



Master's Thesis

Design of Half band Filters for Decimation and Anti-Aliasing

by

Anusha Gundarapu

Supervisors: Professor Peter Nilsson and
MSc Yasser Sherazi

Department of Electrical and Information Technology
Faculty of Engineering, LTH, Lund University
SE-221 00 Lund, Sweden

Abstract

Power consumption awareness began worldwide around 1990–1992. Before that, only few markets required low-power integrated circuits (ICs). Today, every circuit has to face the power consumption issue, for both portable devices aiming at longer battery life and high-end circuits avoiding cooling packages and reliability issues that are too complex.

Digital signal processing (DSP) chips are used in audio applications, digital cameras, mobile phones, base stations, and in many other devices. Each of these devices has their own requirements for performance, power dissipation, and energy usage, which typically implements a particular trade-off among these entities.

The main objective of the thesis is to analyze half band wave-digital filters (WDF) in a STM 65nm complementary metal oxide semiconductor (CMOS) technology. The filters are simulated for energy dissipation, power consumption and performance. The core area is also estimated. The static power and dynamic power consumption has been estimated for Low Power High Threshold Voltage (LPHVT) by using different clock frequencies. The designs are synthesized using Synopsys Design Compiler, Place and Route (PNR) is done using SOC Encounter and power analysis is done using Prime Time PX tool.

Acknowledgments

First and foremost, I would like to thank my Professor **Mr. Peter Nilsson** for not only giving me the opportunity to do thesis work but also being a great mentor. His constant guidance and support made me to finish my master thesis successfully and helped me to improve my skills. Without his patience it would be impossible to finish this thesis.

I would also like to convey special thanks to **Mr. Yasser Sherazi** for his suggestions and help throughout the project.

Lastly, but not least, I would like to thank my husband and parents for their love, support, especially financial aid and hopes on me to finish my masters.

Anusha Gundarapu

Table of Contents

Abstract	3
Acknowledgments	5
Table of Contents	7
1 Introduction	9
1.1 Interpolation	12
1.2 Decimation	13
2 Power consumption	15
2.1 Basic Definitions	15
2.2 Power consumption components	15
2.2.1 Static power dissipation.....	16
2.2.2 Dynamic power dissipation.....	17
2.3 Power or Energy reduction	20
3 Design & Implementation	21
3.1 The Original Filter	21
3.2 The Trivial Filter	23
3.3 The Half-band Digital Filter	24
3.4 The Cascaded Filter	25
3.5 Multiplication unit for the original filter	27
3.5.1 The Coefficient a_0	27
3.5.2 The Coefficient a_1	27
3.5.3 The Coefficient $-a_2$	28
3.6 Truncation of the word length in the half-band filter	29
3.7 Simulation Results	31
4 Synthesis Results	33
4.1 Synthesis results of the original filter	33
4.1.1 Area Constraint.....	33
4.1.2 Speed constraint.....	34
4.1.3 Critical Path.....	35
4.2 Synthesis results of the cascaded filter	35
4.2.1 Area Constraint.....	35
4.2.2 Speed constraint.....	36
4.2.3 Critical Path.....	37
5 Place and Route Results	39
6 Power Analysis Results	43

6.1	Power consumption for impulse input.....	43
6.2	Power consumption for square wave input.....	45
6.3	Power consumption for random input.....	46
6.4	Energy dissipation results.....	48
7	Analysis of the results	51
8	Conclusions	53
9	Future Work.....	55
	References.....	57
	List of Figures.....	58
	List of Tables.....	60
	List of Acronyms.....	61
A.1	Converting Decimal number to Binary number	63
A.2	Synthesis Script.....	65
A.3	Place and Route Script.....	66
A.4	Power Analysis Script	68

CHAPTER 1

1 Introduction

Digital signal processing (DSP) and digital filtering has been developed a lot from last several decades and now it is the core for many diverse applications and products [1]. A block diagram of a DSP receiver system is shown in Fig. 1. The receiver system contains a RF front end, an analog-to-digital converter (ADC), a digital baseband for demodulation and control, finally a decoder that process the received data packets. The main focus is on the digital base band part of the receiver system [2].

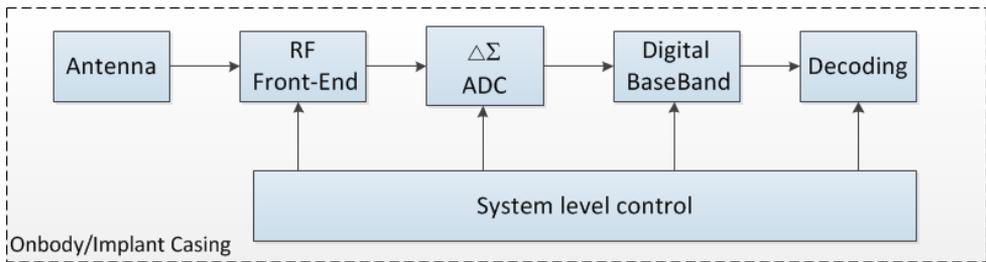


Fig. 1. DSP Receiver system [2]

Digital filtering is an integral part of many DSP applications. Digital filters are classified into finite impulse response (FIR) filters and infinite impulse response (IIR) filters. Non-recursive filters are FIR filters and the recursive filters are IIR filters. The IIR filters require a smaller order for the same set of specifications compared to the FIR filters. As the IIR filters approximate the gain and phase response of analog filters, they are used primarily where analog filters are used [3]. The FIR filters provide inherent stability and linear phase property. However, the IIR filters provide much more flexibility by easily converting analog filter to digital filters, eliminate degradation and produce a specific accuracy based on the number of bits used [3]. If the number of bits is not chosen properly it is difficult to control the IIR filters. It is an advantage to find ways to implement IIR filters that are stable and scalable [1].

Compared to other recursive filters, WDF's maintain stability under finite arithmetic conditions. A particularly suitable WDF is the lattice wave

digital filter (LWDF) [4], because LWDF exhibits excellent stability properties under several nonlinear operating conditions [1]. The LWDF is completely characterized by a set of coefficients (γ) that have excellent dynamic range and low word-length requirements [1]. LWDFs show low sensitivity in the pass band and high sensitivity in the stop band [5].

Except for having low sensitivity to coefficient variations in the stop band, LWDFs have good properties that make them well suited for implementation of broadband digital filters: [5]

- WDF's are derived from real lossless reference analog filters by preserving the order of original analog system
- Exhibits excellent stability properties under several nonlinear operating conditions
- LWDFs are characterized by a set of coefficients that have excellent dynamic range and low word-length requirements
- The LWDF (if designed correctly) is also free from round-off and overflow conditions
- Low sensitivity to coefficient round off error
- Design with simple equations and iterations
- The multiplication coefficients can be implemented with few shifters and adders
- In most of the cases, half of the multiplication coefficients are zero.

The basic building blocks of LWDFs are digital approximations of analog components like capacitors and inductors. In the digital domain the capacitor is represented by a delay, or in transform notation by Z^{-1} . The inductor is in the same way represented by a negated delay $-Z^{-1}$. To interconnect those components an adaptor is needed. There exist different kinds of adaptors like two-port and three-port adaptors are the most common. In LWDFs only two-port adaptors are used, its symbol and internal representation is shown in Fig. 2. To each port a component is connected. The adaptor coefficient ' α ' defines the ratio between the incident and the reflected wave from each port. If it is zero, there is no reflected wave and the adaptor acts as a feed through. If it equals to one the whole wave is reflected and nothing passes through the adaptor [5].

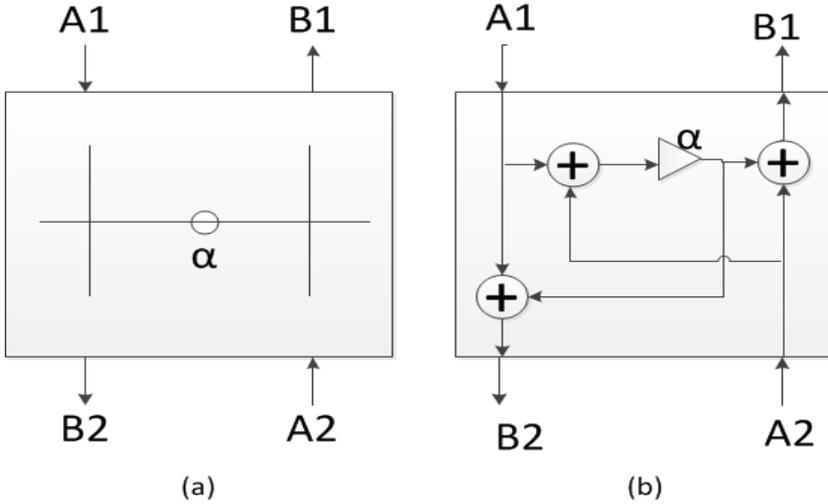


Fig. 2. (a) Adaptor symbol (b) Internal structure of adaptor [5]

In Fig. 2, the A and B represent the inputs and outputs of the adaptor. In this design, two-port adaptors that have two inputs and two outputs are considered. Inside each adaptor there are three adders and a multiplier. The multipliers are the filter coefficients alpha that characterizes the LWDF [5].

The Fig. 3 shows the filter design using three adaptors. The number of multipliers in Fig. 3 is equal to the filter order. For order N there are $(N+1)/2$ stages and a maximum of N adaptors [1].

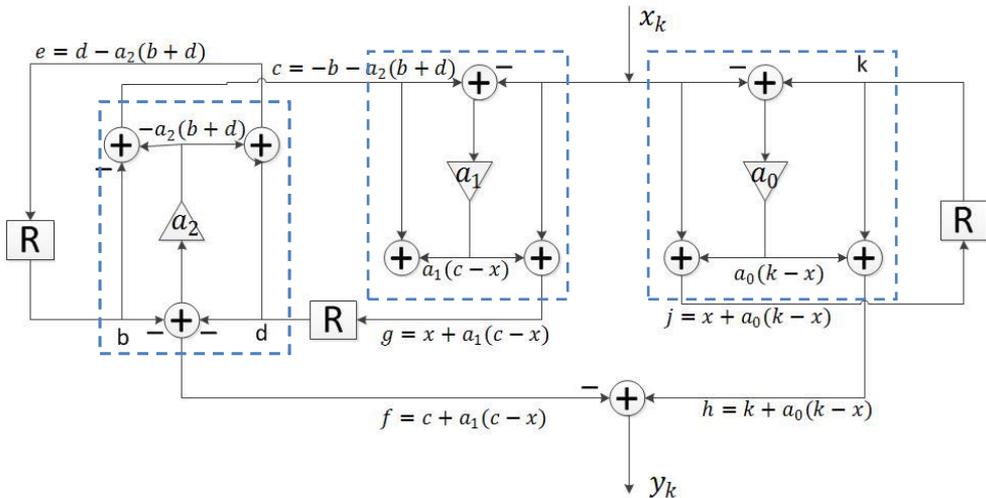


Fig. 3. Filter design using adaptors

Half-band filters are widely used in multi-rate signal processing applications when interpolating/decimating by a factor of two [6]. Multi-rate signal processing is done by changing the sampling rate of the system. The process of converting a signal from a given rate to a different rate is called sampling rate conversion. The systems that employ multiple sampling rates in the processing of digital signals are called multi-rate digital signal processing systems [7]. The primary motivation for using half-band filters is the existence of very efficient, stable and linear phase recursive and non-recursive structures for their realization. These structures have approximately half of the complexity of conventional filter structures primarily due to the fact that half of their multiplier coefficients are zero [8].

1.1 Interpolation

The process of up sampling the low-rate signal $x(n)$ into a high-rate signal $y(n)$ is called interpolation. The equation for an upsampler is

$$y(n) = x(n)|_{n=nI} = x(nI); \quad n, I \in \{integers\}$$

The block diagram for the up sampler is shown in fig. 4, The rate of input signal $x(n)$ is F_x , by passing through the interpolator the rate of output signal is $I F_x$, Where I is the interpolation factor.



Fig. 4. An up sampling element

In fig. 5, the plot shows an example of up sampling. The sampling rate of input $x(n)$ has up sampled by a factor of 2 and plotted in $y(n)$.

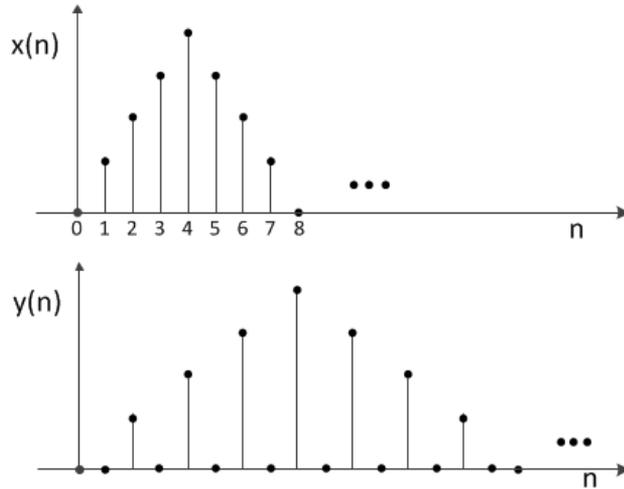


Fig. 5. Demonstration of Interpolation by a factor of 2

1.2 Decimation

The process of down sampling the high-rate signal $x(n)$ into a low-rate signal $y(n)$ is called decimation. The equation for the down sampler is

$$y(n) = x(n)|_{n=nD} = x(nD); \quad n, D \in \{integers\}$$

The block diagram for the down sampler is shown in fig. 6, The rate of input signal $x(n)$ is F_x , by passing through the decimator the rate of output signal is F_x/D , Where D is the decimation factor.



Fig. 6. A down sampling element

In fig. 7, the plot shows an example of down sampling. The sampling rate of input $x(n)$ has down sampled by a factor of 2 and plotted in $y(n)$.

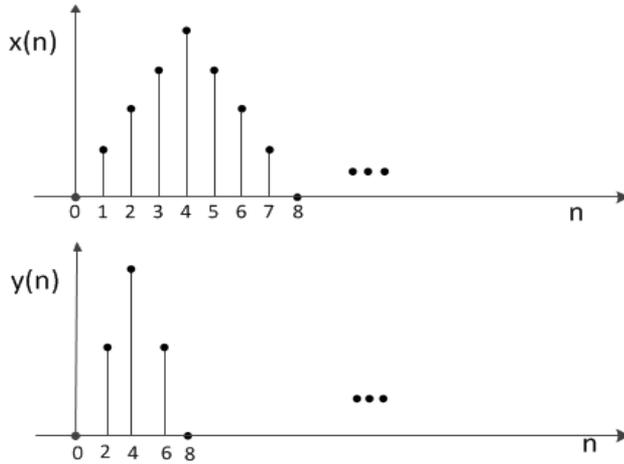


Fig. 7. Demonstration of Decimation by a factor of 2

Half-band filters have two important characteristics, the passband and stopband ripples must be the same, and the passband-edge and stopband-edge frequencies are equidistant from the half-band frequency $\pi/2$ [6], which is shown in Fig. 8.

