

# AD9708 方波产生例程

## 1 实验简介

本实验练习将使用“小眼睛科技”的 MES-ADDA 模块与 FPGA 开发板生成方波，然后使用示波器观察输出方波的波形。MES-ADDA 模组上采用的数模转换器 MS9708 是一款 8bit 高速、低功耗 D/A 转换器，采样率高达 125MSPS。

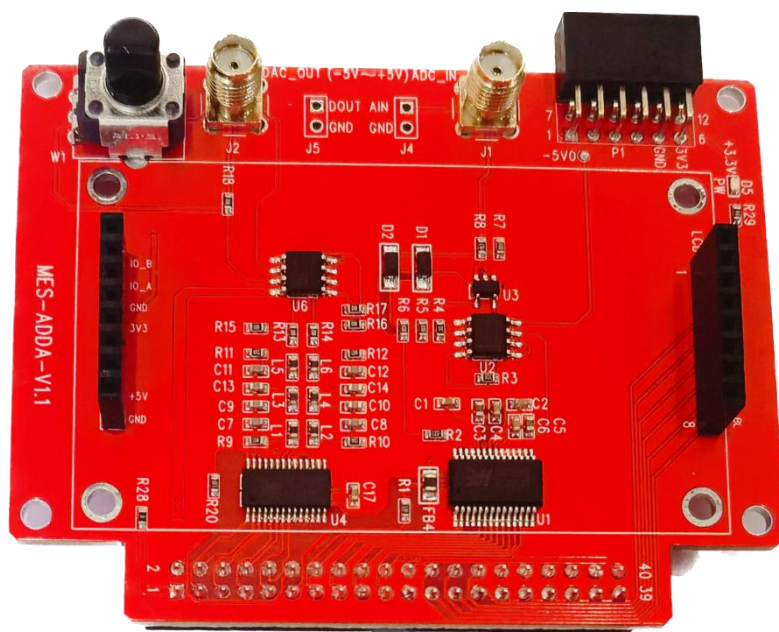


图 1 MES-ADDA 模块

实验器件准备：

FPGA 开发板、MES-ADDA 模块、示波器。

实验流程如下：

- 1、生成方波 ROM 初始化数据文件。
- 2、进行代码编写，生成 8bit 方波数据。
- 3、烧录程序，将数据送入 MS9708 模块。
- 4、使用示波器观察方波波形。

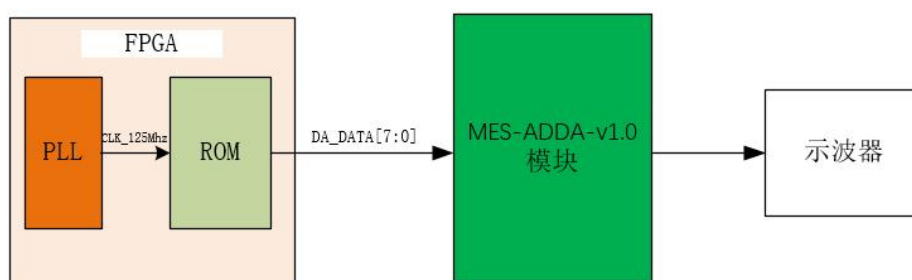


图 2 实验流程图

MS9708 芯片参数：

- 8bit 分辨率
- 更新速率：125MSPS
- 功耗：175mW @ 5V 到 45mW @ 3V
- 掉电模式：20mW @ 5V
- 内部基准：1.2V
- 边沿触发锁存器
- TSSOP28 封装

## 2、实验原理

MES-ADDA 模组采用的 D/A 芯片 MS9708 将 FPGA 提供的数字信号数据转换为模拟信号，但仍需要一些处理，首先需要经过一系列的低通滤波进行降噪处理，然后再经过两级运放进行幅值调节等处理，最后才能将模拟信号输出。

需注意的是，两级运放均采用的是负反馈的方式，并且第一级运放的反向输入端连接的为 D/A 芯片的互补输出端，因此实际输出的模拟信号与输入的数字信号数据对应的波形在相位上相差  $180^\circ$ 。

