2009

Third order CMOS decimator design for sigma delta modulators

Hemalatha Mekala
Louisiana State University and Agricultural and Mechanical College

THIRD ORDER CMOS DECIMATOR DESIGN FOR SIGMA DELTA MODULATORS

A Thesis

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering

in

The Department of Electrical and Computer Engineering

by
Hemalatha Mekala
B.Tech, Jawaharlal Nehru Technological University, India, 2006,
December 2009

Acknowledgments

I would like to take this opportunity to thank everyone who contributed to the successful completion of this project. First of all, I would like to thank Dr. Ashok Srivastava, my major professor, for providing extensive support and encouragement throughout this work. I am very grateful for his guidance, patience and understanding throughout this work. I like to thank Professor Guoxiang Gu and Professor Shuangqing Wei for being a part of my thesis committee.

I am very thankful to the Department of Electrical and Computer Engineering. I would also like to thank Rajiv Soundararajan and Siva Yellampalli for taking time to answer technical questions in this work. I also like to thank all my other friends, Varun Kamalakaran, Shilpa Nutanapati, Raghava Alapati, Manthan Malde and Ravikishore Pratapa for their help and encouragement at times.

I am very thankful to Department of Geology and Geophysics, for supporting me financially during my stay at LSU.

I would also like to thank my parents, Mohanrami Reddy Mekala and Prameela Mekala for their support and encouragement throughout my career. I deeply thank my mother for her moral support during all the difficult times. All her prayers have helped me reach this position in my life.

Table of Contents

List of Tables

List of Figures

Abstract

A third order Cascaded Integrated Comb (CIC) filter has been designed in 0.5μm n-well CMOS process to interface with a second order oversampling sigma-delta ADC modulator. The modulator was designed earlier in 0.5μm technology. The CIC filter is designed to operate with 0 to 5V supply voltages. The modulator is operated with ±2.5V supply voltage and a fixed oversampling ratio of 64. The CIC filter designed includes integrator, differentiator blocks and a dedicated clock divider circuit, which divides the input clock by 64. The CIC filter is designed to work with an ADC that operates at a maximum oversampling clock frequency of up to 25 MHz and with baseband signal bandwidth of up to 800 kHz. The design and performance of the CIC filter fabricated has been discussed.

Chapter 1
Introduction

Deep submicron technology has a significant impact on developing trends in VLSI. As CMOS technology advances, the short channel MOS transistors exhibit higher and higher transconductance, making it more attractive for high-speed analog applications [1]. These scaled transistors also require lower power supply. The reduced channel length and increase in switching speed of CMOS has led the digital systems of wireless communications to provide higher data rate, high quality audio, video and interactive multimedia [2]. These digital systems rely highly on advanced Analog-to-Digital converter (ADC) and Digital-to-Analog converter (DAC).

Usually successive approximation or dual slope converters are used when high resolution is desired. But to achieve higher accuracy, trimming is required. Dual-slope converters require high-speed and high accuracy integrators that can only be fabricated using a high-f_T bipolar process. The main constraint using these architectures is the design of high precision sample and hold circuits. The over sampling converters use digital signal processing techniques in place of complex and precise analog components, which, gives scope to achieve much higher resolution than the Nyquist rate converters. The accuracy of these oversampling converters does not depend on the component matching, precise sample and hold circuitry or trimming and they require only a small amount of analog circuitry [3]. Sigma-delta ADC is one such converter. They are low cost converters, which also provide high dynamic range and flexibility in converting low bandwidth input signals.

The sigma-delta ADC works on the principle of sigma-delta modulation. The sigma-delta (ΣΔ) modulation is a method for encoding high-resolution signals into lower resolution signals using pulse-density modulation. It falls under the category of oversampling ADC's as it samples the input signal at a rate much higher than the Nyquist rate. The core blocks of Delta-Sigma ADC are modulator and decimator. The modulator is an analog block used to sample the input signal at an oversampling rate and decimator is a digital filter or down sampler where the actual digital signal processing is done.

The sample rate changes can be very large, with changes from many tens of MHz to around 100 kHz in applications such as software designed radios, cable modems, satellite receivers, 3G base stations, and radar systems. Such a requirement leads to higher order and high-rate digital filters. A CIC filter is typically used in applications where the system sample rate is much larger than the bandwidth occupied by the signal. It is also used in systems where large rate change factors require large amounts of coefficient storage or fast impulse response generation and the memory is either unavailable or too slow to perform the desired application. They are used widely in building digital down converters and digital up converters.

Research on digital filter implementation has concentrated on custom implementation using various VLSI technologies. The architecture of these filters is largely determined by the target applications of the particular implementations [4]. High performance designs for filtering can be implemented in CMOS and BiCMOS technologies for specific application domains.

In the present work, the motivation is to design and implement a customized CMOS ADC. It has an ADC modulator and a decimator. The output of the modulator will be at the oversampling rate and number of bits will be equal to the number of quantization levels of the ADC in the modulator loop [5]. The modulator output represents the input contaminated with quantization noise at higher frequencies. This noise has to be filtered out using a digital filter. A simple RC filter circuit can be used to remove the quantization noise but this will also convert the output to analog at the same time [6]. Therefore, we need a digital filter, which can remove the quantization noise and also preserve the digital nature of the modulator output.

The focus is to design a CIC filter also known as decimator to achieve the required resolution for a particular oversampling frequency for the already designed second order ADC modulator by other researchers in the group. An external sampling clock is used to operate both the modulator and the decimator stage, which will provide the flexibility to operate the ADC with different input signal bandwidths. The research goal is to develop circuitry for a third order decimator and an internal clock divider circuit, which generates a down sampling clock signal and also achieve a low power design at the same time. The decimator design operates with 0 to 5V supply to interface with a second order sigma-delta ADC modulator, which operates with ± 2.5V supply.

Chapter 2
Sigma-Delta ADC Overview

Mixed-signal circuits bridge the gap between the digital and analog circuits and have several advantages. They help in reducing the size of the system, increasing the speed of operation, reducing the power dissipation and increasing the design flexibility. Data converters are the core components of the mixed-signal system. One of the challenges in designing these data converters is to extend the input frequency range, while maintaining a feasible oversampling ratio (OSR).

There are four types of ADC architectures: pipeline, flash-type, successive approximation and oversampled ADCs. Flash or parallel converters have the highest speed of any type of ADC. The pipeline ADC is an N-step converter, with 1-bit being converted per stage. High resolution and fast speeds can be obtained using these ADCs. Successive approximation ADCs are simple to design and allow both high speed and high resolution and require relatively small area. This architecture is used predominantly. But, oversampling ADCs give the highest resolution among all converters. Figure 2.1 shows the comparison of the four architectures mentioned above [5].

2.1 Introduction to Sigma-Delta ADC

Depending on the sampling rate, analog-to-digital converters are categorized into two types namely Nyquist rate converters and oversampling converters [5]. Nyquist rate ADCs sample the analog input at the Nyquist frequency, f_n such that $f_s = f_n = 2 \text{ x } f_b$, where f_s is the sampling frequency and f_b is the bandwidth of the input signal. Oversampling ADCs sample the analog input at much higher frequencies than the Nyquist frequency [7]. Sigma-delta ADCs come under this category. In a sigma-delta ADC, the input signal is sampled at an oversampling frequency $f_s = K \text{ x } f_n$ where K is defined as the oversampling ratio and is given by

$$K= f_s /2\, f_b \qquad (2.1)$$

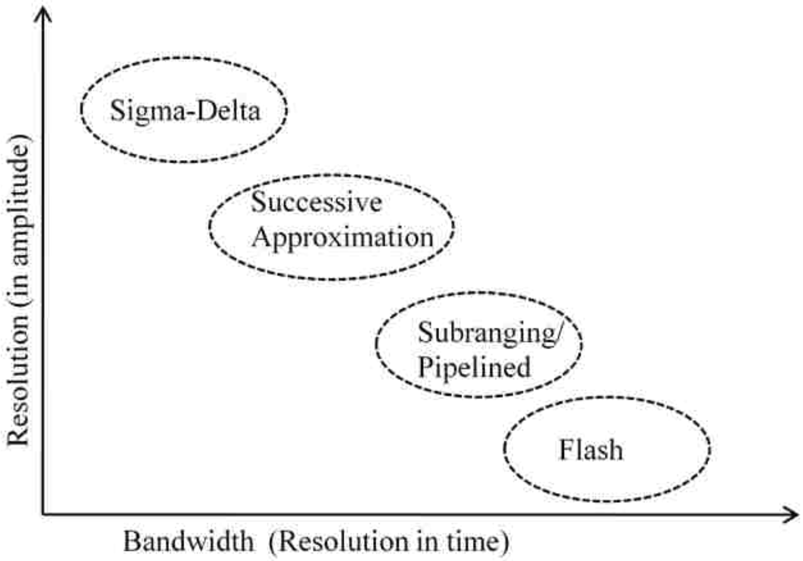


Figure 2.1: Bandwidth versus resolution for ADC's [5].

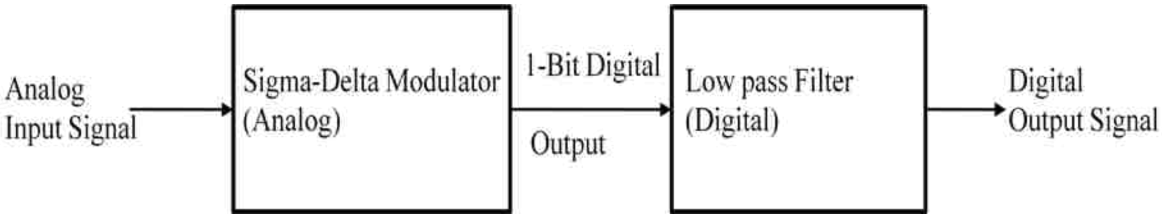


Figure 2.2: Basic block diagram of a 1st order sigma-delta ADC.

Figure 2.2 shows the basic blocks of a sigma-delta ADC. It consists of a sigma-delta modulator and a low pass filter. The modulator will be implemented with analog technique to produce a bit stream and a digital low pass filter will be implemented to achieve a digital output. Similarly for a DAC the modulator will be implemented with digital technique and an analog low pass filter will be implemented to achieve an analog output.

The following sub-sections give an overview of the terms quantization noise, signal sampling and noise shaping which are related to sigma-delta converters.

2.1.1 Quantization Noise

The analog input signal can take any continuous value. But a digital n-bit signal can only settle to 2^n discrete values. It is this difference between the analog value and its digital representation, which causes the distortion known as the quantization noise. Quantization error is defined as a measure of an n-bit converter's failure to represent precisely an analog signal in the digital domain.

