

ADDA 测试例程

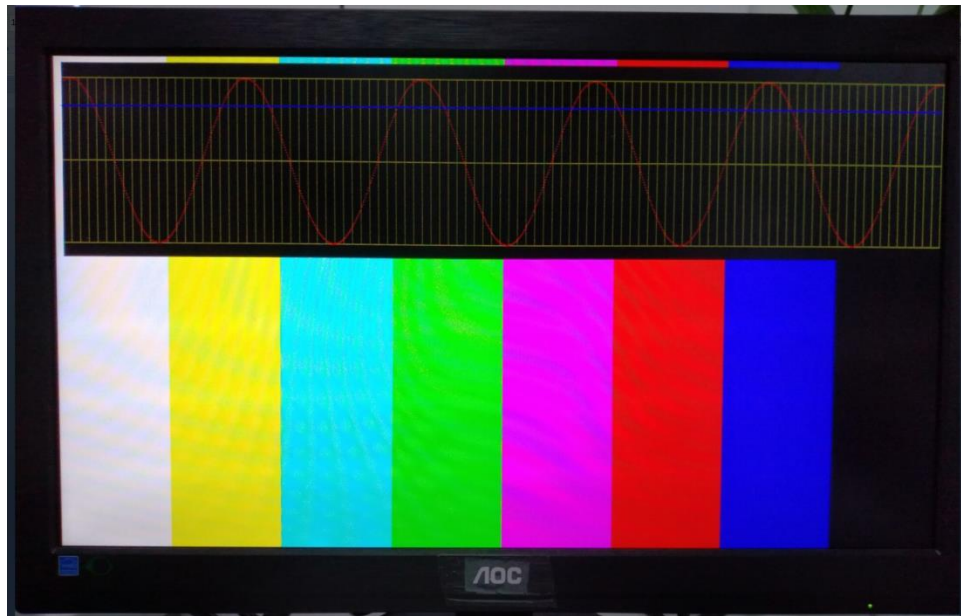
黑金动力社区 2020-03-13

1 实验简介

本实验练习使用 ADC 和 DAC，实验中使用的 ADDA 模块型号为 AN108，ADC 最大采样率 32Mhz，精度为 8 位，DAC 最大采样率 125Mhz，精度为 8 位。实验中用 DAC 输出正弦波，然后使用 ADC 采集并把波形在 HDMI/VGA 显示器显示。

