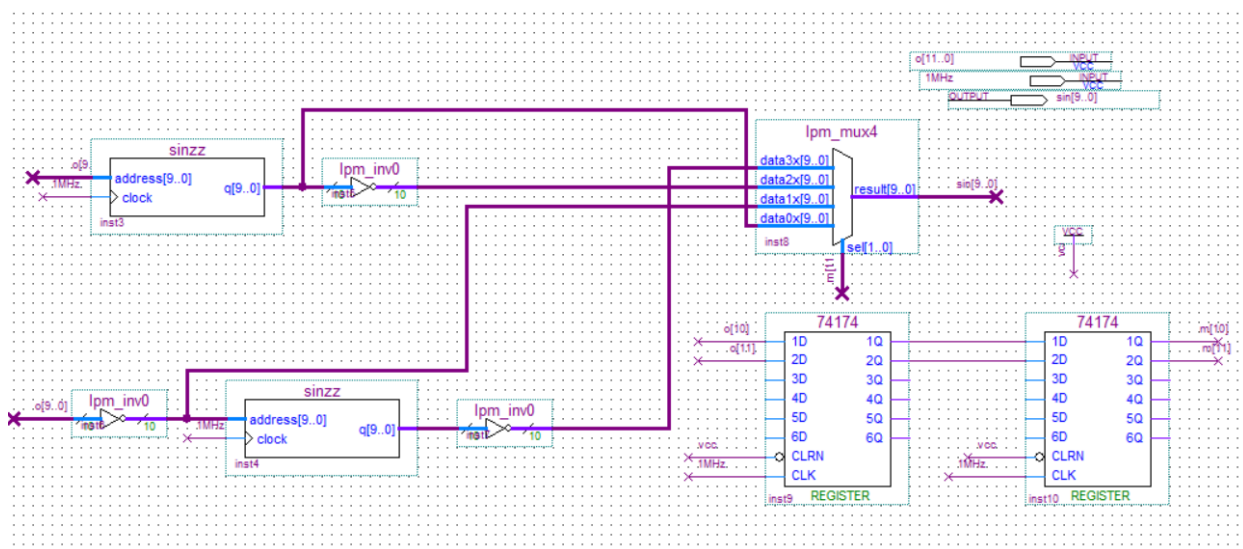
下面建立正弦波波形存储器，其采样数据由 MATLAB 程序获得，其程序如下：

```
clear tic;
t=2*pi/4096;
t=[0:t:2*pi];
y=511*sin(t)+512;
z=y';
round(z);
```

将数据的前 $\frac{1}{4}$ 导入 Excel 中，取整后得到建立 *.mif 文件所用数据再在 Quartus 里新建 mif 文件并生成 $\frac{1}{4}$ ROM，从而实现 10 位相位地址向 10 位 sin 幅度值的转变。



由相位累加器输出的是 12 位地址，即 $0-2\pi$ 的地址。由 \sin 对称性知， $\frac{\pi}{2}-\pi$ 内各对应点的值可由 $0-\frac{\pi}{2}$ 地址取反得到，类似的， $\pi-\frac{3\pi}{2}$ 可由 $0-\frac{\pi}{2}$ 幅度值取反得到，最后， $\frac{3\pi}{2}-2\pi$ 可由 $0-\frac{\pi}{2}$ 先对幅度值取反再地址取反得到。这里的取反电路可由非门得到，最终可实现 \sin 函数 $0-2\pi$ 地址向幅值转换。



四、程序下载、仿真与调试

4.1 调试

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。在调试过程中，切记要设置 TOP-LEVEL，把要调试的原理图设置在顶层再进行调试运行，这是最常犯的错误之一；同时调试并不能代表所设计的原理图适用于所设计要求，只是代表调试成功，在设计过程中需要注意要求注意细心设计，有可能会出现问题级联后不成功的现象，这是我们应该避免的问题。

4.2 仿真

当整个电路设计完成后，需要进行模拟仿真，观察波形。选择“File-New”，打开“other files”标签项，选择“Vector Waveform File”。在弹出对话框右边“Name”下的空白框里双击鼠标，在新弹出的对话框里点击“Node Finder”按钮。在新对话框中的“Filter”中选择“Pins:all”后，点击“Pins:all”后点击“list”按钮，则“Nodes Found”对话框中列出了本工程的所有输入输出节点。双击所要节点，则右边的“selected nodes”框中出现了所选的节点，完成节点的添加。再设置时钟脉冲后，点击进行波形仿真观察波形。

