Digital FIR Prototyping

Prac 5 Report
CSSE4010

Thulith W Mallawa
S44280422

1. Introduction

1.1 Aim

The aim of the following practical is to understand basic DSP algorithms, the graphical design process and to design and prototype a FIR filter using Xilinx System Generator.

The filter is to be created without any hardware optimizations at first, then performance to be benchmarked. This design is to then be optimized using pipelining, register-retiming and/or data broadcast. Followed by a performance benchmark and thus a comparison between the two designs and retuning if **required to improve the critical path delay at the cost of area complexity (resource usage).**

1.2 Filter Spec

$$y(n) = \sum_{i=0}^{16} a_i x(n-i)$$

The filter implemented is a pre-determined low-pass filter. It is to be designed with 16 taps, requiring 17 filter coefficients to satisfy the equation above.

2. Design 1: Direct Form FIR (No Optimization)

2.1 Design Block Diagram

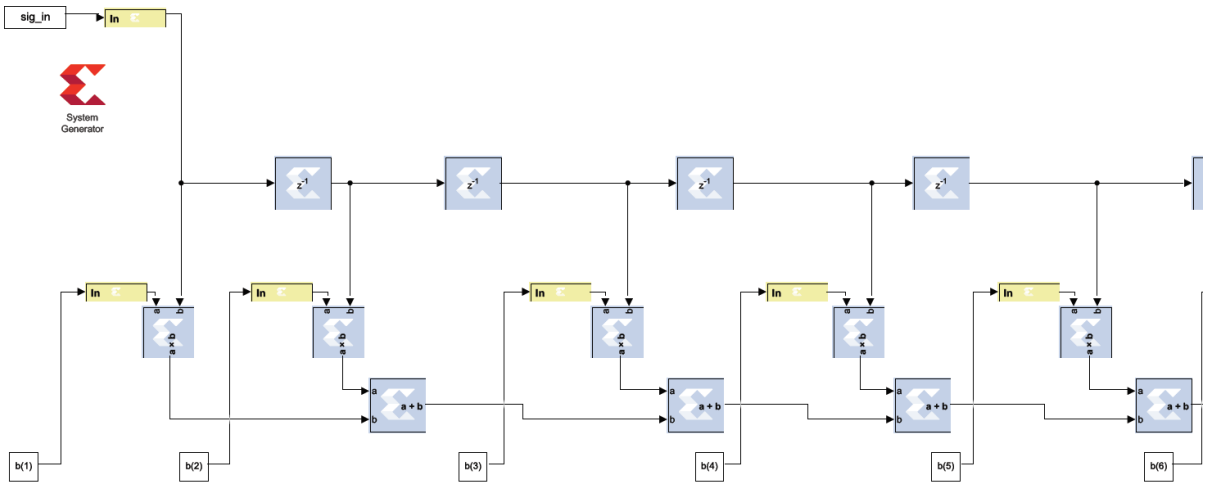


Figure 1 Zoomed in view of the Direct Form Implementation, initial four taps.

