

AGC 的 FPGA 实现

蒋晓东

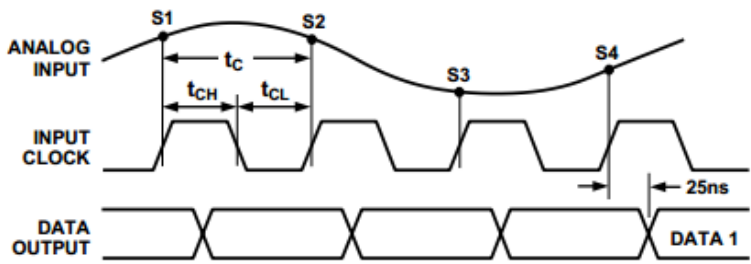
2015 年 5 月 24 日

1 硬件部分

这次用的 FPGA 开发板是淘宝上买的睿智助学板，FPGA 型号：EP4CE6E22C8N，AD_DA 模块是黑金开发板配套的外设模块。

1.1 AD_DA 接口

这次的 AD 芯片为 AD9280，是 AD 公司的 8 位，最大采样率为 32MSPS，就是 32MHz。其时序图如图一所示。可以看出只要输入时钟频率小于 32MHz，在时钟下降沿便会输出 AD 转换值，控制十分简单，与 EDA 实验箱的 TLC5540 比较类似。



图一 AD9280 时序图

使用时将 AD_DA 模块电路当成一个“黑匣子”。AD 部分接口定义如图二所示。

引脚	内容	备注	引脚	内容	备注
1	DCOM	地	2	VCC	+5V
3			4		
5	DACLK	DA 时钟线	6	DADB7	DA 数据线
7	DADB6	DA 数据线	8	DADB5	DA 数据线
9	DADB4	DA 数据线	10	DADB3	DA 数据线
11	DADB2	DA 数据线	12	DADB1	DA 数据线
13	DADB0	DA 数据线	14		
15			16		
17			18		
19			20		
21	ADDB0	AD 数据线	22	ADDB1	AD 数据线
23	ADDB2	AD 数据线	24	ADDB3	AD 数据线
25	ADDB4	AD 数据线	26	ADDB5	AD 数据线
27	ADDB6	AD 数据线	28	ADDB7	AD 数据线
29	ADCLK	AD 时钟线	30		
31			32		
33			34		

图二 AD_DA 接口定义

可以看到，AD 部分只需要 9 线即可，一线时钟输出线，八线数据线。

DA 芯片是 AD 公司的高速 DA 芯片 AD9708，同样也是八位，输出频率可以达到 125MSPS（不知道这个的专业称呼）。同样将其视为一个“黑匣子”。

其接口定义见图二。其也为九线，一线时钟输出线，八线数据线。

具体 AD_DA 模块使用指南，请见《高速 AD/DA 模块使用指南 Rev 1.3》（黑金动力社区 编）。

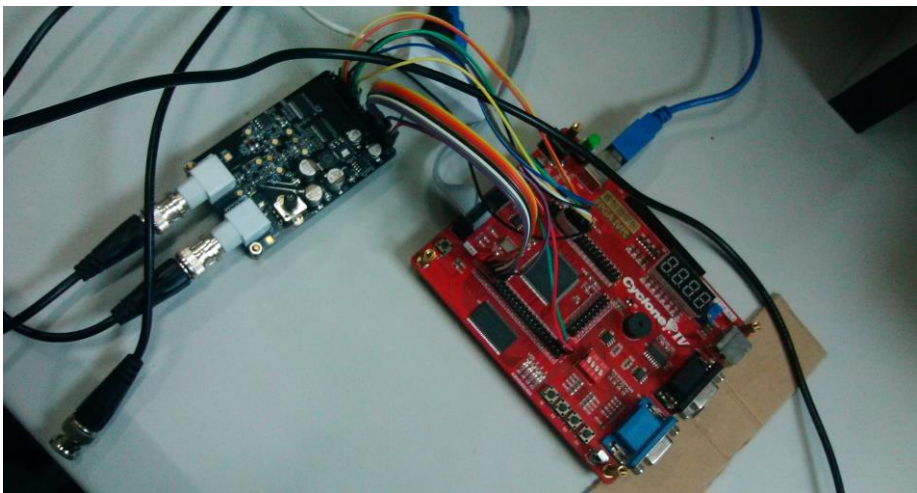
1.2 FPGA 引脚配置

FPGA 的引脚配置请见图三。这次的引脚连接比较简单，注意高地位不要连错即可。其中 clkin_50MHz 为主时钟 50MHz，ADCclk_25Mhz 和 DACclk_25Mhz 为经过 2 分频后输出至 AD 和 DA 的时钟，均为 20MHz。