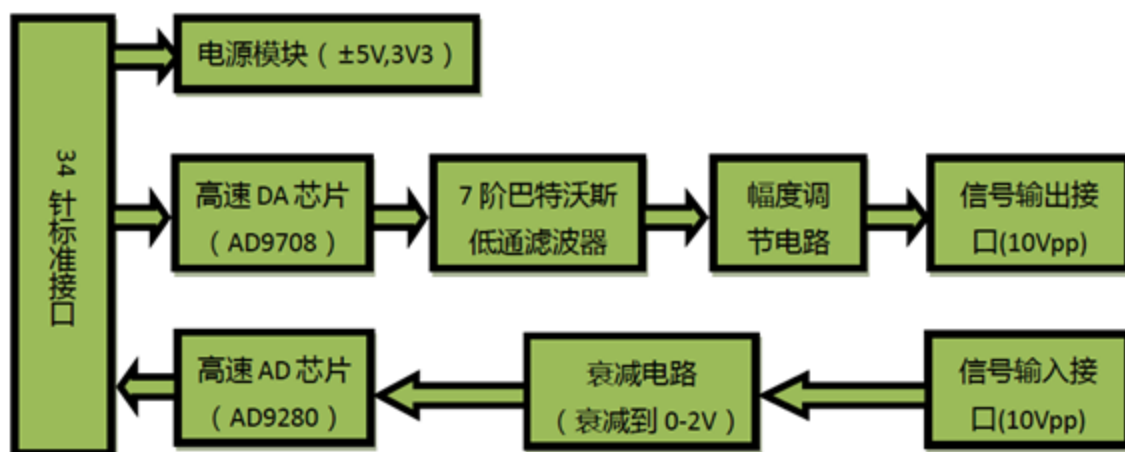


ADDA 模块



实验预期结果

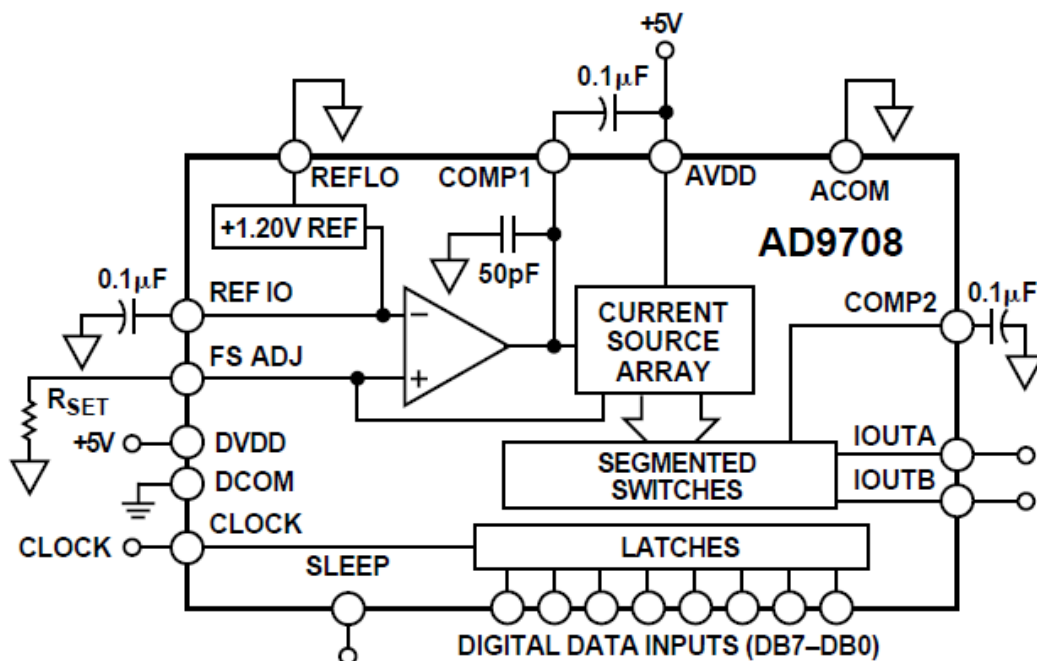
2 ADDA 模块原理



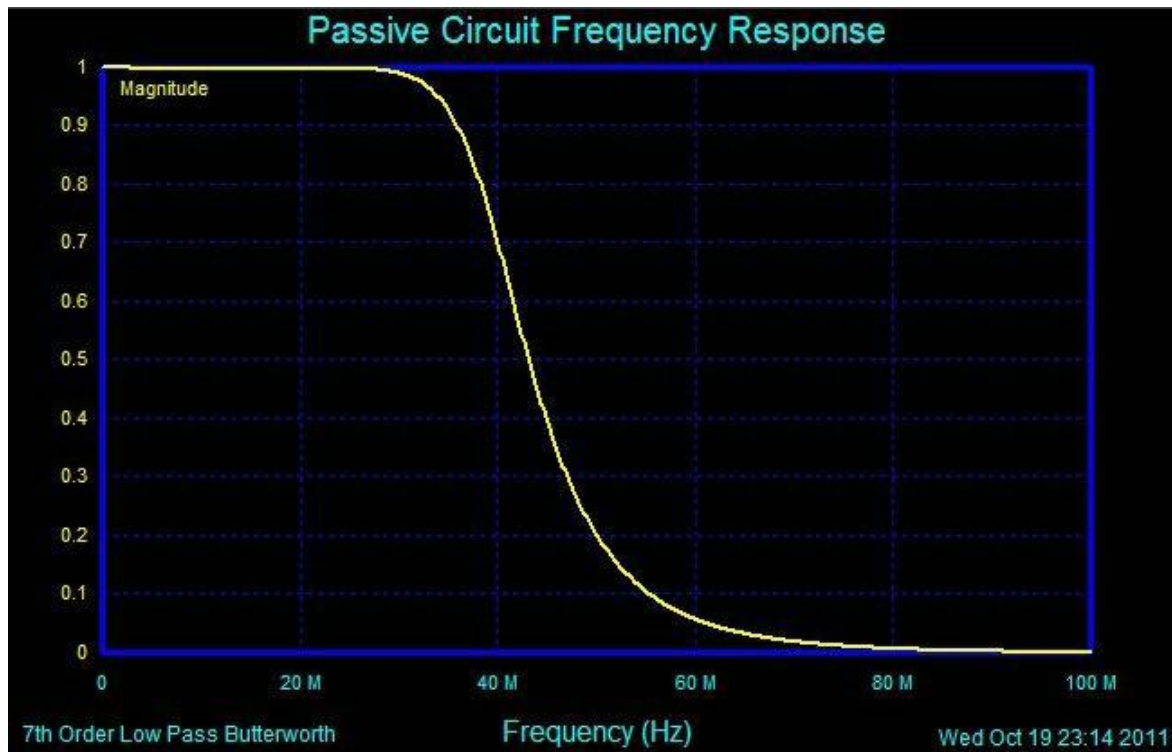
2.1 数模转换 (DA) 电路

如硬件结构图所示，DA 电路由高速 DA 芯片、7 阶巴特沃斯低通滤波器、幅度调节电路和信号输出接口组成。

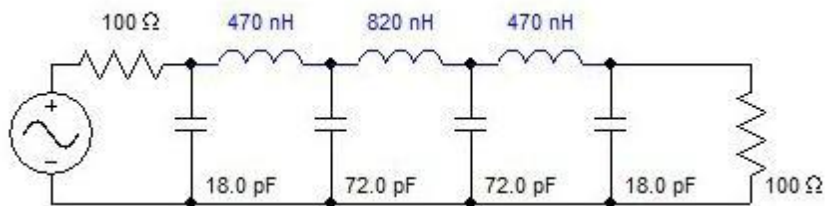
我们使用的高速 DA 芯片是 AD 公司推出的 AD9708。AD9708 是 8 位，125MSPS 的 DA 转换芯片，内置 1.2V 参考电压，差分电流输出。芯片内部结构图如下图所示