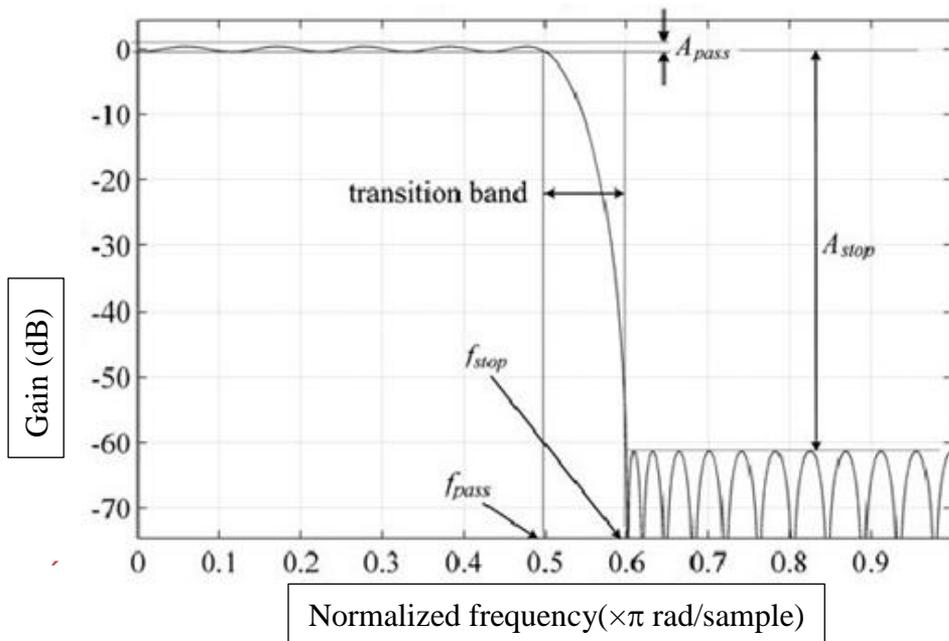


Fig. 8. Frequency response of LPF [8]

CHAPTER 2

2 Power consumption

This chapter explains the basic concepts of power dissipation, power optimization techniques [9] and the power results obtained for the designs.

2.1 Basic Definitions

The instantaneous power is given as:

$$P(t) = i_{DD}(t)V_{DD}$$

Energy over some time interval T is given as:

$$E = \int_0^T i_{DD}(t)V_{DD} dt$$

Average power over time interval T is given as:

$$P_{avg} = \frac{E}{T} = \frac{1}{T} \int_0^T i_{DD}(t)V_{DD} dt$$

Power is measured in Watts (W) and Energy is measured in Joules (J) or Watts hours (Wh).

2.2 Power consumption components

Power dissipation in CMOS circuits comes from two main components [10]

- Static dissipation
- Dynamic dissipation

$$P_{total} = P_{static} + P_{dynamic}$$

Static dissipation is caused by:

- Sub-threshold leakage
- Gate leakage
- Leakage currents through P-N junctions

Dynamic dissipation is caused by:

- Charging and discharging of useful and parasitic load capacitances.
- Short circuit current

2.2.1 Static power dissipation

The static power is dissipated due to the leakage components in the circuit.

2.2.1.1 Sub-threshold leakage

Sub-threshold or weak inversion conduction current is the leakage current that flows in between drain and source when the gate voltage is below the threshold voltage [9]. The sub-threshold leakage current is expressed as

$$I_{sub} = A e^{\frac{q}{n k T} (V_{GS} - V_{TH0} - \gamma' V_{SB} + \eta V_{DS})} (1 - e^{-\frac{q V_{DS}}{k T}})$$

where

$$A = \mu_0 C'_{ox} \frac{W}{L_{eff}} \left(\frac{kT}{q}\right)^2 e^{1.8}$$

V_G , V_D , V_S , and V_B are gate voltage, drain voltage, source voltage and body voltage respectively.

γ' is the linearized body effect coefficient

η is the Drain Induced Barrier Lowering (DIBL) coefficient

C_{ox} is the gate oxide capacitance per unit area

μ_0 is the zero bias mobility

n is the sub-threshold swing coefficient of the transistor

V_{TH0} is the zero bias threshold voltage.

2.2.1.2 Gate leakage

Direct tunneling gate leakage is due to the tunneling of electrons or holes from the bulk silicon through the gate oxide potential barrier into the gate. The tunneling current increases exponentially with decrease in oxide thickness. It also depends on the device structure and the bias condition [9]. The direct tunneling is modeled as

$$J_{DT} = A (V_{ox}/T_{ox})^2 \exp\left(\frac{-B(1-(1-V_{ox}/\phi_{ox})^{3/2})}{V_{ox}/T_{ox}}\right)$$

where

J_{DT} is the direct tunneling current density

V_{ox} is the potential drop across the thin oxide
 ϕ_{ox} is the barrier height of tunneling electron
 T_{ox} is the oxide thickness.

2.2.1.3 Leakage currents through p-n junctions

Drain-to-well and source-to-well junctions are typically reverse biased causing p-n junction leakage currents. A reverse biased p-n junction leakage has two main components: one is minority carrier diffusion/drift near the edge of the depletion region and the other is due to the electron-hole pair generation in the depletion region of the reverse biased junction [9].

In the presence of a high electric field ($>10^6 V/cm$), electrons will tunnel across the reverse biased p-n junction. A significant current can arise as electrons tunnel from the valance band of the p-region to the conduction band of the n-region. Tunneling occurs when the total voltage drop across the junction is greater than the semiconductor band-gap [9].

2.2.2 Dynamic power dissipation

Average dynamic power dissipation is given as [10]

$$P = \alpha \times C_L \times f \times V_{DD}^2$$

Where

α is the switching activity factor
 C_L is the load capacitance
 f is the circuit frequency
 V_{DD} is the supply voltage

2.2.2.1 Charging and discharging of useful and parasitic load capacitances

The dynamic power is caused due to charging and discharging of the useful and parasitic capacitances in the circuit. Capacitances are everywhere in the circuit [10].

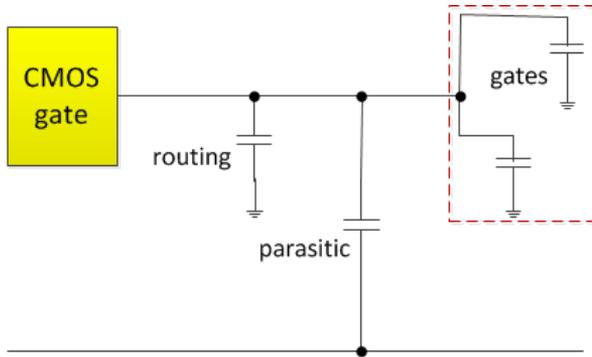


Fig. 9. A CMOS Circuit [10]

- Capacitance due to transistors structure

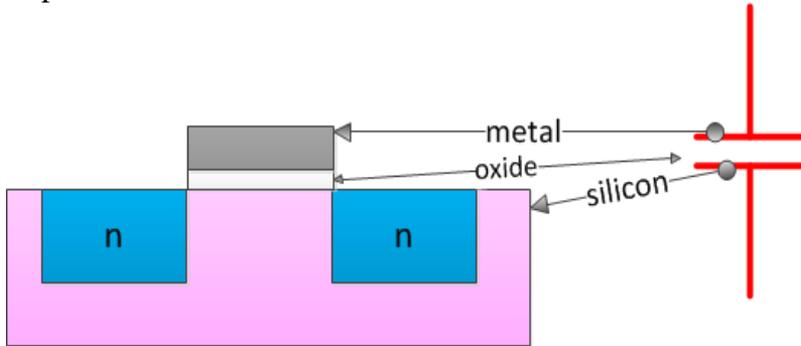


Fig. 10. Transistor structure [10]

- Capacitance due to routing

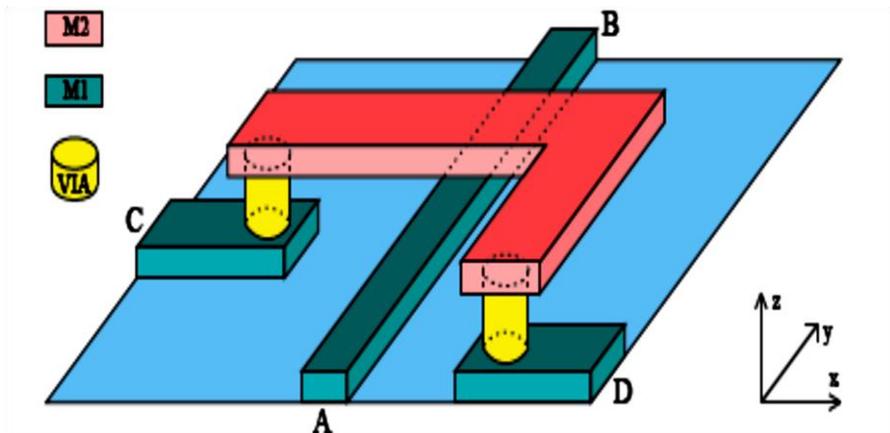


Fig. 11. Routing [10]

- Parasitic capacitance

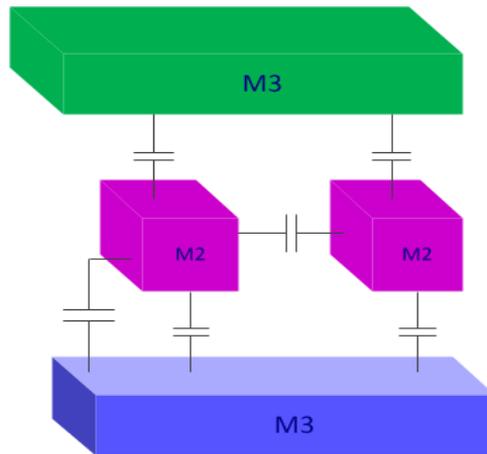


Fig. 12. Parasitic capacitance [10]

- input/output pad capacitance

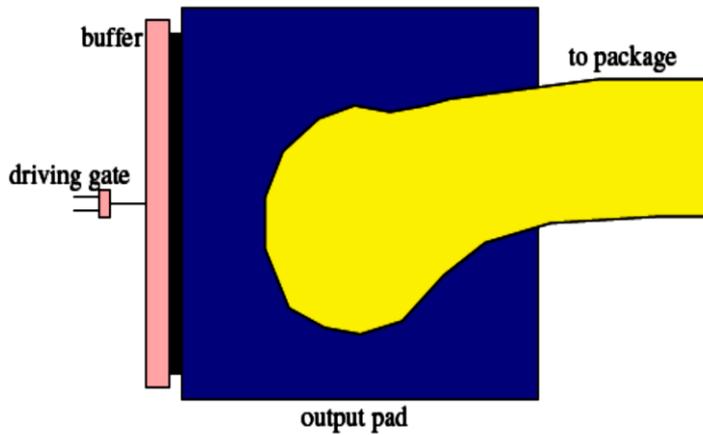


Fig. 13. I/O pad [10]

2.2.2.2 Short circuit current

Short circuit current occurs when both the N and P transistors are ON while the input switches [10] as shown in Fig. 14. The power dissipation due to the short circuit is minor compared to the overall power dissipation, which therefore can be ignored.

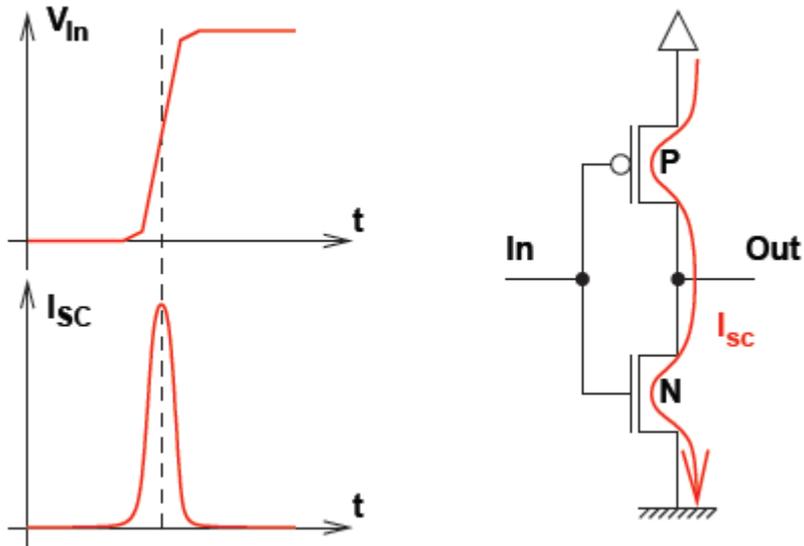


Fig. 14. Short circuit current in CMOS inverter [10]

2.3 Power or Energy reduction

Power can be reduced at all levels of the chip designing. Typical methods for reducing power are [10]

- Use better algorithms and data structures
- Use better technology
- Use smaller gates
- Use better placement and routing
- Reduce power supply
- Reduce frequency
- Reduce switching activity

3 Design & Implementation

3.1 The Original Filter

The structure of a third order original filter with seven bit long coefficients is shown in Fig. 15, considering the multiplication coefficients as shown in Table I. In Fig. 15, x_k is a 12-bit input and y_k represents a 15-bit output of the original filter. The 12 bits input is feed to the adder and the output of the adder is 13 bits by including the carry bit. The adders are implemented as ripple carry adders. The multipliers are implemented by using adders as shown in section 3.5. By selecting the required number of bits from the multiplier output, the output width of the filter is maintained to be 15 bits.

