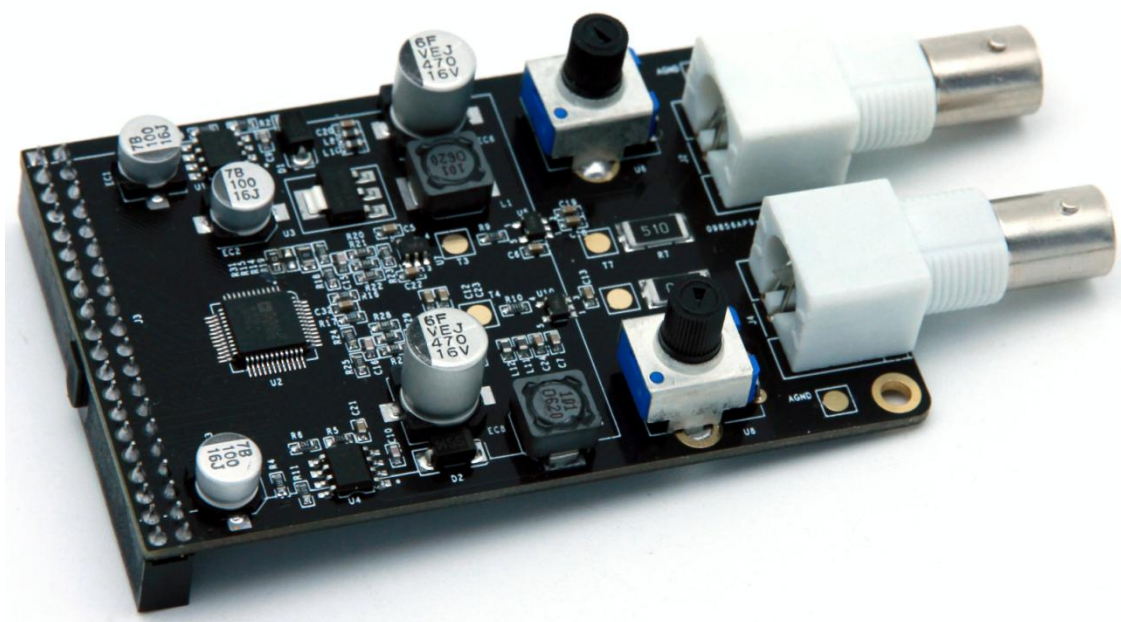


# AD9767 双通道正弦波产生例程

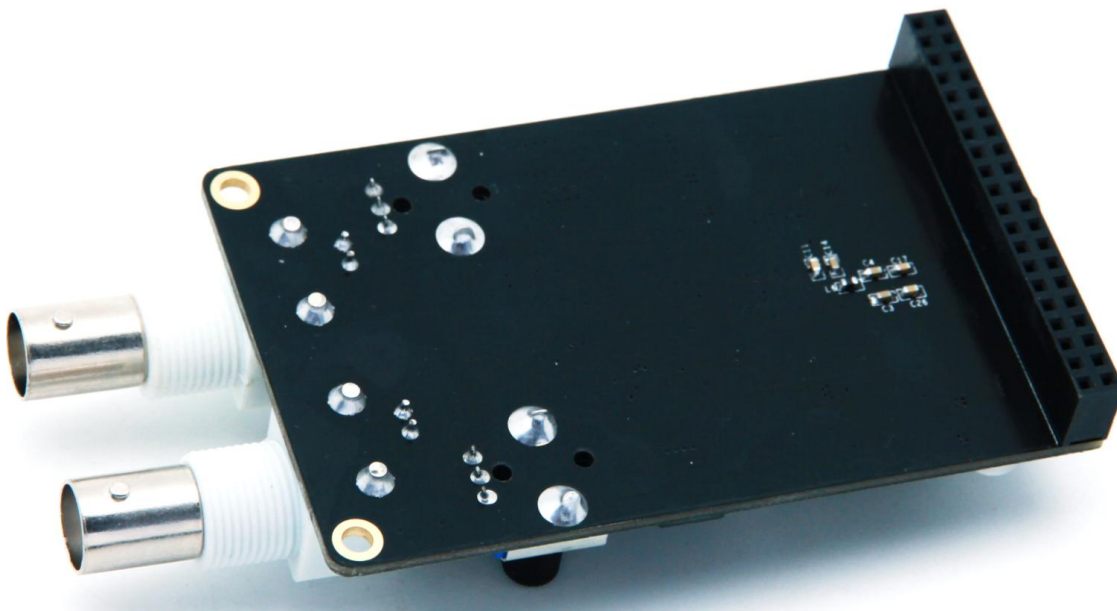
## 1 实验简介

本实验练习使用 AN9767 模块，实验中使用的模块是采用 ANALOG DEVICES 公司的 AD9767 芯片，支持独立双通道、14 位、125MSPS 的数模转换。在教程中介绍了利用该模块与 FPGA 开发板相连输出双通道 14 位的正弦波，然后通过示波器查看把输出的正弦波的波形。

