Digital Signal Processing Integrated Circuit Design

Final Project

系所:通訊所 學號:111064523 姓名:張柏彥

1. 前言

本期末專題依照原作業四的設計步驟，依序進行演算法模擬、浮點數(Floating-point Simulation)與定點數模擬(Fixed-point Simulation)、Verilog 硬體描述語言設計，最後比較 Gate-level Netlist Simulation 與 Fixed-point Simulation 的結果。除此之外也針對原 Digital Filter 進行時序與面積的優化，利用課程中所提及的技術進行架構上的改善，並透過合成結果分析優化後的電路是否達到設定的規格。

2. 設計方法

2.1 系統與演算法模擬

本期末專題使用 Matlab 模擬一濾波系統，將取樣率為 8kHz，24bits/sample 的音訊資料作為輸入訊號，並通過一個低通濾波器後得到輸出訊號，記錄其數值並進行時域與頻域分析。利用 Matlab Filter Design tool 設計符合以下規格的 Low-pass digital FIR filter :

(i)20log(δ_s) < -40dB 阻帶衰減(Stopband Attenuation)

(ii)通帶邊緣頻率(Passband Edge Frequency) f_p = 1.5kHz

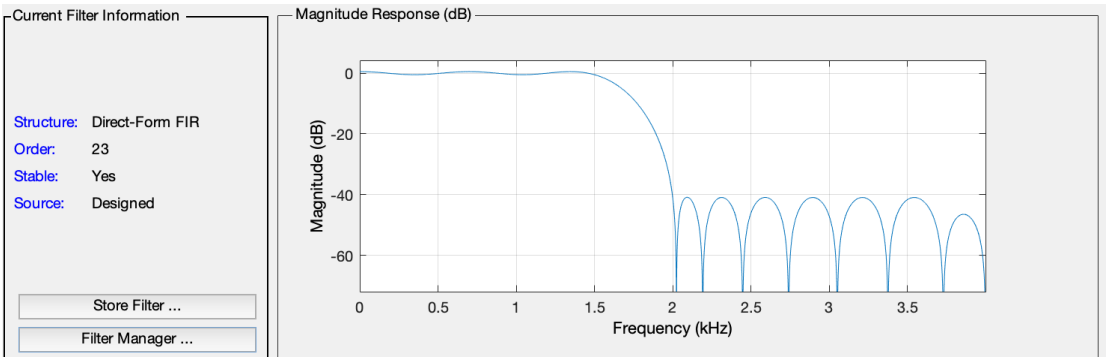
(iii)取樣頻率(Sampling Frequency) f_s = 8kHz

(iv)過渡頻帶(Transition Band) Δf = 0.5kHz

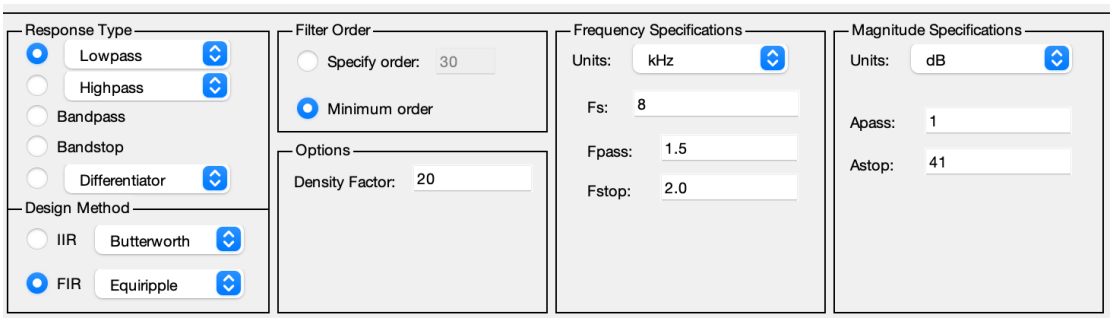
依據圖(三)所設定之參數進行濾波器設計，設計出來的濾波器一共有 24 個 tap 且係數具有對稱性。

```
coeff =

  Columns 1 through 12

    0.0160    0.0233    0.0046   -0.0234   -0.0189    0.0259    0.0435   -0.0156   -0.0854   -0.0256    0.1897    0.3951

  Columns 13 through 24

    0.3951    0.1897   -0.0256   -0.0854   -0.0156    0.0435    0.0259   -0.0189   -0.0234    0.0046    0.0233    0.0160
```