2.1.2 Signal Sampling

Oversampling is a process of sampling the input signal at a frequency much greater than the Nyquist frequency (Nyquist frequency, f_n is defined as twice the input signal bandwidth, f_b). This process greatly reduces the quantization noise in the required band. The sampling theorem or the basic Nyquist sample theory states that the sampling frequency of a signal must be at least twice the input signal frequency in order to reconstruct the sampled signal without distortion. Figure 2.3(a) shows the spectrum of an under sampled signal [8]. Here the sampling frequency, f_s is less than twice the input signal frequency $2f_o$. The shaded portion of the figure shows the aliasing which occurs when the sampling theorem is not followed. We get a distorted signal at the output when a signal contaminated with aliasing is recovered. Figure 2.3(b) shows the spectrum of an oversampled signal [8]. Here the sampled signal in the frequency domain appears as a series of band-limited signals at multiples of sampling frequency that are widely spaced. This process puts the entire input bandwidth at less than $f_S/2$ which reduces the aliasing.

2.1.3 Noise Shaping

Noise shaping is a property of sigma-delta ADCs resulting from the application of feedback that extends dynamic range. A closed loop modulator works as a high-pass filter for quantization-noise and as a low-pass filter for the input signal. When the signal is over-sampled, the quantization noise power in the Nyquist bandwidth ($f_s/2$) spreads over the wider bandwidth, $Kf_s/2$ where; K is the oversampling ratio [9], which is shown in Figure 2.4. The total quantization noise is still the same but the quantization noise in the bandwidth of interest is greatly reduced. The figure also illustrates the noise shaping achieved by using the oversampled sigma-delta modulator.

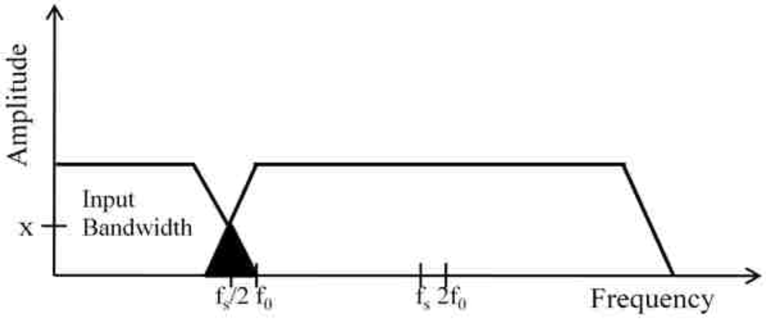


Figure 2.3(a): Under sampled signal spectrum.

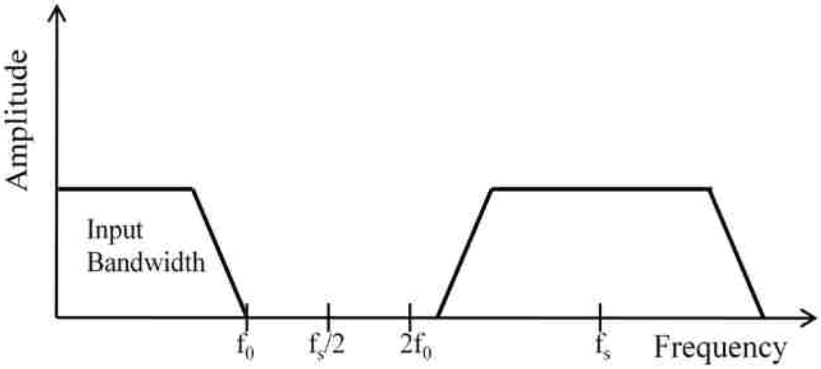


Figure 2.3(b): Oversampled signal spectrum.

In a sigma-delta ADC, the analog modulator samples the input at oversampling ratio and after the input signal passes through the modulator it is fed into the digital filter or a decimator. The function of the digital filter is to provide a sharp cutoff at the bandwidth of interest, which essentially removes out of band quantization noise and signals [9].

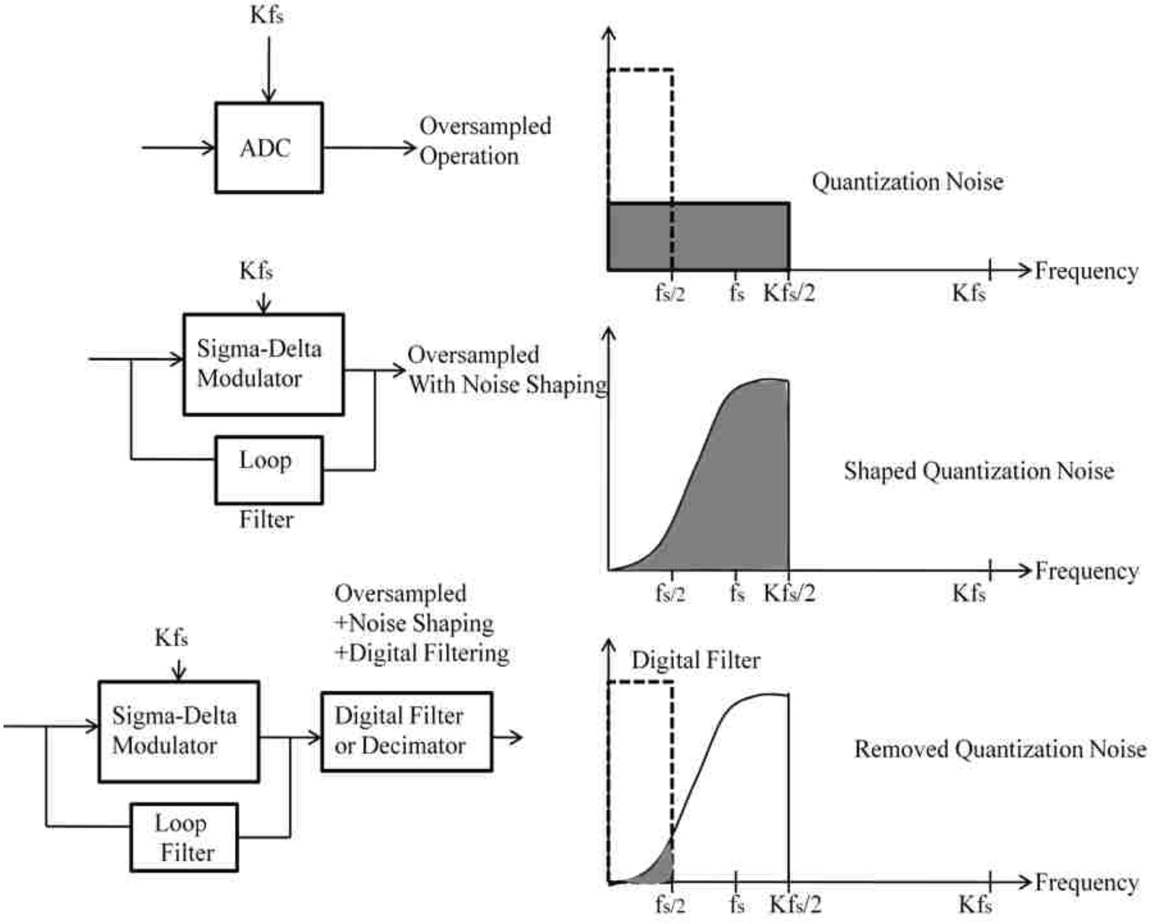


Figure 2.4: Putting noise shaping and digital filtering together [9].

2.2 Modulator Overview

Figure 2.5 shows the block diagram of a first order sigma delta modulator. It consists of an integrator and a comparator in the forward path and a 1-bit digital-to-analog converter (DAC) in the feedback loop. The name first order is derived from the fact that there is only one integrator in the circuit, placed in the forward path. As the Greek letter-Δ (delta) is used to show the deviation or small incremental change, the process came to be known as “delta modulation”. Delta modulation is based on quantizing the change in the signal from sample to sample rather than the absolute value of the signal at each sample. Sigma stands for summing or integrating, which is performed at the input stage on the digital output from 1-bit DAC and the input signal before the delta modulation [11]. Hence this technique of analog to digital conversion is called sigma-delta modulation.

The single-bit feedback DAC output is subtracted from the analog input signal, by the summing amplifier. The resultant error signal from the summing amplifier output is low-pass filtered by the integrator. The comparator works at an over-sampling clock frequency. It compares the input signal against its last sample, to see if this new sample is larger than the last one. If it is larger, then the output is increasing and if it is smaller, then the output is decreasing. This comparator is effectively a 1-bit A/D converter (ADC). The density of ‘1s’ or ‘0s’ forming the pulse stream at the output is a direct digital representation of the input signal. The 1-bit DAC in the feedback loop switches between ± V_{ref} depending on the output bit stream. This architecture achieves higher resolution and bandwidth.

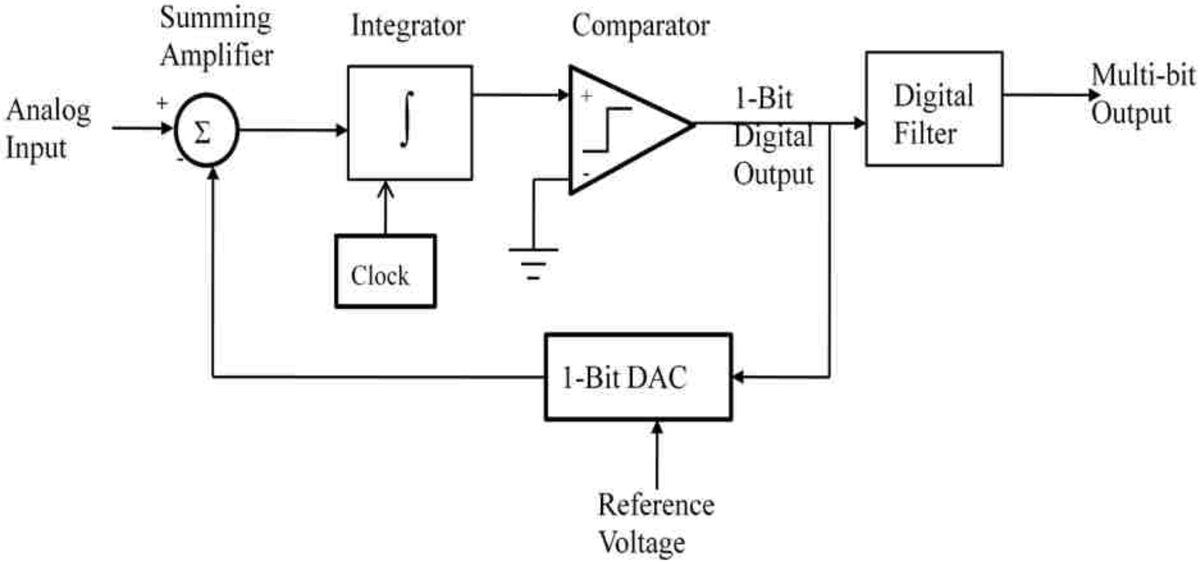


Figure 2.5: Block diagram of a first order analog sigma delta ADC [10].

