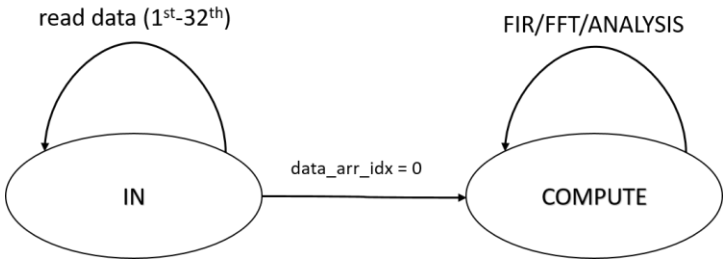
CVSD HW2 Report

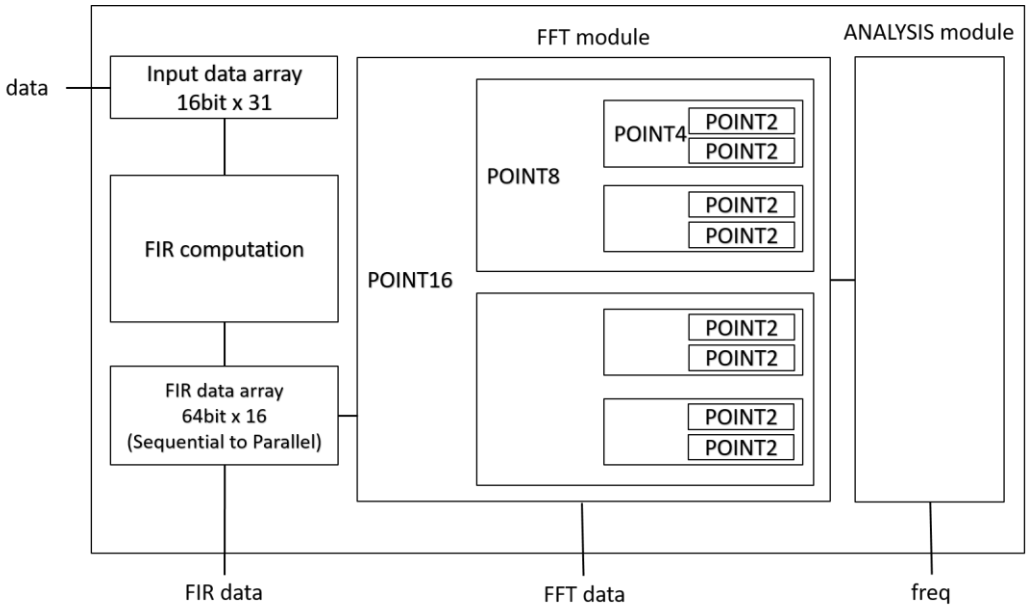
1. Finite State Machine:



a. 主要分為兩個 state。
b. IN:必須等到前 32 筆 data 輸入完後才進行第一次 FIR 運算，所以將此時與後面的運算做切割，設為第一個 state；另外為了 FIR 公式的易讀性、從 data_arr[31]開始輸入 data。
c. COMPUTE:包含開始進行 FIR 後的所有運算，同時也會繼續讀入 data。

2. Modules:

(省略部分接線)



a. Input data array:

為 16bit x 31 的 array，且實現了 shift register 的功能，開始進行 FIR 運算後，每個 clock 皆會將 data 往後 shift 一格，並將新的 input data 讀至 data_arr[0]。

b. FIR computation:

進行 FIR computation，適時地進行 sign extension，最後取 16bit 的結果。除了將 16bit 的結果輸出至 fir_d，也轉成 64bit 的 fir_data(此時沒

有虛部，後 32bit 全為 0)，並存至 fir_data_arr。

```
fir_arr[fir_arr_idx-1] = (temp[65]==1)?  {{{{8'hff},{temp[65], temp[30:16]} + 16'd1}},{8'h00}},{32'h00000000}} :
: {{{{8'h00},{temp[65], temp[30:16]}},{8'h00}}, {32'h00000000}};
```

c. FIR data array:

為 64bit x 16 的 array，為 32bit 實部+ 32bit 虛部、實部和虛部分別為 16bit 整數+ 16bit 小數；同時實現串列轉為並列的功能，每輸入 16 組 fir_data 後，在持續 1 個 cycle 的時間內將 fft_valid、done 拉至 1，即會輸出 FFT、analysis 的運算結果。

d. FFT module:

16-point-FFT 切為 2 個 8-point-FFT，8-point-FFT 中又切為 2 個 4-point FFT，4-point-FFT 中切為兩個 2-point-FFT。總共 4 級，包含

(1) 16-point-FFT x 1

(2) 8-point-FFT x 2

(3) 4-point-FFT x 4

(4) 2-point-FFT x 8

I/O: Input 為 64bit，output 為 32bit，在 8 個 2-point-FFT 做完運算後，將 32bit/32bit 的實部/虛部 truncate 為 16bit/16bit，並組合成 32bit 輸出結果至 fft_d0-fft_d15 及 ANALYSIS module。

e. ANALYSIS module:

先處理每一項的大小，令 temp_aa-temp_pp 為 33bit 的 reg，儲存每項 16bit x 16bit(實部平方)+ 16bit x 16bit(虛部平方) 的結果。

再來，為了比較出最大值，使用了 15 個 mux，並運用 parallel processing 來縮短 critical path。

```
temp_freq_ab = (temp_aa > temp_bb)? 4'b0000:4'b0001;
temp_freq_cd = (temp_cc > temp_dd)? 4'b0010:4'b0011;
temp_freq_ef = (temp_ee > temp_ff)? 4'b0100:4'b0101;
temp_freq_gh = (temp_gg > temp_hh)? 4'b0110:4'b0111;
temp_freq_ij = (temp_ii > temp_jj)? 4'b1000:4'b1001;
temp_freq_kl = (temp_kk > temp_ll)? 4'b1010:4'b1011;
temp_freq_mn = (temp_mm > temp_nn)? 4'b1100:4'b1101;
temp_freq_op = (temp_oo > temp_pp)? 4'b1110:4'b1111;
temp_freq_abcd = (temp_freq_ab > temp_freq_cd)? temp_freq_ab:temp_freq_cd;
temp_freq_efgh = (temp_freq_ef > temp_freq_gh)? temp_freq_ef:temp_freq_gh;
temp_freq_ijkl = (temp_freq_ij > temp_freq_kl)? temp_freq_ij:temp_freq_kl;
temp_freq_mnop = (temp_freq_mn > temp_freq_op)? temp_freq_mn:temp_freq_op;
temp_freq_abcdefgh = (temp_freq_abcd > temp_freq_efgh)? temp_freq_abcd:temp_freq_efgh;
temp_freq_ijklmnop = (temp_freq_ijkl > temp_freq_mnop)? temp_freq_ijkl:temp_freq_mnop;
freq = (temp_freq_abcdefgh > temp_freq_ijklmnop)? temp_freq_abcdefgh:temp_freq_ijklmnop;
```