AN9767 模块实物照片如下：



AN9767 模块正面图



AN9767 模块背面图

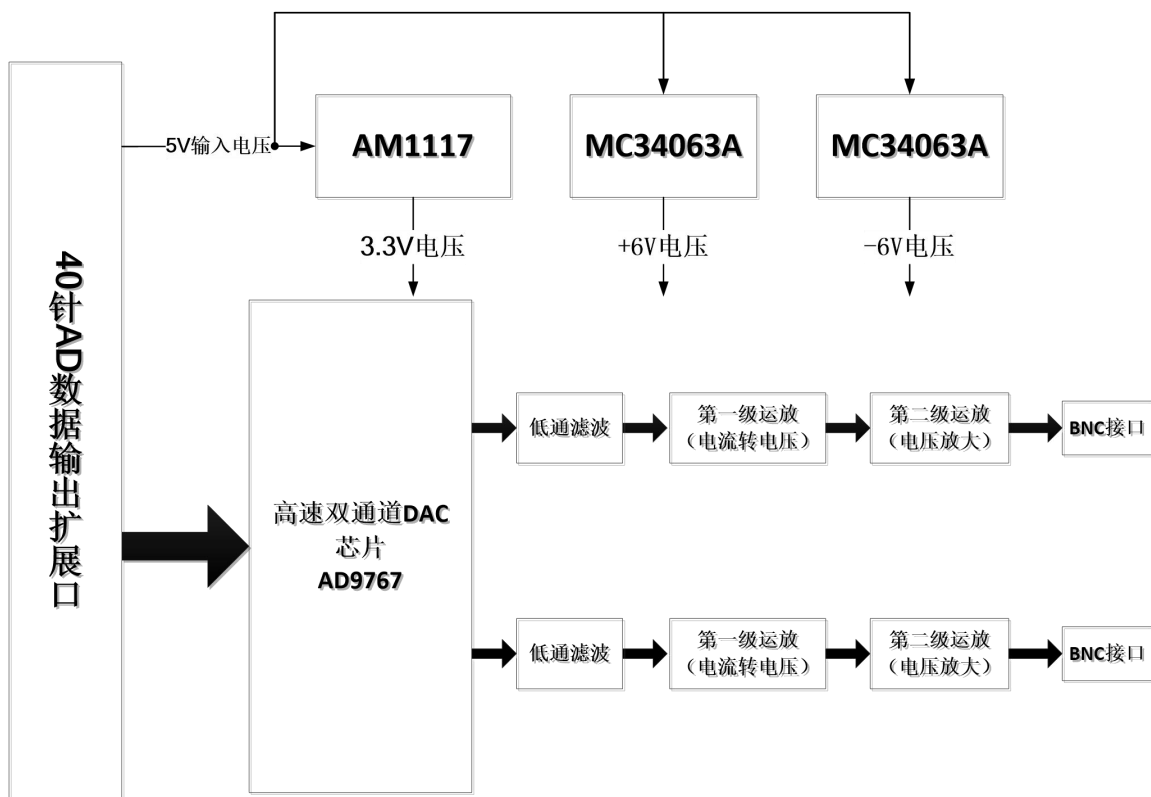
AN9767 双通道 DA 模块的详细参数:

- DA 转换芯片: AD9767;
- 通道数: 2 通道;
- DA 转换位数: 14bit;
- DA 更新速率: 125 MSPS;
- 输出电压范围:  $-5V \sim +5V$ ;
- 模块 PCB 层数: 4 层, 独立的电源层和 GND 层;
- 模块接口: 40 针 2.54mm 间距排座, 方向向下;
- 工作温度:  $-40^{\circ} \sim 85^{\circ}$  模块使用芯片均满足工业级温度范围
- 输出接口: 2 路 BNC 模拟输出接口 (用 BNC 线可以直接连接到示波器) ;