圖(一) Floating-point coefficients of the low-pass digital FIR filter

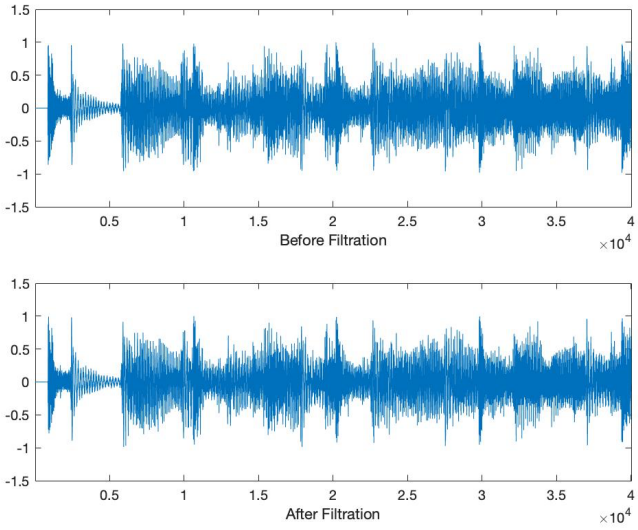


圖(二) Spectral mask of the target filter

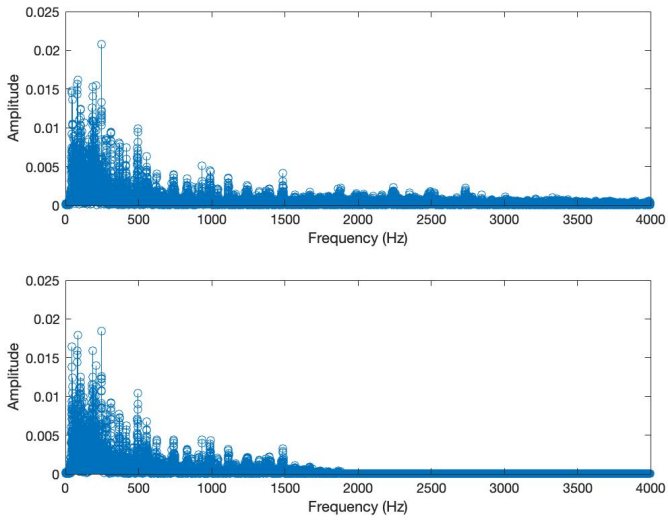


圖(三) Parameters setting of the target filter

圖(四)及圖(五)分別為輸入訊號及輸出訊號的時域與頻域分析圖。從頻域分析圖可以明顯發現高頻訊號增益在經過設計的濾波器訊號後被抑制。



圖(四) Time analysis of input/output signal



圖(五) Frequency analysis of input/output signal

2.2 浮點數模擬 (Floating-point Simulation)

使用 C++實作上述之濾波系統，首先讀取輸入訊號與濾波器係數(即 input_24.txt 及 coeff.txt)，並實現 conv 函數以驗證其運算是否與上述所得到的輸出訊號相同。

```
vector<double> r(size, 0);
for(int i=0; i<s.size(); i++){
    for(int j=0; j<h.size(); j++){
        r[i+j] += s[i]*h[j];
    }
}
```