AD9708 芯片差分输出以后，为了防止噪声干扰，电路中接入了 7 阶巴特沃斯低通滤波器，带宽为 40MHz，频率响应如下图所示



滤波器参数如下图所示



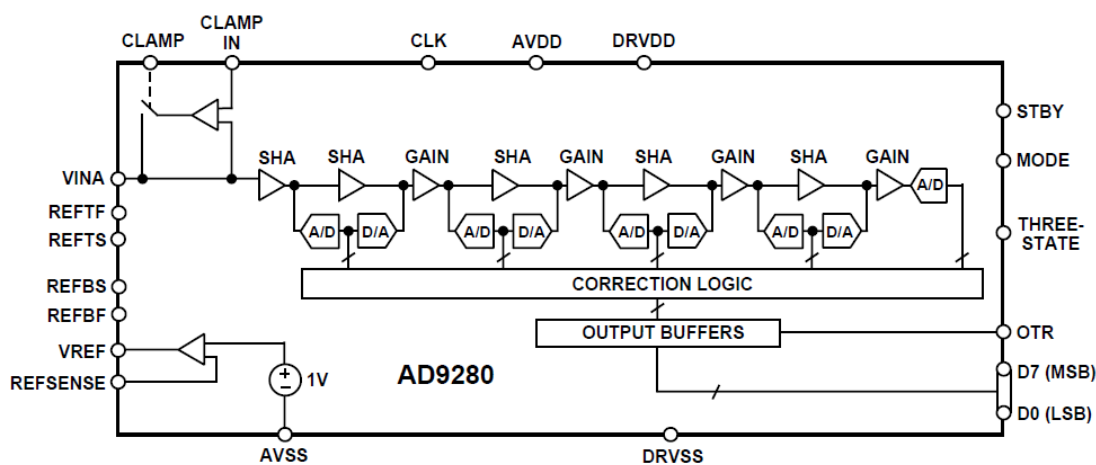
滤波器之后，我们使用了 2 片高性能 145MHz 带宽的运放 AD8056，实现差分变单端，以及幅度调节等功能，使整个电路性能得到了最大限度的提升。幅度调节，使用的是 5K 的电位器，最终的输出范围是-5V~5V（10Vpp）。

注：由于电路器的精度不是很精确，最终的输出有一定误差，有可能波形幅度不能达到 10Vpp，也有可能出现波形削顶等问题，这些都属正常情况。

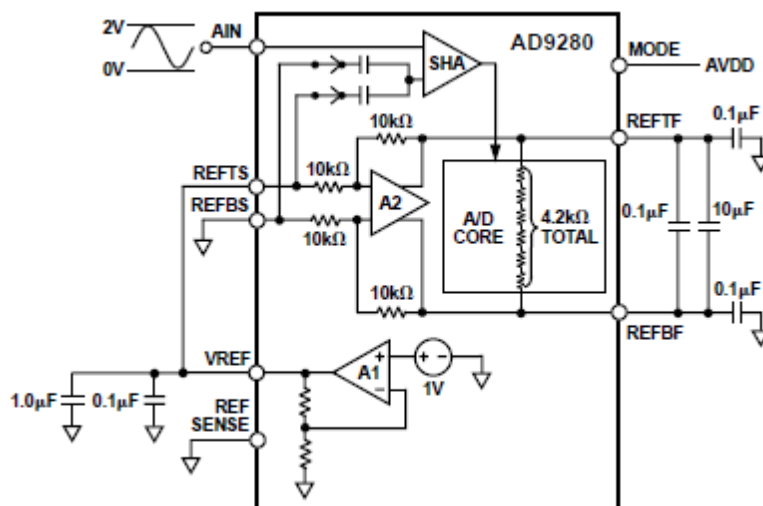
2.2 模数转换（AD）电路

如硬件结构图中所示，AD 电路由高速 AD 芯片、衰减电路和信号输入接口组成。

我们使用的高速 AD 芯片是由 AD 公司推出的 8 位，最大采样率 32MSPS 的 AD9280 芯片。内部结构图如下图所示



根据下图的配置，我们将 AD 电压输入范围设置为：0V~2V



在信号进入 AD 芯片之前，我们用一片 AD8056 芯片构建了衰减电路，接口的输入范围是-5V~+5V(10Vpp)。衰减以后，输入范围满足 AD 芯片的输入范围（0~2V）。转换公式如下：

