

數位實驗 Final Project

第八組

0612020 吳峻陞

0412106 羅宇呈

1. 設計理念

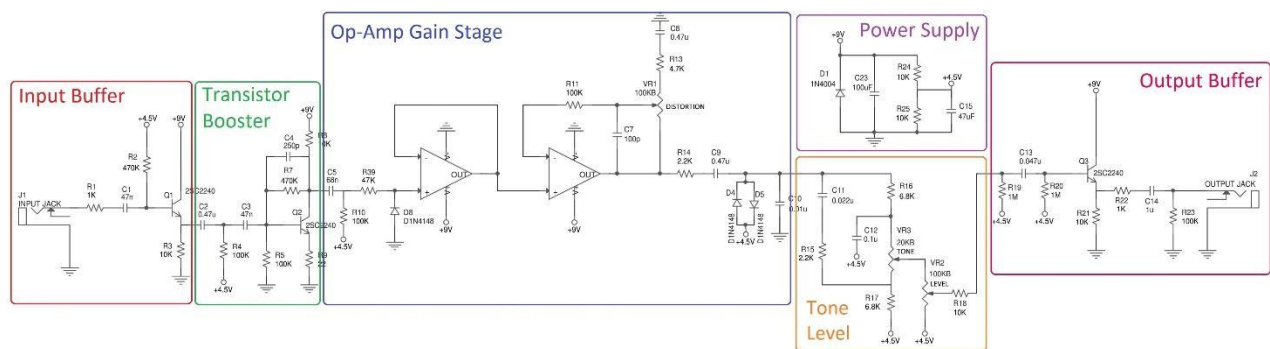
這次的主題是“吉他效果器”，整個設計有三個大部分：

a. 類比效果放大

b. 數位訊號效果處理

c. 輸出訊號放大

在類比效果放大這邊我們參考了一款經典的效果放大器“**BOSS DS1**”，他是一個可以調控音量、gain level 還有 tone 中高频加強的效果器，電路圖如下



Boss DS-1 Distortion Schematic DS-1A ElectroSmash.com

源頭先經過一次 high pa 的 input buffer，透過一個 BJT 實現 boost，然後送入 OP-GAIN，第一個 OP 是用來 buffer 的，以免雜訊放大，第二個 OP 就是真本事拉，從第三腳位接到第六腳位串聯的 RT 是 gain level 的調節頻，以一個可變來調控，尾端利用兩個 DIODE 的串聯做訊號的 CUT。後面進到 TONE LEVEL，電阻值的大小可以影響中高频增強的幅度，最後就通過一個 OUTPUT BUFFER 就輸出至 FPGA。

```
xadc_wiz_0 XADC (
    .dclk_in(clk_2),
    .daddr_in(8'h12),
    .den_in(enable),
    .di_in(0),
    .dwe_in(0),
    .vauxp2(vauxp2),
    .vauxn2(vauxn2),
    .vn_in(0),
    .vp_in(0),
    .do_out(data_adc),
    .reset_in(0),
    .eoc_out(enable),
    .drdy_out(ready)
);
```

ADC 模組的參數設定

數位訊號處理的部分都是在 FPGA 板子上面進行的，從類比過來的訊號先經過板子內建的 WIZARD 轉成數位訊號，之後以 FIR、DELAY、PWM 接續處理訊號。

```
FIR_Filter FIR (
    .data_in(data_adc[15:4]),
    .data_out(data_fir),
    .clk(clk_48k),
    .rst_n(rst_n)
);
delay delay_module(
    .clk_48k(clk_48k),
    .clk_195k(clk_195k),
    .rst_n(rst_n),
    .data_in(data_fir[27:16]),
    .data_out(data_delay)
);
PWM PWM1 (
    .in(data_delay[13:5]),
    .clk(clk),
    .rst_n(rst_n),
    .out(PWM_out)
```