A converter usually requires a sample rate of more than twice the highest input frequency to reproduce the sampled signal without distortion. The density of the pulses represents the average value of the signal over a specific period. Figure 2.6 illustrates the sample output of the modulator for a sine wave input. It can be observed that for the positive peak of the sine wave, most of the pulses are high and for the negative peak of the sign wave most of the pulses are low. As the sine wave decreases in value, the pulses become distributed between high and low according to the sine wave value.

The resolution of the sigma-delta ADC can be increased by balancing three design aspects. They are the over-sampling ratio, order of the sigma-delta modulator and quantizer resolution. By using a higher order modulator, the superimposed noise on the bit stream can also be reduced. Doubling the sampling frequency will also decrease the in-band quantization noise by 3(2N+1) dB where N is the order of the modulator [8]. Also the input signal bandwidth may be higher than the first order. A second order sigma-delta modulator can be derived by placing two integrators in series as shown in Figure 2.7. The operation of the second order modulator blocks is similar to that of the first order modulator except that the integration is performed twice on the data.

2.3 Decimator Overview

Decimation is a technique used to reduce the number of samples in a discrete-time signal. The process of decimation is used in a sigma-delta converter to eliminate redundant data at the output. In practice, this usually implies low pass filtering a signal and then down sampling to decrease its effective sampling rate. The function of the digital filter in a sigma-delta ADC is to attenuate the out of band quantization noise and signal components that are pushed out to the higher frequencies by modulator as shown in Figure 2.8(a) and Figure 2.8(b) [8].

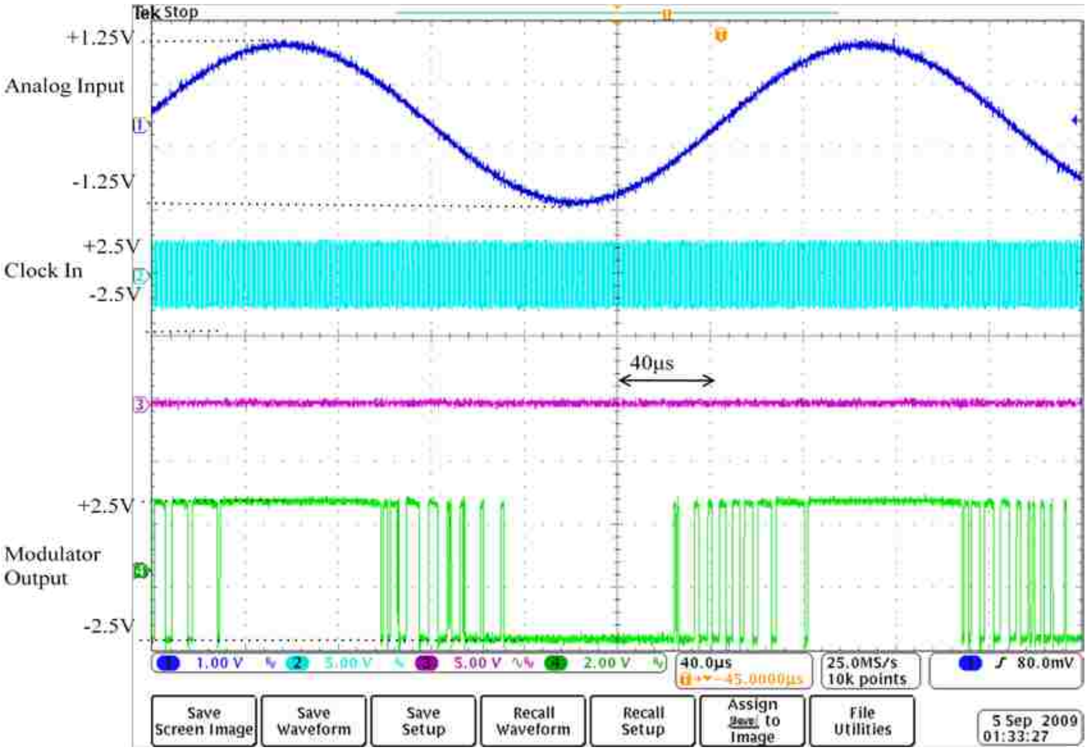


Figure 2.6: Estimated signals within a first order analog modulator.

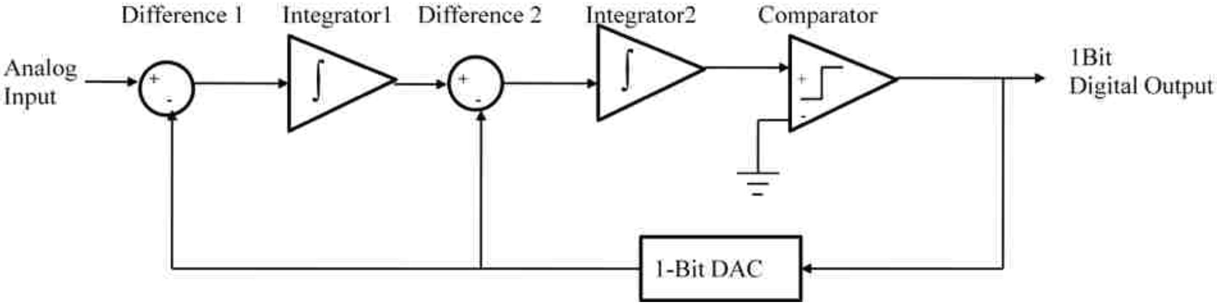
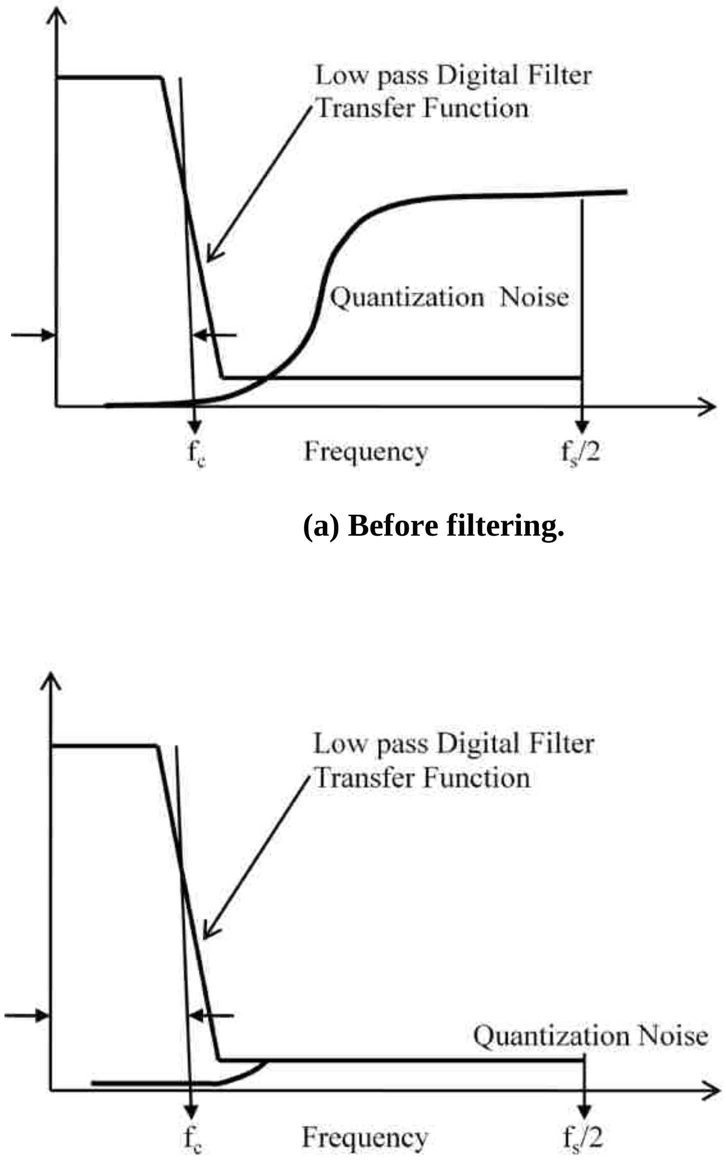


Figure 2.7: Block diagram of a second order sigma-delta modulator [8].

Since the sigma-delta modulator oversamples the input signal to reduce the quantization noise, this results in redundant data that has to be eliminated. The decimation process for an input signal decimated by a factor of four in time domain is as shown in Figure 2.9(a). The frequency domain spectrum of a decimated signal is shown in Figure 2.9(b) [8]. The MATLAB code required to generate Figure 2.9(a) is given in Appendix C. Here y = decimate (x,r) function is used. It reduces the sample rate of input signal, x by the oversampling ratio factor, r. By default, decimate function employs an eighth-order low pass Chebyshev Type I filter with a cutoff frequency of $0.8(f_s/2)/r$ [12]. It filters the input sequence in both the forward and reverse directions to remove all phase distortion, effectively doubling the filter order.


Low pass Digital Filter Transfer Function
Quantization Noise
f_c
Frequency
$f_s/2$
Low pass Digital Filter Transfer Function
Quantization Noise
f_c
Frequency
$f_s/2$

(a) Before filtering.

(b) After filtering.

Figure 2.8: In-band quantization noise before and after filtering [8].