TABLE I. MULTIPLICATION COEFFICIENTS OF THE ORIGINAL AND THE CASCADED FILTER

Coefficient	Original filter		Cascaded filter	
	Decimal	Binary	Decimal	Binary
a_0	0.375000	0011000	0.0	00
a_1	0.578125	0100101	0.5	01
a_2	-0.328125	1101011	0.0	00

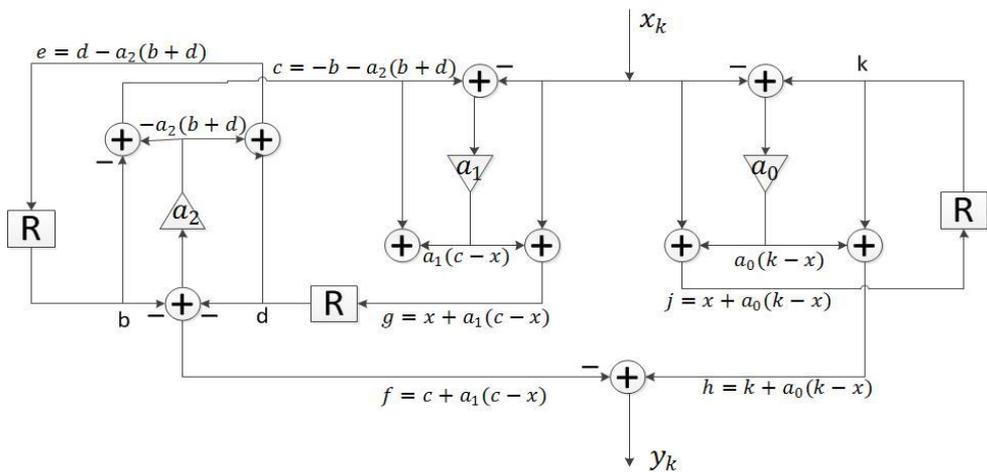


Fig. 15. The third order original filter

The frequency response of the original filter is shown in Fig. 16, which is obtained from MATLAB simulations.

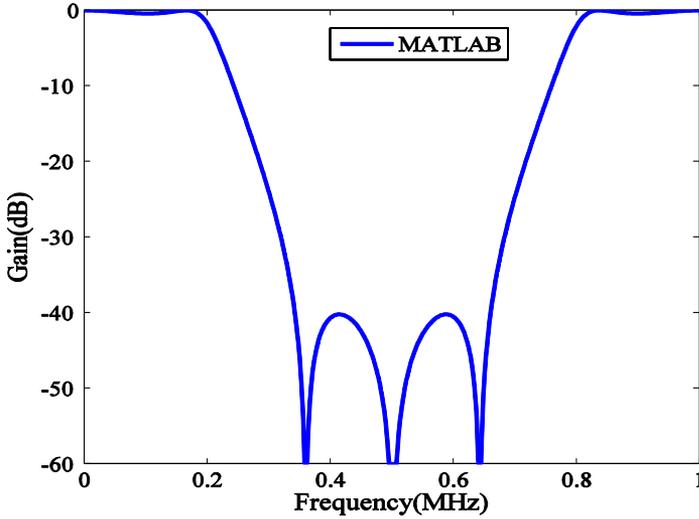


Fig. 16. The frequency response of the original filter

The sign digit can be skipped by changing the a_2 coefficient from negative to positive number. An alternative filter form of original filter with 7-bit coefficients $a_0 = 0.375$ (0.011000), $a_1 = 0.578125$ (0.100101) and $-a_2 = 0.328125$ (0.010101), which gives the same frequency response as the original filter is shown in Fig. 17.

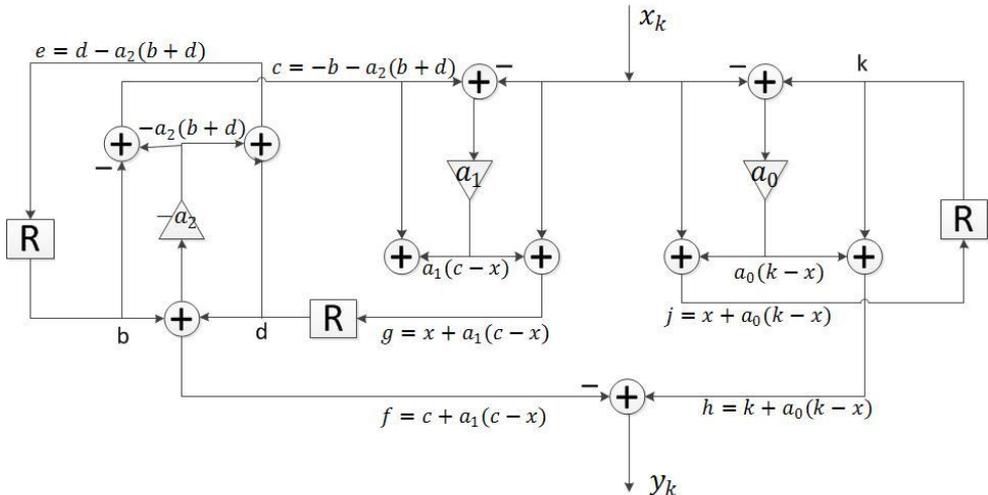


Fig. 17. The alternative filter extracted from the original filter

3.2 The Trivial Filter

With the trivial coefficients $a_0 = 0$, $a_1 = 0.5$ and $a_2 = 0$, the architecture is reduced to Fig. 18,

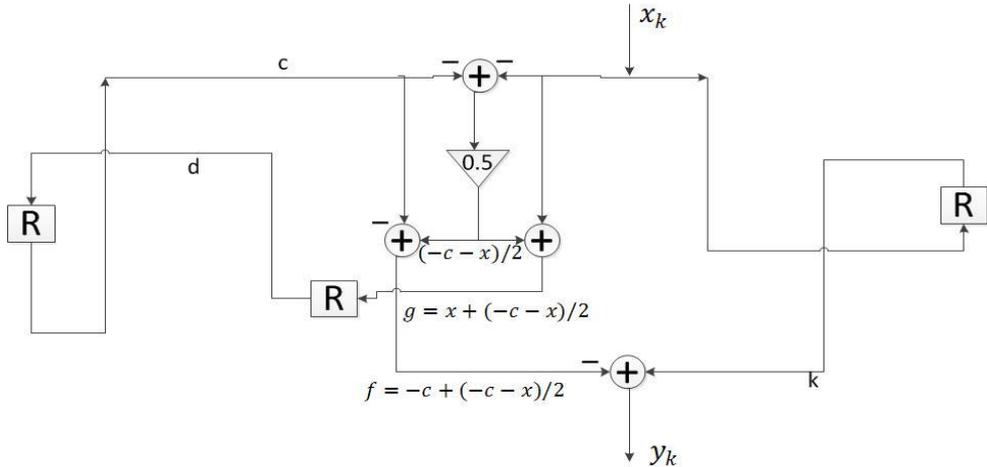


Fig. 18. The filter with trivial coefficients in the original filter

Fig. 18 can be redrawn properly as in Fig. 19, which is called the trivial filter. It has only one multiplier and four adders and three registers. Compared to the original filter in Fig. 17, the multipliers are reduced in Fig. 19.

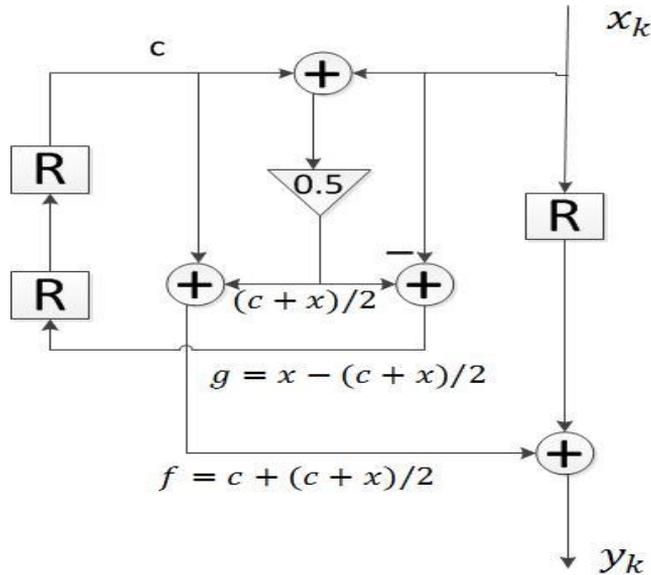


Fig. 19. The trivial filter

3.3 The Half-band Digital Filter

The trivial filter can be further simplified by recalculating the equations,

$$g(i) = x - (c + x) / 2 = 0.5c + 0.5x$$

$$\begin{aligned} f(i) &= c + (c + x) / 2 = 1.5c + 0.5x \\ &= 1.5c + 0.5c + (-0.5c + 0.5x) \\ &= 2c + g \end{aligned}$$

this gives the half-band digital (HBD) filter as shown in Fig. 20, is a third-order bi-reciprocal lattice wave digital filter. The transfer function of the HBD filter is

$$H_z = \frac{1 + 2z^{-1} + 2z^{-2} + z^{-3}}{2 + z^{-2}}$$

The advantage of the HBD filter is that the filter coefficients can be implemented by simple shifts, thereby reducing the area and the energy dissipation [2].

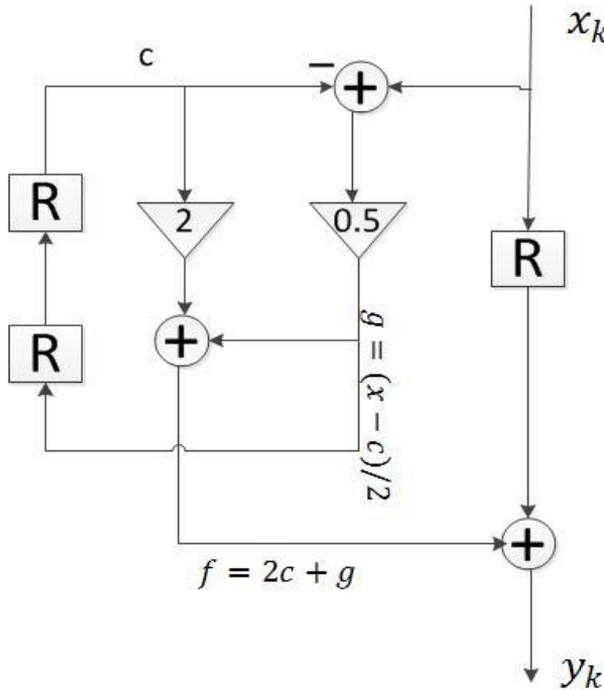


Fig. 20. The HBD filter

The frequency response of the HBD filter is shown in Fig. 21, which is obtained from MATLAB simulations.

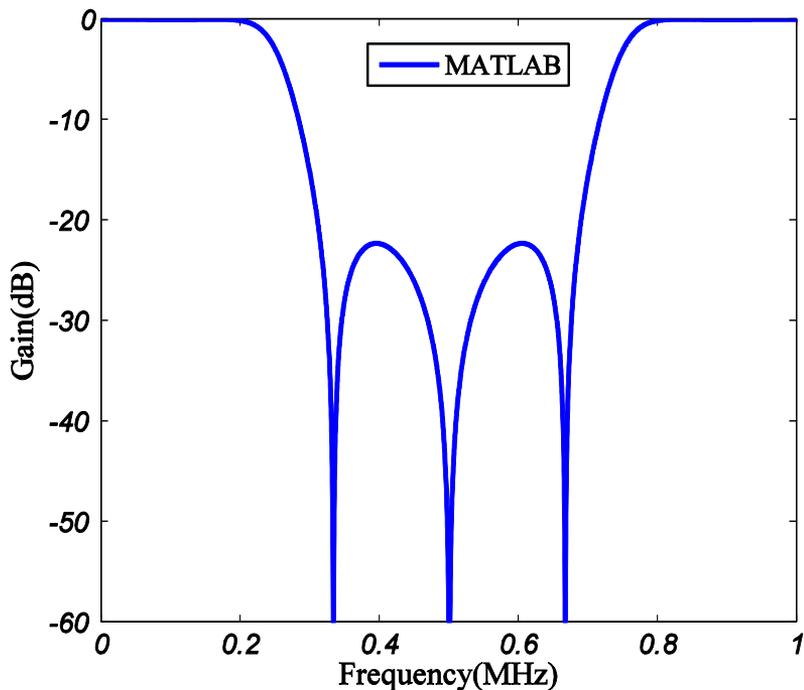


Fig. 21. The frequency response of the HBD filter

3.4 The Cascaded Filter

The order of the filter is increased by cascading the two half-band filters, which gives an 6th order filter, and is called as the cascaded filter. The architecture for the cascaded filter is shown in Fig. 22. By cascading, the filters have very sharp roll-off and very high signal-to-noise ratio (SNR).

To reduce the gain of the signal, the output of the first third-order filter is made half and given as input to the second third-order filter.

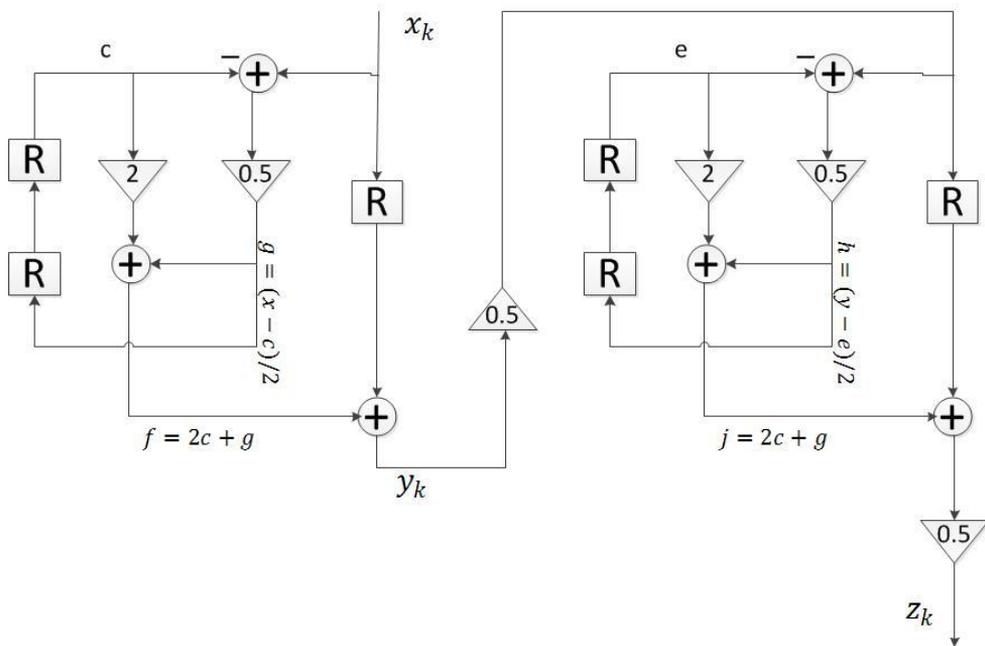


Fig. 22. The cascaded filter

The frequency response of the cascaded filter is shown in Fig. 23, which is obtained from MATLAB simulations.