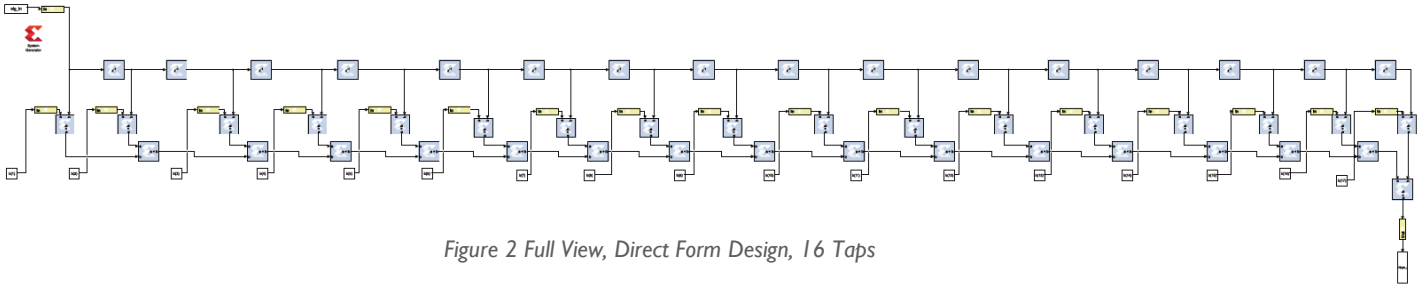


Figure 2 Full View, Direct Form Design, 16 Taps

2.2 Input / Output Characteristic – Functionality Testing

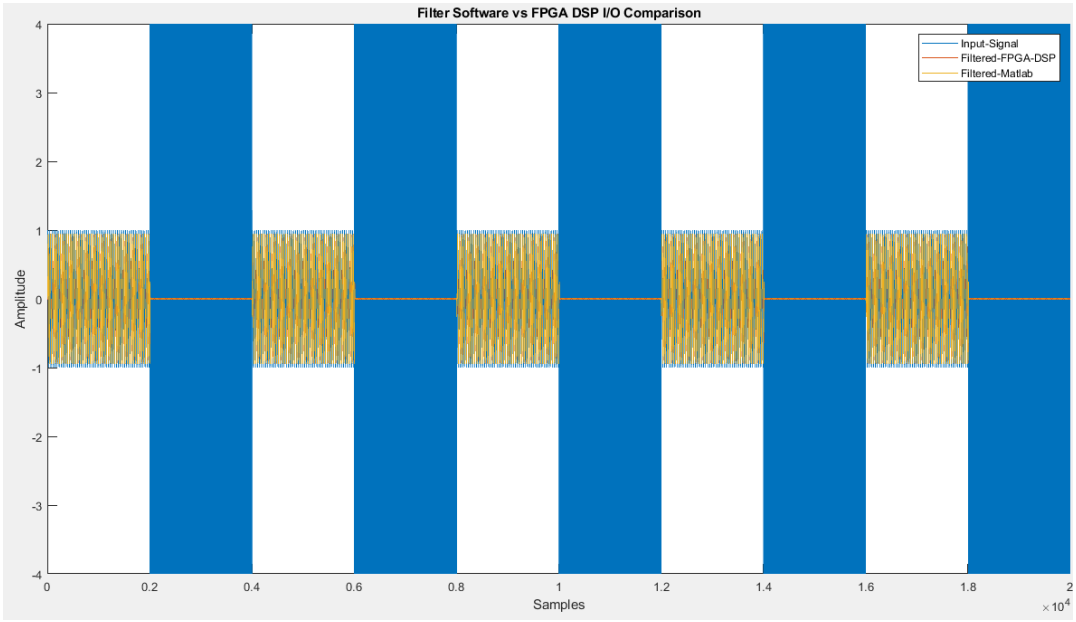


Figure 3 Filter Response I/O compared with software filtered in MATLAB vs FPGA output, refer to legend

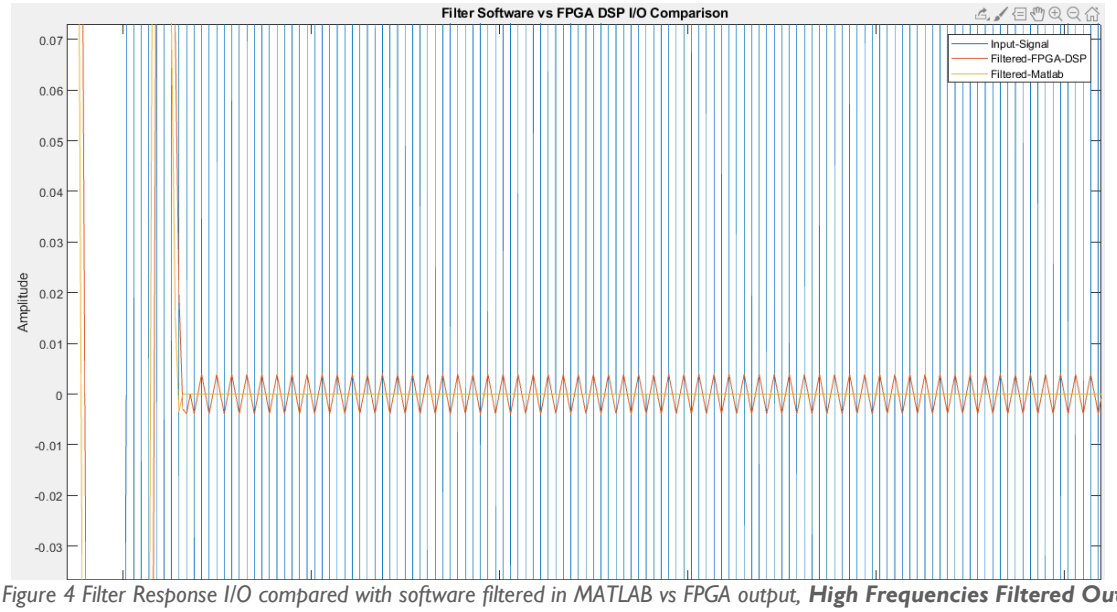


Figure 4 Filter Response I/O compared with software filtered in MATLAB vs FPGA output, ***High Frequencies Filtered Out***

Comparing the signals seen in the figures above, it is seen that fundamental functionality is achieved. The input signal *blue* contains the high frequency component of the input wave as opposed to the software filtered signal *yellow*, and the FPGA filtered signal *orange*. However, some quantization error is visible with the FPGA, this is expected as the software filters supposes a more ideal nature unlike the hardware implementation which takes quantization error into account.