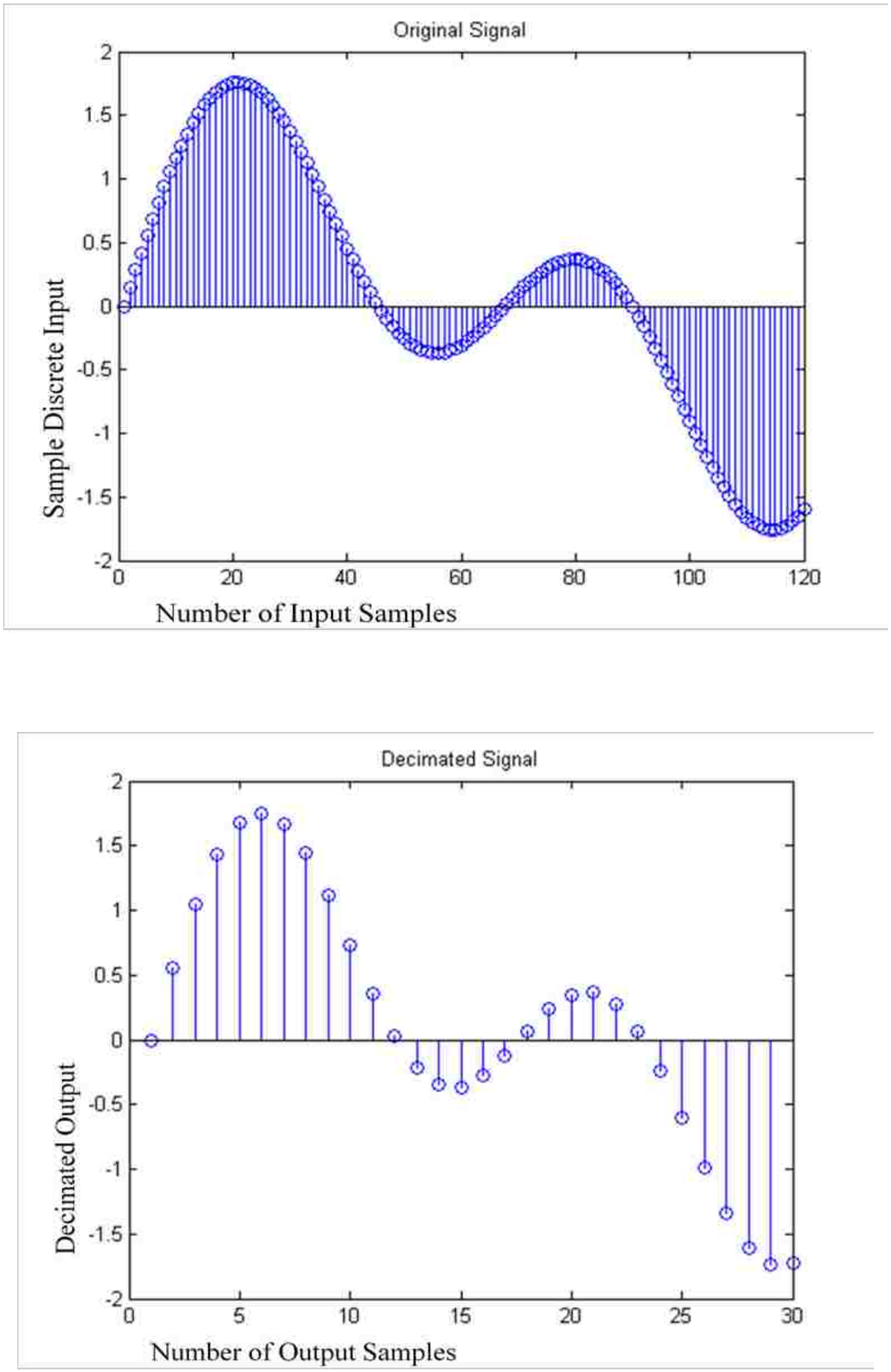


Figure 2.9(a): Decimation in time domain-original and decimated signals.

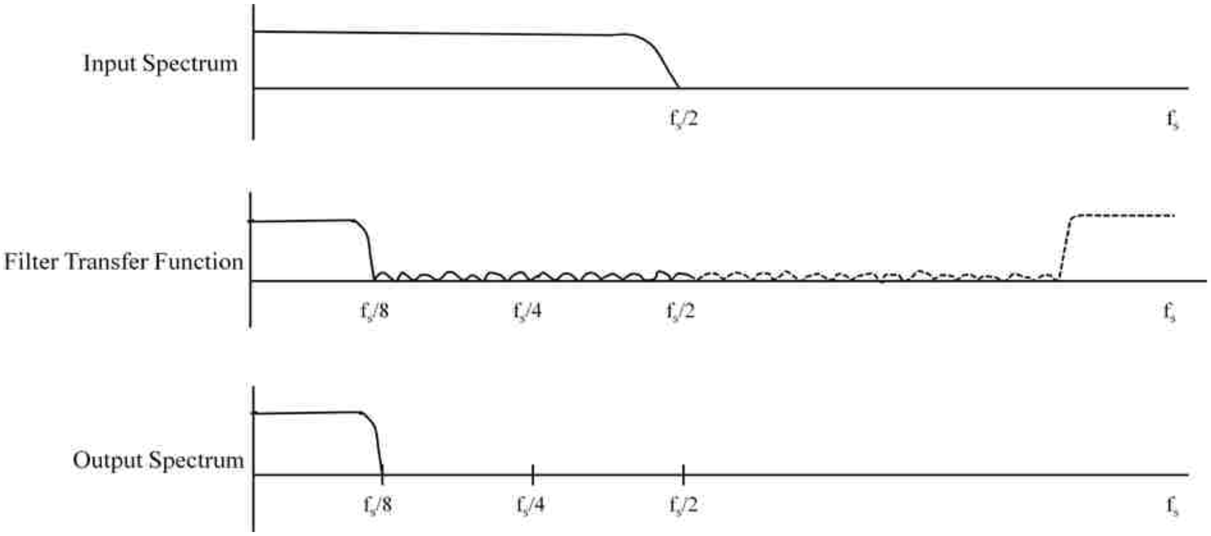


Figure 2.9(b): Decimation by 4 in frequency domain.

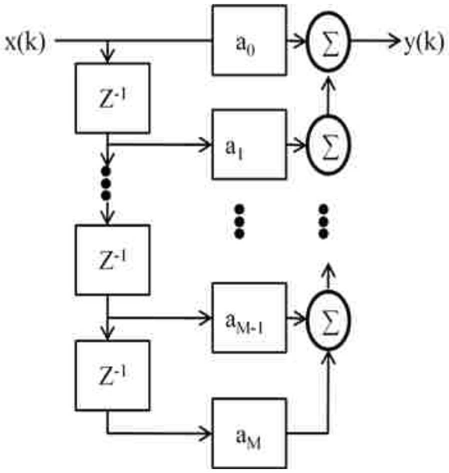
2.4 Literature Review on Low-pass Digital Filters and Decimators

Digital filtering is a computational process, or an algorithm which is used in transforming a discrete sequence of numbers (the input) into another discrete sequence of numbers (the output) having a modified frequency domain spectrum. Digital filtering can be in the form of software routine operating on the data stored in computer memory or can be implemented with a dedicated digital hardware. Digital filtering algorithms are most commonly implemented using general purpose digital signal processing chips for audio applications, or special purpose digital filtering chips and application-specific integrated circuits (ASICs) for higher sampling rates [4]. A digital filter can be analyzed in the time or frequency domain.

Filtering is necessary for decimation or interpolation of a digital signal. To implement the filtering part, one can use either finite impulse response (FIR) or infinite impulse response (IIR) filters. The first class of filters is known as FIR filters, also known as non-recursive filters and can be represented by equation (2.2). For the FIR, the output, y(n) is dependent only on past and present values of the input. Figure 2.10 shows a block diagram example and derived transfer function of an FIR filter [8].

$$y(n)=\sum\nolimits_{k=0}^{M}a_{k}x(n-k) \tag{2.2}$$

In equation (2.2), a_k's are the filter coefficients, x(n) is the input signal, y(n) is the output signal and M is the filter order. An M^{th} order filter has (M+1) terms. A decimating FIR is the same as a regular FIR. It stores K samples in the delay line and calculates the decimated output as the sum-of-products of the delay line values and the filter coefficients. Then it shifts the delay line by K places for the inputs for the next decimation. Since here the filter computes only one of every K outputs, we save K-1 operations per output.



$H(z)= a_0 + a_1Z^{-1} +….+a_MZ^{-M}$

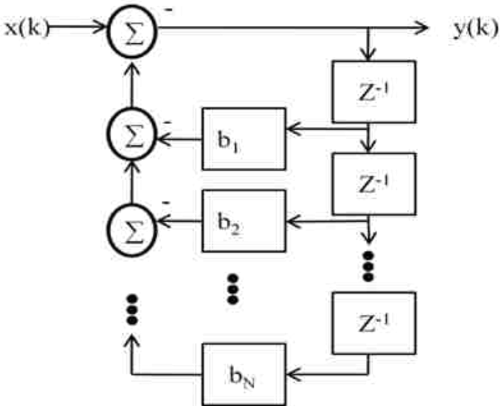
Figure 2.10: Block diagram of a FIR filter.

The other class of filters is known as IIR filters, also known as recursive filters, whose response is given by equation (2.3).

$$y(n)=\sum\nolimits_{k=0}^{M}a_{k}x(n-k)+\sum\nolimits_{k=0}^{N}b_{k}y(n-k) \tag{2.3}$$

In equation (2.2), a_k's are the feedforward filter coefficients, b_k's are the feedback filter coefficients, x(n) is the input signal, y(n) is the output signal and M is the feedforward filter order and N is the feedback filter order. An M^{th} order filter has (M+1) terms. Here the output, y(n) for the IIR filter is dependent on past and present values of both the input and the output. Figure 2.11 shows a block diagram example and derived transfer functions of an IIR filter [8]. IIR filters can use less memory and calculations to achieve a given filter characteristics than a similar FIR filter.

The advantages and disadvantages of these filters are given in Table 2.1 [8]. A serious problem with a sharp FIR filter is its complexity. FIR filters occupy a large chip area due to the multiplier circuits. IIR filters with sharp transition bands suffer from extremely high sensitivities of transfer function poles and are costly to implement [13].



$$H(z)=1/(1+ b_1Z^{-1} +\ldots.+b_NZ^{-N})$$

Figure 2.11: Block diagram of a IIR filter.

Table 2.1: Advantages and disadvantages of FIR and IIR filters.

FIR FILTERS	**IIR FILTERS**
Easy to Design	More Difficult to Design
Always Stable	May be Unstable
Linear Phase Response	Nonlinear Phase Response
Easy to Incorporate Decimation	Cannot Incorporate Decimation
Less Efficient	More Efficient

There is another class of filters known as half-band filters. They are important in applications of multirate signal processing and wavelets. They provide improved efficiency and are also used for decimation in A/D conversion. Half-band filters are used in sampling rate conversion (particularly in cases where the conversion factor is a multiple of two). They are also used as the basic building blocks in multistage systems. They divide the operating frequency range of a discrete-time system into two equal parts. IIR half-band filters provide higher computation speed than the regular FIR and IIR filters. The only disadvantage of this filter is the phase nonlinearity. The advantage of IIR half-band filters is evident for the applications where the computation speeds and low power consumption are of primary concern. Half-band filters are particularly efficient for octave changes in sampling rate [14]. The characteristics of a half-band filter are shown in Figure 2.12.