ADCclk_25MHz	Output	PIN_141	2.5 V (default)	8mA (default)
ADCin[7]	Input	PIN_10	2.5 V (default)	8mA (default)
ADCin[6]	Input	PIN_2	2.5 V (default)	8mA (default)
ADCin[5]	Input	PIN_3	2.5 V (default)	8mA (default)
ADCin[4]	Input	PIN_144	2.5 V (default)	8mA (default)
ADCin[3]	Input	PIN_1	2.5 V (default)	8mA (default)
ADCin[2]	Input	PIN_142	2.5 V (default)	8mA (default)
ADCin[1]	Input	PIN_143	2.5 V (default)	8mA (default)
ADCin[0]	Input	PIN_138	2.5 V (default)	8mA (default)
DACclk_25MHz	Output	PIN_43	2.5 V (default)	8mA (default)
DACout[7]	Output	PIN_38	2.5 V (default)	8mA (default)
DACout[6]	Output	PIN_39	2.5 V (default)	8mA (default)
DACout[5]	Output	PIN_33	2.5 V (default)	8mA (default)
DACout[4]	Output	PIN_34	2.5 V (default)	8mA (default)
DACout[3]	Output	PIN_31	2.5 V (default)	8mA (default)
DACout[2]	Output	PIN_32	2.5 V (default)	8mA (default)
DACout[1]	Output	PIN_28	2.5 V (default)	8mA (default)
DACout[0]	Output	PIN_30	2.5 V (default)	8mA (default)
altera_reserved_tck	Input		2.5 V (default)	8mA (default)
altera_reserved_tdi	Input		2.5 V (default)	8mA (default)
altera_reserved_tdo	Output		2.5 V (default)	8mA (default)
altera_reserved_tms	Input		2.5 V (default)	8mA (default)
clkin_50MHz	Input	PIN_23	2.5 V (default)	8mA (default)
<<new node>>				

图三 FPGA 引脚配置

实物连接如图四所示。



图四 实物连接图示

2 HDL 部分（以 verilog 编写，字体较小）

2.1 先贴 HDL

```
module Autogreat(clkin_50MHz,ADCclk_25MHz,DACclk_25MHz,ADCin,DACout);
    //50MHz 分频 25MHz，ADCin 读入 8 位 ADC 数据，DACout 发送 8 位转换数据，进行 DAC 转换
    input clkin_50MHz;
    input [7:0]ADCin;
    output reg ADCclk_25MHz;
    output reg DACclk_25MHz;
    output reg [7:0]DACout;
    reg [3:0]counter;
    reg [8:0]cnt500;//2^9=512，采集 500 次，进行冒泡
    //reg [7:0]max,tmp;//冒泡使用
    reg [7:0]tmp;
    reg [7:0]great;//定义增益使用
    reg [7:0]great1;//接受 great 的高八位
    reg [15:0]tmp16;
    reg [7:0]realgreat;//真正增益，相当于 great 右移 8 位
    //---@分频开始@---//
    always@(posedge clkin_50MHz)
    begin
        if(counter==1)
            begin
                ADCclk_25MHz<=~ADCclk_25MHz;
                DACclk_25MHz<=~DACclk_25MHz;
                counter<=0;
            end
        else if(counter==0)
            begin
                counter<=counter+1;
            end
    end
    //---@分频结束@---//
    //---@获取自动增益，并调整输出电压开始@---//
    always@(posedge DACclk_25MHz)
    begin
        tmp16<=(ADCin-128)*great1;//ADCin-128,取中线上面部分，
        //128 即 1000_0000 为正电压中心线，乘以增益，单边放大
        realgreat<=tmp16[15:8];//取得 tmp 高 8 位，即浮点数整数部分
        DACout<=128+realgreat;//真正 DAC 输出部分
        //---@实验成功语句@---//
        //---@DACout<=(ADCin-128)*1+128;@---//
        //---@实验成功语句@---//
    end
end
```

```

end
//---@对 ADCin 进行处理@---//
always@(posedge ADCclk_25MHz)
begin
    //---@冒泡取大开始@---//
    if(cnt500>=500)
        begin
            cnt500<=0;
            //max<=tmp;//取得 500 次采样最大值
            //great<=(64*256)/(max-128);//64 为设定电压值，相当于放大倍数，
            //64 乘以 256 相当于左移 8 位，需要 16 位储存，max-8'b10000000
            great<=(36*256)/(tmp-128);//此时 great 为 16 位，需要其高 8 位，
            //64 为 0100_0000，当设置输出为正负 2.5V 时用 64，比较时运用差值
            great1<=great[7:0];
        end
    else if(cnt500<500)//在两个信号周期内采样
        begin
            if(tmp<=ADCin)//当前 tmp 小于等于 ADCin
                begin
                    tmp<=ADCin;//冒泡取得比当前大的值
                end
            else
                begin
                    tmp<=tmp;//保持当前值
                end
            cnt500<=cnt500+1;//cnt500 自增
        end
    //---@冒泡取大结束@---//
end
//---@获取自动增益，并调整输出电压结束@---//
endmodule