圖(六) C++ code of conv function

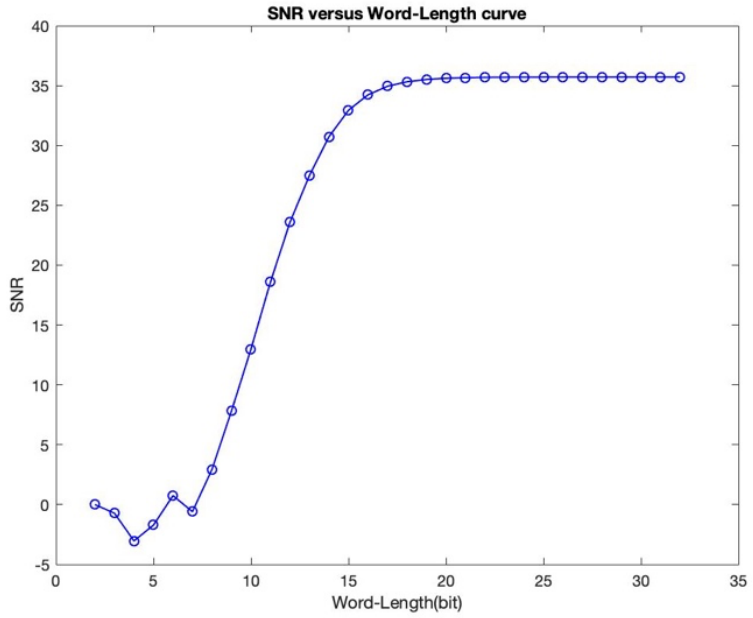
2.3 定點數模擬(Fixed-point Simulation)

同樣使用 C++實現，將輸入訊號改成取樣率為 8kHz，8bits/sample 的音訊資料以方便觀察 SNR 飽和的情形，建構 gFix 資料型態已自行定義定點數的乘法與加法運算。首先決定整數部分的長度，若資料經過四則運算後造成溢位(overflow)，則增加 1bit 的長度，反覆此步驟直到不會產生溢位。本期末專題設定整數部分長度為 2，其中最高位元為符號位(signed bit)。接著決定資料的 fraction word-length，其長度越長則精確度越高，SNR loss 也較低，而相對需付出的硬體成本也越高。因此本期末專題從 word-length=2 開始模擬至 word-length=32 並繪製 SNR versus word-length 曲線圖，其中 SNR 的計算公式如下:

$$SNR = 10\log_{10}\left(\frac{\sum_{n=0}^{N-1}[y(n)]^2}{\sum_{n=0}^{N-1}[y(n)-y'(n)]^2}\right)$$

圖(七) SNR formula

由圖(八)中的 SNR curve 所示，當 word-length=14 時，開始出現飽和的趨勢，而當 word-length=30 時，SNR 會飽和在 35.1004dB。本期末專題最後選擇 word-length=16，其中整數部分為 2bits，小數部分為 14bits，對應的 SNR 為 34.0692dB，SNR loss 為 1.0312dB。



圖(八) Output SNR versus input word-length curve

2.4 硬體描述語言設計

本期末專題設定的目標規格為：(i)提升至少 2 倍的 Clock rate 以及 (ii)降低至少 2 倍的 Area cost。為了優化原電路已達到設定之規格，分別使用了 Folding、Fine-Grain pipelining 及 Retiming technique 優化時序及面積。

2.4.1 Folded Architecture

由於卷積運算即乘法與加法運算，因此乘加器為 FIR 濾波器中最主要的運算單元，濾波器電路的大小也幾乎由乘加器的數量決定，在原電路中一共使用了 24 個乘法器與 23 個加法器。為了減少電路面積， 因此採用課程第四章提及的 Folded 架構，藉由減少乘法與加法器的使用數量以降低整體濾波器電路的面積成本。本期末專題僅使用 1 個乘法器與 1 個加法器，將 sampling rate 減少 1/24，透過降低吞吐率以換取電路面積的優化。