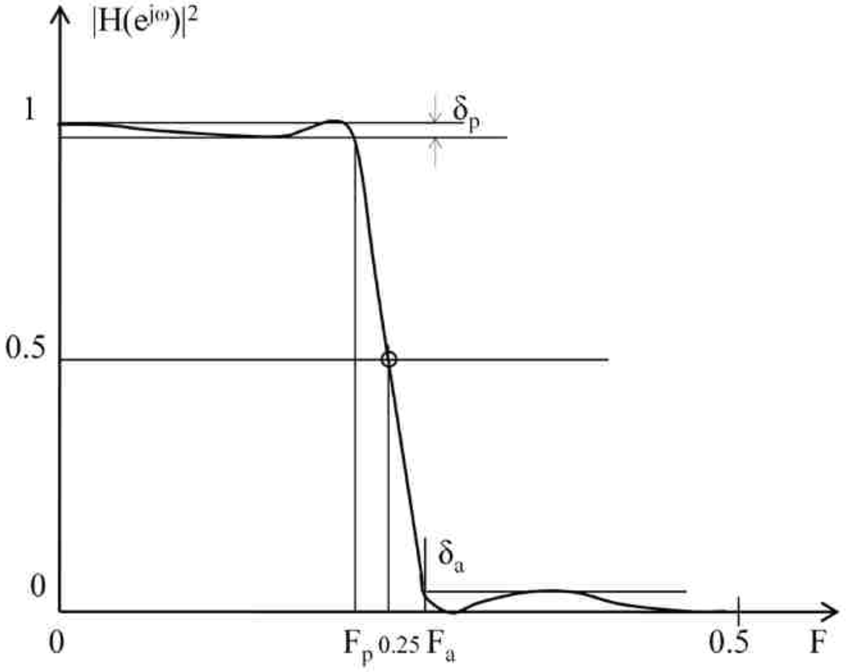


Figure 2.12: Half-band filter characteristics [13].

In the above characteristics the angular digital frequency is designated as θ and $0 \leq \theta \leq 2\pi$. The digital frequency is defined as $f = \theta/2\pi$, $0 \leq f \leq 1$. And also the frequency spectrum is symmetric assuming the coefficients are real. Hence we consider the frequency f only between $0 \leq f \leq 1/2$.

A basic decimator is an averaging circuit. It can also be implemented using the sync filter. One-way of implementation is by using an accumulate and dump circuit. Accumulate and dump circuit is a sync filter and also acts as a digital low pass filter. Better attenuation in the stop band could be achieved by cascading of decimation stages. Accumulate and dump circuit limits the bandwidth specification with cascading more number of stages.

The filters designed using common window techniques as the ones mentioned above also have severe problem when implementing in hardware because of the number of taps the filters required. They might need multipliers and storage elements, which occupy a large area [15].

A special class of filters known as CIC filter is being used widely for decimation. This filter is most economical when implementing in hardware and have better filter characteristics. The following chapters give a detailed description about the CIC filter, its advantages and the design and implementation of a third order CIC filter in 0.5µm CMOS technology.

Chapter 3
Cascaded Integrator Comb Filter Theory and Description

3.1 Cascaded Integrator Comb (CIC) Filter Theory

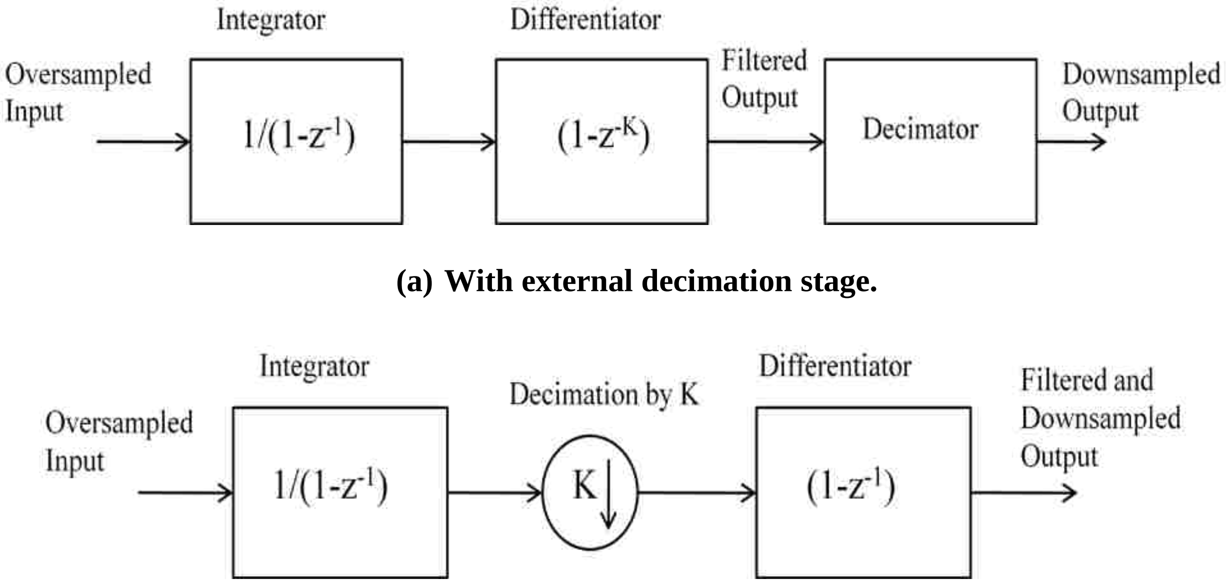
The CIC filter is a multiplier free filter that can handle large rate changes. It was proposed by Eugene Hogenauer in 1981 [16]. It is formed by integrating basic 1-bit integrators and 1-bit differentiators. It uses limited storage as it can be constructed using just adders and delay elements. It is also well suited for FPGA and ASIC implementation due to the same reason. The CIC filter can also be implemented very efficiently in hardware due to its symmetric structure.

This filter is a combination of digital integrator and digital differentiator stages, which can perform the operation of digital low pass filtering and decimation at the same time. The transfer function of the CIC filter in z-domain is given in equation (3.1) [6].

$$H(z)=\left(\frac{1-z^{-K}}{1-z^{-1}}\right)^{L} \qquad (3.1)$$

In equation (3.1), K is the oversampling ratio and L is the order of the filter. The numerator $(1\text{-}z^{-k})^{L}$ represents the transfer function of a differentiator and the denominator $1/(1\text{-}z^{-1})^{L}$ indicates the transfer function of an integrator.

The CIC filter first performs the averaging operation then follows it with the decimation. A simple block diagram of a first order CIC filter is shown in Figure 3.1. In Figure 3.1(a), the differentiator circuit needs K (oversampling ratio) delay elements, which are implemented using registers. The number of delay elements increases as oversampling ratio increases, and as well the number of registers bits that are used to store the data. This type of implementation becomes complex and requires more area as we go for higher order and higher sampling rates. This problem can be overcome by implementing a decimation stage between the integrator and differentiator stages as shown in Figure 3.1(b). Here the clock divider circuit divides the oversampling clock signal by the oversampling ratio, K after the integrator stage. The same output can be achieved by having the decimation stage between integrator stage and comb stage. By dividing the clock frequency by K the delay buffer depth requirement of the comb section is reduced. In Figure 3.1(b), the integrator operates at the sampling clock frequency, f_s while the differentiator operates at down sampled clock frequency of f_s/K. By operating the differentiator at lower frequencies, a reduction in the power consumption is achieved.



(a) With external decimation stage.

(b) Without external decimation stage.

Figure 3.1: Block diagram of first order CIC filter.

3.1.1 First Order Digital Integrator

An integrator is a single-pole IIR filter with a unity feedback coefficient which also acts as an accumulator. The transfer function for an integrator on the z-plane is shown in equation (3.2) [16].

$$H_{I}(z)=\frac{1}{1-z^{-1}} \tag{3.2}$$

The time domain representation of equation (3.2) is shown in equation (3.3). From equation (3.3), it can be seen that the function of the integrator is to add the present input to the past output.

y[n] = y[n - 1] + x[n] (3.3)

In equation (3.3), x[n] is the present input, y[n] is the present output and y[n-1] is the past output.

The integrator derived using equation (3.3) is constructed as shown in Figure 3.2 The delay element is used to delay the output signal by one clock period and can be implemented using a memory element. A simple register can be used to achieve the delay [15]. The magnitude response of the integrator in frequency domain is given by equation (3.4) [17].

$$\left.\begin{aligned} &\left|H_I\left(e^{j\omega}\right)\right|^2 = \frac{1}{2(1-cos\omega)} \\ &ARG\left[H_I\left(e^{j\omega}\right)\right] = -tan^{-1}\left[\frac{sin\omega}{1-cos\omega}\right] \\ &grd\left[H_I\left(e^{j\omega}\right)\right] = \left\{\begin{matrix} undefined & \omega = 0 \\ -\frac{1}{2} & \omega \neq 0 \end{matrix}\right. \end{aligned}\right\} \quad (3.4)$$

In equation (3.4), $H_I\left(e^{j\omega}\right)$ is the magnitude response, $\left|H_I\left(e^{j\omega}\right)\right|^2$ gives the squared magnitude and $ARG\left[H_I\left(e^{j\omega}\right)\right]$ gives the phase of integrator characteristics in frequency domain where ω is the angular frequency. At $\omega = 0,\ \cos\omega = 1$, hence the denominator on the right hand side of the magnitude response equation is zero and hence the magnitude response is undefined at this point.

The magnitude response plot of the integrator is as shown in Figure 3.3. It is basically a low-pass filter with a -20 dB per decade roll off and infinite gain at dc and f_s. A single integrator is unstable due to the single pole at z=1. There is a chance of register overflow and data may be lost. To avoid this problem with register overflow, 2's complement coding scheme is used. By using the 2’s complement number representation, the data will not be lost due to register overflow as long as the register used to store the data is long enough to store the largest word given by $Kx2^N$. Here N is the number of input bits to that particular integrator stage. Internal word width (W) needed to ensure not run time overflow is estimated from equation (3.5) [18].

W = (1Sign bit) + (Number of input bits) + (Number of stages,N)$\log_2$ (Decimator factor)

In this design,

$W = 1 + 1 + 3\log_2(64)$

$W = 20,$ (3.5)

where, N is the input bit resolution. In order to boost the word size to that shown in equation (3.5), the integrator stage is preceded with a coder circuit. A coder circuit could be a simple multiplexer circuit. The coder circuit operation is discussed in detail in Section 3.2 of Chapter 3.

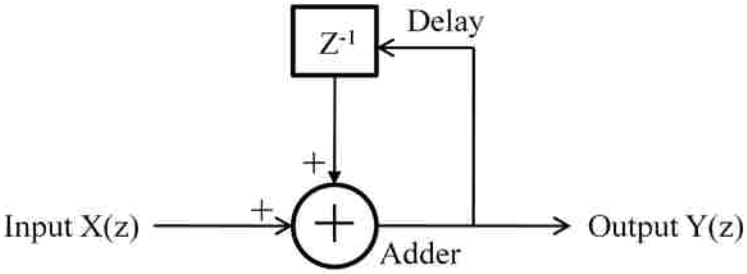


Figure 3.2: Basic integrator.