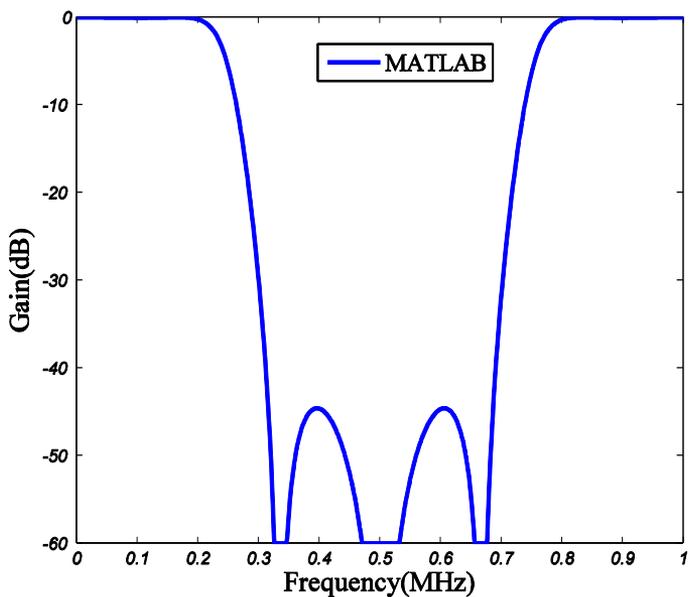


Fig. 23. The frequency response of the cascaded filter

3.5 Multiplication unit for the original filter

The original filter has three fixed coefficient multiplications. These filters are implemented in hardware by using adders. Converting the decimal number into binary number is shown in Appendix A.1.

3.5.1 The Coefficient a_0

The decimal value of the coefficient a_0 is 0.375, and the binary representation is 0011000. The hardware implementation of coefficient a_0 is shown in Fig. 24. The coefficient $a_0=0011000$ contain two “1: s”, so the multiplier can be realized by using one ripple carry adder. The three LSB ‘0: s’ are added because the coefficient contains three ‘0: s’ before the two ‘1: s’. The Most Significant Bit (MSB) is used to represent the sign extension of the number. The result of the multiplier is 18 bits. The a_0 multiplier contains 12 full adders (FA’s) and one half adders (HA).

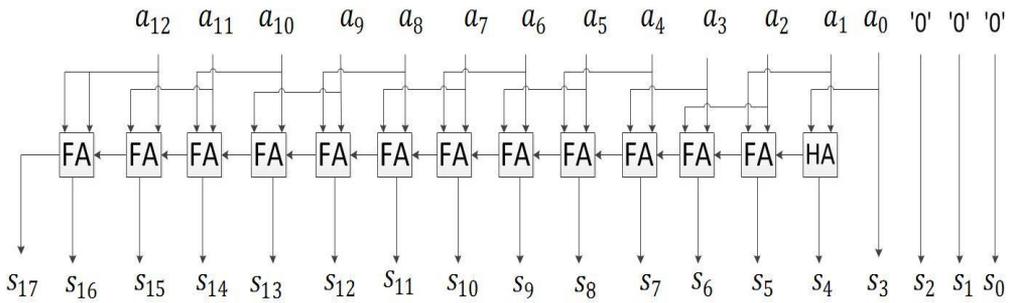


Fig. 24. Hardware Implementation of the coefficient a_0

3.5.2 The Coefficient a_1

The decimal value of the coefficient a_1 is 0.578125, and the binary representation is 0100101. The hardware implementation of coefficient a_1 is shown in Fig. 25. The coefficient $a_1=0100101$ contain three “1: s”, so the multiplier can be realized by using two ripple carry adders, where the upper adder is used for adding the input data together with the two-step left-shifted input data. After that the result of the upper adder to added together with the five-step left-shifted input data. The result of the multiplier is 19 bits. The a_1 multiplier has 24 full adders (FA) and 2 half adders (HA).

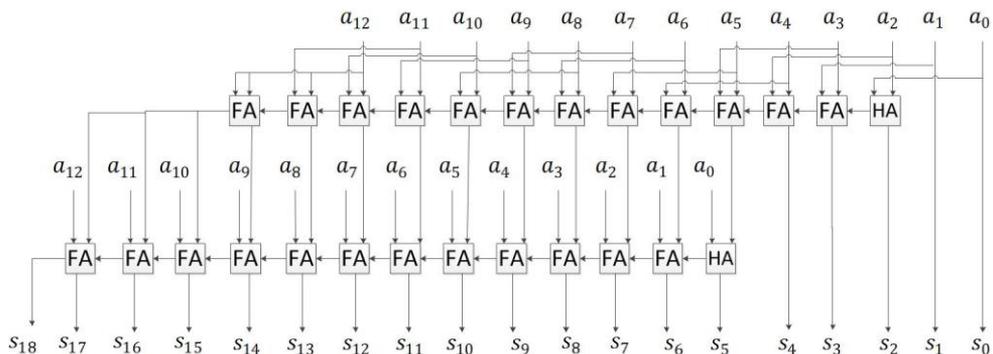


Fig. 25. Hardware Implementation of the coefficient a_1

3.5.3 The Coefficient $-a_2$

The decimal value of the coefficient a_2 is -0.328125 and in binary representation is 1101011 . By multiplying with ‘-’ on both sides, the negative value is changed to positive value. The decimal value of the coefficient $-a_2$ is 0.328125 , and the binary representation is 0010101 . The hardware implementation of coefficient $-a_2$ is shown in Fig. 26. The coefficient $a_2=0010101$ contain three “1: s”, so the multiplier can be realized by using two ripple carry adders, where the upper adder is used for adding the input data together with the two-step left-shifted input data. After that the result of the upper adder to added together with the four-step left-shifted input data. The result of the multiplier is 20 bits. The a_2 multiplier has 28 full adders (FA) and 2 half adders (HA).

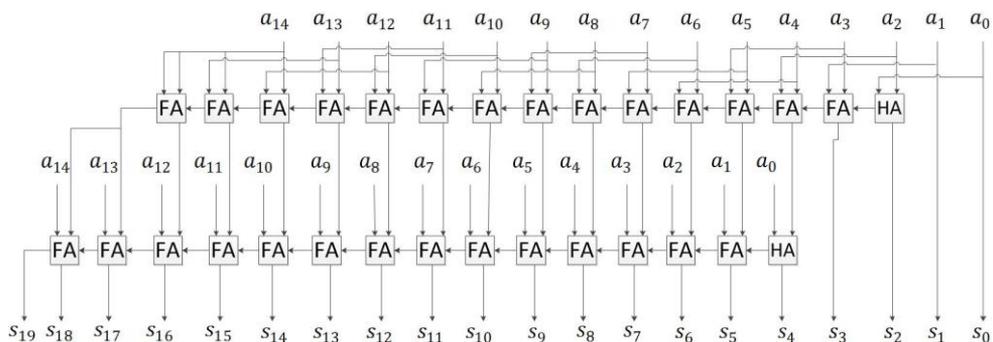


Fig. 26. Hardware Implementation of the coefficient $-a_2$

3.6 Truncation of the word length in the half-band filter

The half-band filter is tested by varying the number of fractional bits to check the precision in the filter output.

By truncating to 4 fractional bits, the frequency response of the half-band filter is shown in Fig. 27. Here, the total number of bits is 5, one integer bit and four fractional bits.

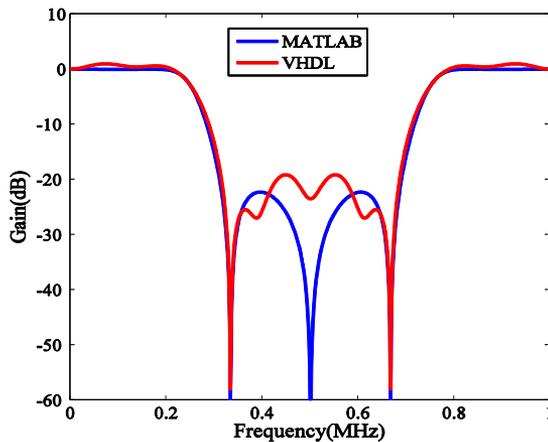


Fig. 27. The frequency response of HBF truncated to 4 fractional bits

By truncating to 7 fractional bits, the frequency response of the half-band filter is shown in Fig. 28. Here, the total number of bits is 8, one integer bit and seven fractional bits.

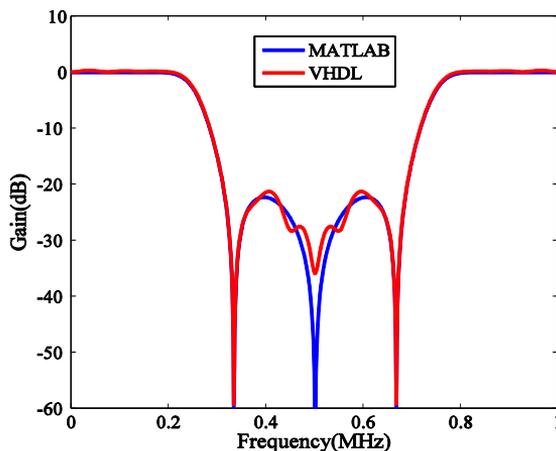


Fig. 28. The frequency response of HBF truncated to 7 fractional bits

By truncating to 9 fractional bits, the frequency response of the half-band filter is shown in Fig. 29. Here, the total number of bits is 10, one integer bit and nine fractional bits.

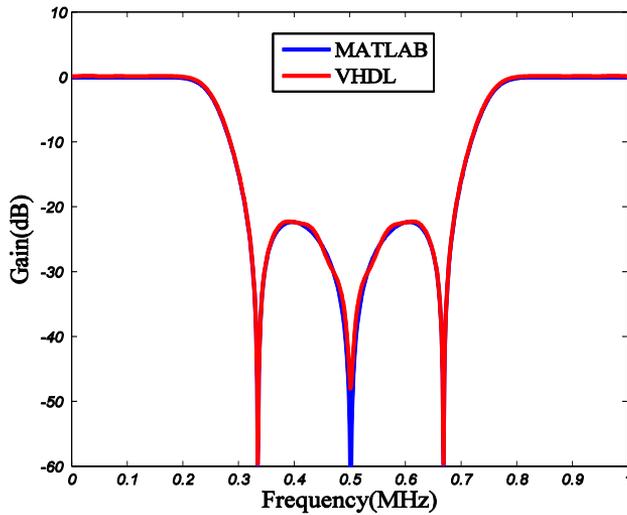


Fig. 29. The frequency response of HBF truncated to 9 fractional bits

By truncating to 11 fractional bits, the frequency response of the half-band filter is shown in Fig. 30. Here, the total number of bits is 12, one integer bit and eleven fractional bits. With 12 bits the full precision is obtained.

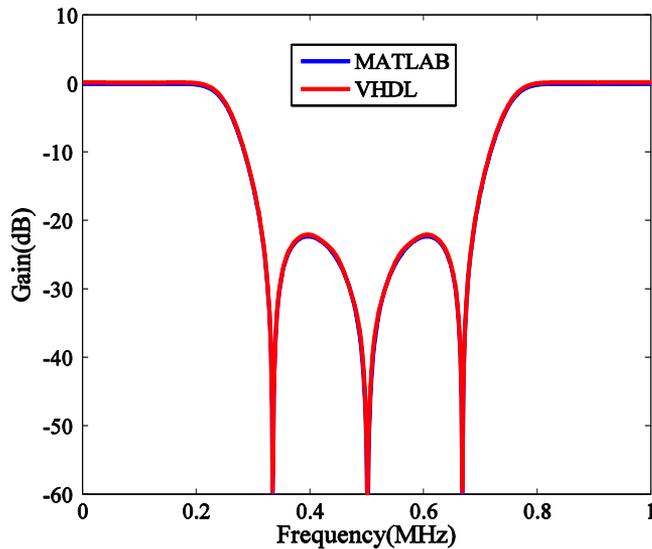


Fig. 30. The frequency response of HBF truncated to 11 fractional bits

3.7 Simulation Results

Both the filters are implemented in hardware using VHSIC hardware description language (VHDL) and the results are compared with the MATLAB results. The input width of both the filters is 12 bits and the output width is 15 bits. Fig. 31 & 32, shows the comparison plots of original filter and the cascaded filter respectively. The variation in the plot is occurred due to stop band attenuation. The data observed from the plots is presented in Table II.

TABLE II. FILTER DATA

Filter	Number of taps	Sampling Freq. (MHz)	Pass band		Stop band	
			Edge (MHz)	Ripple (dB)	Edge (MHz)	Ripple (dB)
Original	3	100	0.210	0.2	0.338	39
Cascaded	6	100	0.238	0.1	0.317	43

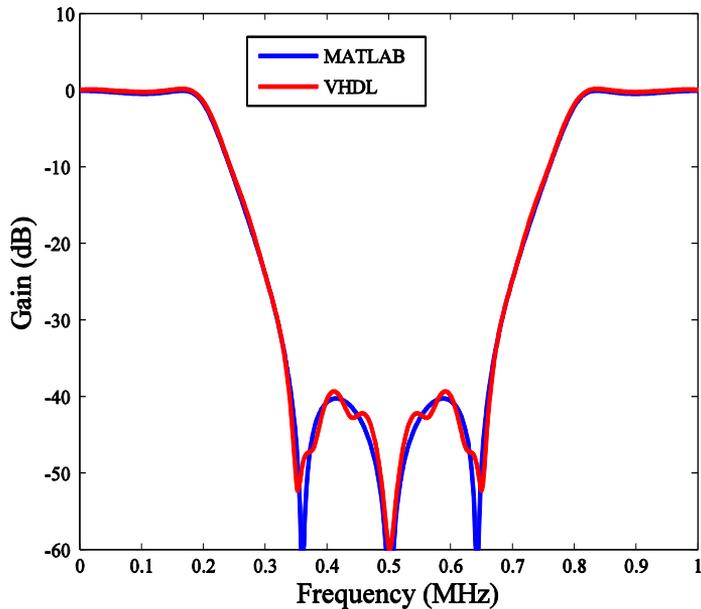


Fig. 31. MATLAB reference model output vs. post-simulation output of the original filter

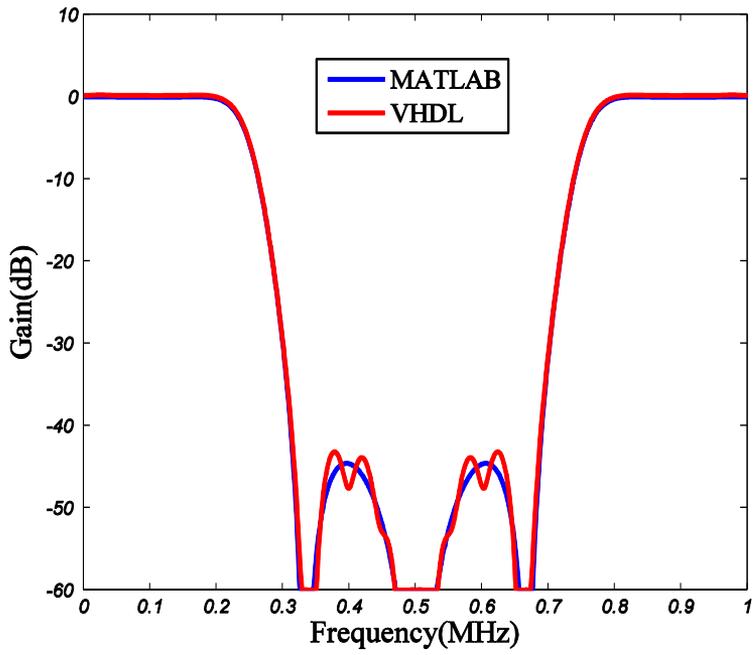


Fig. 32. MATLAB reference model output vs. post-simulation output of the cascaded filter

CHAPTER 4

4 Synthesis Results

Synthesis is carried out using Synopsys Design Compiler. The synthesis process converts Register Transfer Level (RTL) descriptions to the gate-level netlists. The original filter and the cascaded filter are synthesized for two design constraints.