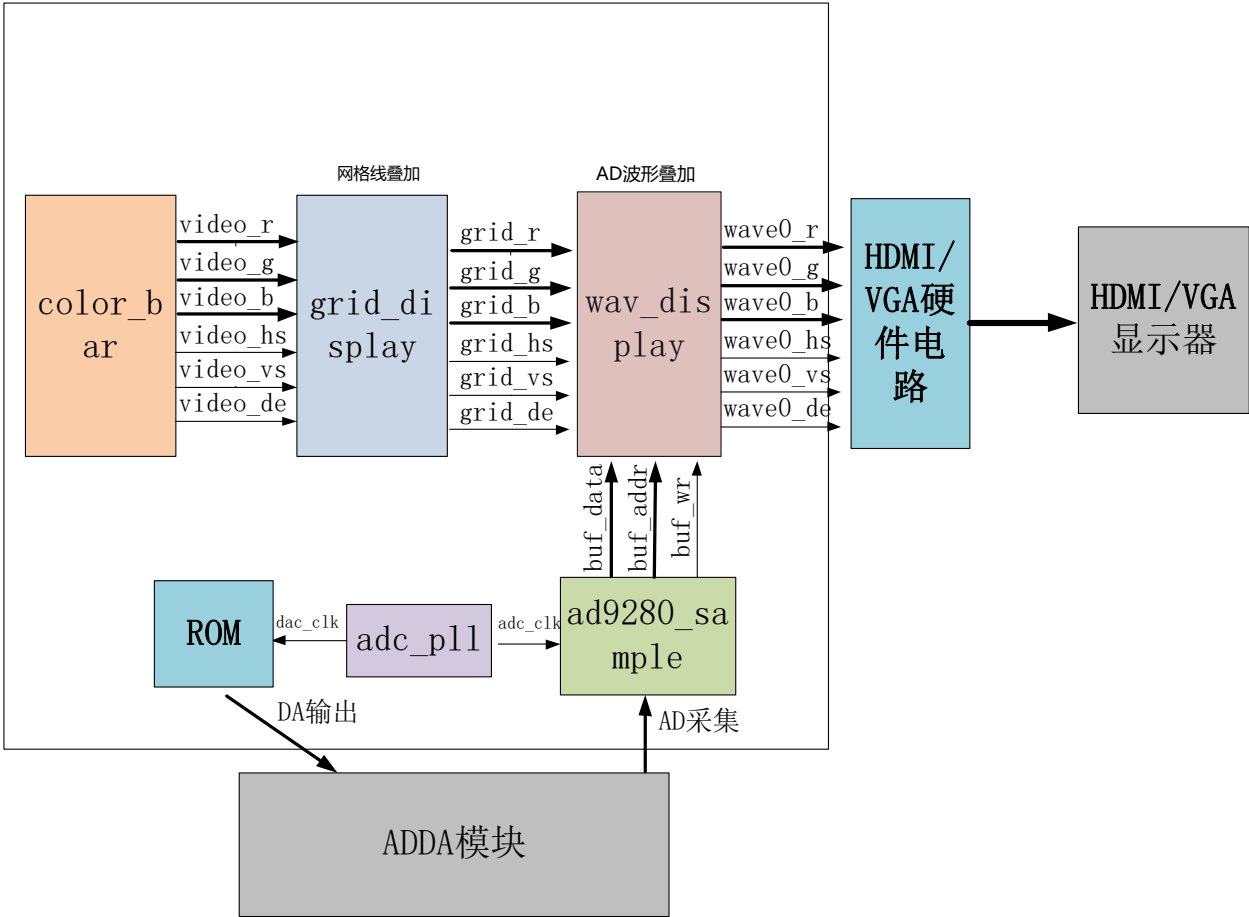
$$V_{AD} = \frac{1}{5} V_{IN} + 1$$

当输入信号 $V_{in}=5(V)$ 的时候，输入到 AD 的信号 $V_{ad}=2(V)$ ；

当输入信号 $V_{in}=-5(V)$ 的时候，输入到 AD 的信号 $V_{ad}=0(V)$ ；

3 程序设计

本实验程序设计跟 AD9226 波形显示实验基本类似，只是 ADDA 模块是单通道的 AD，这里只是一路采集波形的叠加。另外 FPGA 通过 ROM IP 产生正选波数据输出到 DA 芯片进行 DA 转换，产生正选波模拟信号，用户只有用 BNC 线把模块的 AD 和 DA 端口连接起来就形成环路。这样 HDMI/VGA 显示器上显示的就是 DA 正选波的信号了。



ad9280_sample 模块主要完成 ad9280 的 AD 8 位数据采集和转换，每次采集 1024 个数据，然后等待一段时间再继续采集下次的 1024 个数据。

信号名称	方向	宽度 (bit)	说明
adc_clk	in	1	adc 系统时钟
rst	in	1	异步复位，高复位
adc_data	in	12	ADC 数据输入，MSB 在最低位，需要倒序
adc_buf_wr	out	1	ADC 数据写使能
adc_buf_addr	out	12	ADC 数据写地址
adc_buf_data	out	8	无符号 8 位 ADC 数据

ad9280_sample 模块端口

grid_display 模块主要完成视频图像的网格线叠加，本实验将彩条视频输入，然后叠加一个网格后输出，这一块网格区域提供给后面的波形显示模块使用，这个网格区域是位于显示器水平方向（从左到右）从 9 到 1018，垂直方向（从上到下）从 9 到 308 的视频显示位置。

```
if(pos_y >= 12'd9 && pos_y <= 12'd308 && pos_x >= 12'd9 && pos_x <= 12'd1018)
    region_active <= 1'b1;
```