## 2 实验原理

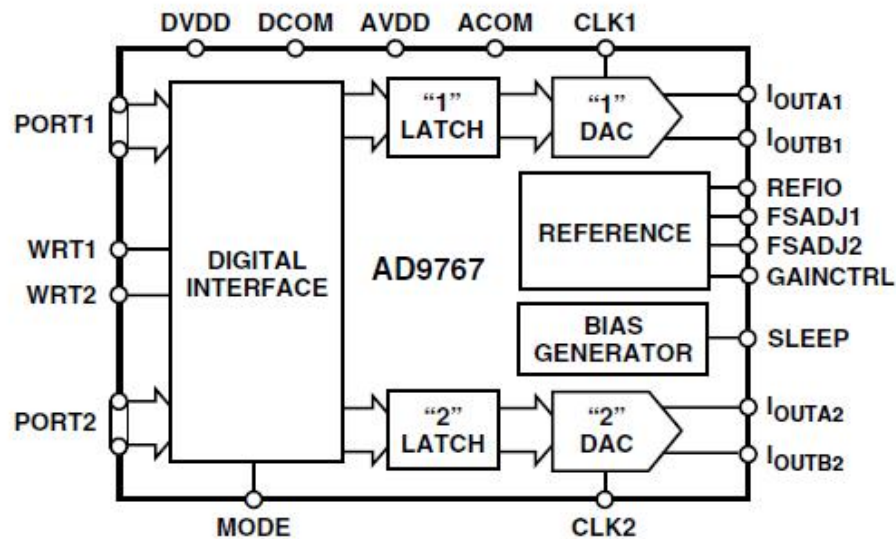
### 2.1 AN9767 模块原理框图

AN9767 模块的原理设计框图如下:



## 2.2 AD9767 芯片简介

AD9767 是双端口、高速、双通道、14 位 CMOS DAC, 芯片集成两个高品质 TxDAC+®内核、一个基准电压源和数字接口电路, 采用 48 引脚小型 LQFP 封装。器件提供出色的交流和直流性能, 同时支持最高 125 MSPS 的更新速率。AD9767 的功能框图如下:



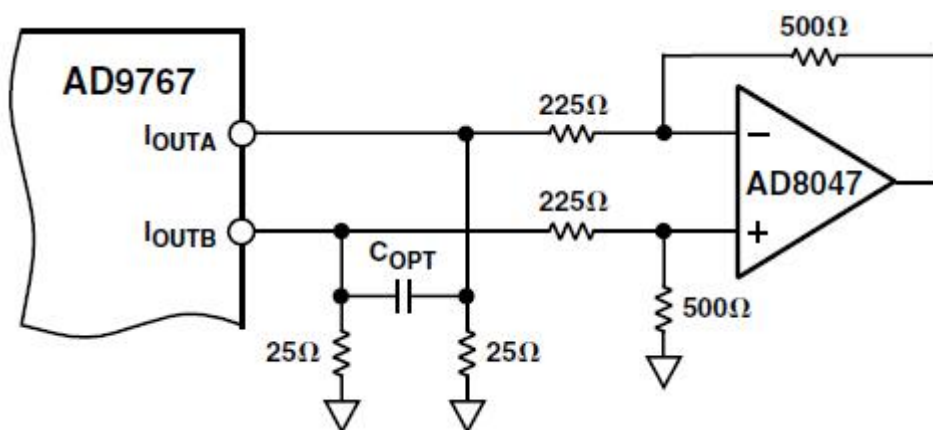
## 2.3 电流电压转换及放大

AD9767 的两路 DA 输出都为补码形式的电流输出  $I_{OUTA}$  和  $I_{OUTB}$ 。当 AD9767 数字输入为满量程时（DAC 的输入的 14 位数据都为高）， $I_{OUTA}$  输出满量程的电流输出 20mA。 $I_{OUTB}$  输出的电流为 0mA。具体的电流和 DAC 的数据的关系如下公式所示：

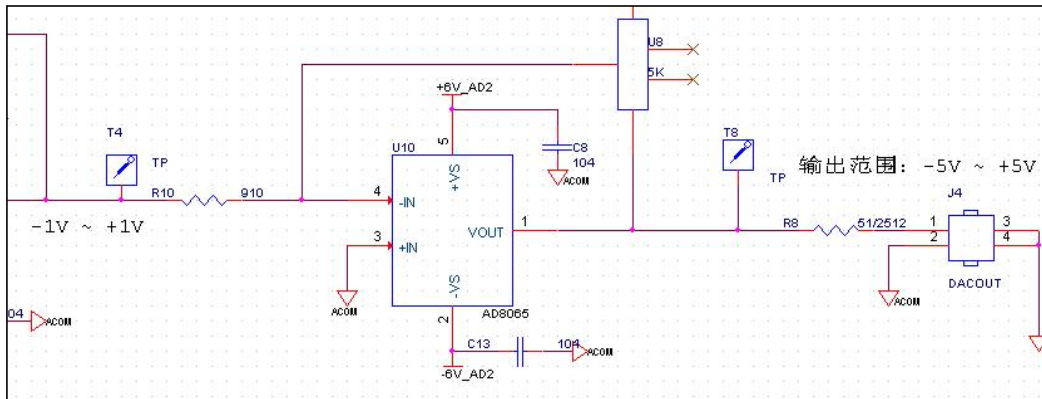
$$I_{OUTA} = (DAC\ CODE / 16384) \times I_{OUTFS}$$
$$I_{OUTB} = (16383 - DAC\ CODE) / 16384 \times I_{OUTFS}$$