1. Minimum area
2. Maximum speed

The synthesis script is presented in the Appendix A.2.

4.1 Synthesis results of the original filter

4.1.1 Area Constraint

To achieve minimum area, the area constraint is set to $0 \mu\text{m}^2$, which is practically impossible to achieve but the tool will fit the design in the possible minimum area, and specify high clock period, which is $T_{clk} = 6.4\text{ns}$ in this case, such that slack met is positive.

TABLE III. AREA CONSTRAINT OF THE ORIGINAL FILTER

Cell	Minimum Area [μm^2]	Maximum speed Area [μm^2]
Number of ports	29	29
Number of nets	29	29
Number of cells	1	1
Number of references	1	1
Combinational area	2356.63	4349.27
Non-Combinational area	415.99	421.19
Total cell area	2772.63	4770.47

After synthesized for minimum area and maximum speed constraints, by comparing the area for both the constraints, it is clear that the synthesis tool optimizes for minimum area. The area is 58% minimum compared to the area of maximum speed. The comparison of the minimum area and maximum speed for the original filter is shown in Table III.

4.1.2 Speed constraint

The maximum speed of the design is achieved by specifying the timing constraints. While the design is constraint for maximum speed, the area constraints are not specified and the slack met is maintained to be positive. The timing information for maximum speed and minimum area are presented in Table IV. The time slack is the difference between the data required time and the data arrival time. The clock period applied is 3ns and the minimum required time for the original filter is 2.8ns.

TABLE IV. THE CRITICAL PATH TIMING INFORMATION OF THE ORIGINAL FILTER

Original Filter		
Critical path	Minimum Area (ns)	Maximum Speed (ns)
Register 3	0.17	0.16
Adder 7	2.50	0.48
Coefficient a2	3.51	1.24
Adder 9	4.20	1.73
Adder 1	4.68	1.90
Coefficient a1	5.19	2.59
Adder 3	6.26	2.84
Register 1	6.44	2.89
Data required time	7.84	2.89
Data arrival time	-6.44	-2.89
Slack (MET)	1.39	0.00

The maximum clock frequency for the original filter is

$$f_{max} = \frac{1}{T_{clk}} = 346\text{MHz.}$$

The clock frequency for minimum area is $f_{max} = \frac{1}{T_{clk}} = 155\text{MHz.}$

The clock frequency for the maximum speed of the original filter is 45% more than the clock frequency of the minimum area.

4.1.3 Critical Path

The critical path is the longest path chosen or minimum time required to produce the output of a design. The critical path information for the original filter is shown in Table IV and drawn in Fig.33.

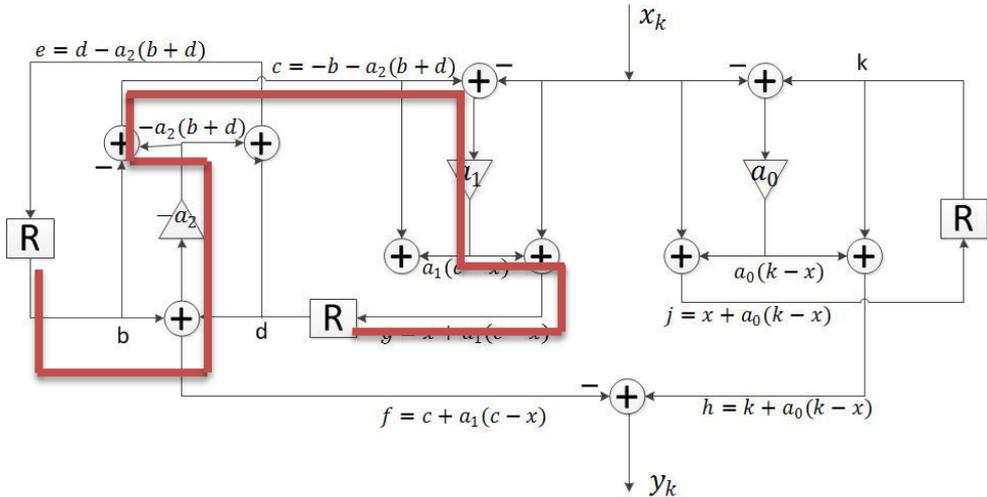


Fig. 33. critical path of the original filter

4.2 Synthesis results of the cascaded filter

4.2.1 Area Constraint

To achieve minimum area, the area constraint is set to $0 \mu\text{m}^2$, which is practically impossible to achieve but the tool will fit the design in the possible minimum area, and specify high clock period, which is $T_{clk} = 3.5\text{ns}$ in this case, such that slack met is positive.

TABLE V. THE AREA CONSTRAINT OF THE CASCADED FILTER

Cell	Minimum Area [μm^2]	Maximum speed Area [μm^2]
Number of ports	29	29
Number of nets	29	29
Number of cells	1	1
Number of references	1	1
Combinational area	920.39	1387.35
Non-Combinational area	811.19	811.19
Total cell area	1731.60	2198.56

After synthesized for minimum area and maximum speed constraints, by comparing the area for both the constraints, it is clear that the synthesis tool optimizes for minimum area. The area is 78% minimum compared to the area of maximum speed. The comparison of the minimum area and maximum speed for the cascaded filter is shown in Table V.

4.2.2 Speed constraint

The maximum speed of the design is achieved by specifying the timing constraints. While the design is constraint for maximum speed, the area constraints are not specified and the slack met is maintained to be positive. The timing information is presented in Table VI. The clock period applied is 2ns and the minimum required time for the cascaded filter is 1.86ns.

The maximum clock frequency for the cascaded filter is

$$f_{max} = \frac{1}{T_{clk}} = 537\text{MHz.}$$

The clock frequency for minimum area is $f_{max} = \frac{1}{T_{clk}} = 281\text{MHz.}$

The clock frequency for the maximum speed of the cascaded filter is 53% more than the clock frequency of the minimum area.

CHAPTER 5

5 Place and Route Results

Place and Route is done using the SOC Encounter tool. This tool takes the input from the net list generated from the synthesis tool and its associated timing information in the form of an SDC file. The structure of both the filters is same. Floor planning is the primary importance of the place and route, which is done by specifying the number of IO (Input and Output) pads and their placements around the core. The inputs are placed on the top and left side, and the outputs are placed on the bottom and right side for the designs. The metal layers 5 and 6 are used for core power rings and power strips, as they are wider, they cause very little voltage drop across the distribution. The physical layout from the tool for the original filter and the cascaded filter are presented in Fig. 35 and 36 respectively. As the design is pad constraint the area required is set by the pads. Since these modules are included with other modules the core area is then reduced. The script used for place and route is added in Appendix A.3. The gate count and the wire length observed from the tool are presented in Table VII. The gate count and the wire length of the original filter are more than the cascaded filter, as the number of adders is more in the original filter.

TABLE VII. THE GATE COUNT AND WIRE LENGTH FROM PNR

Design	Gate count	Wire length [μm]
Original Filter	3324	14572.82
Cascaded Filter	1131	8732.59

The Place and Route area report is presented in the Table VIII. The standard cells are placed in the core for the adders and multipliers in the design. The pad cells are placed for the number of input/output pads in the design. The chip area is the whole area required for the chip to manufacture.

TABLE VIII. THE AREA INFORMATION FROM PNR

Placement	Original Filter Area [μm^2]	Cascaded Filter Area [μm^2]
Standard cells	14760.20	9449.44
Pad cells	204982.40	193715.20
Core	14762.47	9459.99
Chip	324330.37	296614.77

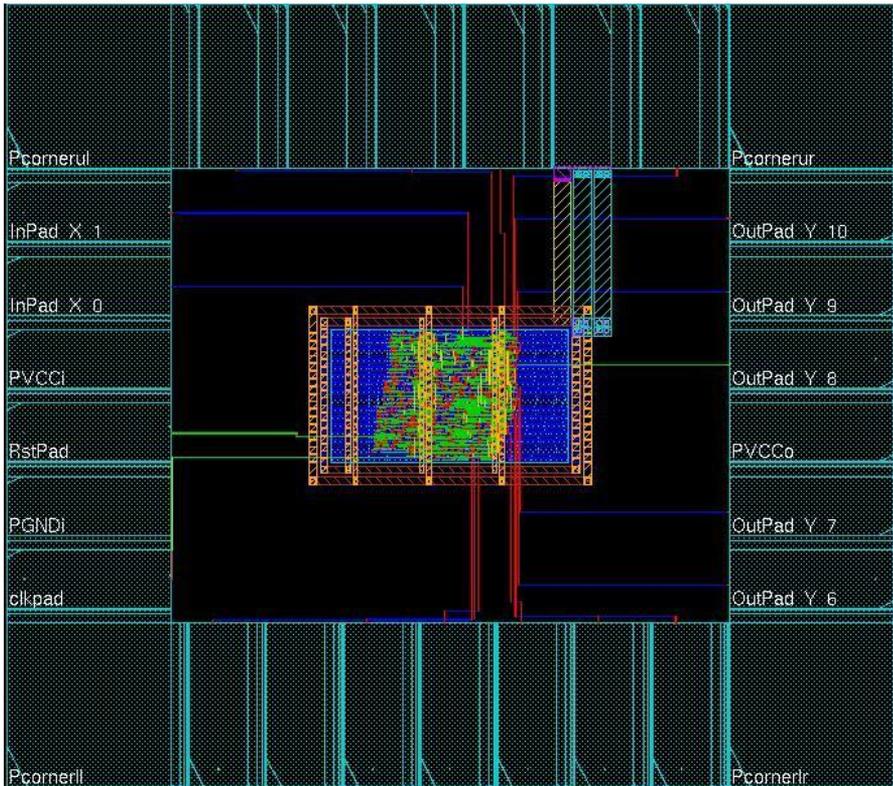


Fig. 35. The physical layout of the original filter

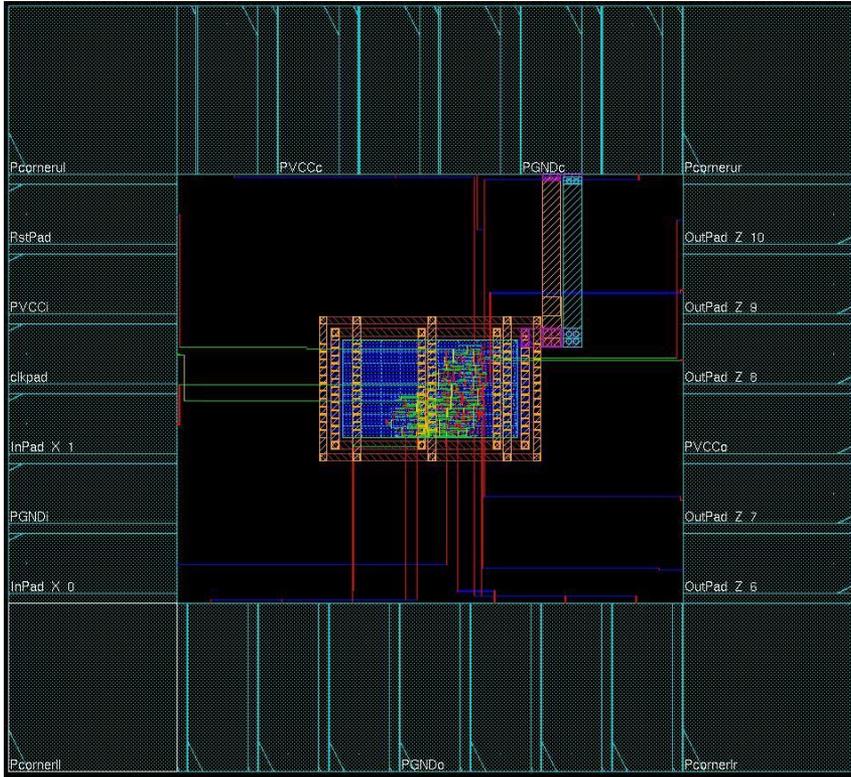


Fig. 36. The physical layout of the cascaded filter

CHAPTER 6

6 Power Analysis Results

Power analysis is carried out by using Synopsys Prime Time PX tool. This chapter presents the power simulations and energy simulations of both the original filter and the cascaded filter for three different inputs, the impulse response, a square wave input, and the random input. Power simulations are done by generating the netlist from the synthesis tool and the VCD file from the post-synthesis simulation. The VCD file contains the switching activity of the data. Switching activity is based on the input changes in the data from $0 \rightarrow 1$ or $1 \rightarrow 0$. The script for the power analysis is presented in Appendix A.4.

The power supply for the designs is 1.20V. The designs are synthesized to 65 nm CMOS technology. The designs are simulated for different frequencies in the range 0.01 KHz to 100MHz. The total power P_{Total} calculated includes the combinational power and the register power. The power consumed by the clock network was not included in the total power because clock network was surrounded outside the filter. The energy dissipation was calculated by multiplying the propagation delay with the total power. Frequency is plotted on the horizontal axis and the total power using a logarithmic scale is plotted on vertical axis. The power simulation tables and the comparison plots for both the original filter and the cascaded filter for different inputs are shown in section 6.1, 6.2 and 6.3. The energy dissipation plots are shown in section 6.4.