信号名称	方向	宽度 (bit)	说明
pcclk	in	1	像素时钟
rst_n	in	1	异步复位，低电平复位
i_hs	in	1	视频行同步输入
i_vs	in	1	视频场同步输入
i_de	in	1	视频数据有效输入
i_data	in	24	视频数据输入
o_hs	out	1	带网格视频行同步输出
o_vs	out	1	带网格视频场同步输出
o_de	out	1	带网格视频数据有效输出
o_data	out	24	带网格视频数据输出

grid_display 模块端口

wav_display 显示模块主要是完成波形数据的叠加显示，模块内含有一个双口 ram，写端口是由 ADC 采集模块写入，读端口是显示模块。在网格显示区域有效的时候，每行显示都会读取 RAM 中存储的 AD 数据值，跟 Y 坐标比较来判断显示波形或者不显示。

```
79 if(region_active == 1'b1)
80     if(12'd287 - pos_y == {4'd0,q})
81         v_data <= wave_color;
82     else
83         v_data <= pos_data;
84     else
85         v_data <= pos_data;
```


信号名称	方向	宽度 (bit)	说明
pclk	in	1	像素时钟
rst_n	in	1	异步复位，低电平复位
wave_color	in	24	波形颜色，rgb
adc_clk	in	1	adc 模块时钟
adc_buf_wr	in	1	adc 数据写使能
adc_buf_addr	in	12	adc 数据写地址
adc_buf_data	in	8	adc 数据，无符号数
i_hs	in	1	视频行同步输入
i_vs	in	1	视频场同步输入
i_de	in	1	视频数据有效输入
i_data	in	24	视频数据输入
o_hs	out	1	带网格视频行同步输出
o_vs	out	1	带网格视频场同步输出
o_de	out	1	带网格视频数据有效输出
o_data	out	24	带网格视频数据输出

wav_display 模块端口

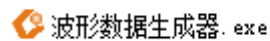
timing_gen_xy 模块为其它模块的子模块，完成视频图像的坐标生成，x 坐标，从左到右增大，y 坐标从上到下增大。

信号名称	方向	宽度 (bit)	说明
clk	in	1	系统时钟
rst_n	in	1	异步复位，低电平复位
i_hs	in	1	视频行同步输入
i_vs	in	1	视频场同步输入
i_de	in	1	视频数据有效输入
i_data	in	24	视频数据输入
o_hs	out	1	视频行同步输出

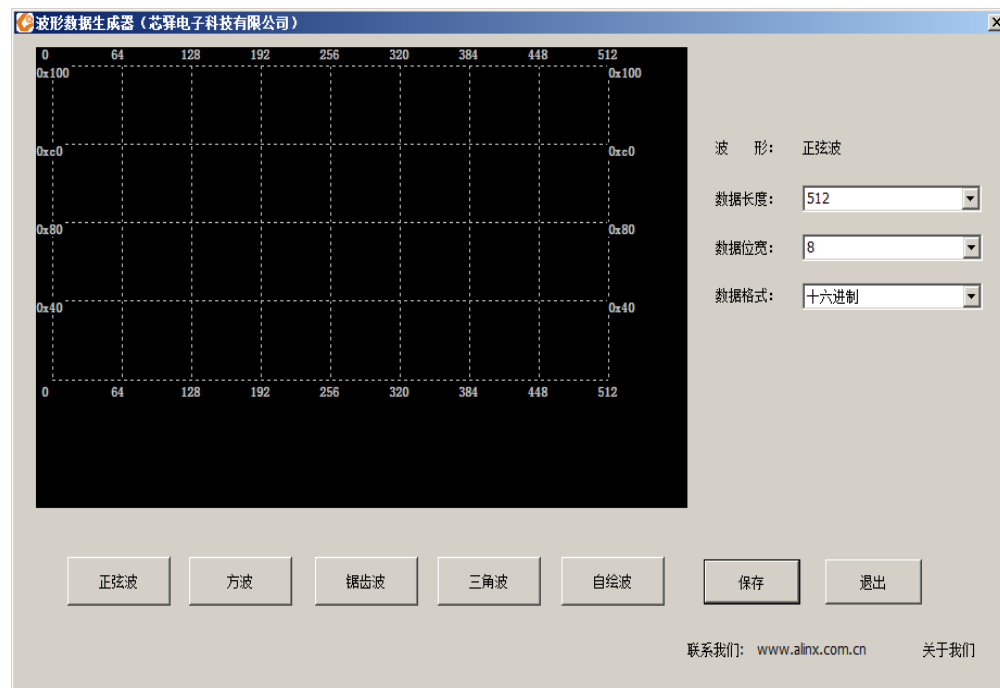
o_vs	out	1	视频场同步输出
o_de	out	1	视频数据有效输出
o_data	out	24	视频数据输出
x	out	12	坐标 x 输出
y	out	12	坐标 y 输出