2.4.2 Fine-Grain pipelining

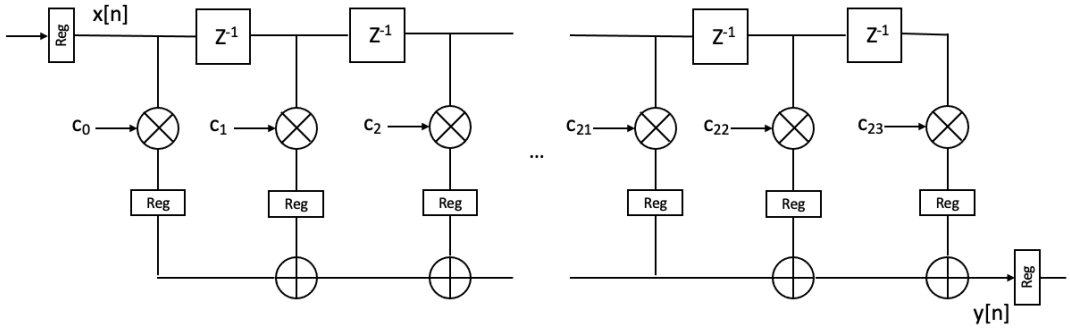
原電路的 critical path 為 $23T_A$，為了提高 clock rate，利用課程第三章提到的 pipeline processing 進行時序上的優化。Feed-forward cutset 並加上 pipelining registers，其 critical path 減少至 T_M，經實驗得到此電路可運作的最大 clock rate 約為 MHz(即 clock period 為 3.7ns)。為了更加接近 Fine-Grain pipelining，需使用 two-stage pipelined multiplier 來及縮短 critical path，並搭配第四章所提及的 retiming 技巧以平衡不同路徑。其中的 two-stage pipelined multiplier 使用 Synopsis DesignWare 提供的 DW02_mult_2_stage IP 實現，在 RTL simulation 時需注意匯入 DW behavior model，並輸入 -y /usr/cad/synopsys/synthesis/cur/dw/ sim_ver+libext+.v 指令。

2.4.3 Retiming

時序重置(Retiming)是在 Sequential circuit 中將 register 重新置放，用以降低 clock period 或 area 的最佳化方法。本期末專題在合成腳本 dc_syn.tcl 中，輸入 compile_ultra -scan -retime -timing 指令，針對一般邏輯優化，根據 register 前後相鄰的 path 進行數據計算分析，最後也成功將 clock period 降低至 2.4ns。