```

2.2 再来说说 HDL 思路

FPGA 与外部端口在 1.2 节已经说明。整个 HDL 分为分频部分、读取 ADC 转换值部分、增益处理和 DAC 输出波形部分。

2.2.1 分频部分

第一个部分请见 always@(posedge clkin_50MHz)

```
begin
```

```
.....
```

end 部分，为二分频，分别产生 20MHz 和 AD 和 DA 时钟。

2.2.2 读取 ADC 转换值部分

第二个部分请见 always@(posedge clkin_50MHz)

```
begin
```

end 部分。其中输入信号为 100KHz，ADC 采样时钟为 25MHZ，所以一个信号周期可以采样 250 次，设置采集 500 次，利用冒泡法取得最大值，信号最大值的 AD 转换值储存在 tmp 当中；记录 500 次后，利用 tmp 取得正半周期的幅度，即“tmp-128”，因为根据《高速 AD/DA 模块使用指南 Rev 1.3》，其 AD 电路将正弦信号负半周衰减后抬升到正电压，“tmp-128”即为 tmp-8'b100_0000，即得到正半周期的幅度。

此时，再利用“great<=(36*256)/(tmp-128);”，得到与预设电压的增益或者衰减系数，注意 great 为 8 位，而(36*256)是将 36 左移 8 位，这里运用了 8 位拓展为 16 位，来就利用 FPGA 完成浮点数运算。具体浮点数运算，我会写在后面。如果没有写，就是我觉得写不清楚。

紧接着这句话，是“great1<=great[7:0];”，这句话其实并没有什么现实意义，但是最初写的时候，great 是 16 位的，后来写我就没有改。注意 great1 也是 8 位的。

2.2.3 增益处理和 DAC 输出波形部分

第二个部分请见 always@(posedge DACclk_25MHz)

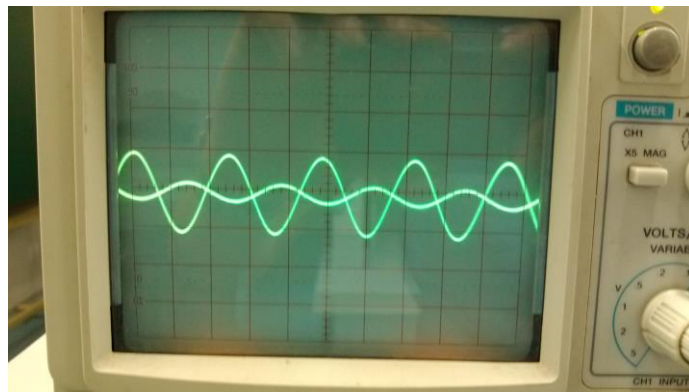
begin

end 部分。首先是“tmp16<=(ADCin-128)*great1;”，这句话中 tmp16 是 16 位数据，“ADCin-128”为 ADCin 的值减去轴的值（见 2.2.2），然后乘以增益（great1 大于等于 1 为增益，小于 1 则为衰减，这里都叫为增益），8 位乘以 8 位得到 16 位，恰巧这里又用了拓展位数进行浮点数运算。