6.1 Power consumption for impulse input

Power simulations for the impulse input of the original filter are shown in Table IX and for the cascaded filter are shown in Table X. The comparison of the total power in log scale for the impulse response is plotted in Fig. 37.

TABLE IX. POWER SIMULATIONS FOR THE ORIGINAL FILTER (IMPLUSE RESPONSE)

<i>Freq (MHz)</i>	<i>P_{clk} (W)</i>	<i>P_{reg} (nW)</i>	<i>P_{comb} (nW)</i>	<i>LOG (P_{Total} (nW))</i>	<i>LOG (Energy(nJ))</i>
0.00001	3.37E-12	1.65	7.91	0.980	0.452
0.0001	3.37E-11	1.67	8.00	0.985	0.454
0.001	3.48E-10	1.59	7.19	0.943	0.435
0.01	3.54E-09	3.58	18.4	1.342	0.588
0.1	3.53E-08	7.03	37.6	1.649	0.678
1	3.48E-07	7.04	37.5	1.649	0.678
10	3.48E-06	7.04	37.5	1.649	0.678
100	3.48E-05	7.04	37.5	1.649	0.678

TABLE X. POWER SIMULATIONS FOR THE CASCADED FILTER (IMPLUSE RESPONSE)

<i>Freq (MHz)</i>	<i>P_{clk} (W)</i>	<i>P_{reg} (nW)</i>	<i>P_{comb} (nW)</i>	<i>LOG (P_{Total} (nW))</i>	<i>LOG (Energy (nJ))</i>
0.00001	5.42E-12	2.81	2.83	0.751	0.145
0.0001	5.42E-11	2.82	2.85	0.753	0.146
0.001	5.92E-10	2.69	2.88	0.745	0.141
0.01	5.96E-09	4.61	4.51	0.959	0.251
0.1	5.98E-08	12.8	14.2	1.431	0.425
1	5.92E-07	12.8	14.2	1.430	0.425
10	5.91E-06	12.8	14.2	1.430	0.425
100	5.91E-05	12.8	14.2	1.430	0.425

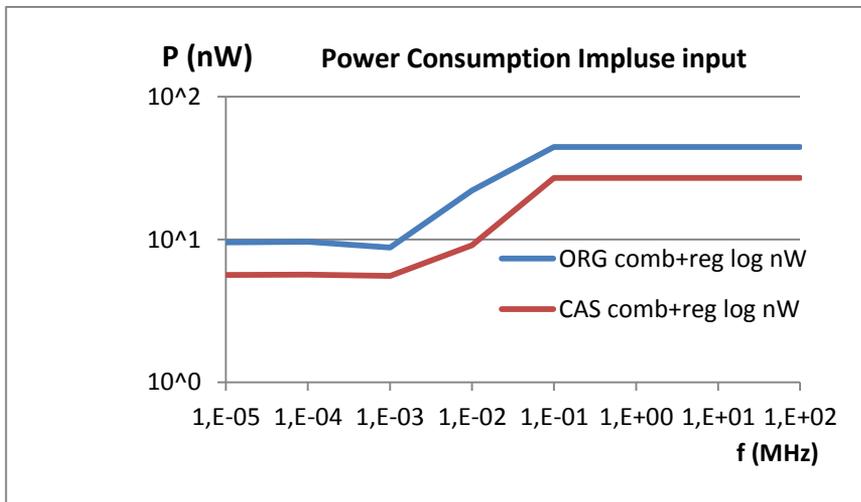


Fig. 37. Comparison of the total power for the impulse input

6.2 Power consumption for square wave input

Power simulations for the square wave input of the original filter are shown in Table XI and for the cascaded filter in Table XII. The comparison of the total power in log scale for the square wave input is plotted in Fig. 38.

TABLE XI. POWER SIMULATIONS FOR ORIGINAL FILTER (SQUARE WAVE INPUT)

<i>Freq</i> <i>(MHz)</i>	<i>P_{clk}</i> <i>(W)</i>	<i>P_{reg}</i> <i>(nW)</i>	<i>P_{comb}</i> <i>(nW)</i>	<i>LOG</i> <i>(P_{Total} (nW))</i>	<i>LOG</i> <i>(Energy(nJ))</i>
0.00001	1.74E-12	1.82	7.19	0.954	0.440
0.0001	1.74E-11	3.64	7.17	1.033	0.475
0.001	1.74E-10	1.82	7.19	0.954	0.440
0.01	1.74E-09	1.83	7.37	0.963	0.444
0.1	1.74E-08	4.01	44.2	1.683	0.687
1	1.74E-07	8.67	123.0	2.119	0.787
10	1.74E-06	28.7	463.0	2.692	0.890
100	1.74E-05	978.0	16500.0	4.243	1.088

TABLE XII. POWER SIMULATIONS FOR CASCADED FILTER (SQUARE WAVE INPUT)

<i>Freq</i> <i>(MHz)</i>	<i>P_{clk}</i> <i>(W)</i>	<i>P_{reg}</i> <i>(nW)</i>	<i>P_{comb}</i> <i>(nW)</i>	<i>LOG</i> <i>(P_{Total} (nW))</i>	<i>LOG</i> <i>(Energy (nJ))</i>
0.00001	3.39E-12	3.55	2.69	0.795	0.170
0.0001	3.39E-11	3.55	2.69	0.795	0.170
0.001	3.39E-10	3.55	2.69	0.795	0.170
0.01	3.39E-09	3.56	2.70	0.796	0.170
0.1	3.39E-08	3.61	2.77	0.804	0.175
1	3.39E-07	4.18	3.56	0.888	0.218
10	3.39E-06	17.4	21.7	1.592	0.471
100	3.39E-05	113.0	153.0	2.425	0.654

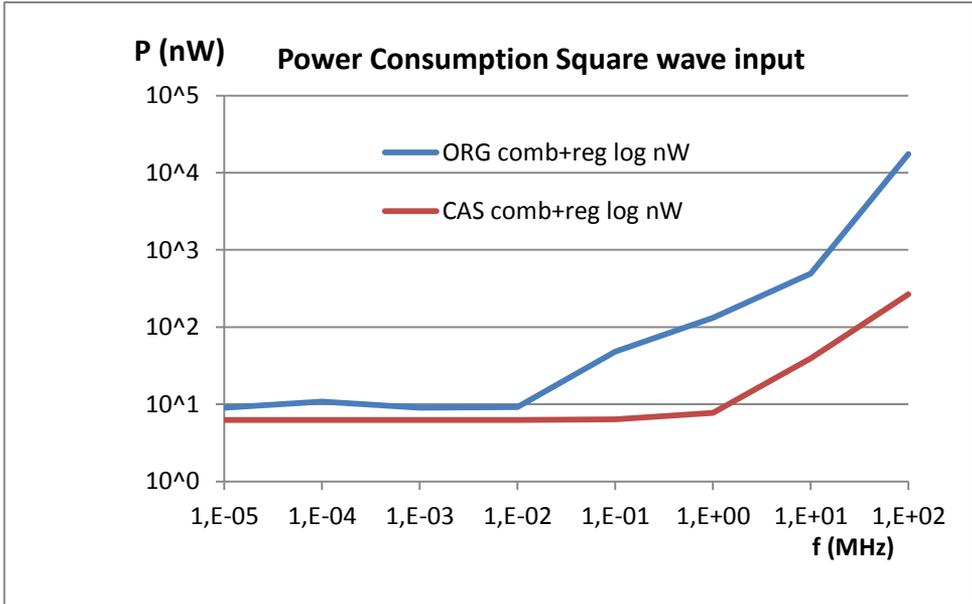


Fig. 38. Comparison of the total power for the square wave input

6.3 Power consumption for random input

Power simulations for the random input of the original filter are shown in Table XIII and for the cascaded filter are shown in Table XIV. The comparison of the total power in log scale for the random input is plotted in Fig. 39.

TABLE XIII. POWER SIMULATIONS FOR ORIGINAL FILTER (RANDOM INPUT)

<i>Freq</i> (MHz)	<i>P_{clk}</i> (W)	<i>P_{reg}</i> (nW)	<i>P_{comb}</i> (nW)	<i>LOG</i> (<i>P_{Total}</i> (nW))	<i>LOG</i> (<i>Energy</i> (nJ))
0.00001	1.74E-12	1.87	21.0	1.358	0.593
0.0001	1.74E-11	1.87	21.0	1.358	0.593
0.001	1.74E-10	1.87	21.0	1.358	0.594
0.01	1.74E-09	1.89	21.2	1.362	0.595
0.1	1.74E-08	1.88	21.1	1.361	0.594
1	1.74E-07	2.92	33.8	1.565	0.655
10	1.74E-06	4.50	53.2	1.761	0.706
100	1.74E-05	126.0	1540.0	3.221	0.968

TABLE XIV. POWER SIMULATIONS FOR ORIGINAL FILTER (RANDOM INPUT)

<i>Freq</i> (MHz)	<i>P_{clk}</i> (W)	<i>P_{reg}</i> (nW)	<i>P_{comb}</i> (nW)	<i>LOG</i> (<i>P_{Total}</i> (nW))	<i>LOG</i> (<i>Energy</i> (nJ))
0.00001	3.39E-12	3.53	5.66	0.963	0.253
0.0001	3.39E-11	3.53	5.66	0.963	0.253
0.001	3.39E-10	3.53	5.66	0.963	0.253
0.01	3.39E-09	3.54	5.68	0.964	0.253
0.1	3.39E-08	3.59	5.79	0.972	0.257
1	3.39E-07	4.80	8.25	1.115	0.316
10	3.39E-06	7.87	14.5	1.350	0.399
100	3.39E-05	57.3	116.0	2.237	0.619

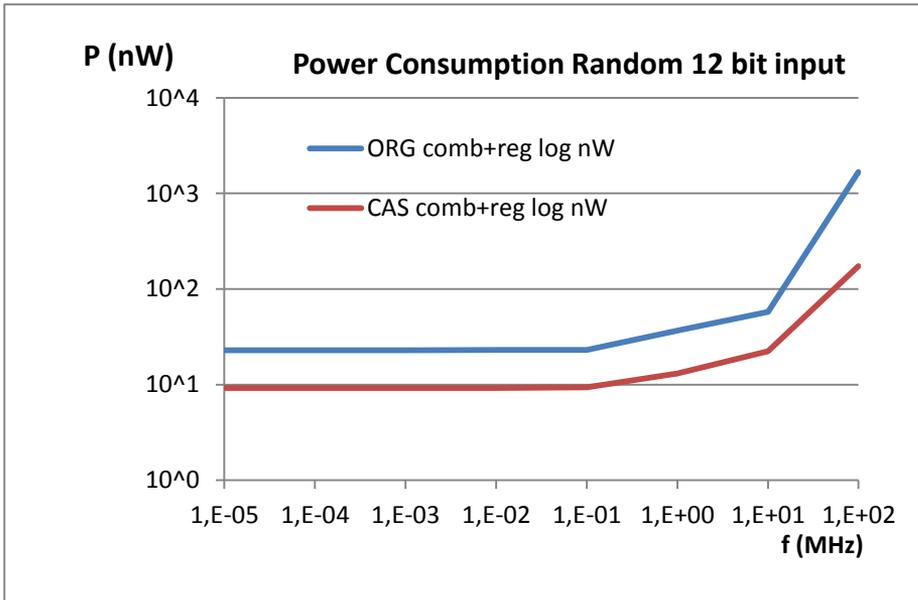


Fig. 39. Comparison of the total power for the random input

From the Fig. 37, 38, 39, it can be observed that the power consumed by the original filter is more compared to the cascaded filter. Hence, it is better to use the cascaded filter in the devices where low power and high speed is necessary.

6.4 Energy dissipation results

Energy dissipation is calculated using the formulae below:

$$E = t_p * P$$

where, t_p = propagation delay
P = power consumption

The energy dissipation is measured in Joules (J) or Watt hours (Wh).

In order to evaluate the energy dissipation the designs are synthesized in 65 nm CMOS technology. The calculated energy dissipation values for the impulse response, the square wave input and the random input are shown in Table IX to XIV. The comparison of energy dissipation observed in the original filter and the cascaded filter for the impulse input, the square wave input and the random input are plotted in Fig. 40, 41, 42 respectively. Frequency is plotted on the horizontal axis and the energy dissipated using a logarithmic scale is plotted on vertical axis.

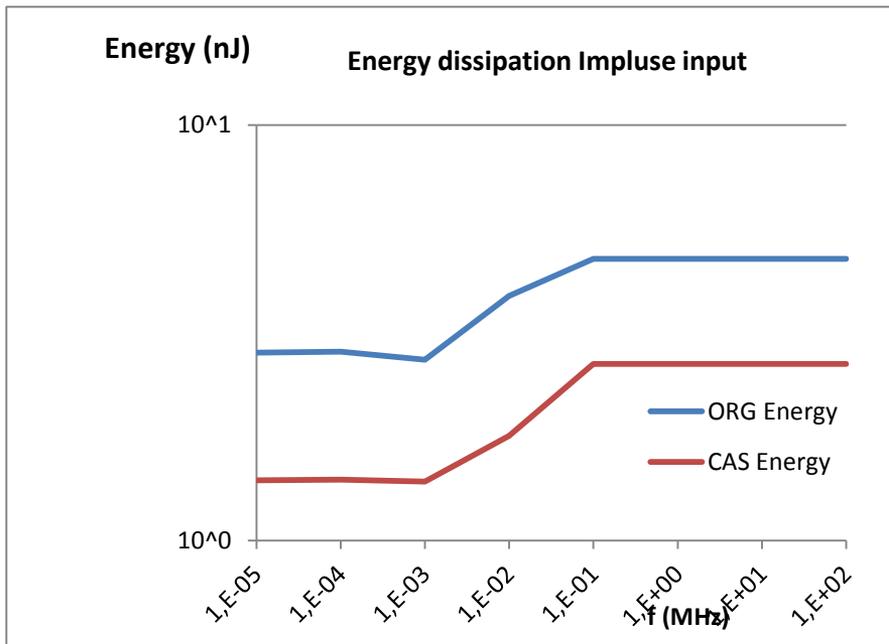


Fig. 40. The energy consumption for the impulse input

From the Fig. 40, it can be observed that the energy dissipated by the original filter is more compared to the cascaded filter. The filters have leakage power until 1 KHz and then the dynamic power is increased linearly until 100 KHz. After 100 KHz the filters have stopped working for the impulse input.