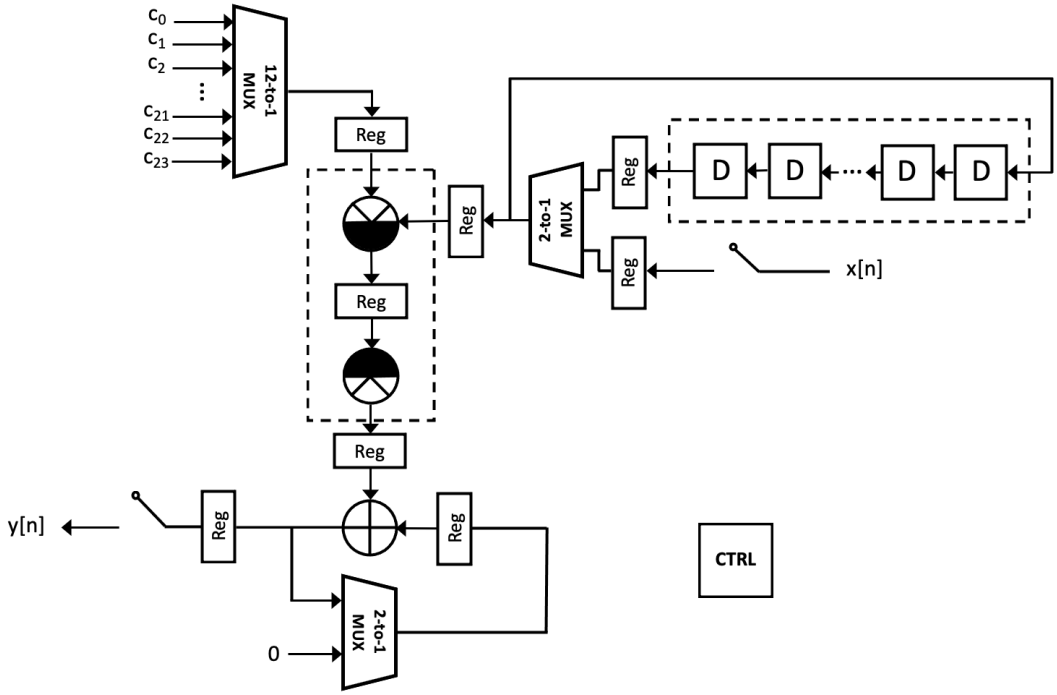
3. 電路架構設計

原電路的架構圖如圖(九)，為一個 Direct Form 架構。在輸入與輸出各連接了一個 reg 作為 input 與 output buffer。利用 pipeline processing，在每個乘法器運算後也連接 reg，最後再經過 23 個加法器運算。此電路的最大 Clock rate 為 166.67MHz，且由於加入了 input buffer, output buffer 及 1 個 pipeline register，其 latency 為 3。



圖(九) Architecture of Direct Form FIR filter (Benchmark circuit)

優化後的電路架構圖如圖(十)，為一個 Folded 架構，同樣在輸入與輸出加上 input 與 output buffer。此電路僅有一個乘法器與加法器，當 sample_in 訊號為 0 時，Mux 會選擇 FIFO(即 24 個 shift register)輸出的值，當 sample_in 訊號為 1 時，Mux 會選擇 input buffer 的值，並將該值與相對應的濾波器係數相乘同時儲存於 FIFO 當中，用於計算下一個 iteration 的輸出值，此外控制濾波器係數的 Mux 由 CTRL 中的 counter 值決定。訊號經過二級流水線乘法器後經過 1 個累加器，同樣由 CTRL 中的 a_select 訊號決定累加器是否歸零。最後在 sample_out 訊號為 1 時對輸出訊號進行取樣。



圖(十) Architecture of Folded FIR filter (Proposed circuit)

4. 實驗結果

利用 Synopsys Design Compiler 合成電路並執行 Gate-level Netlist Simulation 後，優化前與優化後的電路整體性能分析如圖所示。在設定輸入訊號長度為 40000，Tap 數同為 24 的條件下，兩電路皆使用 TSMC 0.13μm CMOS 製程技術進行合成。優化後的電路之最高操作頻率為 416.67MHz，時脈週期為 2.4ns，合成面積為 36343.030961μm^2，整體功耗為 7.6306mW，吞吐量為 277.8Mbps，模擬時間為 2305.4044μs。

Hardware Specification	Direct Form FIR filter	Folded FIR filter
Technology	TSMC 0.13μm CMOS	TSMC 0.13μm CMOS
Max. frequency (MHz)	166.67	416.67
Clock period (ns)	6.0	2.4
Total cell area (μm^2)	94507.837173	36343.030961
Power (mW)	4.8945	7.6306
Throughput(Mbps)	2666.72	277.8
Simulation time (μs)	240.181	2305.4044

圖(十一) Hardware Characteristics of Benchmark and Proposed FIR implementation

比較優化前後電路設計，在 Clock rate、Area、Power 及 Throughput 的取捨下，其最大操作頻率提升了 2.5 倍，且面積成本降低了 2.6 倍，達到了設定的目標規格。然而在功耗及吞吐量方面相較於原電路的表現較差，其功耗提升了 1.56 倍且由於取樣率只有原本的 1/24，因此其吞吐量降低了 9.6 倍。

面積分析：由於優化後的電路僅使用了一個乘法器與一個加法器，因此減少大量的面積成本，然而面積下降的比例並非完全由乘法器與加法器的數目決定。由圖(十)中可以發現相較於圖(九)的電路多了 1 個控制電路以及 3 個多工器，使用 folded 架構雖然能大量減少運算單元的使用，但是需要額外的控制訊號在不同相位時能夠選擇相對應的輸入訊號以及在正確的時間點對輸出訊號取樣，因此這些電路的作用在於控制輸入訊號以及濾波器係數的選擇。