4.3 管脚分配

编译通过后要进行管脚分配，选择“Processing-start complication”进行全编译，编译通过后要进行管脚分配，选择“Assignments→Pins”，在打开的对话框中的“Location”栏中选择相应的管脚填入，并将未用到的管脚置为三态，最后将程序下载到 SmartSOPC 实验系统中运行，检验结果是否正确。

48MHz	Input	PIN_A10	7	B7_N0	2.5 V (default)	
a	Output	PIN_F10	7	B7_N0	2.5 V (default)	
b	Output	PIN_E11	7	B7_N0	2.5 V (default)	
c	Output	PIN_F12	7	B7_N0	2.5 V (default)	
d	Output	PIN_C12	7	B7_N0	2.5 V (default)	
DA1	Output	PIN_F6	8	B8_N0	2.5 V (default)	
DA2	Output	PIN_H6	1	B1_N0	2.5 V (default)	
DACLK	Output	PIN_A2	8	B8_N0	2.5 V (default)	
dig[7]	Output	PIN_J13	6	B6_N0	2.5 V (default)	
dig[6]	Output	PIN_H16	6	B6_N0	2.5 V (default)	
dig[5]	Output	PIN_L16	5	B5_N0	2.5 V (default)	
dig[4]	Output	PIN_L14	5	B5_N0	2.5 V (default)	
dig[3]	Output	PIN_D16	7	B7_N0	2.5 V (default)	
dig[2]	Output	PIN_A18	7	B7_N0	2.5 V (default)	
dig[1]	Output	PIN_E14	7	B7_N0	2.5 V (default)	
dig[0]	Output	PIN_H15	6	B6_N0	2.5 V (default)	
e	Output	PIN_D12	7	B7_N0	2.5 V (default)	
f	Output	PIN_E13	7	B7_N0	2.5 V (default)	
g	Output	PIN_C14	7	B7_N0	2.5 V (default)	
k5	Input	PIN_R18	5	B5_N0	2.5 V (default)	
k6	Input	PIN_R17	5	B5_N0	2.5 V (default)	
k7	Input	PIN_P18	5	B5_N0	2.5 V (default)	
plbaochi	Input	PIN_V9	3	B3_N0	2.5 V (default)	
plqingling	Input	PIN_U10	4	B4_N0	2.5 V (default)	

Q1[9]	Output	PIN_P2	2	B2_N0	2.5 V (default)
Q1[8]	Output	PIN_F7	8	B8_N0	2.5 V (default)
Q1[7]	Output	PIN_C1	1	B1_N0	2.5 V (default)
Q1[6]	Output	PIN_E1	1	B1_N0	2.5 V (default)
Q1[5]	Output	PIN_G1	1	B1_N0	2.5 V (default)
Q1[4]	Output	PIN_H1	1	B1_N0	2.5 V (default)
Q1[3]	Output	PIN_K5	2	B2_N0	2.5 V (default)
Q1[2]	Output	PIN_L1	2	B2_N0	2.5 V (default)
Q1[1]	Output	PIN_M1	2	B2_N0	2.5 V (default)
Q1[0]	Output	PIN_P1	2	B2_N0	2.5 V (default)
Q2[9]	Output	PIN_M2	2	B2_N0	2.5 V (default)
Q2[8]	Output	PIN_L2	2	B2_N0	2.5 V (default)
Q2[7]	Output	PIN_K2	2	B2_N0	2.5 V (default)
Q2[6]	Output	PIN_H2	1	B1_N0	2.5 V (default)
Q2[5]	Output	PIN_G2	1	B1_N0	2.5 V (default)
Q2[4]	Output	PIN_F3	1	B1_N0	2.5 V (default)
Q2[3]	Output	PIN_D2	1	B1_N0	2.5 V (default)
Q2[2]	Output	PIN_M3	2	B2_N0	2.5 V (default)
Q2[1]	Output	PIN_B2	1	B1_N0	2.5 V (default)
Q2[0]	Output	PIN_C2	1	B1_N0	2.5 V (default)
xwbaochi	Input	PIN_B9	8	B8_N0	2.5 V (default)
xwqingling	Input	PIN_B10	7	B7_N0	2.5 V (default)