数模转换功能流程图如下：

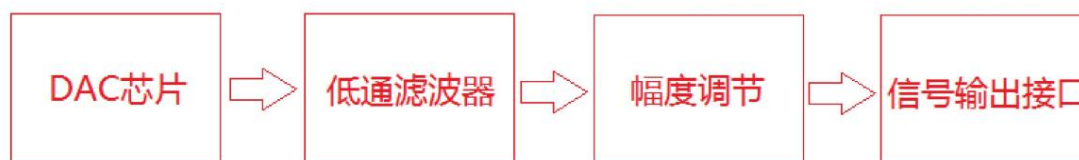


图 3 DAC 转换流程

### 2.1.1、DAC 芯片

DAC 芯片将数字信号转换为模拟信号，内部包含了一个 PMOD 电流源阵列，最大可产生 20mA 电流，芯片输出两路信号，DAC 电流输出 IOUTA、互补 DAC 电流输出 IOUTB，信号转换的计算方式如下，具体内容请参考 MS9708 数据手册。

$$IOUTA = (DAC\ CODE/256) \times IOUTFS \quad (1)$$

$$IOUTB = (255 - DAC\ CODE)/256 \times IOUTFS \quad (2)$$

$$IOUTFS = 32 \times IREF \quad (3)$$

$$IREF = VREFIO / RSET \quad (4)$$

注：

1、RSET 为图中 R19，REFIO 接地，故使用的是内部 1.2V 基准，VREFIO 为 1.2V。

2、电阻 RSET 在原理图中为 R19。

DAC 芯片原理图如下图所示，详情请参考 MES-ADDA 原理图。

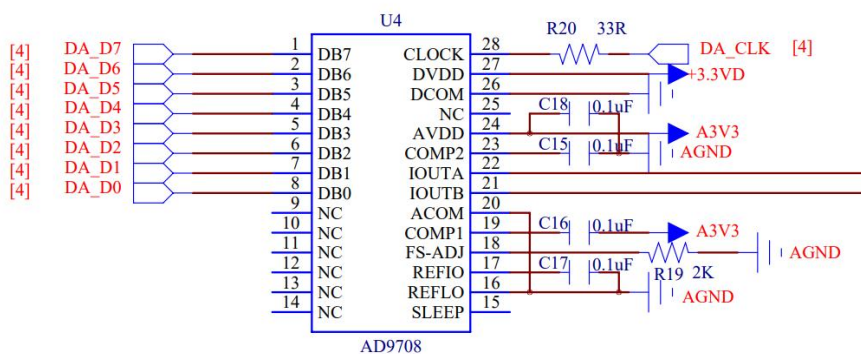


图 4 DAC 芯片

## 2.1.2、低通滤波电路

低通滤波电路主要用于滤波去噪，使输出波形更加圆滑，电路图如下图所示：

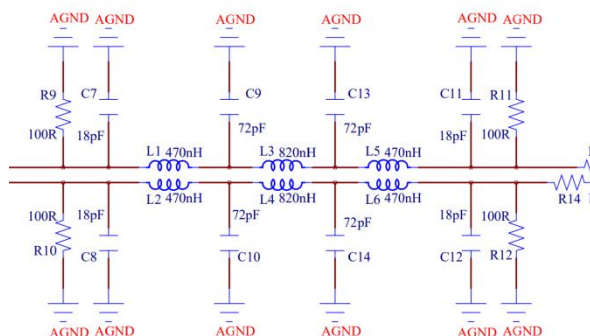


图 5 低通滤波电路

## 2.1.3、运算放大器

第一级负运算放大器主要将输出电流信号进行减法运算，并且转换为电压信号，输出电压幅值范围：-1v ~ +1v。

需要注意的是第一级运算放大器采用负反馈的方式，因此运放输出与 DAC 互补输出信号相位相差 180°。

电路如下图所示：

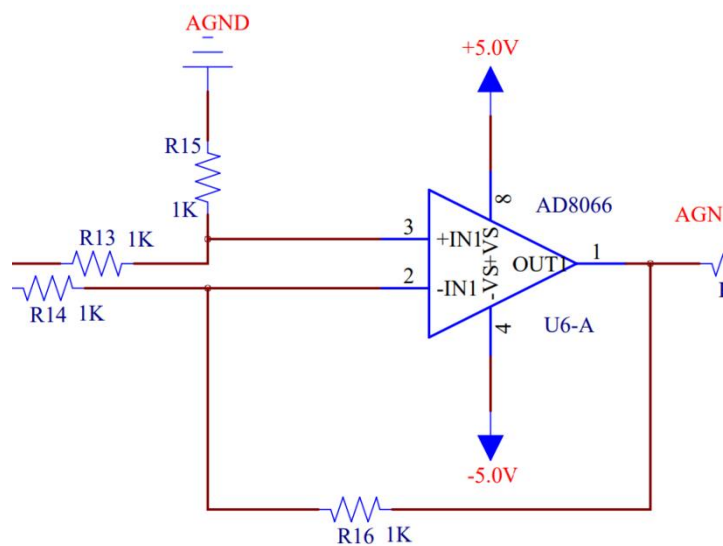


图 6 第一级放大电路

第二级运算放大器可以通过对 W1 旋钮电阻的调节控制放大倍数。测试显示第二级放大电路输出电压控制在 -5V ~ +5V 范围内，输出波形不失真。

须注意的是第二级运放仍采用负反馈的方式，运放输出与输入相位仍相差 180°。

设 W1 的阻值为  $R_f$ ，输出电压  $V_{out}$ ，输入电压  $V_{in}$ ，它们的关系如下：

$$V_{out} = - (R_f / R_{17}) * V_{in} ;$$

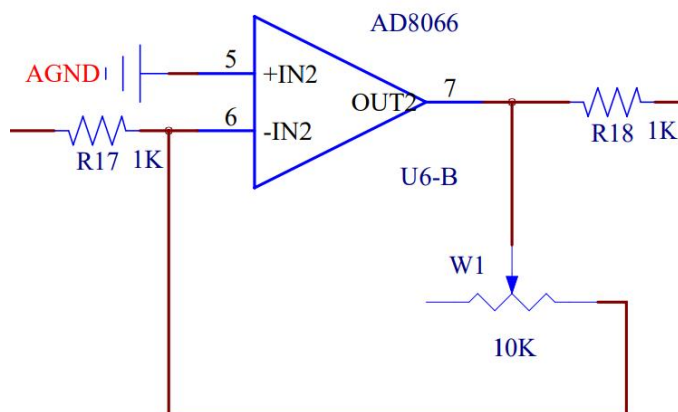
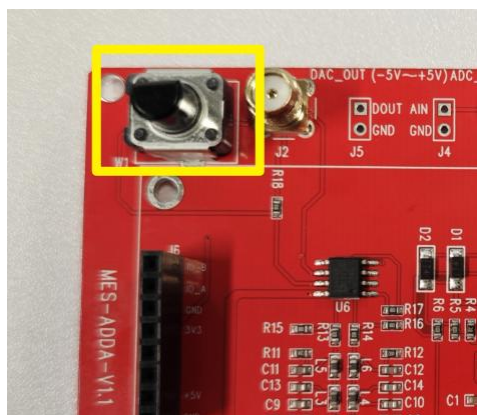


图 7 第二级负反馈放大电路

旋钮电阻的旋钮位置如下：



### 2.1.4、信号输出接口

模拟信号输出有两个接口，一个为 SMA 接口，一个为测试点接口，位置为下图所标：



## 3、程序设计

### 3.1、使用 MATLAB 生成 ROM 初始化文件

MATLAB 源码如下所示：

```
clc;close all;clear all;

rom_depth    = 2^10;
rom_widths   = 8;
N = 0 : (rom_depth-1) ;
squarex = square(2*pi*N/rom_depth) ;
square_wave = fopen('D:\square_1024.dat','w') ;
fprintf(square_wave,'%X\n',round((2^(rom_widths -
1)-1)*squarex+2^((rom_widths - 1))));
fclose(square_wave);
```