其中  $I_{OUTFS} = 32 \times I_{REF}$ ，在 AN9767 模块设计中， $I_{REF}$  的值由电阻 R16 的值决定，如果 R16=19.2K，那  $I_{REF}$  的值就是 0.625mA。这样  $I_{OUTFS}$  的值就是 20mA。

AD9767 输出的电流通过第一级运放 AD6045 转换成 -1V~+1V 的电压。具体的转换电路如下图所示：



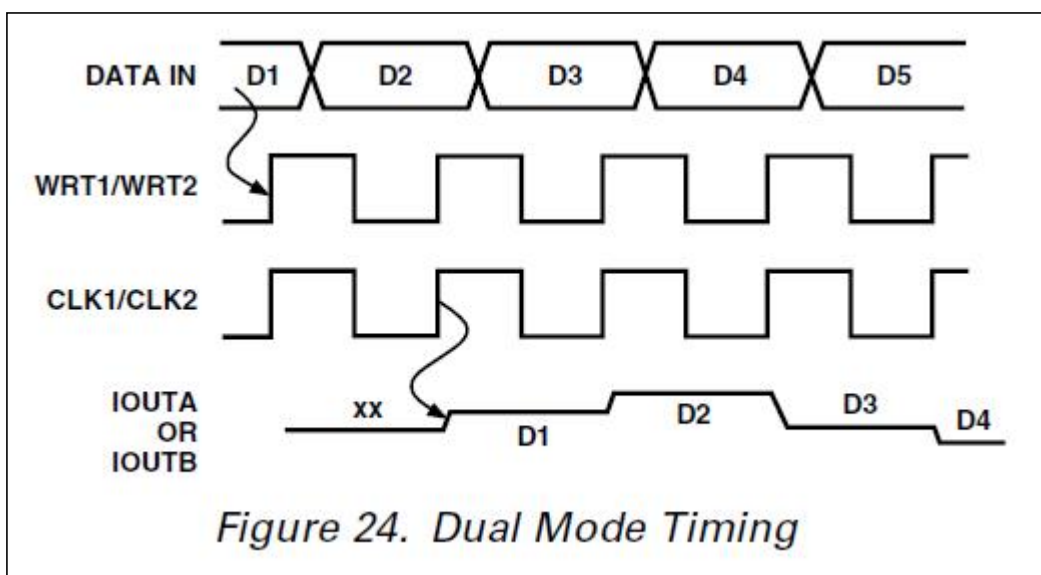
第一级运放转换后的 -1V~+1V 的电压通过第二级运放变换到更高幅度的电压信号，这个运放的幅度大小可以通过调整板上的可调电阻来改变。通过第二级运放，模拟信号的输出范围高达 -5V~+5V。



下表为数字输入信号和各级运放输出后的电压对照表：

DAC 数据输入值	AD9767 电流输出	第一级运放输出	第二级运放输出
3fff(14 位全高)	+20mA	-1V	+5V
0(14 位全低)	-20mA	+1V	-5V
2000 (中间值)	0mA	0V	0V

AD9767 芯片的数字接口可以通过芯片的模式管脚(MODE)来配置成双端口模式(Dual)或者交叉 (Interleaved)模式。在 AN9767 模块设计中，AD9767 芯片是工作在双端口模式，双通道的 DA 数字输入接口是独立分开的。双端口模式(Dual)的数据时序图如下图所示：