五、实验中遇到的问题及解决办法

1、做节省 ROM 时会有毛刺出现，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，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，我决定采用两个寄存器，这样起到了延时的作用，可以避免毛刺的出现。

2、在做 AM 调制时候也遇到了不少的问题，在一开始的时候，并没有考虑到载波信号频率要比调制信号的频率大得多的问题，所以调制出来的波形一直都不太理想，后来经过向同学请教还有自己查阅书籍，找到了这个问题，并作了相应的修改，从而解决了这个问题。

3、AM 调制的时候我们的 ROM 用到的是有符号数，所以所有的 ROM 必须重做，一开始我并没有意识到这个问题，所以做完之后调试，发现出来的波形并不正确。经过一步步的检查问题和思考问题，我最终发现了这个问题，最终对其进行了修改，才出现了正确的波形。

4、在一开始加入 AM 调制的时候并没有对 DA1_MODE、DA2_MODE 进行修改，DA_MODE=0（无符号数）DA_MODE=1（有符号数），因为 AM 调制的时候我们用到的是有符号数，而正常工作的时候我们用到的事无符号数，这个时候就应该通过

开关 k7 来控制 DA_MODE 的取值，当 k7 为 1 的时候，DA_MODE 为 1，当 k7 为 0 的时候，DA_MODE 为 0。

六、实验感想

这是第二次使用 Quartus2 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，这就需要我们另外查阅相关的资料进行设计。经过第一次的实验，我明白到，我们设计的电路是一个整体，在设计电路的时候必须要考虑到开关及数码管的数量是否够用，如果不够时必须要使用复用。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，这都是需要我们去考虑的。

首先要把基础电路搭出来必须要有以下几个模块：分频、控制字产生、累加器、寄存器、测频、显示译码。与第一个实验相比较，没有涉及到的就是累加器和寄存器，所以这两个需要另外设计，其他模块相对来说比较容易实现，分频模块的原理以及显示译码更是基本上差不多。但是显示译码模块这时要输出的是 8 个数码管，而实验一中我只用到了 6 个，所以这个需要对电路进行一点小修改。基本电路搭好之后，最后通过程序下载、仿真与调试又遇到了新的问题，那就是图像中存在着很长的毛刺，我又考虑了出现毛刺的原因是在输出 $\frac{\pi}{2} - \pi$ 的波形时，会从 $0 - \frac{\pi}{2}$ 的第一个点开始取反，这是波形输出从最高点一下落到最低点就会产生的。可不可以先使地址从 $0 - \frac{\pi}{2}$ 变化到 $\frac{\pi}{2} - \pi$ ，然后再输出各点对应的幅度值，后面两个段类似。通过查阅相关的书籍以及文献资料，我决定采用两个寄存器，这样起到了延时的作用，可以避免毛刺的出现。

完成了基础电路的调试之后，我又尝试着做基于 DDS 的 AM 调制，我就在基础电路的基础上，加入了 AM 调制。因为我的编程能力比较弱，所以使用的是逻辑原理图。其实 AM 调制并没有想象中的那么难，关键是要弄懂它的原理。需要注意的是，AM 调制中要求使用有符号的 ROM，所以需要再次使用 matlab 编程，生成*.mif 文件形成有符号 ROM，还有需要考虑的是载波信号的频率要求比调制信号大，所以我在载波信号频率产生的电路里的高位加了数值，而在调制信号频率产生的电路里的低位加了数值，最后经过量化卷积输出，在示波器上得到调制后

的信号。

第二次实验，我发现只要有了正确的设计思路，遇到问题一步一步地解决，采用自上而下的设计方法，设计电路时考虑到全局，那么一定可以实现所涉及的功能的。还有就是一定要细心，有条不紊地进行，不能过于急于求成。

两次的成功设计相信是个很大的收获，也满满地激发了我动手实践的热情，对以后的学习和生活起到了很大的促进作用。