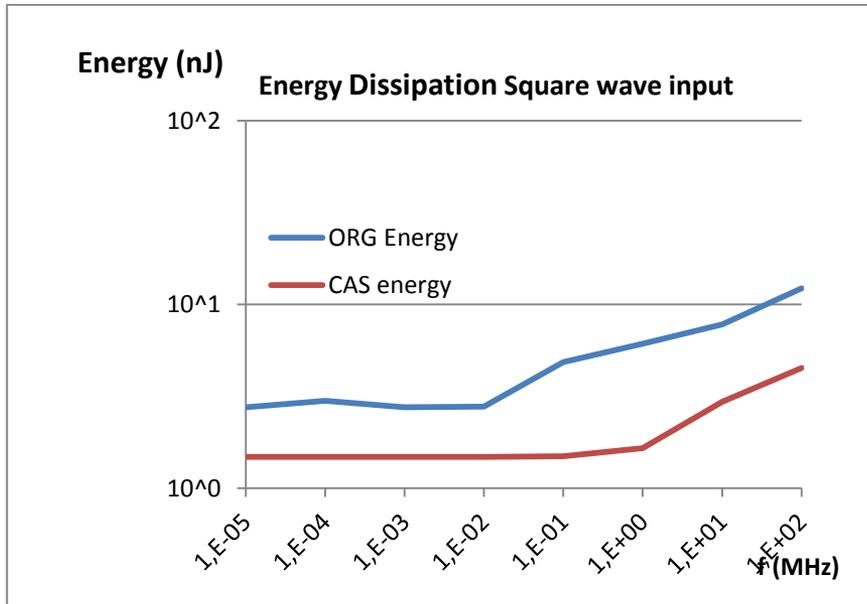


Fig. 41. The energy consumption for the square wave input

From the Fig. 41, it can be observed that the energy dissipated by the original filter is more compared to the cascaded filter. The original filter has leakage power until 10 KHz and the cascaded filter has leakage power until 1 MHz. The cascaded filter has more leakage than the original filter. The dynamic power is linearly increased for the cascaded filter but is almost linear for the original filter.

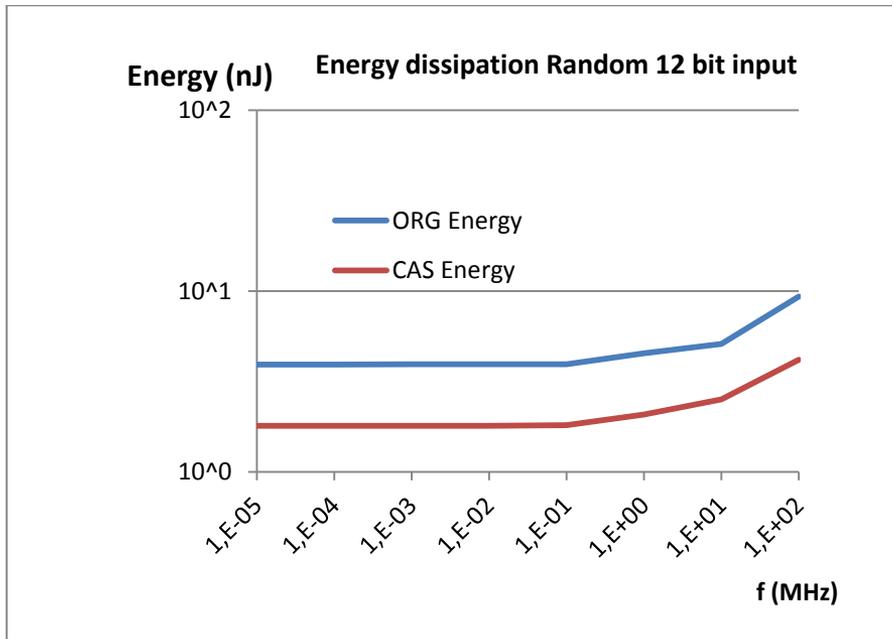


Fig. 42. The energy consumption for the random input

From the Fig. 42, it can be observed that the energy dissipated by the original filter is more compared to the cascaded filter. Both the filters have leakage power until 0.1 MHz and then the dynamic power is linearly increased.

The energy dissipation is more in the original filter than the cascaded filter, because the number of cells in the original filter is more than the cascaded filter. Considering all the performance characteristics, the cascaded filter is better choice for low energy dissipation and high speed devices.

CHAPTER 7

7 Analysis of the results

The filters are synthesized using STM 65 nm CMOS technology. Both filters are verified using the test bench, which allows changing the input data and the clock frequency easily. The designs are tested for different clock frequencies and for three different input signals, which are the impulse response, random input, and the square wave input. The inputs are generated from MATLAB. The input word length is 12 bits and the output word length is 15 bits for the original filter and the cascaded filter, the block diagrams for the filters are presented in chapter 3.

Initially the filters are tested with the impulse response, to check if the filters are working properly. The square wave input is used to check the functionality of the filter with immediate change in the input from a minimum value to the maximum value. The random input is used to check how the filter acts for some random input data.

The original filter contains 10 adders, 3 registers, 3 multipliers whereas the cascaded filter contains 6 adders, 6 registers and the multipliers are implemented using shifters. The 3 multipliers in the original filter are implemented as ripple carry adders.

The area of the cascaded filter is 62% less compared to the original filter. In the original filter the total of 13 adders consume more area. Also, the cascaded filter is 64% faster than the original filter. The power consumption for the cascaded filter is 69% less than the original filter. The energy dissipation in the cascaded filter is 63% low than the original filter. Hence, the best performance was found with the cascaded filter, which is built out of two half band third order filters. The switching activity for the filters is very low.

	Original Filter	Cascaded Filter	Percentage (%)
Area [μm^2]	2772	1731	62
t_p [ns]	2.89	1.86	64
Power [nW]	3.221	2.237	69
Energy [nJ]	0.968	0.619	63
α	0.000096	0.000058	61

CHAPTER 8

8 Conclusions

- The minimum required area, maximum speed, power consumption and the energy dissipation of the original filter and the cascaded filter has been estimated.
- It is proved that the cascaded filter is more efficient than the original filter.

CHAPTER 9

9 Future Work

- The filters can be designed in Cadence and simulate in Spectre.
- Can use bit serial or digit serial arithmetic for the multipliers in the original filter and reduce the power consumption.
- Input of a pipeline stage in the multipliers.
- Can use the simplified filter for decimation or interpolation.
- Can extend the power analysis for LPSVT and LPLVT cells.

References

- [1] V. Kripasagar, “Wave Digital Filtering using the MSP430,” *Texas Instruments*, September 2006.
- [2] S. M. Yasser Sherazi, Joachim N. Rodrigues, Omer C. Akgun, H. Sjöland, and P. Nilsson, “Ultra Low Energy Design Exploration of Digital Decimation Filters in 65 nm Dual- V_T CMOS in the Sub- V_T Domain,” *Department of Electrical and Information Technology*, Lund University, Sweden, February 2012.
- [3] S. White, *Digital Signal Processing*, Delmar Cengage Learning Publication, ISBN 0766815315, March 2000.
- [4] H. Johansson, L. Wanhammar, “Design of linear-phase Lattice Wave Digital Filters”, *Department of Electrical Engineering*, Linköping University, Sweden, March 1997.
- [5] P. Åstrom, P. Nilsson, and M. Torkelson, “Power Reduction in Custom CMOS Digital Filter Structures,” *Department of Applied Electronics*, Lund University, Sweden, March 1998.
- [6] By the mathworks website www.mathworks.se
- [7] Vinay K. Ingle, John G. Proakis, *Digital signal processing using MATLAB*, Northeastern University, ISBN-10: 1-111-42737-2, april 2007.
- [8] H. Samueli, “A Low-Complexity Multiplierless Half-Band Recursive Digital Filter Design,” *IEEE transactions on Acoustics, Speech, and Signal processing*, VOL 37, NO.3, March 1989.
- [9] C. Piguet, *Low-power Electronics Design*, CRC Press Publication, E-Book ISBN 978-1-4200-3955-9, November 2004.
- [10] A. Tisserand, “Introduction to Power Consumption in Digital Integrated Circuits,” CNRS, IRISA laboratory, CAIRN research team.
- [11] M.Morris Mano, *Digital Design*, California State University, Los Angeles, Prentice Hall Publication, ISBN 0-13-062121-8, year 2002.

List of Figures

Fig. 1. DSP Receiver system [2].....	9
Fig. 2. (a) Adaptor symbol (b) Internal structure of adaptor [5].....	11
Fig. 3. Filter design using adaptors.....	11
Fig. 4. An up sampling element.....	12
Fig. 5. Demonstration of Interpolation by a factor of 2.....	13
Fig. 6. A down sampling element.....	13
Fig. 7. Demonstration of Decimation by a factor of 2.....	14
Fig. 8. Frequency response of LPF [8].....	14
Fig. 9. A CMOS Circuit [10].....	18
Fig. 10. Transistor structure [10].....	18
Fig. 11. Routing [10].....	18
Fig. 12. Parasitic capacitance [10].....	19
Fig. 13. I/O pad [10].....	19
Fig. 14. Short circuit current in CMOS inverter [10].....	20
Fig. 15. The third order original filter.....	21
Fig. 16. The frequency response of the original filter.....	22
Fig. 17. The alternative filter extracted from the original filter.....	22
Fig. 18. The filter with trivial coefficients in the original filter.....	23
Fig. 19. The trivial filter.....	23
Fig. 20. The HBD filter.....	24
Fig. 21. The frequency response of the HBD filter.....	25
Fig. 22. The cascaded filter.....	26
Fig. 23. The frequency response of the cascaded filter.....	26
Fig. 24. Hardware Implementation of the coefficient a_0	27
Fig. 25. Hardware Implementation of the coefficient a_1	28
Fig. 26. Hardware Implementation of the coefficient $-a_2$	28
Fig. 27. The frequency response of HBF truncated to 4 fractional bits.....	29
Fig. 28. The frequency response of HBF truncated to 7 fractional bits.....	29
Fig. 29. The frequency response of HBF truncated to 9 fractional bits.....	30
Fig. 30. The frequency response of HBF truncated to 11 fractional bits.....	30
Fig. 31. MATLAB reference model output vs. post-simulation output of the original filter.....	31
Fig. 32. MATLAB reference model output vs. post-simulation output of the cascaded filter.....	32
Fig. 33. critical path of the original filter.....	35
Fig. 34. critical path of the cascaded filter.....	37
Fig. 35. The physical layout of the original filter.....	40
Fig. 36. The physical layout of the cascaded filter.....	41
Fig. 37. Comparison of the total power for the impulse input.....	44

Fig. 38. Comparison of the total power for the square wave input.....	46
Fig. 39. Comparison of the total power for the random input	47
Fig. 40. The energy consumption for the impulse input	48
Fig. 41. The energy consumption for the square wave input.....	49
Fig. 42. The energy consumption for the random input.....	50

List of Tables

TABLE I. MULTIPLICATION COEFFICIENTS OF THE ORIGINAL AND THE CASCADED FILTER	21
TABLE II. FILTER DATA	31
TABLE III. AREA CONSTRAINT OF THE ORIGINAL FILTER.....	33
TABLE IV. THE CRITICAL PATH TIMING INFORMATION OF THE ORIGINAL FILTER.....	34
TABLE V. THE AREA CONSTRAINT OF THE CASCADED FILTER	36
TABLE VI. THE CRITICAL PATH TIMING INFORMATION OF THE CASCADED FILTER.....	37
TABLE VII. THE GATE COUNT AND WIRE LENGTH FROM PNR..	39
TABLE VIII. THE AREA INFORMATION FROM PNR	40
TABLE IX. POWER SIMULATIONS FOR THE ORIGINAL FILTER (IMPLUSE RESPONSE).....	44
TABLE X. POWER SIMULATIONS FOR THE CASCADED FILTER (IMPLUSE RESPONSE).....	44
TABLE XI. POWER SIMULATIONS FOR ORIGINAL FILTER (SQUARE WAVE INPUT)	45
TABLE XII. POWER SIMULATIONS FOR CASCADED FILTER (SQUARE WAVE INPUT)	45
TABLE XIII. POWER SIMULATIONS FOR ORIGINAL FILTER (RANDOM INPUT)	46
TABLE XIV. POWER SIMULATIONS FOR ORIGINAL FILTER (RANDOM INPUT)	47

List of Acronyms

CMOS	Complementary Metal Oxide Semiconductor
IC	Integrated Circuit
SoC	System on Chip
PDA	Personal Digital Assistants
DSP	Digital Signal Processing
IIR	Infinite-duration Impulse Response
FIR	Finite-duration Impulse Response
LPHVT	Low Power High Threshold Voltage
WDF	Wave Digital Filter
LWDF	Lattice Wave Digital Filter
PNR	Place and Route
MSB	Most Significant Bit
LSB	Least Significant Bit
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuits

Appendix 1

A.1 Converting Decimal number to Binary number

In general, a number with a decimal point is represented by a series of coefficients as follows: [11]

$$a_5a_4a_3a_2a_1a_0.a_{-1}a_{-2}a_{-3}$$

The a_j coefficients are any of 10 digits (0, 1, 2, ..., 9), and the subscript value j gives the place value. Hence, the power of 10 by which the coefficient must be multiplied. This can be expressed as

$$10^5a_5 + 10^4a_4 + 10^3a_3 + 10^2a_2 + 10^1a_1 + 10^0a_0 + 10^{-1}a_{-1} + 10^{-2}a_{-2} + 10^{-3}a_{-3}$$

The decimal number is said to be base, or radix, 10 because it uses 10 digits and the coefficients are multiplied by powers of 10.

The binary system is different number system. The coefficients of the binary number system have only two possible values: 0 or 1. Each coefficient a_j is multiplied by 2^j .