Additionally, signal to noise ratio was calculated to fine tune filter coefficients, it was calculated using the following equation:

```
SNR = 10*log10(sum((sig_out_filt - filtered_signal_sw).^2)/sum(filtered_signal_sw.^2));
```

It was calculated to be **-16.136dB**

```
% Fixed-point word lengths for signal
 W=16;
 D=10;

% Fixed-point word lengths for filter coefficients
 Wc=12;
 Dc=10;
```

Through iterative testing (trying to minimize the word length while maintaining a good SNR), the above word lengths yielded the best SNR (**-16.136dB). Lower implies error signal energy to be small compared to the reference.**

2.3 Post Implementation Synthesis Data

The following data was calculated using an **FPGA Clock of 100ns, 10MHZ.**

Slack (ns)	Delay (ns)	Logic Delay (ns)	Routing Delay (ns)	Levels of Logic
38.324	61.621	30.704	30.917	90

The critical path delay is calculated as:

$$Tcpd = -(Slack - ClockConstrait)$$
$$Tcpd = -(38.324 - 100)$$
$$Tcpd = 61.676nS$$

This is apx. equivalent to the *Delay(ns)* column within the timing report, since *System Generator* measures latency **only** between delay blocks, slight modification was made (temporarily) to the design to measure the critical path delay accurately.

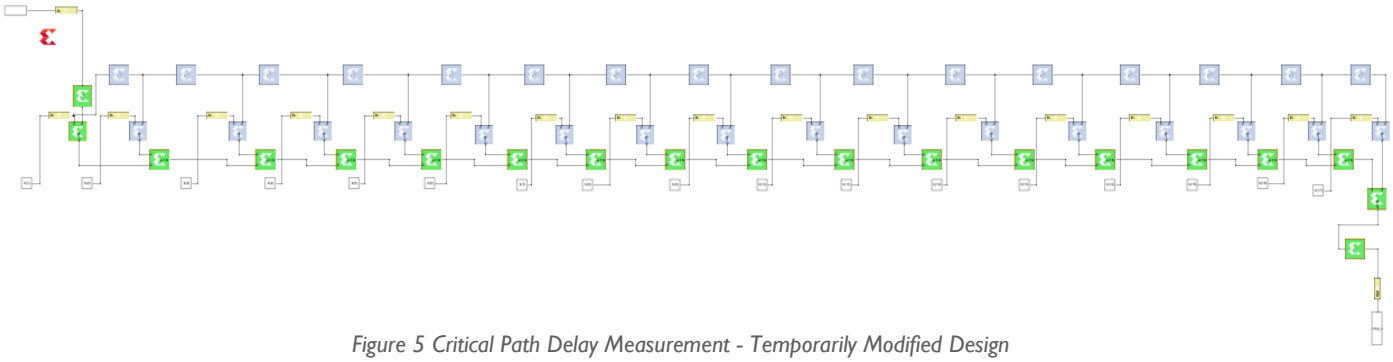


Figure 5 Critical Path Delay Measurement - Temporarily Modified Design

$$\therefore \boldsymbol{Maximum\ Frequency} = \frac{1}{Tcpd} = \frac{1}{61.676ns} Hz$$

$$= 16.2137\ MHz = \boldsymbol{Max\ Throughput}$$

The following figure shows the **area complexity** of this design which shows the resources used on the FPGA

Name	BRAMs (135)	DSPs (240)	LUTs (63400)	Registers (126800)
Prototype_Prac5	0	17	654	256