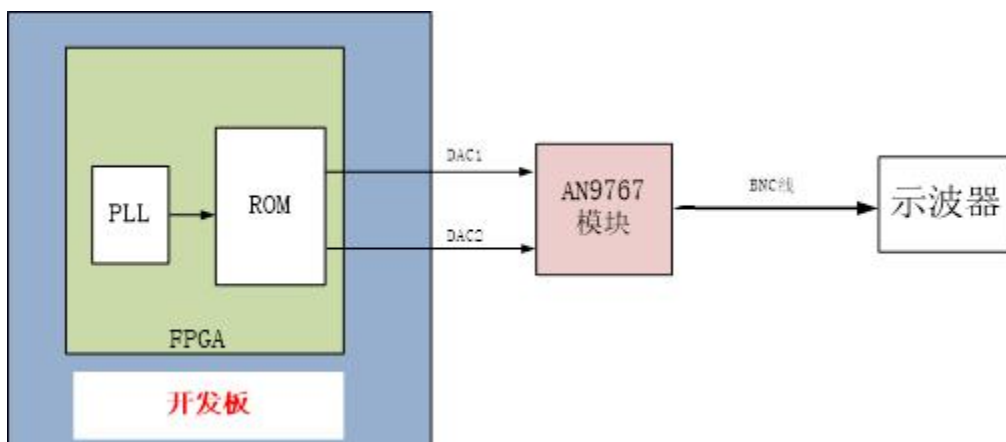


给 AD9767 芯片的 DA 数据通过时钟 CLK 和写信号 WRT 的上升沿输入到芯片进行 DA 转换。

### 3 程序设计

例程中提供了 AN9767 模块的 DA 测试程序，通过 AN9767 模块来实现正弦波信号的输出。

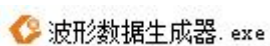
正弦波测试程序是通过读取 FPGA 内部的一个 ROM 中存储的正弦波数据，然后把正弦波的数据输出到 AN9767 模块进行数模的转换，从而得到正弦波的模拟信号。正弦波测试程序的示意图如下：