## 3.2、实验代码

使用 PLL 生成数模转换时钟，生成 ROM IP 核，将准备好的方波 ROM 初始化文件导入进 IP 核中，从 ROM 中读出数据，输出 8bit 方波数据。

```
1  module square_wave(  
2      input    wire    clk_50M      ,  
3      output   wire    da_clk       ,  
4      output   wire    [7:0]da_data  
5  );  
6  
7      wire rst_n ;  
8      wire clk_125M ;  
9  
10     reg    [10:0]rom_addr ;  
11     wire   [7:0]rom_data_out ;  
12  
13     assign da_clk = clk_125M ;  
14     assign da_data = rom_data_out ;  
15  
16     always @(negedge clk_125M or negedge rst_n) begin  
17         if (!rst_n)  
18             rom_addr <= 11'd0 ;  
19         else if (rom_addr >= 11'd1023)  
20             rom_addr <= 11'd0 ;  
21         else  
22             rom_addr <= rom_addr + 10'd1 ;           //The output wave frequency is 122Khz  
23         // rom_addr <= rom_addr + 10'd2 ;           //The output wave frequency is 244Khz  
24         // rom_addr <= rom_addr + 10'd4 ;           //The output wave frequency is 488Khz  
25         // rom_addr <= rom_addr + 10'd32 ;          //The output wave frequency is 3.9Mhz  
26         // rom_addr <= rom_addr + 10'd128 ;         //The output wave frequency is 15.6Mhz  
27     end  
28  
29     ad_clock_125m u_pll (  
30         .clk_in1(clk_50M),           // input  
31         .pll_lock(rst_n),           // output  
32         .clk_out0(clk_125M)         // output  
33     );  
34  
35     rom_square_wave u_rom (  
36         .addr(rom_addr[9:0]),         // input [9:0]  
37         .clk(clk_125M),              // input  
38         .rst(1'b0),                  // input  
39         .rd_data(rom_data_out)       // output [7:0]  
40     );  
41  
42     endmodule
```