Figure 6 Area Complexity, Resource Usage

3. Design 2: Re-Timed and Pipelined FIR Design

3.1 Design Block Diagram

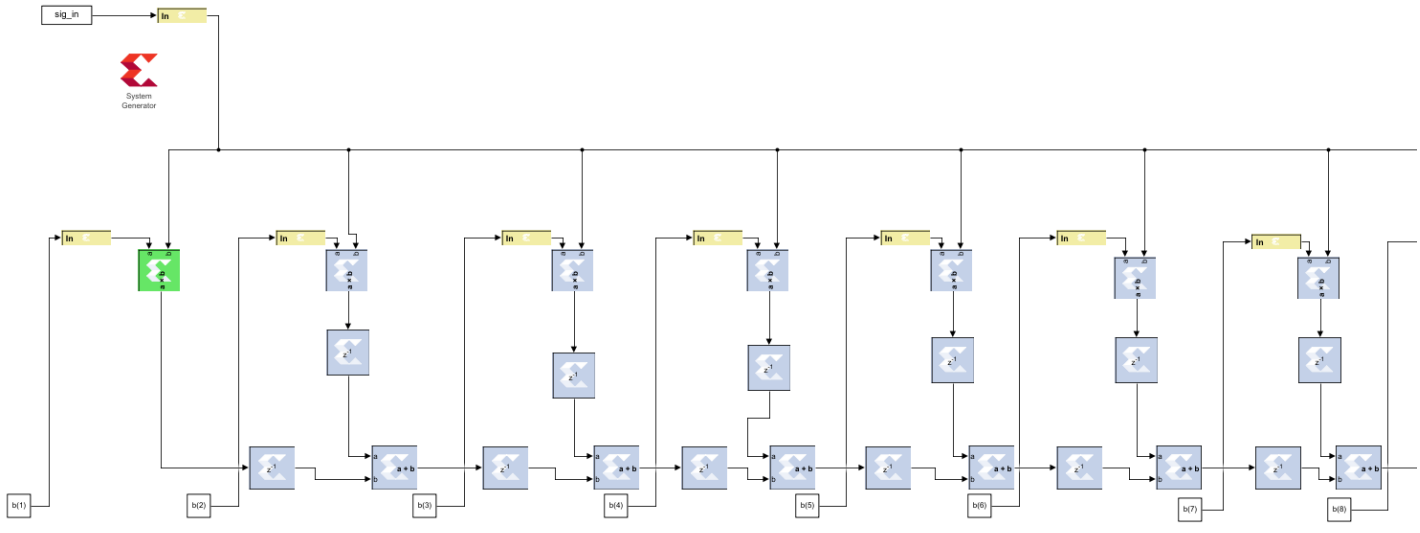
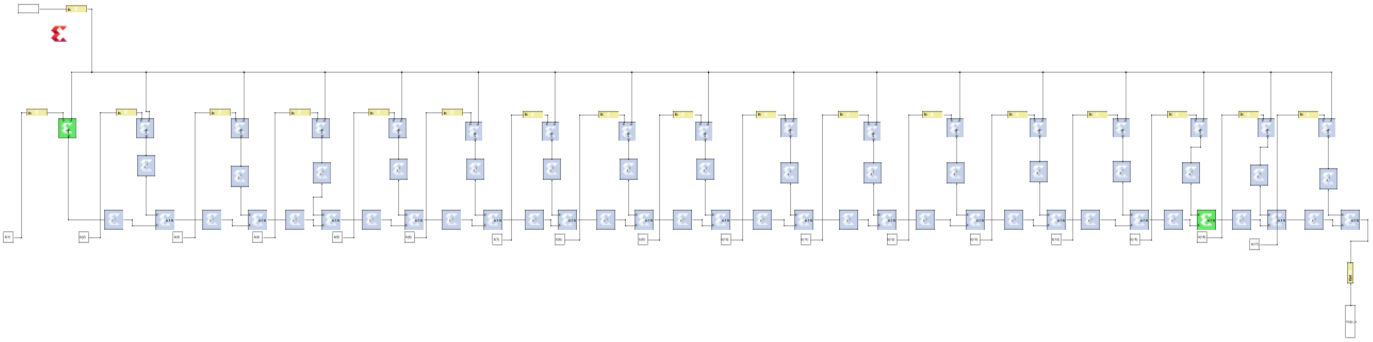