#### 3.1 生成 ROM 初始化文件

程序中我们会用到一个 ROM 用于存储 1024 个 14 位的正弦波数据, 首先我们需要准备 ROM 的初始化文件。以下为生成正弦波 ROM 数据文件的方法：

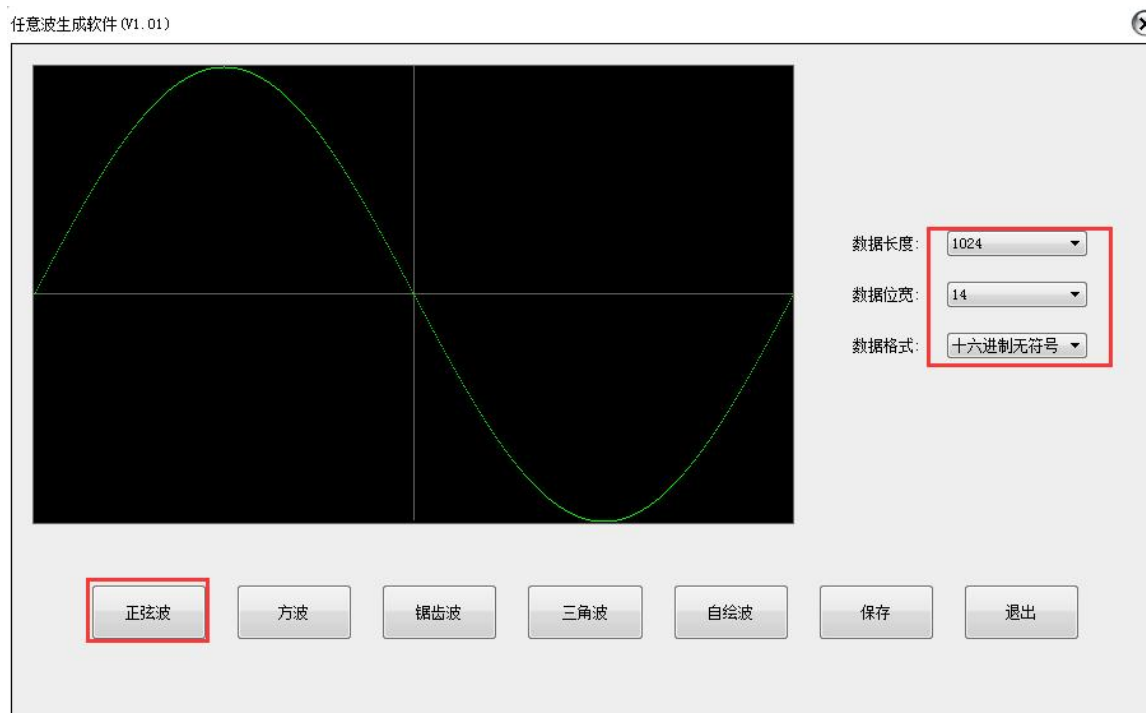
在软件工具及驱动文件夹下找到工具，其图标如下所示：



1. 双击.exe 打开工具，打开界面如下：

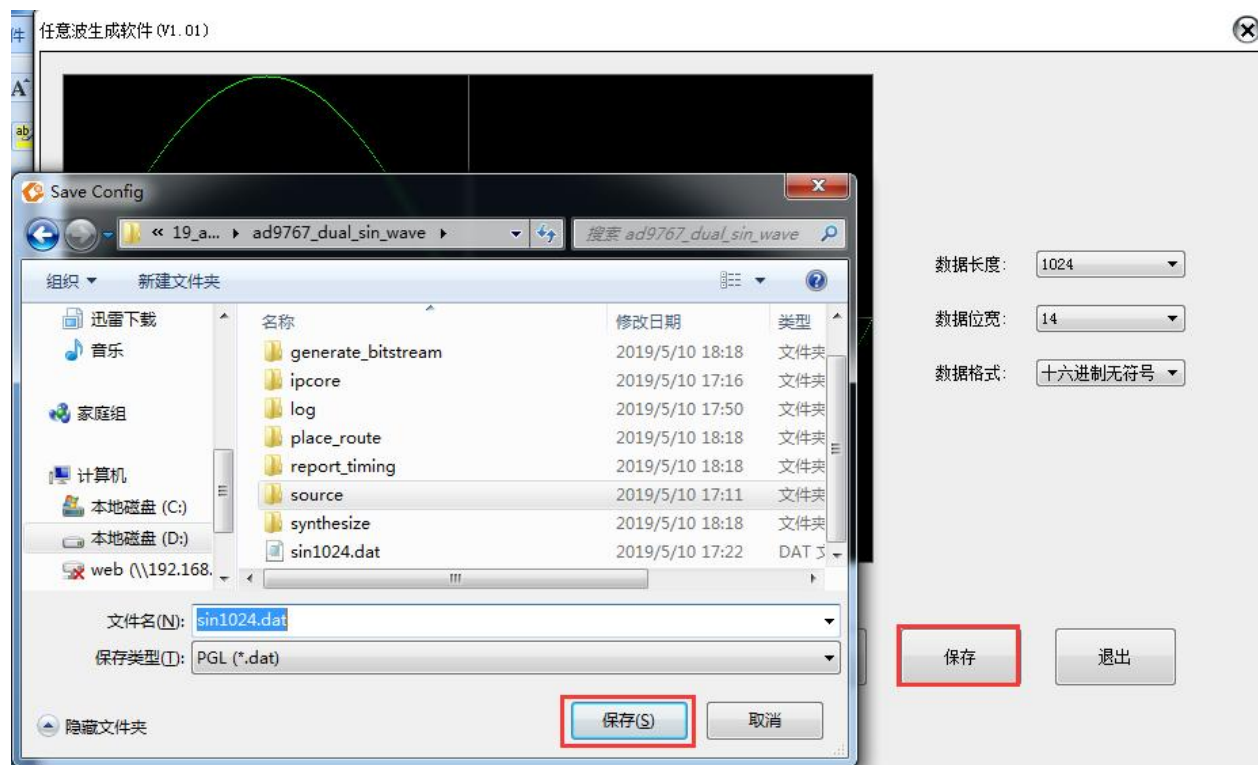


2. 可以根据需要自选波形，本例程中选择正弦波，数据长度 1024，数据位宽 14，其它默认：

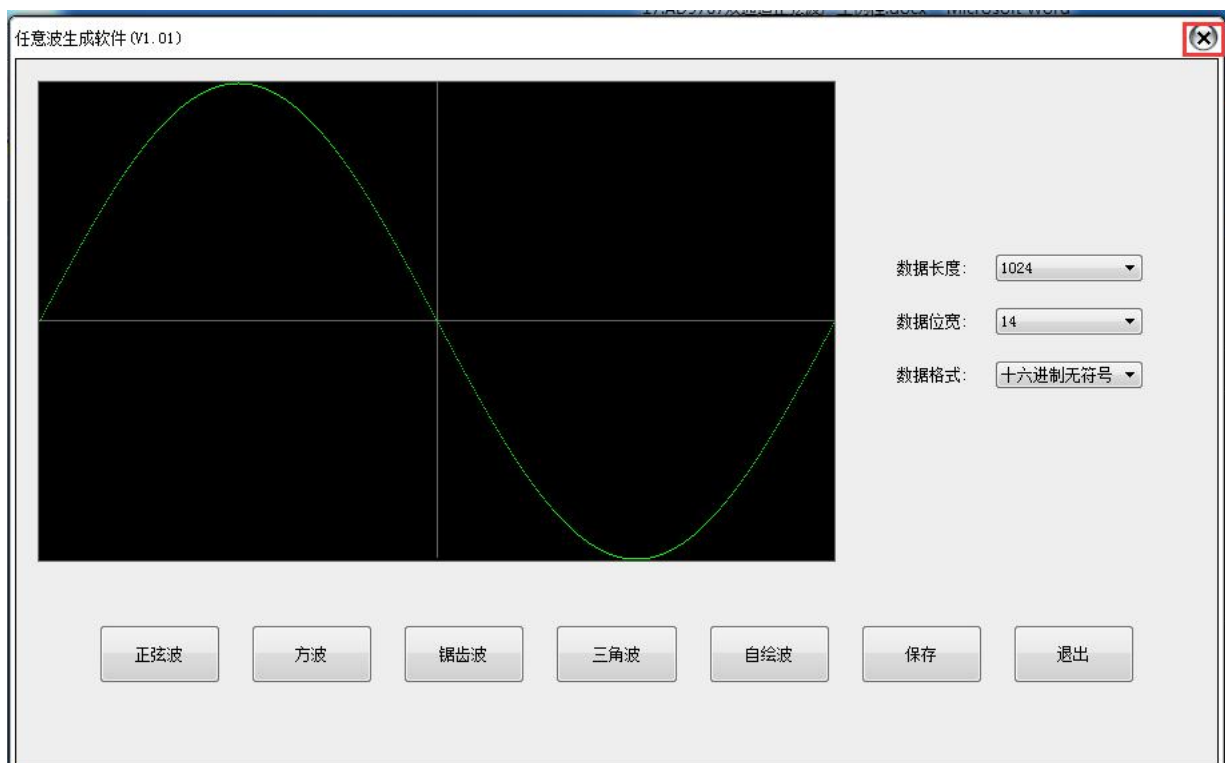


3. 点击保存按钮，将生成的数据文件保存到工程目录文件下（注意保存的文件类型.dat）：





#### 4. 保存后点击确定后关闭工具



将 .dat 文件保存到生成的 Rom IP 核中即可，在字符显示实验教程中已做介绍，这里不再重复。



## 3.2 双通道正弦波发生程序

```
`timescale 1ns / 1ps
/////////////////////////////////////////////////////////////////
//Two sine wave outputs -10V ~ +10V
/////////////////////////////////////////////////////////////////
module ad9767_test
(
input  sys_clk_p,           //system clock positive
input  sys_clk_n,           //system clock negative

output da1_clk,             //AD9767 CH1 clock
output da1_wrt,             //AD9767 CH1 enable
output [13:0] da1_data,     //AD9767 CH1 data output

output da2_clk,             //AD9767 CH2 clock
output da2_wrt,             //AD9767 CH2 enable
output [13:0] da2_data      //AD9767 CH2 data output

);

reg [9:0] rom_addr;

wire [13:0] rom_data;
wire clk_125M;