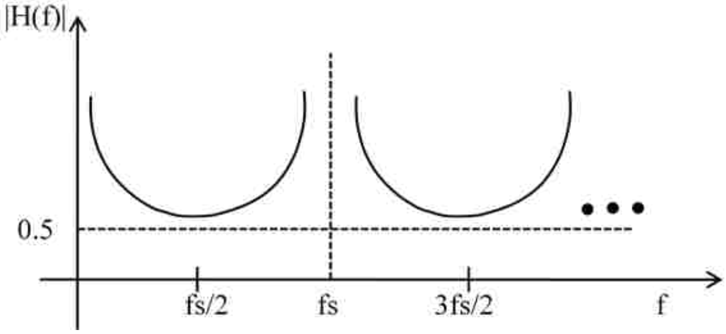


Figure 3.3: Magnitude response of a digital integrator.

3.1.2 First Order Digital Differentiator

The differentiator also known as comb filter is an odd symmetric FIR filter. The transfer function of the differentiator is given as in equation (3.6) [17].

$$H_C(z) = 1 - z^{-KM} \tag{3.6}$$

In equation (3.6), K is the oversampling ratio and M is a design parameter known as the differential delay. M can be any positive integer, but it is usually limited to 1 or 2. The time domain representation of equation (3.6) is given in equation (3.7).

$$y[n] = x[n] – x[n-K] \tag{3.7}$$

In equation (3.7), x[n] is present input, y[n] is present output and x[n-K] is present input delayed by K (oversampling ratio) instants. The function of the differentiator is to obtain the difference between the present input and past input. The magnitude response of the comb in frequency domain is given by equation (3.8) [17].

$$\left.\begin{array}{l} |H_C(e^{j\omega})|^2 = 2(1 - cosKM\omega) \\ ARG[H_C(e^{j\omega})] = -\dfrac{KM\omega}{2} \\ grd[H_C(e^{j\omega})] = \frac{KM}{2} \end{array}\right\} \tag{3.8}$$

In equation (3.8), $H_C(e^{j\omega})$ is the magnitude response, $|H_C(e^{j\omega})|^2$ is the squared magnitude and $ARG[H_C(e^{j\omega})]$ is the phase of differentiator characteristics in frequency domain and ω is the angular frequency. When K=1 and M=1, equation (3.7) becomes equation (3.9).

$$y[n] = x[n] – x[n-1] \tag{3.9}$$

Using equation (3.9), the comb section can be built as shown in Figure 3.4. The magnitude response of the comb is as shown in Figure 3.5. It is basically a high-pass function with a 20 dB per decade gain [17]. The 2’s complement output of the integrator is applied as the input to the differentiator. Hence the differentiator also uses the 2’s complement scheme of coding. So the output at the end of the differentiator will be in 2’s complement form and has to be converted back to binary form.

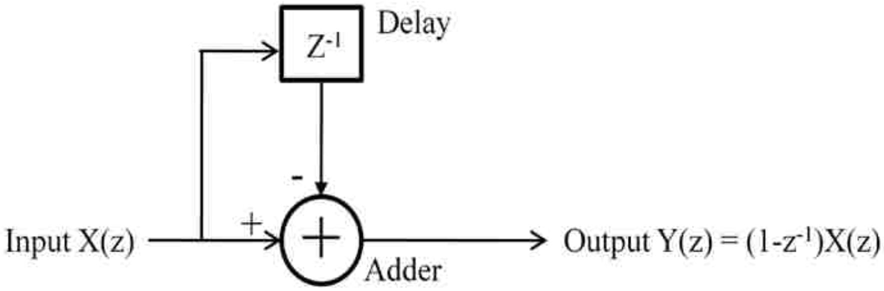


Figure 3.4: Basic differentiator.

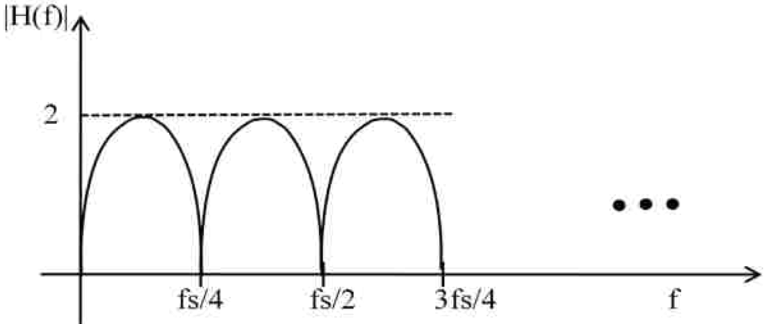


Figure 3.5: Magnitude response of a digital differentiator.

Here, each integrator contributes a pole to the CIC transfer function while each comb section contributes a zero of order K, where K is the frequency decimation ratio. The total response of a CIC filter at frequency, f_s is given by equation (3.10) [19].

$$H(z) = H^{N}{}_{I}(z)H^{N}{}_{C}(z^{K}) = \left(\sum_{K=0}^{KM-1} z^{-K}\right)^{N} \tag{3.10}$$

In equation (3.10), N is the order of CIC filter, M is the differential delay and K is the rate change factor or oversampling ratio. Equation (3.10) implies the equivalent time domain impulse response of a CIC filter. It can be viewed as a cascade of N rectangular pulses. Each rectangular pulse has KM taps. Equation (3.11) gives the magnitude response of a CIC filter at frequency, f where N is the order of the filter.

$$|H(f)| = \left| \frac{Sin(\pi Mf)}{Sin(\frac{\pi f}{K})} \right|^{N} \tag{3.11}$$

Figure 3.6 shows frequency response of the CIC filter obtained using equation (3.11) [19]. The figure shows the aliasing bands $2f_c$ centered around multiples of the low sampling rate. As the number of stages in a CIC filter is increased, the frequency response has a smaller flat pass band [19]. To overcome the magnitude droop, an FIR filter that has a magnitude response that is the inverse of the CIC filter can be applied to achieve frequency response correction. Such filters are called “compensation filters".

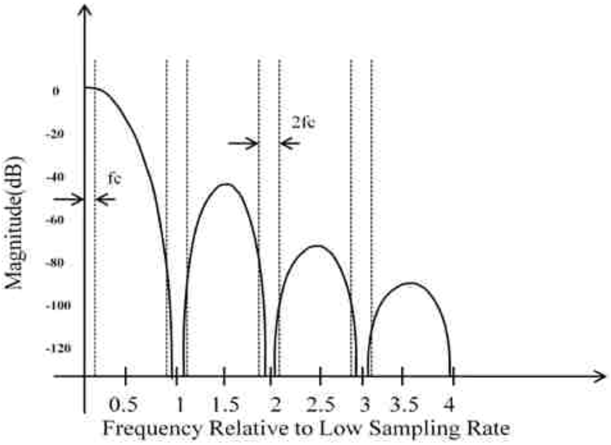


Figure 3.6: Frequency response of a CIC filter.

3.1.3 Multi-Order CIC filter

Higher order CIC is a cascade of digital integrators followed by a cascade of combs (digital differentiators) in equal numbers. The N^{th} order CIC filter is made by integrating N integrators followed by down sampling the signal by a oversampling ratio K followed by N comb filters. To lower the sampling frequency of the combs signal with respect to the sampling frequency of the integrators, a decimator is used between the integrators and the combs.

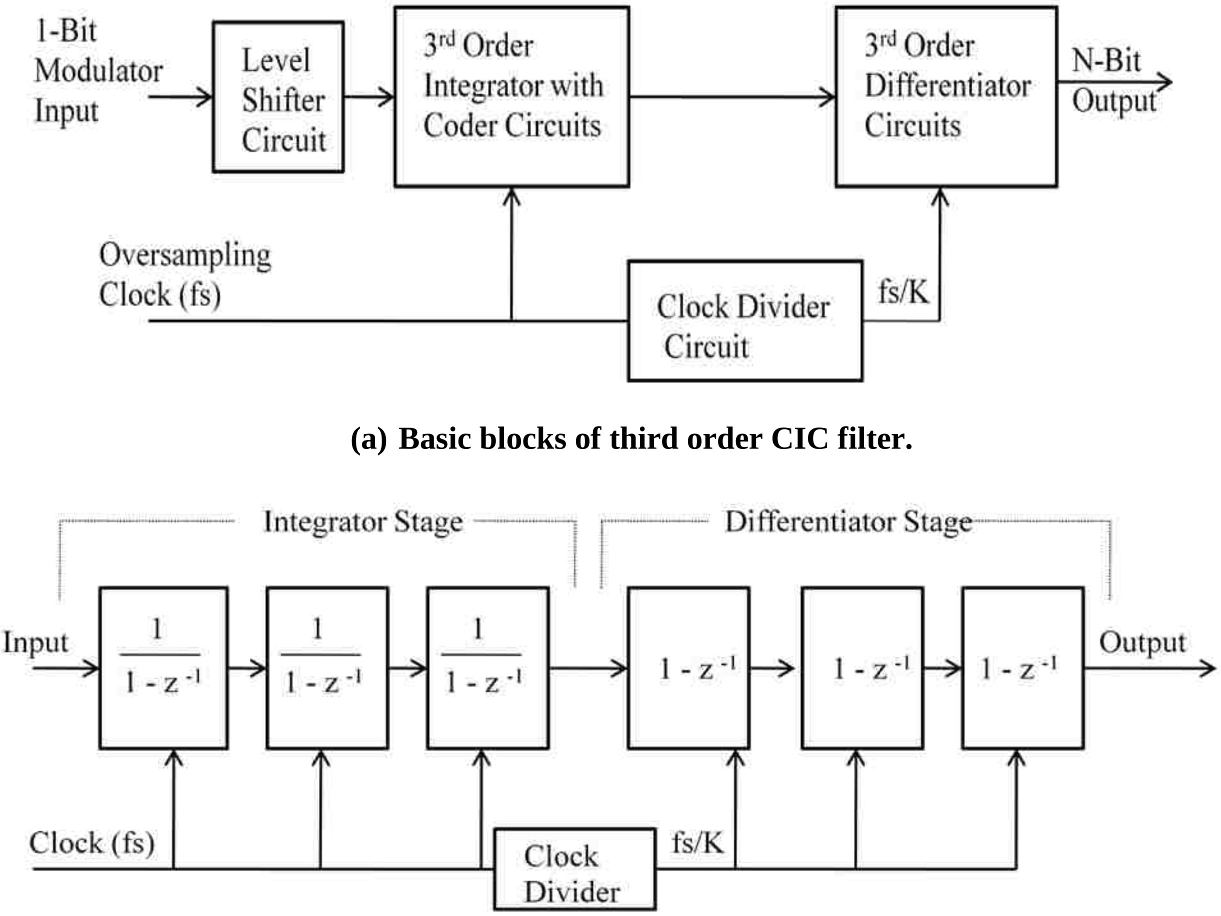
For an N^{th} order CIC filter:

- The filter gain is approximately K^N.
- The transfer function has nulls at each multiple of the output sampling frequency, $f=f_S/K$.
- There are only two control parameters, the number of integrator/comb stages N and the decimation ratio K.
- The transition band is wide.
- Usually accompanied by an anti-aliasing filter or used for narrow-band spectrums.