## 3.3、调整方波频率

通过对 ROM 读地址的调整，可以调整输出方波的频率。ROM 中存储的波形数据为 1024 点。实验中 ROM 的读时钟为 125Mhz。调整方式举例如下：

rom\_addr <= rom\_addr + 10'd1 : 方波的频率为  $125\text{Mhz}/1024 = 122\text{Khz}$  ；

rom\_addr <= rom\_addr + 10'd2 : 方波的频率为  $125\text{Mhz}/1024 = 244\text{Khz}$  ；



```

always @(negedge clk_125M or negedge rst_n) begin
    if (!rst_n)
        rom_addr <= 11'd0 ;
    else if (rom_addr >= 11'd1023)
        rom_addr <= 11'd0 ;
    else
        rom_addr <= rom_addr + 10'd1 ;           //The output wave frequency is 122Khz
// rom_addr <= rom_addr + 10'd2 ;           //The output wave frequency is 244Khz
// rom_addr <= rom_addr + 10'd4 ;           //The output wave frequency is 488Khz
// rom_addr <= rom_addr + 10'd32 ;          //The output wave frequency is 3.9Mhz
// rom_addr <= rom_addr + 10'd128 ;         //The output wave frequency is 15.6Mhz
end

```

## 4、实验现象

### 4.1、观察实验结果前步骤

- 1、将 MES-ADDA 模块插入到 FPGA 开发板上（40pin 扩展脚位置），如图 8。
- 2、使用示波器对模拟信号输出接口进行测量
- 3、调整 MES-ADDA 模块的 W1 电阻旋钮，调整输出波形幅度，如图 9。