wire sys_clk;           //single end clock

assign da1_clk=clk_125M;
assign da1_wrt=clk_125M;
assign da1_data=rom_data;

assign da2_clk=clk_125M;
assign da2_wrt=clk_125M;
assign da2_data=rom_data;

//DA output sin waveform
always @(negedge clk_125M)
begin
    rom_addr <= rom_addr + 1'b1 ;           //The output sine wave frequency is 122Khz
    // rom_addr <= rom_addr + 4 ;           //The output sine wave frequency is 488Khz
    // rom_addr <= rom_addr + 128 ;        //The output sine wave frequency is 15.6Mhz
end

/*****
generate single end clock
*****/
GTP_INBUFGDS sys_clk_ibufgds
(
.O                               (sys_clk                               ),
.I                               (sys_clk_p                             ),
.IB                              (sys_clk_n                             )
);

ROM ROM_inst
(
.addr(rom_addr),
.clk(clk_125M),
```

```
.rst(1'b0),  
.rd_data(rom_data)  
);  
  
PLL PLL_inst  
(  
.rst(1'b0),  
.clkln1(sys_clk),  
.lock(),  
.clkout0( ),  
.clkout1(clk_125M)  
);  
  
endmodule
```

程序中通过一个 PLL IP 来产生 125M 的 DA 输出时钟，然后就是循环读取存放在 ROM 中的 1024 个数据，并同时输出到通道 1 和通道 2 的 DA 数据线上。程序中可以通过地址的加 1，加 4，或者加 128 来选择输出不同的频率的正弦波。