3.2 Third Order CIC Filter Operation

In the present work, a third order CIC filter has been designed. The basic block diagram of a third order filter is as shown in Figure 3.7 and Figure 3.8. It should be noted that the differentiator operates at a different clock frequency compared to the integrator as explained above. Because of this, both the circuits act as individual blocks and can be used for cascading in order to form a cascaded integrator comb filter. The filter is, in fact, a linear phase moving average filter that provides inherent alias rejection, as the frequencies aliased into the baseband are centred around nulls in the frequency response [20].

In the present work, the input to the decimator, which is the output of the second order sigma-delta modulator, is a 1-bit oversampled binary signal. The decimator shown is for an oversampling ratio set at K=64. The operation of the design is explained by considering K=64. In the differentiator circuit, the difference operation is implemented by performing the summation of first input sequence and the complement of the second input sequence. The coder circuit is responsible for achieving the increase in resolution.



(a) Basic blocks of third order CIC filter.

(b) Blocks with transfer functions for integrators and differentiators.

Figure 3.7: Block diagrams of a third order CIC filter.

As discussed earlier equation (3.5) gives the register size required by each integrator, W. Where, W = (1Sign bit) + (Number of input bits, N) + (Number of stages)log2(K). So a coder circuit is used to encode the input data to the respective resolution. The coder circuit before the integrator stage boosts the 1-bit input to 20-bit input. Hence the integrator 1 works with 20-bit data. The integrators 2 and 3 also work with 20 bits of data. The coder circuit not only increases the resolution but also converts the binary data into 2’s complement data. In order to convert the 1-bit binary data input of the coder circuit to 20-bit 2’s complement form, the representation shown in equation (3.12) is used.

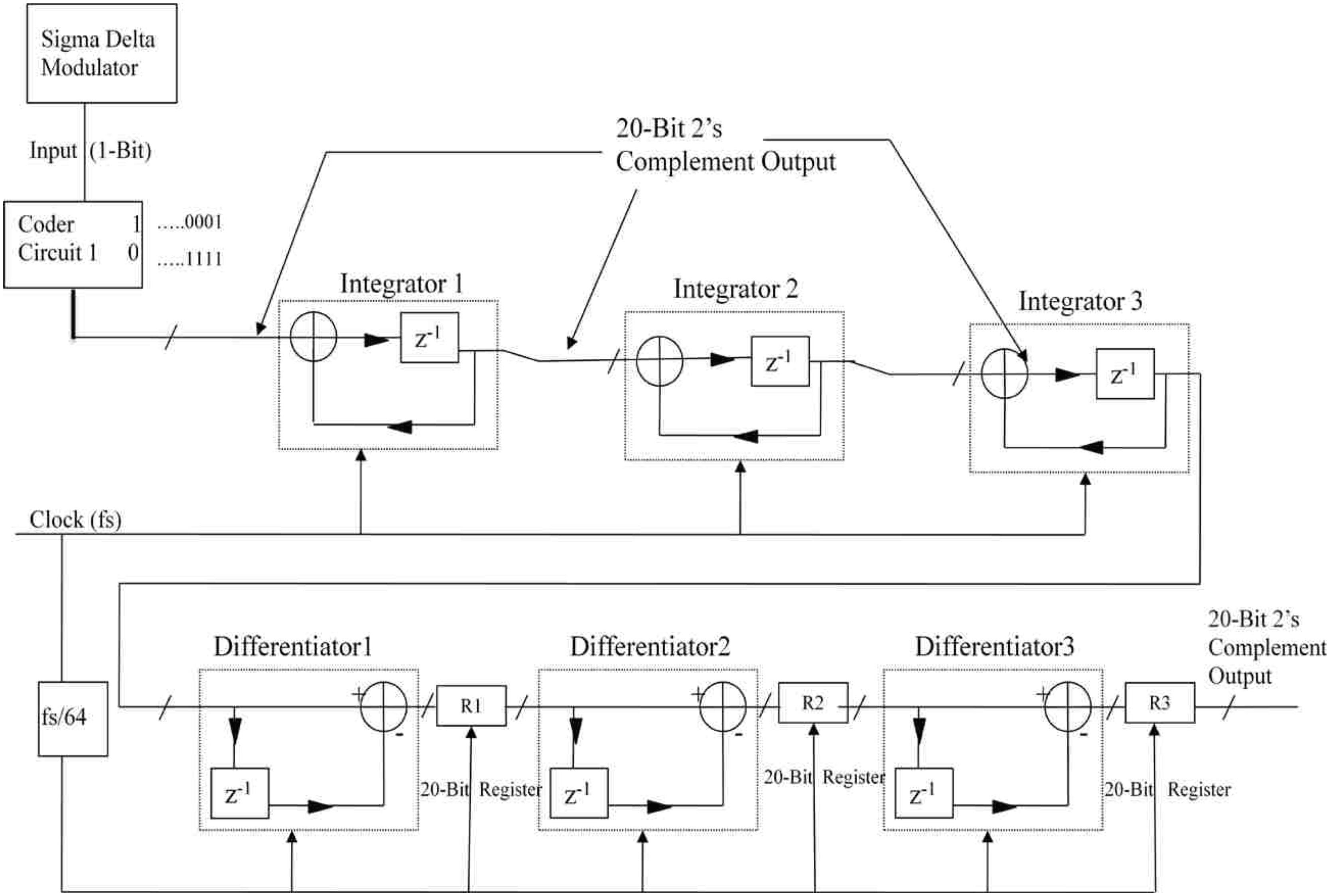


Figure 3.8: Block diagram of a third order CIC filter, K=64.

1 - (20-bit representation) –……00000001 - (+1 in 2'scomplement) (3.12 a)

0 - (20-bit representation) --……1111111 - (-1 in 2's complement) (3.12 b)

Due to the use of 2's complement scheme no data will be lost due to register overflow. The differentiator circuit is operated at a lower frequency compared to that of the integrator stage. This is made possible by using a clock divider circuit, which generates a clock signal at a reduced frequency. The clock divider circuit generates clock for an oversampling ratio of 64. The integrator circuit could cause register overflow but since the differentiator circuit follows the integrator, it performs the difference operation, which eliminates the problem with register overflow [15]. The detailed explanation and hardware implementation of the third order low power decimator circuit is presented in Chapter 4.

3.3 Third Order CIC Decimation Filter Using MATLAB Filter Design Toolbox 4.6

3.3.1 Using SIMULINK

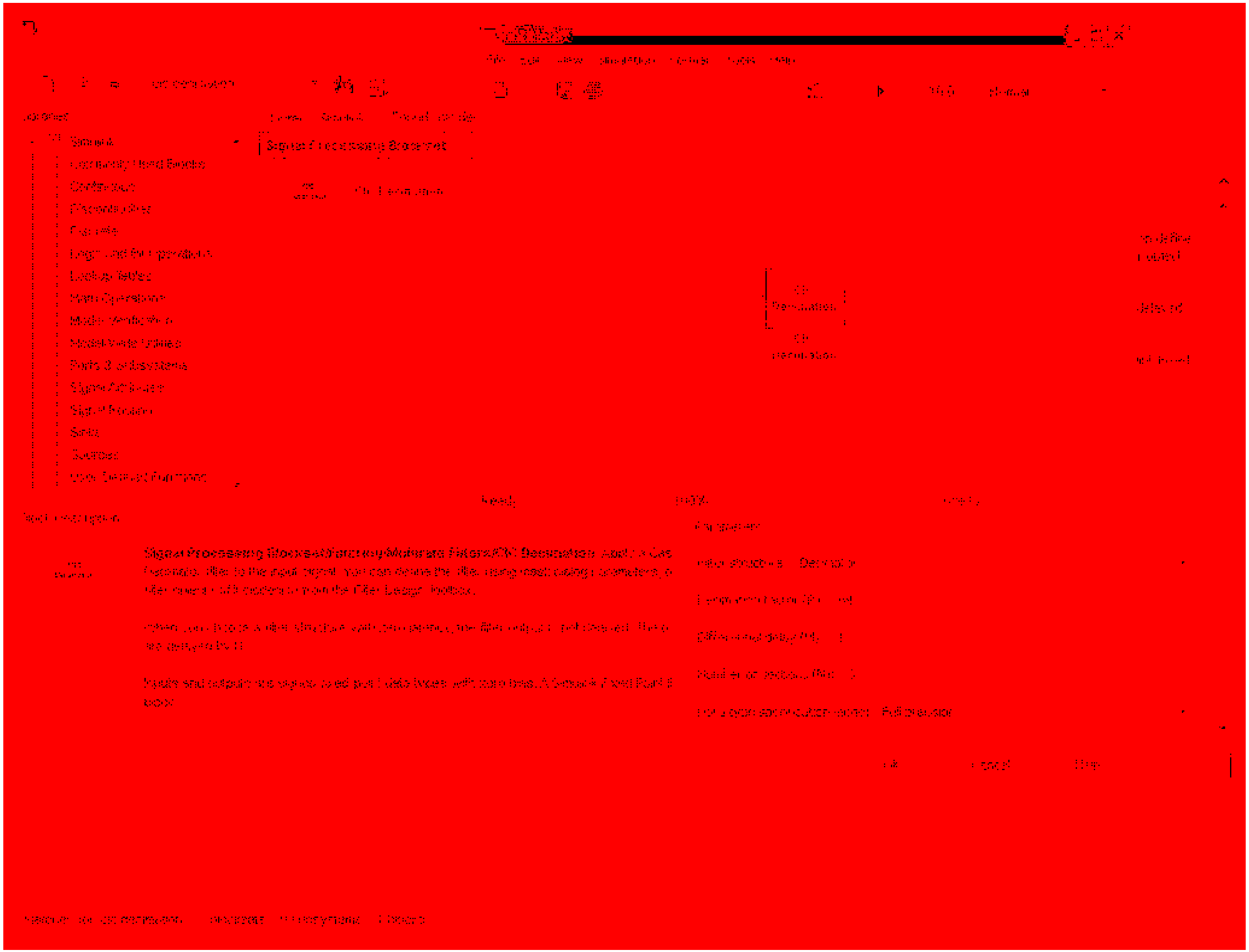
Figure 3.9 shows a screenshot of the simulink tools used to obtain a CIC filter. We choose the filter structure as decimator and specify the other filter parameters for the CIC filter. The data type specification is chosen as full precision mode. The list below gives the description of the parameters that can be controlled. The word length of the CIC filter designed by this method is given by equation 3.13 [21].

$$wordlengt\square = ceil(N \log_2(MR) + I) \quad (3.13)$$

In equation (3.13),

- I = input word length
- M = differential delay
- N = number of sections
- R = decimation factor

Figure 3.10 shows the magnitude response of a CIC filter designed using signal processing block 'CIC-decimation' in simulink. The parameters for the filter designed are R=64, M=1 and N=3.