timing_gen_xy 模块端口

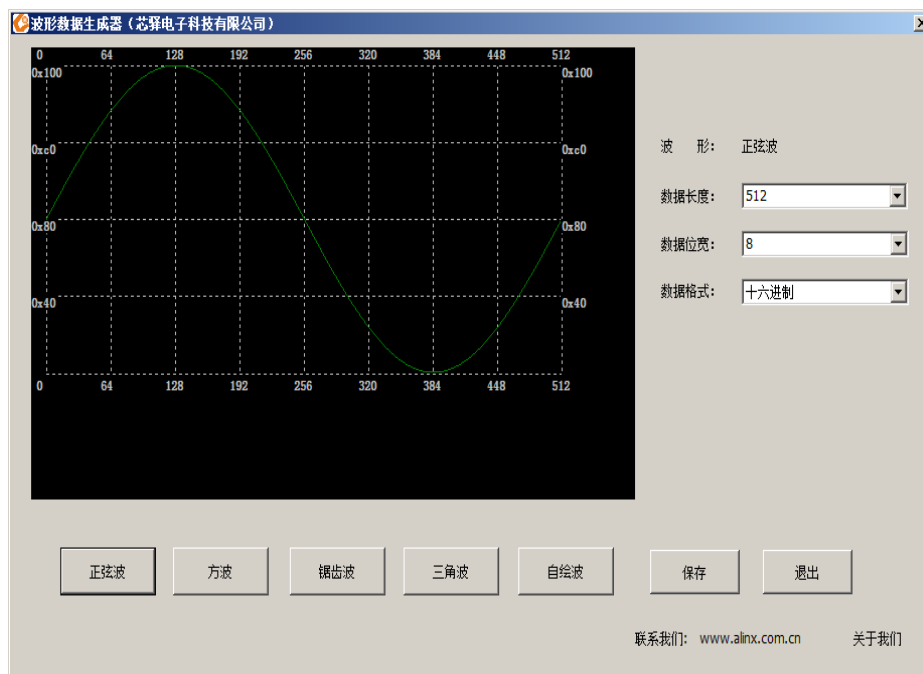
另外在本例程中添加了一个 ROM IP 模块，需要对 ROM IP 初始化数据。这里仅介绍如何使用波形数据生成工具，在软件工具及驱动文件夹下找到工具，其图标如下所示：