時序分析：clock rate 的快慢幾乎取決於 critical path 的長短，由於原電路的 critical path 為 23 個加法器的運算時間，在經過 folded 並進行適當的 pipeline processing 之後，critical path 即變成乘法器的運算時間。為了盡可能接近 Fine-Grain pipelining，因此將原本的沒有 latency 的乘法器(即 DW_02_mult)，置換為二級流水線乘法器(即 DW_02_mult_2_stage)。在過程中找尋了置換 DesignWare IP 的方法，搭配 Design Compiler User Guide 中提供的 retiming 方法，成功將 Clock period 減少至 2.4ns。

```
# operating conditions and boundary conditions #

set cycle 2.4
create_clock -name clk  -period $cycle   [get_ports  clk]
```

圖(十二) Operating conditions in FIR.sdc file

```
Combinational area:              15478.590573
Buf/Inv area:                     1580.279408
Noncombinational area:           20864.440388
Macro/Black Box area:                0.000000
Net Interconnect area:          186900.996643

Total cell area:                 36343.030961
Total area:                     223244.027604
```

圖(十三) Area report of the proposed circuit

Power Group	Internal Power	Switching Power	Leakage Power	Total Power	(%)	Attrs	Cell Count
io_pad	0.0000	0.0000	0.0000	0.0000	(0.00%)		0
memory	0.0000	0.0000	0.0000	0.0000	(0.00%)		0
black_box	0.0000	0.0000	0.0000	0.0000	(0.00%)		0
clock_network	0.0000	0.0000	0.0000	0.0000	(0.00%)		0
register	5.9066	0.1129	1.9612e+07	6.0391	(79.14%)		552
sequential	0.0000	0.0000	0.0000	0.0000	(0.00%)		0
combinational	0.6045	0.9707	1.6329e+07	1.5915	(20.86%)		954
Total	6.5111 mW	1.0836 mW	3.5940e+07 pW	7.6306 mW			

圖(十四) Power consumption of the proposed circuit

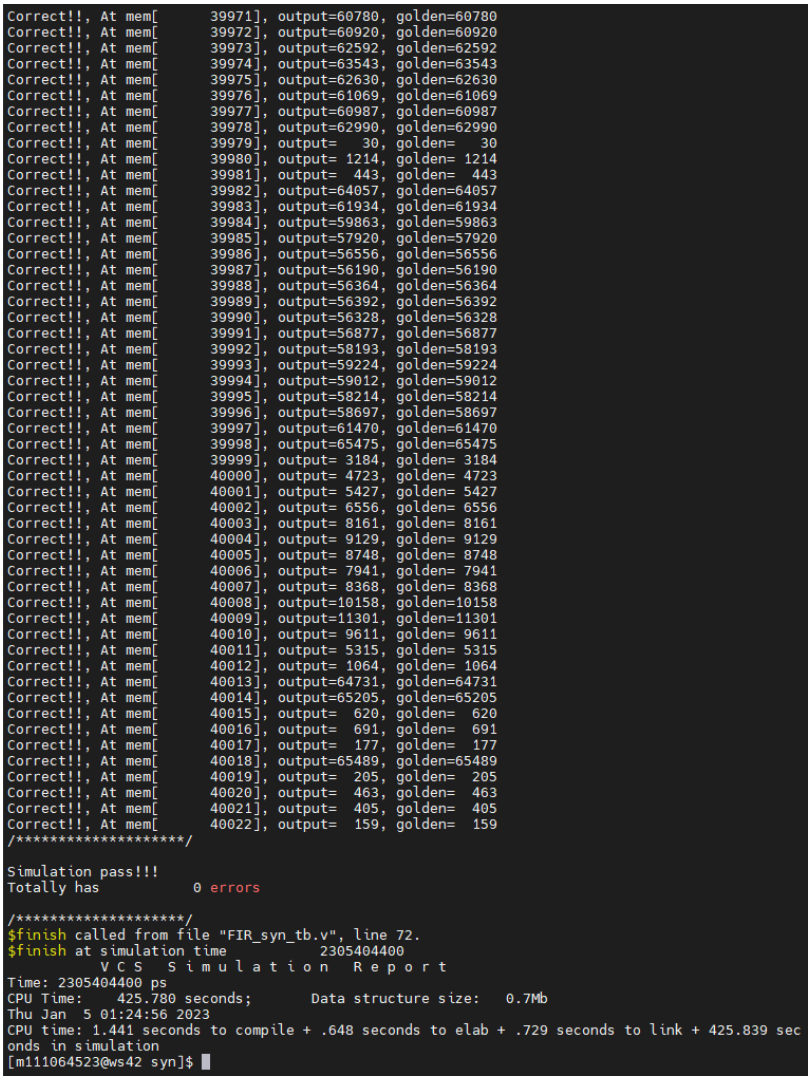
```
compile_ultra -scan -retime -timing

set_leakage_optimization false
compile -inc
```