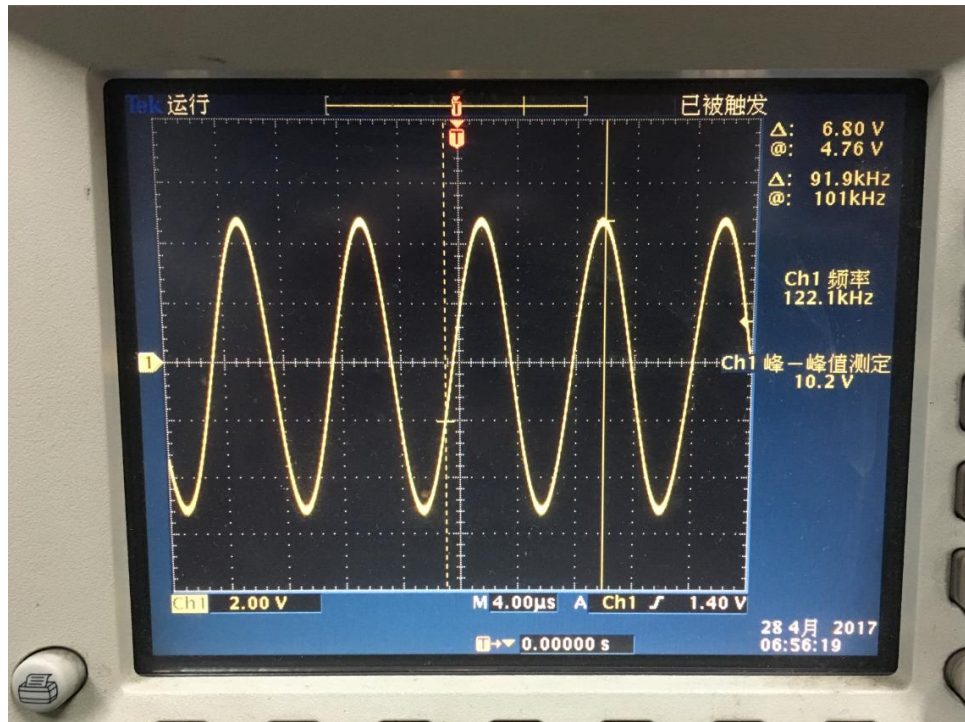
## 4 实验现象

- (1) 将 AN9767 模块插入开发板**扩展口 J13**，**注意 1 脚对齐，不要插错、插偏，不能带电操作。**
- (2) 用我们提供的 BNC 线连接 AN9767 的输出到示波器的输入如下图，然后开发板上电，下载程序就可以从示波器上观察从 DA 模块输出的模拟信号的波形了。



AN9767 与开发板连接图

(3) 示波器上看到的正弦波如下：

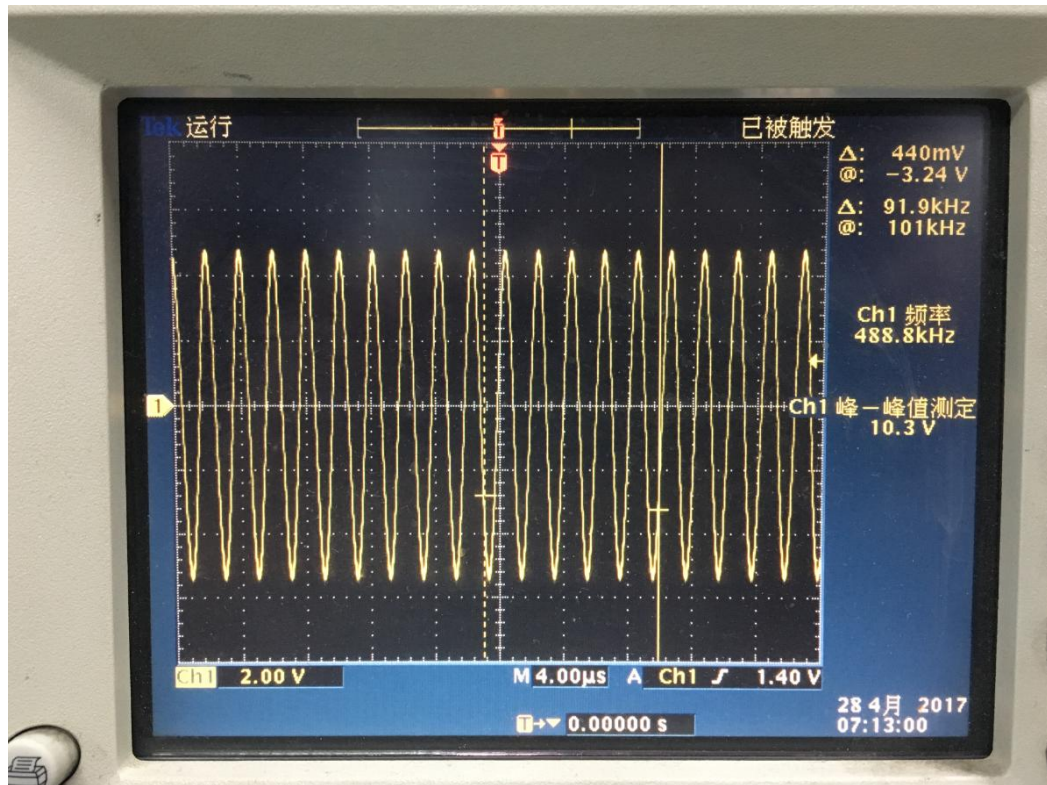


- (4) 我们可以把程序中的地址修改成+4的方式，如下修改，这样一个正弦波的输出的点为 256 个，输出的正弦波的频率会提高 4 倍：

```
35  always @(negedge clk_125)
36  begin
37      // rom_addr <= rom_addr + 1'b1 ;
38      rom_addr <= rom_addr + 4 ;
39      // rom_addr <= rom_addr + 128 ;
40
41
42
```

程序修改后，重新下载 FPGA 后，正弦波的频率变高，示波器显示的波形如下：





用户也可以通过调节 AN9767 模块上的可调电阻来改变 2 个通道输出波形的幅度。