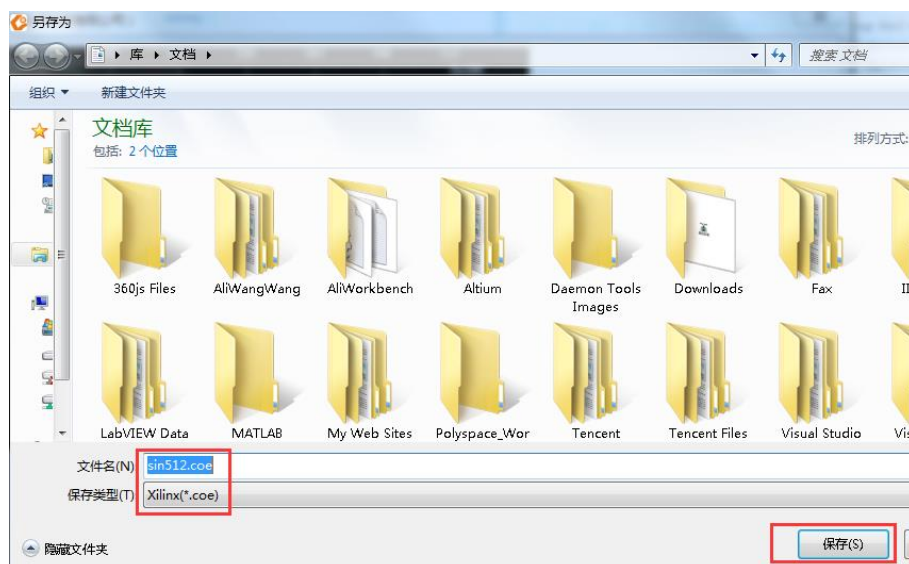
1. 双击.exe 打开工具，打开界面如下：



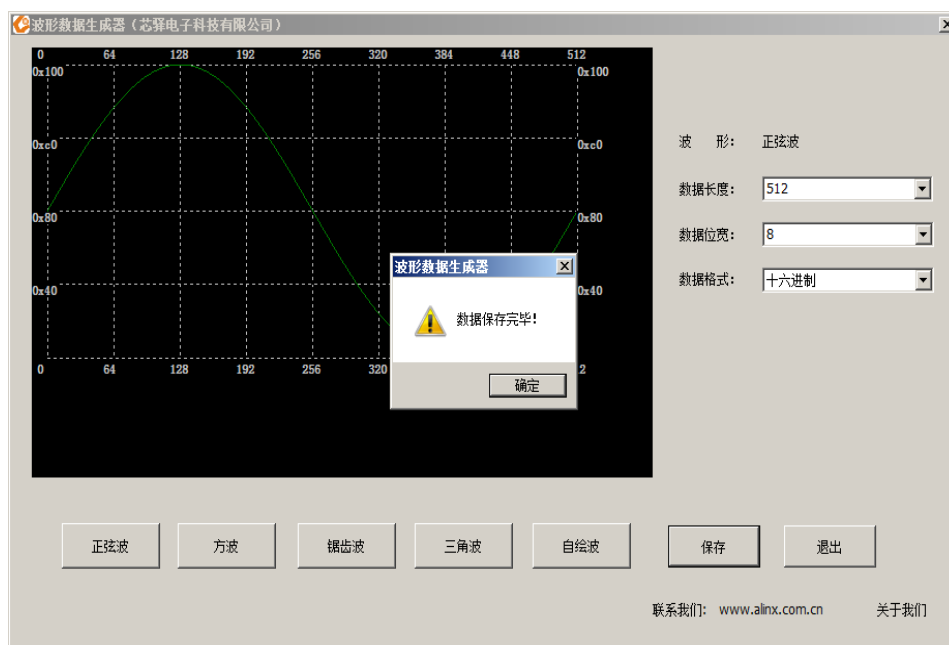
2. 可以根据需要自选波形，本例程中选择正弦波，数据长度和位宽保持默认



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）：



4. 保存后出现如下对话框表示保存成功，点击确定后关闭工具



将 .coe 文件保存到生成的 Rom IP 核中即可，这里不再重复介绍

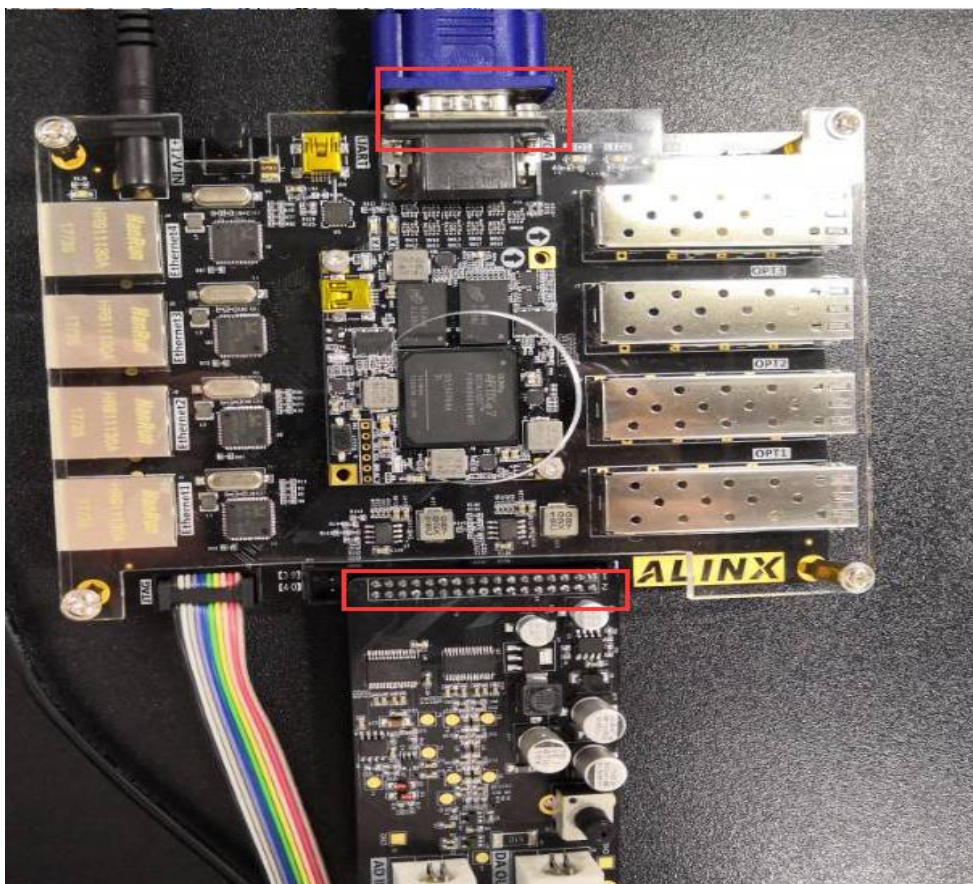
4 实验现象

- (1) 将 AN108 模块插入开发板，**AX7101 接 J11、AX7102 接 J5，AX7103 接 J13，注意 1 脚对齐，不要插错、插偏，不能带电操作。**
- (2) 连接 AN108 的 DAC 输入到信号发生器的输出，**这里使用的是专用屏蔽线，如果使用其他线可能会有较大干扰。**