三個 MODULE 的宣告處理

FIR 的參數 (由 MATLAB 求得) 和 CONVOLUTION 的部分

```
assign H[0] = 16'b1011111111110111;
assign H[1] = 16'b0011111111110110;
assign H[2] = 16'b1011111111110110;
assign H[3] = 16'b1011111111110000;
assign H[4] = 16'b0011111111110001;
assign H[5] = 16'b1011111111110101;
assign H[6] = 16'b0011111111110111;
assign H[7] = 16'b0011111111110101;
assign H[8] = 16'b1011111111110101;
assign H[9] = 16'b0011111111111000;
assign H[10] = 16'b0011111111111110;
assign H[11] = 16'b0011111111111100;
assign H[12] = 16'b1011111111111101;
assign H[13] = 16'b0011111111111010;
assign H[14] = 16'b0011111111111011;
assign H[15] = 16'b1011111111111010;
assign H[16] = 16'b0011111111111001;
assign H[17] = 16'b1011111111111000;
assign H[18] = 16'b1011111111111010;
assign H[19] = 16'b0011111111110110;
assign H[20] = 16'b1011111111110111;

assign data_out = data_reg[0]*H[0]+data_reg[1]*H[1]+data_reg[2]*H[2]+data_reg[3]*H[3]+data_reg[4]*H[4]+
    data_reg[5]*H[5]+data_reg[6]*H[6]+data_reg[7]*H[7]+data_reg[8]*H[8]+data_reg[9]*H[9]+
    data_reg[10]*H[10]+data_reg[11]*H[11]+data_reg[12]*H[12]+data_reg[13]*H[13]+data_reg[14]*H[14]+
    data_reg[15]*H[15]+data_reg[16]*H[16]+data_reg[17]*H[17]+data_reg[18]*H[18]+data_reg[19]*H[19]+data_reg[20]*H[20];
```

```

case(counter)
  2'd0 : begin
    write_en <= 0;
    data <= 0;
    cnt <= cnt;
    address <= cnt-15'd5000;
  end
  2'd1 : begin
    write_en <= 0;
    data <= 0;
    cnt <= cnt;
    address <= cnt-15'd10000;
    data_reg[0] <= data_read;
  end
  2'd2 : begin
    write_en <= 0;
    data <= 0;
    cnt <= cnt;
    address <= cnt-15'd15000;
    data_reg[1] <= data_read;
  end
  2'd3 : begin
    write_en <= 1;
    cnt <= cnt + 1;
    data <= data_in;
    address <= cnt + 1;
    data_reg[2] <= data_read;
  end
end
endcase

```

□DELAY 利用 4 個 CYCLE，先一個寫，後面 3

個

讀，就可以成功製作出 DELAY 效果音

```

case(cnt)
  9'd0 : begin
    clk_out <= 0;
    cycle <= 1;
    cnt <= count;
  end
  9'd511 : begin
    clk_out <= 1;
    cycle <= 0;
    cnt <= count;
  end
  default : begin
    cnt <= (cycle)? 9'd511-count : count;
    if (counter < cnt) begin
      counter <= counter + 1;
      clk_out <= clk_out;
      cycle <= cycle;
    end else begin
      counter <= 0;
      clk_out <= ~clk_out;
      cycle <= ~cycle;
    end
  end
end
endcase

```

□PWM 切割成 512 份來判讀

PWM 轉出來的訊號再接到 CLASS_D 放大做最後的聲頻和類比化，就可以得到我們想要的吉他效果器了！

2. 設計過程 & 碰到問題 & 解決方法

設計的過程中相當相當不順利，一部份是因為我們對這塊板子真的不是很熟，一部分是因為我們訂的題目難度有點太高 XDD，整個架構的藍圖一下子就出來了，很純粹的以真正的吉他效果當作底來執行，我們兩個就分頭工作，一邊把 DS1 的類比線路接出來，一邊完成 FIR、DELAY、PWM。

把類比的東西搞出來還蠻順利的，但是一下子就碰到了第一個問題“如何執行 ADC”，我們原本想說上網找人家音訊的專題看一下有沒有轉換的方法，偶而間就看到這個網址 <https://reurl.cc/WZ75D>，發現原來 FPGA 的 IP CATALOG 有內建的 XADC WIARD，參照他的數字執行了一遍，恩...沒用，只好繼續爬文以及修改 VERILOG 的頻率和 CONSTRAINTS，經過一個通宵的努力終於把 ADC 轉換的東西弄好了，可以前進到下一個步驟。(累積 12 小時)

後面將 ADC 弄好後變成 DELAY 的東西跑不出來，硬體的部分完全支撐不了這麼大型 CONV 的運算，但幸好有發現“SRAM”這個東西，讓硬體限制得以解決，但解決了不代表 CODE 沒有問題，沒弄好的話，會造成 OUTPUT 那邊直接沒有東西，費了千辛萬苦後，又發現 PWM 的輸出有問題，切割份數也有點不夠，接著的大概三天就一直陷入這個無止盡的 LOOP。(累積 24 小時)

在現場 DEMO 的前兩天我們把 FPGA 上面的東西算是告一個完結了，INPUT 那邊也多加了一個 HIGH PA 讓整個訊號穩定一點，但有一個致命的問題“不夠大聲而且雜訊太多”，只好搬動救兵把“CLASS_D 放大器”拿出來，看能不能透過增加 SWING 的方式把雜訊濾掉，後面如何讓整個東西和 CLASS_D 相容的部分我就不贅述了。(累積 30 小時)

這個中間還有包括了很多什麼板子突然沒聲音阿，FPGA 插 PMOD 插一差就突然 SHUT DOWN 的拉，元件莫名其妙壞掉諸如此類“神奇”的事情。

3. 可改善的地方

我們兩個有討論一下覺得應該是 FPGA 那邊 DELAY 和 PWM 的東西沒有做的很好，讓現場 DEMO 的地方不是很完美，雜訊還是很多，可能有時間一點可以把精度的部分做的再詳細一點點。

還有一個地方是類比的電路接線，沒有成功的用焊板 DEMO，麵包板本身雜訊比焊板多很多，使用焊板照理講應該可以再濾掉一部分的雜訊。

4. 結論

為了這個 PROJECT，我們兩個真的可以說是把期末考全部都賭上了，在現場 DEMO 的前一刻還是在改良，但是看到自己把這個困難的題目做出來，真的是有無比的成就：)