圖(十五) Design Compiler instructions for Retiming

```
Startpoint: mult1_r_reg[11]
            (rising edge-triggered flip-flop clocked by clk)
Endpoint: __tmp150/U1/mult_x_1/clk_r_REG22_S1
          (rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: max

Des/Clust/Port     Wire Load Model       Library
------------------------------------------------
FIR                tsmc13_wl10           slow

Point                                                    Incr       Path
-----------------------------------------------------------------------------
clock clk (rise edge)                                    0.00       0.00
clock network delay (ideal)                              0.50       0.50
mult1_r_reg[11]/CK (SDFFRX4)                             0.00       0.50 r
mult1_r_reg[11]/Q (SDFFRX4)                              0.48       0.98 f
__tmp150/U1/A[11] (FIR_DW02_mult_2_stage_J1_0)           0.00       0.98 f
__tmp150/U1/U283/Y (BUFX20)                              0.18       1.16 f
__tmp150/U1/U202/Y (XNOR2X1)                             0.42       1.59 r
__tmp150/U1/U409/Y (OAI22X4)                             0.23       1.82 f
__tmp150/U1/U509/S (ADDFHX4)                             0.30       2.11 f
__tmp150/U1/U589/CO (ADDFHX2)                            0.24       2.35 f
__tmp150/U1/U510/S (ADDFHX4)                             0.27       2.62 f
__tmp150/U1/mult_x_1/clk_r_REG22_S1/D (SDFFHQX4)         0.00       2.62 f
data arrival time                                                   2.62

clock clk (rise edge)                                    2.40       2.40
clock network delay (ideal)                              0.50       2.90
clock uncertainty                                       -0.10       2.80
__tmp150/U1/mult_x_1/clk_r_REG22_S1/CK (SDFFHQX4)        0.00       2.80 r
library setup time                                      -0.18       2.62
data required time                                                  2.62
-----------------------------------------------------------------------------
data required time                                                  2.62
data arrival time                                                  -2.62
-----------------------------------------------------------------------------
slack (MET)                                                         0.00
```

圖(十六) Timing report of the proposed circuit

圖(十七)僅擷取 Gate-level Netlist Simulation 與 Fixed-point Simulation 的部分結果(39991 筆～40022 筆)，其中 golden 為 Fixed-point Simulation 的結果。可以從實驗結果看出兩者完全一致。



圖(十七) Simulation result on the terminal

5. 結論

在這個期末專題中，主要使用課程當中所提及的技巧，搭配閱讀一些論文以及 Design Compiler User Guide，了解 Folded FIR Filter Architecture 的運作原理及優化電路的實作方法，使用 Matlab 及 C++模擬濾波器系統並執行 Floating point 及 Fixed-point Simulation，透過 SNR curve 決定資料的 word-length，接著配合 Folding, Fine-Grain pipelining 及 retiming 技巧優化原電路的時序與面積，最後利用 Synopsys Design Compiler 合成電路並執行 Gate-level Netlist Simulation。除了模擬結果與 Fixed-point Simulation 一致之外，Clock rate 與 Area 相較於原電路皆有所優化並達到目標規格。

6. 參考文獻

[1] P.Premkumar, Dr.S.Kavitha, and S.Nandhini, “Design and Implementation of Folded FIR Filter Structures using High Speed Multiplier”, in International

Journal of Advanced Research in Electronics and Instrumentation Engineering, vol.3, Issue.12, Dec 2014.

[2] Synopsys, “Design Compiler User Guide”, Version D-2010.03-SP2, June 2010