Figure 7 Retimed and Pipelined FIR Design

Figure 8 Zoomed Out, Retimed and Pipelined FIR Design



3.2 Input / Output Characteristic – Functionality Testing

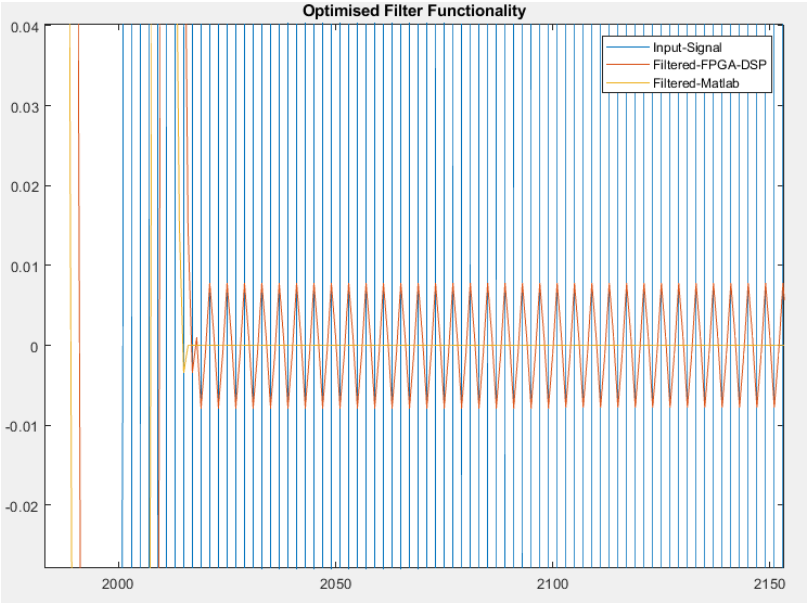


Figure 9 Optimized Filter: High Frequencies Filtered Out: Comparison

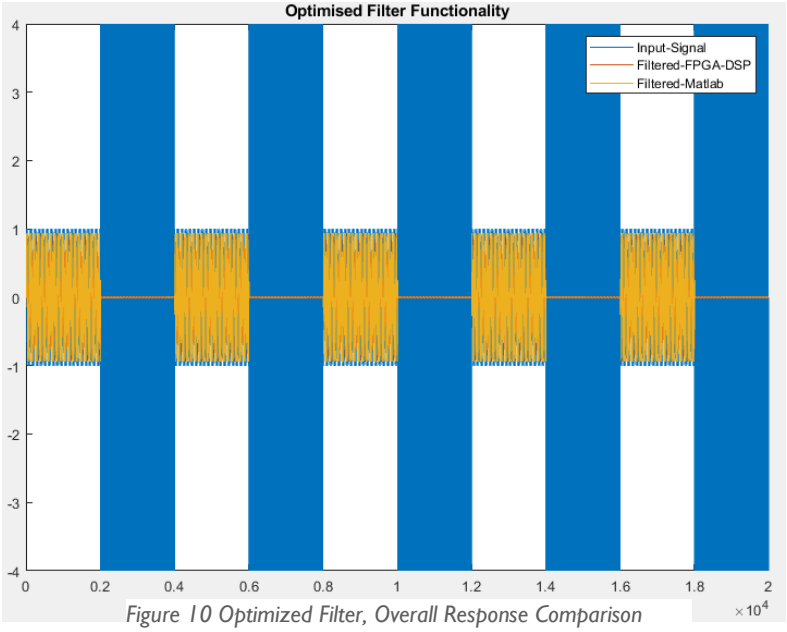


Figure 10 Optimized Filter, Overall Response Comparison

Comparing *figures 10 and 9* previously attained figures for the not optimized design *(figures 3, 4)*, no significant different are seen here. However, looking **the error metric**, an SNR of -16.136dB was calculated for the un-optimized design. For the optimized design the SNR had worsened and increased up to -10.1194dB.

Given that the pipelining was implemented at a depth of 2 (M = 2), the worsening of the SNR is caused by the additionally delay added to the system. **NOTE: The SNR remained the same when ONLY re-timing was done, it was only decreased after pipelining was implemented**.