The decimal numbers, which are used in the original filter, are converted to binary numbers as follows:

To convert the decimal number $(0.375)_{10}$ to the binary number. First, 0.375 is multiplied by 2 to give an integer and a fraction. The new fraction is multiplied by 2 to a new integer and a new fraction. This process is continued until the fraction becomes 0 or until the number of digits has sufficient accuracy. The coefficients of the binary number are obtained from the integers as follows:

Coefficient $a_0 = 0.375$

	Integer		Fraction	Coefficient
$0.375 \times 2 =$	0	+	0.750	$a_{-1} = 0$
$0.750 \times 2 =$	1	+	0.500	$a_{-2} = 1$
$0.500 \times 2 =$	1	+	0.000	$a_{-3} = 1$

Therefore, the answer is $(0.375)_{10} = (0.a_{-1}a_{-2}a_{-3})_2 = (0.011)_2$

Coefficient $a_1 = 0.578125$

	Integer		Fraction	Coefficient
$0.578125 \times 2 =$	1	+	0.156250	$a_{-1} = 1$
$0.156250 \times 2 =$	0	+	0.312500	$a_{-2} = 0$
$0.312500 \times 2 =$	0	+	0.625000	$a_{-3} = 0$
$0.625000 \times 2 =$	1	+	0.250000	$a_{-4} = 1$
$0.250000 \times 2 =$	0	+	0.500000	$a_{-5} = 0$
$0.500000 \times 2 =$	1	+	0.000000	$a_{-6} = 1$

Therefore, the answer is $(0.578125)_{10} = (0.a_{-1}a_{-2}a_{-3}a_{-4}a_{-5}a_{-6})_2$
 $= (0.100101)_2$

Coefficient $-a_2 = 0.328125$

	Integer		Fraction	Coefficient
$0.328125 \times 2 =$	0	+	0.656250	$a_{-1} = 0$
$0.656250 \times 2 =$	1	+	0.312500	$a_{-2} = 1$
$0.312500 \times 2 =$	0	+	0.625000	$a_{-3} = 0$
$0.625000 \times 2 =$	1	+	0.250000	$a_{-4} = 1$
$0.250000 \times 2 =$	0	+	0.500000	$a_{-5} = 0$
$0.500000 \times 2 =$	1	+	0.000000	$a_{-6} = 1$

Therefore, the answer is $(0.328125)_{10} = (0.a_{-1}a_{-2}a_{-3}a_{-4}a_{-5}a_{-6})_2$
 $= (0.010101)_2$

A.2 Synthesis Script

Maximum speed synthesis script

```
remove_design -all
analyze -library WORK -format vhdl
{/home/piraten/sx08ag3/thesiswork/original_filter/components_pack.vhd
/home/piraten/sx08ag3/thesiswork/original_filter/mul_a0.vhd
/home/piraten/sx08ag3/thesiswork/original_filter/mul_a1.vhd
/home/piraten/sx08ag3/thesiswork/original_filter/mul_a2.vhd
/home/piraten/sx08ag3/thesiswork/original_filter/orginal_filter.vhd}
elaborate original_filter -architecture STRUCTURAL -library WORK -update
create_clock -name "clk" -period 3 -waveform { 0 1.5 } { clk }
set_clock_uncertainty 0.01 {clk}
compile -map_effort high
report_constraints -all_violators
remove_unconnected_ports -blast_buses [get_cells "*" -hier]
remove_unconnected_ports [get_cells "*" -hier]
report_timing -max_paths 1
report_area
report_cell
report_net -verbose -connections
sizeof_collection [all_registers]
change_names -rules verilog -hierarchy
write -format verilog -hierarchy -output ./netlists/original_filter.v
write_sdf ./netlists/original_filter.sdf
write_sdc ./netlists/original_filter.sdc
```

Minimum area synthesis script

```
remove_design -all
analyze -library WORK -format vhdl
{/home/piraten/sx08ag3/thesiswork/original_filter/components_pack.vhd
/home/piraten/sx08ag3/thesiswork/original_filter/mul_a0.vhd
/home/piraten/sx08ag3/thesiswork/original_filter/mul_a1.vhd
/home/piraten/sx08ag3/thesiswork/original_filter/mul_a2.vhd
/home/piraten/sx08ag3/thesiswork/original_filter/orginal_filter.vhd}
elaborate original_filter -architecture STRUCTURAL -library WORK -update
create_clock -name "clk" -period 8 -waveform { 0 1.5 } { clk }
set_clock_uncertainty 0.01 {clk}
set_max_area 0
compile -map_effort high
report_constraints -all_violators
remove_unconnected_ports -blast_buses [get_cells "*" -hier]
remove_unconnected_ports [get_cells "*" -hier]
report_timing -max_paths 1
report_area
report_cell
report_net -verbose -connections
sizeof_collection [all_registers]
change_names -rules verilog -hierarchy
write -format verilog -hierarchy -output ./netlists/original_filter.v
write_sdf ./netlists/original_filter.sdf
write_sdc ./netlists/original_filter.sdc
```

A.3 Place and Route Script

```
#----LOAD DESIGN
loadConfig /home/piraten/sx08ag3/thesiswork/original_filter/netlists/original.conf 0
commitConfig
setDrawViewplan
fit

#----FLOORPLAN SETTINGS
floorPlan -site CORE -r 0.589444825223 0.329418 110.0 110.0 110.0 110.0
setObjFPlanBox Module DUT 222.4 222.588 383.4 310.988

#----CONNECTING GLOBAL VCC & GND
clearGlobalNets
globalNetConnect VCC -type pgpin -pin VCC -inst *
globalNetConnect VCC -type tiehi -pin VCC -inst *
globalNetConnect GND -type pgpin -pin GND -inst *
globalNetConnect GND -type tielo -pin GND -inst *
saveFPlan /home/piraten/sx08ag3/thesiswork/original_filter/netlists/iopads_original_top.fp

#----POWER PLANNING
addRing -spacing_bottom 3 -width_left 4.9 -width_bottom 4.9 -width_top 4.9 -spacing_top 3 -layer_bottom M5 -
stacked_via_top_layer AP -width_right 4.9 -around core -jog_distance 2.5 -offset_bottom 2.5 -layer_top M5 -
threshold 2.5 -offset_left 2.5 -spacing_right 3 -spacing_left 3 -offset_right 2.5 -offset_top 2.5 -layer_right M6 -
nets {GND VCC } -stacked_via_bottom_layer M1 -layer_left M6

#----ADD POWER STRIPES
addStripe -block_ring_top_layer_limit M7 -max_same_layer_jog_length 6 -padcore_ring_bottom_layer_limit M5
-set_to_set_distance 50 -stacked_via_top_layer AP -padcore_ring_top_layer_limit M7 -spacing 2 -xleft_offset 10
-merge_stripes_value 2.5 -layer M6 -block_ring_bottom_layer_limit M5 -width 3 -nets {GND VCC } -
stacked_via_bottom_layer M1

#----place standard cells
placeDesign -prePlaceOpt
#----optimise the placed standard cells
optDesign -preCTS
set DrawView place
clockDesign -specFileClock.ctstch -outdirclock_report -fixedInstBeforeCTS
set DrawView ameaba

#---- FILLER CELLS
getFillerMode -quiet
findCoreFillerCells
addFiller -cell HS65_LS_FILLERSNPWFPF4 HS65_LS_FILLERSNPWFPF3 HS65_LS_FILLERPF4
HS65_LS_FILLERPF3 HS65_LS_FILLERPF2 HS65_LS_FILLERPF1 HS65_LS_FILLERPFOP9
HS65_LS_FILLERPFOP8 HS65_LS_FILLERPFOP64 HS65_LS_FILLERPFOP32 HS65_LS_FILLERPFOP16
HS65_LS_FILLERPFOP12 HS65_LS_FILLERNPWFPF8 HS65_LS_FILLERNPWFPF4
HS65_LS_FILLERNPWFPF4 HS65_LS_FILLERNPWFPF32 HS65_LS_FILLERNPWFPF3
HS65_LS_FILLERNPWFPF16 HS65_LS_FILLERNPW4 HS65_LS_FILLERNPW3 HS65_LS_FILLERCELL4
HS65_LS_FILLERCELL3 HS65_LS_FILLERCELL2 HS65_LS_FILLERCELL1
HS65_LL_FILLERSNPWFPF4 HS65_LL_FILLERSNPWFPF3 HS65_LL_FILLERPF4
HS65_LL_FILLERPF3 HS65_LL_FILLERPF2 HS65_LL_FILLERPF1 HS65_LL_FILLERPFOP9
HS65_LL_FILLERPFOP8 HS65_LL_FILLERPFOP64 HS65_LL_FILLERPFOP32 HS65_LL_FILLERPFOP16
HS65_LL_FILLERPFOP12 HS65_LL_FILLERNPWFPF8 HS65_LL_FILLERNPWFPF64
HS65_LL_FILLERNPWFPF4 HS65_LL_FILLERNPWFPF32 HS65_LL_FILLERNPWFPF3
HS65_LL_FILLERNPWFPF16 HS65_LL_FILLERNPW4 HS65_LL_FILLERNPW3 HS65_LL_FILLERCELL4
HS65_LL_FILLERCELL3 HS65_LL_FILLERCELL2 HS65_LL_FILLERCELL1
HS65_LH_FILLERSNPWFPF4 HS65_LH_FILLERSNPWFPF3 HS65_LH_FILLERPF4
HS65_LH_FILLERPF3 HS65_LH_FILLERPF2 HS65_LH_FILLERPF1 HS65_LH_FILLERPFOP9
HS65_LH_FILLERPFOP8 HS65_LH_FILLERPFOP64 HS65_LH_FILLERPFOP32
HS65_LH_FILLERPFOP16 HS65_LH_FILLERPFOP12 HS65_LH_FILLERNPWFPF8
HS65_LH_FILLERNPWFPF64 HS65_LH_FILLERNPWFPF4 HS65_LH_FILLERNPWFPF32
```

```

HS65_LH_FILLERNPWFPF3 HS65_LH_FILLERNPWFPF16 HS65_LH_FILLERNPW4
HS65_LH_FILLERNPW3 HS65_LH_FILLERCELL4 HS65_LH_FILLERCELL3 HS65_LH_FILLERCELL2
HS65_LH_FILLERCELL1 HS65_LH_DECAP9 HS65_LH_DECAP8 HS65_LH_DECAP64
HS65_LH_DECAP4 HS65_LH_DECAP32 HS65_LH_DECAP16 HS65_LH_DECAP12
HS65_GS_FILLERSNPWFPF4 HS65_GS_FILLERSNPWFPF3 HS65_GS_FILLERPF4
HS65_GS_FILLERPF3 HS65_GS_FILLERPF2 HS65_GS_FILLERPF1 HS65_GS_FILLERPFOP9
HS65_GS_FILLERPFOP8 HS65_GS_FILLERPFOP64 HS65_GS_FILLERPFOP32 HS65_GS_FILLERPFOP16
HS65_GS_FILLERPFOP12 HS65_GS_FILLERNPWFPF8 HS65_GS_FILLERNPWFPF64
HS65_GS_FILLERNPWFPF4 HS65_GS_FILLERNPWFPF32 HS65_GS_FILLERNPWFPF3
HS65_GS_FILLERNPWFPF16 HS65_GL_FILLERSNPWFPF4 HS65_GL_FILLERSNPWFPF3
HS65_GL_FILLERPF4 HS65_GL_FILLERPF3 HS65_GL_FILLERPF2 HS65_GL_FILLERPF1
HS65_GL_FILLERPFOP9 HS65_GL_FILLERPFOP8 HS65_GL_FILLERPFOP64 HS65_GL_FILLERPFOP32
HS65_GL_FILLERPFOP16 HS65_GL_FILLERPFOP12 HS65_GL_FILLERNPWFPF8
HS65_GL_FILLERNPWFPF64 HS65_GL_FILLERNPWFPF4 HS65_GL_FILLERNPWFPF32
HS65_GL_FILLERNPWFPF3 HS65_GL_FILLERNPWFPF16 HS65_GH_FILLERSNPWFPF4
HS65_GH_FILLERSNPWFPF3 HS65_GH_FILLERPF4 HS65_GH_FILLERPF3 HS65_GH_FILLERPF2
HS65_GH_FILLERPF1 HS65_GH_FILLERPFOP9 HS65_GH_FILLERPFOP8 HS65_GH_FILLERPFOP64
HS65_GH_FILLERPFOP32 HS65_GH_FILLERPFOP16 HS65_GH_FILLERPFOP12
HS65_GH_FILLERNPWFPF8 HS65_GH_FILLERNPWFPF64 HS65_GH_FILLERNPWFPF4
HS65_GH_FILLERNPWFPF32 HS65_GH_FILLERNPWFPF3 HS65_GH_FILLERNPWFPF16
HS65_50_DECAP9 HS65_50_DECAP64 HS65_50_DECAP32 HS65_50_DECAP16 HS65_50_DECAP12
HS65_28_DECAP9 HS65_28_DECAP64 HS65_28_DECAP32 HS65_28_DECAP16 HS65_28_DECAP12 -
prefix FILLER -markFixed

```

```
#----IO FILLER CELLS
```

```

addIoFiller -cell IOFILLERCELL64_ST_SF_LIN -prefix if64
addIoFiller -cell IOFILLERCELL32_ST_SF_LIN -prefix if32
addIoFiller -cell IOFILLER16_ST_SF_LIN -prefix if16
addIoFiller -cell IOFILLER8_ST_SF_LIN -prefix if8
addIoFiller -cell IOFILLER4_ST_SF_LIN -prefix if4
addIoFiller -cell IOFILLER2_ST_SF_LIN -prefix if2
addIoFiller -cell IOFILLER1_ST_SF_LIN -prefix if1
redraw
#----signal Routing
sroute -noPadRings -jogControl { preferWithChangesdifferentLayer }
setNanoRouteMode -quiet -routeWithTimingDriven true
setNanoRouteMode -quiet -drouteEndIteration default
setNanoRouteMode -quiet -routeWithSiDriven false
routeDesign -globalDetail
#----verify geometry violations
verifyGeometry
#----verify Antenna violations
violationBrowser -all -no_display_false
#----save netlist
Save Netlistnetlists/from_pnr_original1.v
delayCal -sdfnetlists/from_pnr_original1.sdf -idealclock

```

A.4 Power Analysis Script

```
remove_design -all
setpower_enable_analysis true
setsearch_path "$env(STM065_DIR)/IO65LPHVT_SF_1V8_50A_7M4X0Y2Z_7.0/libs \
$env(STM065_DIR)/CORE65LPHVT_5.1/libs \
    $env(STM065_DIR)/CORE65LPSVT_5.1/libs \
$search_path"

setlink_library "* IO65LPHVT_SF_1V8_50A_7M4X0Y2Z_nom_1.00V_1.80V_25C.db \
CORE65LPHVT_nom_1.20V_25C.db CORE65LPSVT_nom_1.20V_25C.db"

settarget_library "IO65LPHVT_SF_1V8_50A_7M4X0Y2Z_nom_1.00V_1.80V_25C.db \
CORE65LPHVT_nom_1.20V_25C.db CORE65LPSVT_nom_1.20V_25C.db "

read_verilog /home/piraten/sx08ag3/power/netlists/original_filter.v
current_design original_filter_top
create_clock "clk_top" -name "clk_top" -period 8
set_clock_uncertainty 0.01 {clk_top}
report_vcd_hierarchy ./original_filter.vcd
read_vcd -strip_path tb_original/dut ./original_filter.vcd
update_power
report_power
report_power> ./report/power_report_original_filter.rpt
report_timing> ./report/timing_report_original_filter.rpt
```