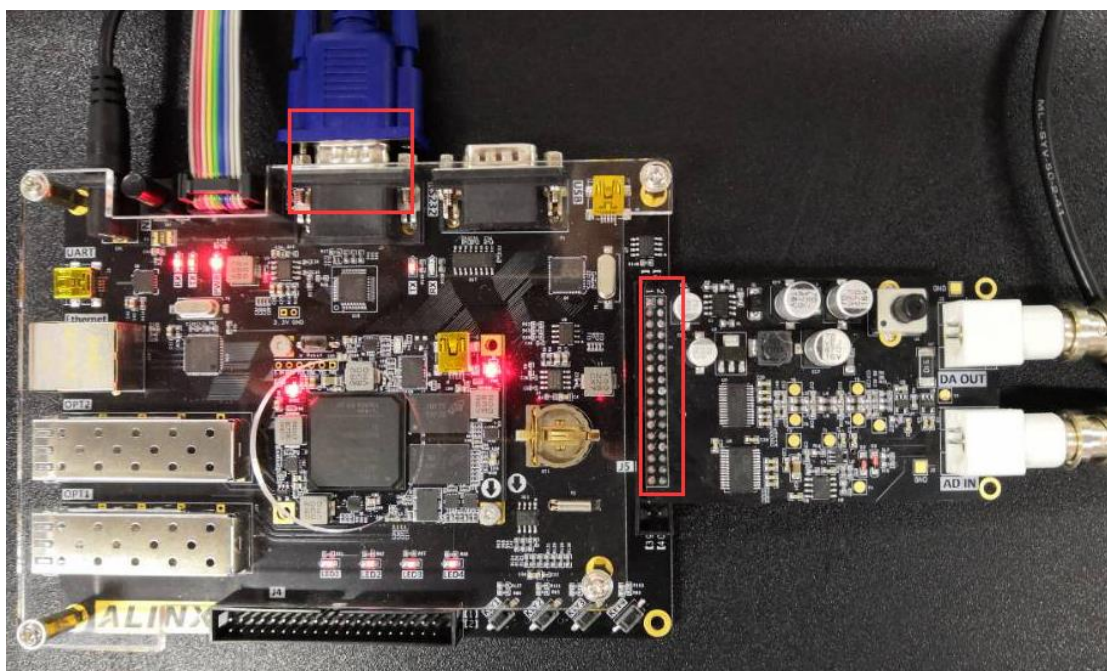


AN108 连接示意图

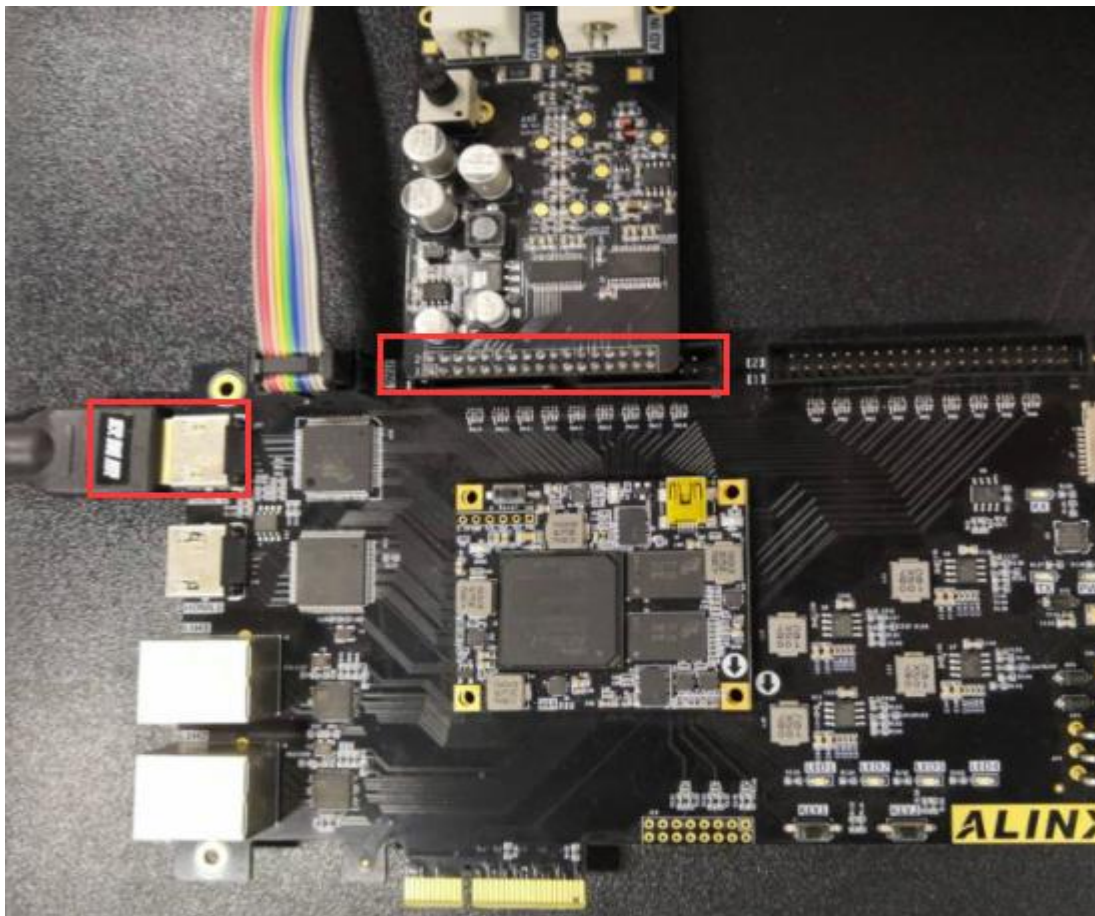
(3) 连接 HDMI/VGA 显示器，**注意：连接 HDMI 的是显示器，不是笔记本电脑接口**



AX7101 开发板连接图



AX7102 开发板连接图



AX7103 发板连接图

- (4) 下载程序，旋转 AN108 模块上的旋钮，可以看到波形幅度在变化，波形如下图所示，黄色网格最上面横线代表 5V，最下面横线代表 -5V，中间横线代表 0V，每个竖线间隔是 10 个采样点。