3.3 Post Implementation Synthesis Data

The following data was calculated using an **FPGA Clock of 100ns, 10MHZ.**

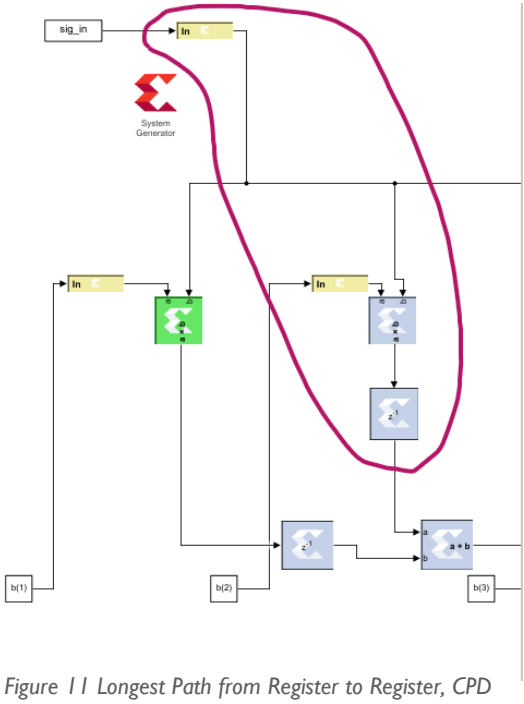
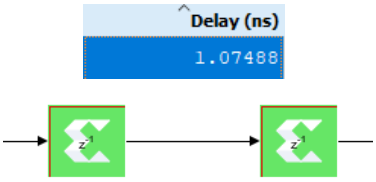
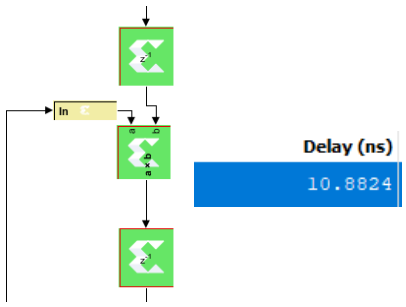


Figure 11 Longest Path from Register to Register, CPD

For this pipe lined designed, given that multipliers take more computational time (specially over adders), Tcpd = Tm as this is longest path from register to register (*figure 11*). To calculate the critical path delay, the design was temporarily modified to add delays as measuring points.





Figures above show that the time between the two delay blocks are ~1.07ns = TP1, and that the delay between the two-delay block with the multiplier is ~10.8824ns = TP2.

Therefore, Tm = TP2 – TP1 = 9.80752ns = Tcpd.

$$\therefore \boldsymbol{Maximum\ Frequency} = \frac{1}{Tcpd} = \frac{1}{9.80752ns} Hz$$

$$= 101.9625\ MHz$$

$$Fsamp = k * Fmax$$
$$k = 1$$
$$Fsamp = Fmax = Max\ Throughput$$

Name	BRAMs (135)	DSPs (240)	LUTs (63400)	Registers (126800)
Prototype_Prac5	0	34	1636	1024

Figure 12 Area Complexity, FPGA Resource Usage

4. Design Comparison

System Characteristics Timing:

	Un-Optimized	Re-Timed and Pipelined
Critical Path Delay	$61.676ns$	$9.80752ns$
Maximum Frequency	$16.2137\ MHz$	$101.9625\ MHz$
Maximum Throughput	$16.2137\ MHz$	$101.9625\ MHz$

System Characteristics Area Complexity:

Un-Optimized Design

Name	BRAMs (135)	DSPs (240)	LUTs (63400)	Registers (126800)
Prototype_Prac5	0	17	654	256

Re-Timed and Pipelined

Name	BRAMs (135)	DSPs (240)	LUTs (63400)	Registers (126800)
Prototype_Prac5	0	34	1636	1024

Filter Characteristics:

	Un-Optimized	Re-Timed and Pipelined
SNR	-16.136dB	-10.1194dB
Signal Word length	Bits: 16 Binary Points: 10	Bits: 16 Binary Points: 10
Filter Coeff Word length	Bits: 12 Binary Points: 10	Bits: 12 Binary Points: 10

5. Conclusions

The aim of this practical was to analyze and compare the differences between an un-optimized FPGA DSP implementation to an optimized one with pipelining and retiming. Comparing the results in *section 4*, a significant increase in the throughput of the filter can be seen once optimized however at the cost of increased use in register, look up tables and DSP slices on the FPBA. Thus, it is seen that, improved timing complexity is achieved at the cost of area complexity. Additionally, the register timing can likely be further improved to obtain a lower (better) SNR value in the optimized design.

Marking Criteria

Marks	Criteria
Simulation	
0	Not sufficient results are provided
1	Results shown only for the un-optimised design – working correctly
2	Results shown only for the optimised design – working correctly
3	Results shown for both designs - working correctly
4	Both designs are working correctly, and a comparison is provided
Report	
0	No evidence of content or work
1	Some content, insufficient explanation of circuit
2	Reasonable content, some explanation of circuit (both approaches are explained)
3	Good content, reasonable explanation of circuit (both approaches are explained)
4	Excellent content, good explanation of circuit (both approaches are explained)
Oral assessment	
0	No knowledge of the design
1	Very little knowledge of the design
2	Reasonable knowledge of the design
3	Good knowledge of the design.
4	Excellent knowledge of the design.
Total (12):	**Marker Initials:** **Date:**