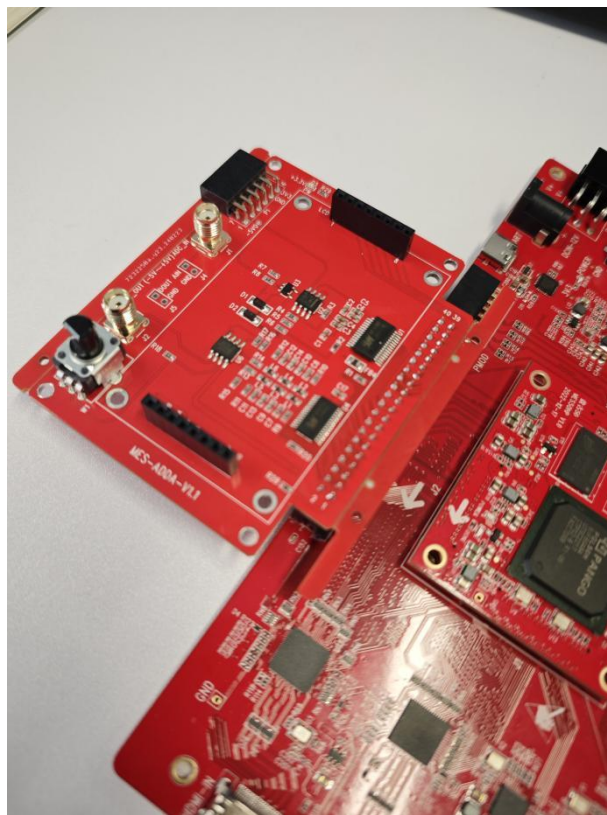


图 8 步骤 1

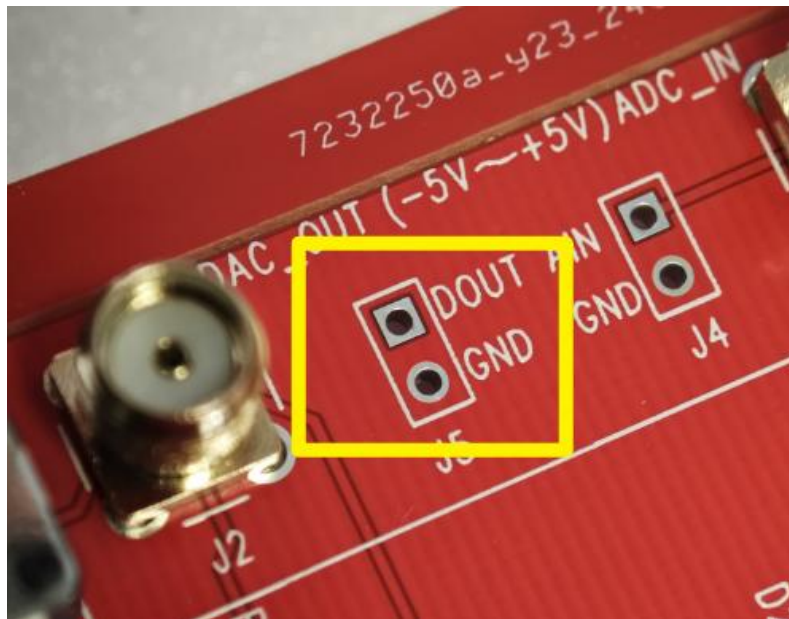


图 9 步骤 3

## 4.2、观察实验现象

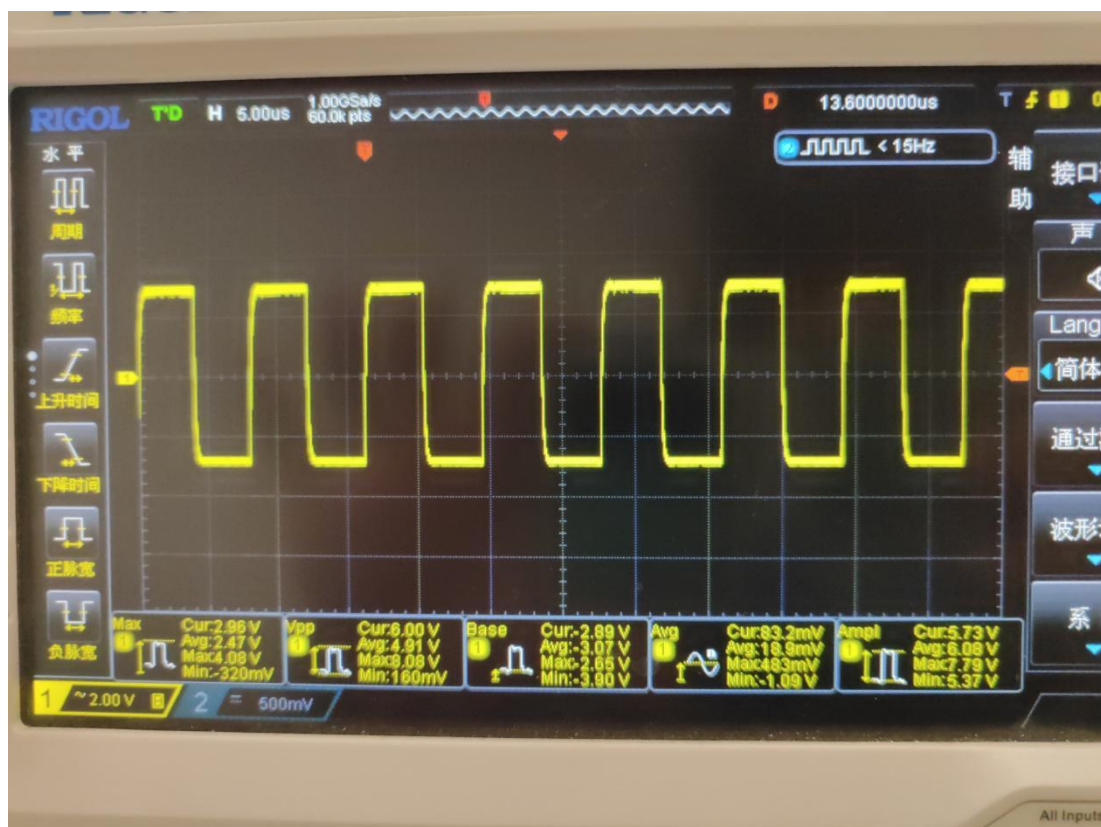


图 10 实验结果