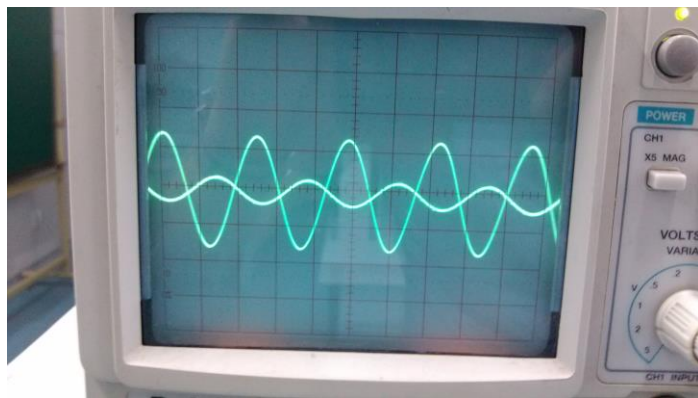
最后“realgreat<=tmp16[15:8];”和“DACout<=128+realgreat;”则为将 tmp1 的高八位给 realgreat，即将整数部分给 realgreat，而后一句则为 DACout 是当前计算得到的值再加上轴的值（128，即 8'b1000000）。最后 DAC 输出，得到波形。

3 实验结果（使用模拟示波器，读数没有直接显示在屏幕）

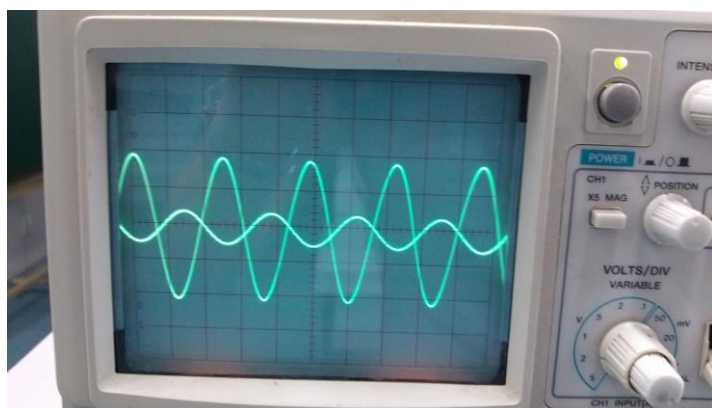
最后的实验效果可以见图五、图六、图七和图八。由于未将直接增益实验结果拍照，故无法在此对比效果。另外可以从我录制的视频中看到，随着输入信号幅值变化，输出信号可以稳定在一个设定值（设定为 36，即输出用十进制表示为：92~154）附近，但是波形稳定不是很好，另外当输入信号电压峰峰值超过 AD 模块限定的 10V 时，出现削顶。



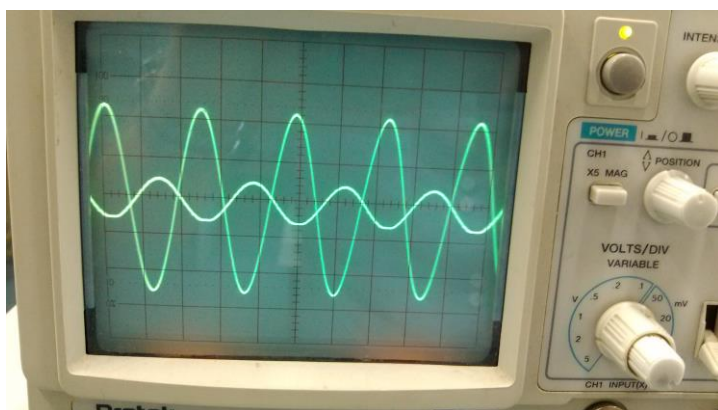
图五



图六



图七



图八