Signal Processing Blockset/Filtering/Multirate Filters/CIC Decimation: Apply a Cas [illegible] filter to the input signal. You can define the filter using mask dialog parameters, o [illegible] from the Filter Design Toolbox.

When you [illegible] the filter output is [illegible] by [illegible]

Inputs and outputs [illegible] data types [illegible] Fixed-Point [illegible] box.

Filter structure: Decimator

[illegible] factor (R): [illegible]

Differential delay (M): 1

Number of sections (N): 3

[illegible]: Full precision

Figure 3.9: Screen shot from SIMULINK-CIC filter R=64, M=1, N=3.

Similar characteristics can also obtained using mfilt.cicdecim (K,M,N) function,where, K is the oversampling ratio, M is the differential delay and N is the filter order. The MATLAB code to achieve this is given below.

3.3.2 USING "mfilt.cicdecim" in MATLAB

- **To plot magnitude response of CIC filter, the following MATLAB code is used [21]**

```
Hm = mfilt.cicdecim(64,1,3);
hfvt = fvtool(Hm);

set(hfvt,'showreference','off','Color','white');
legend(hfvt, 'CIC Decimator: K=64, M=1, N=3');
```

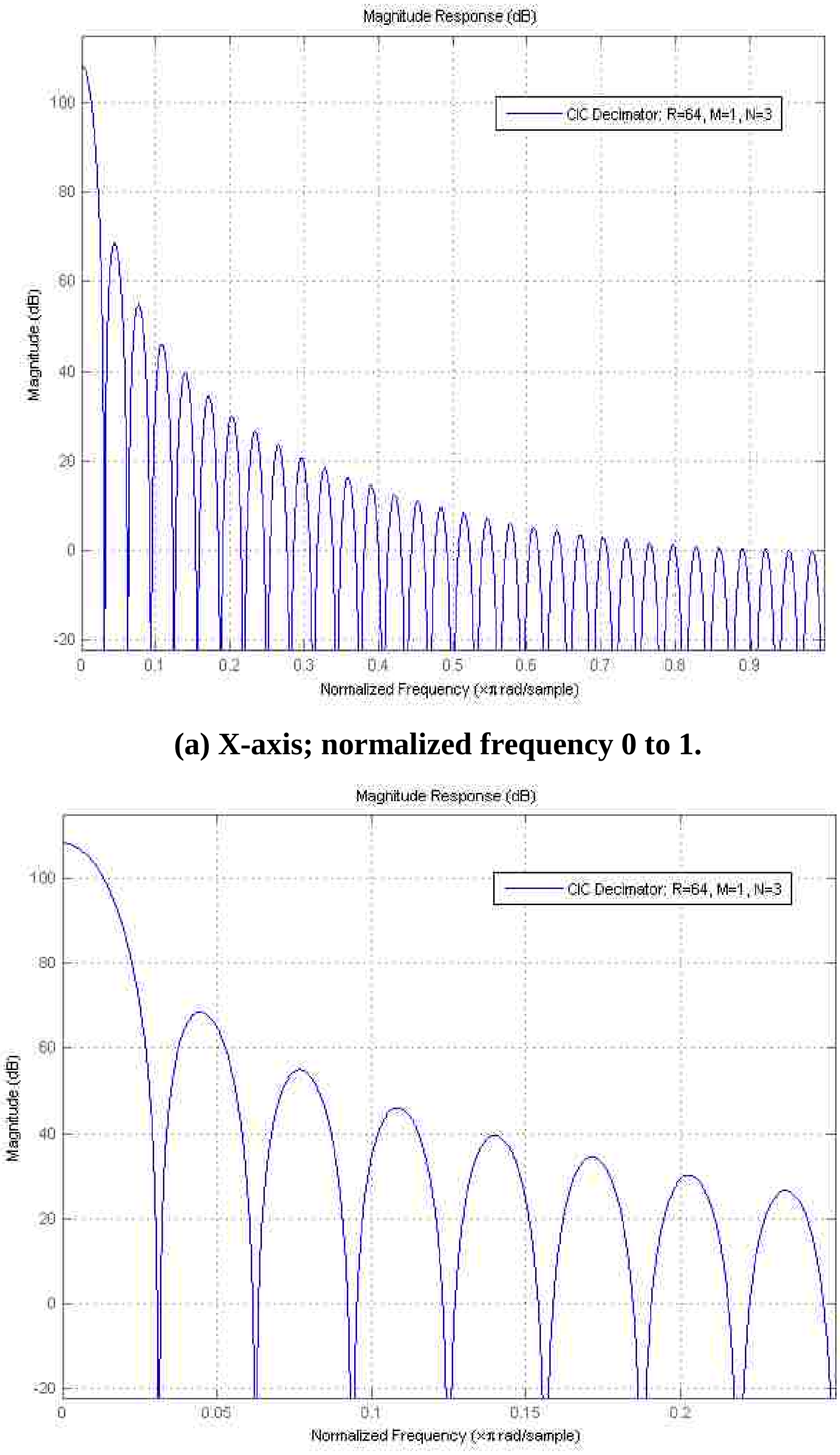
Where K: Oversampling Ratio, M: Differential Delay, N: Filter Order.

Hm gives the magnitude response of the CIC filter with K=64, M=1 and N=3. Figure 3.10 shows the magnitude response, Hm plotted against the normalized frequency.

- **To plot impulse response of CIC filter, the following MATLAB code is used [21]**

```
M = 1;  % Differential delays in the filter.
N = 3;  % Filter sections
K = 64;   % Decimation factor

x = int16(zeros(1280,1)); x(1) = 1;    % Create a 1280-point impulse signal.
Hm = mfilt.cicdecim(K,M,N,20,20); % 20-bit input by default.

y = filter(Hm,x);
stem(double(y));    % Plot output as a stem plot.

xlabel('Samples'); ylabel('Amplitude');

title('Decimated Signal');
```

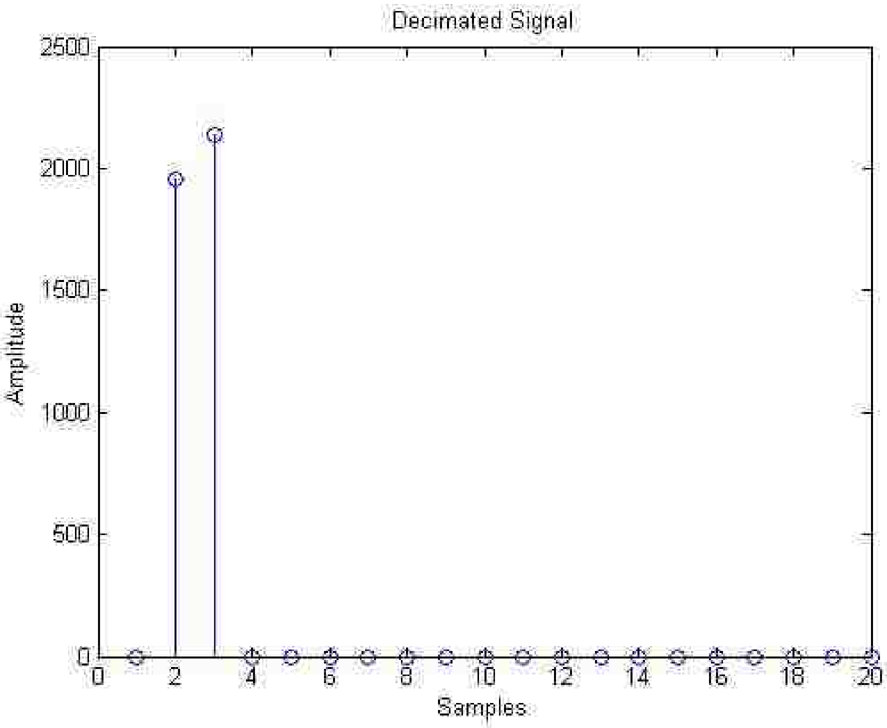
The impulse response of CIC filter is obtained by plotting the magnitude response, Hm against a 1280 point input impulse signal x. Figure 3.11(a) shows the impulse response of the CIC filter with oversampling ratio, K=64, differential delay, M=1 and filter order, N=3. Figure 3.11(b) shows the impulse response for K=64, M=2 and N=3.



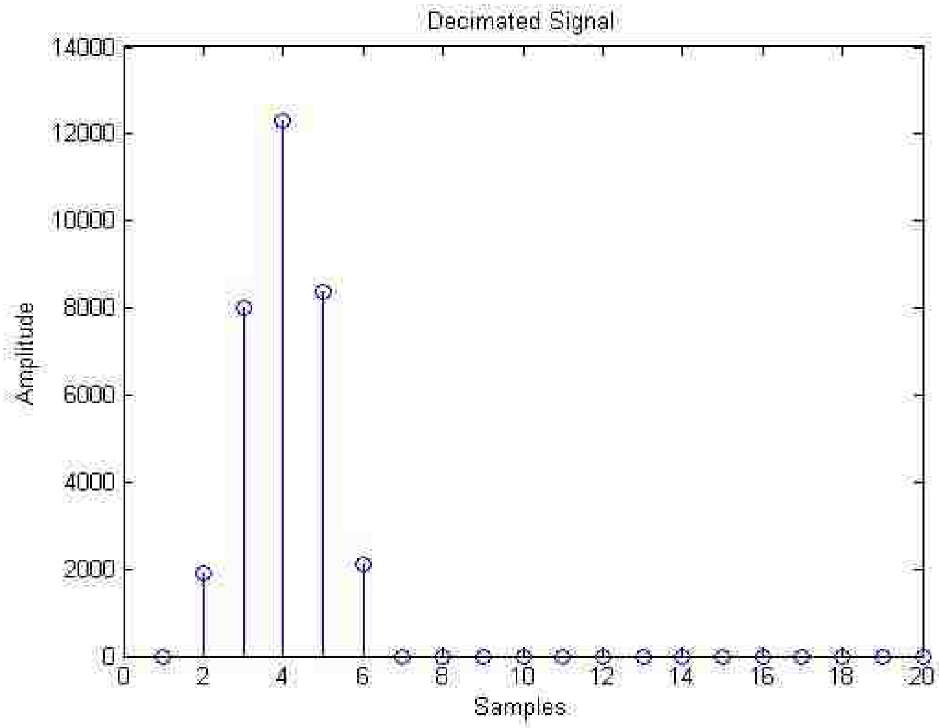
(a) X-axis; normalized frequency 0 to 1.

(b) Expanded X-axis; normalized frequency 0 to 0.25.

Figure 3.10: Magnitude response of a CIC filter R=64, M=1, N=3 using mfilt.cicdecim or signal processing block CIC decimation in simulink.



(a) For differential delay M=1.



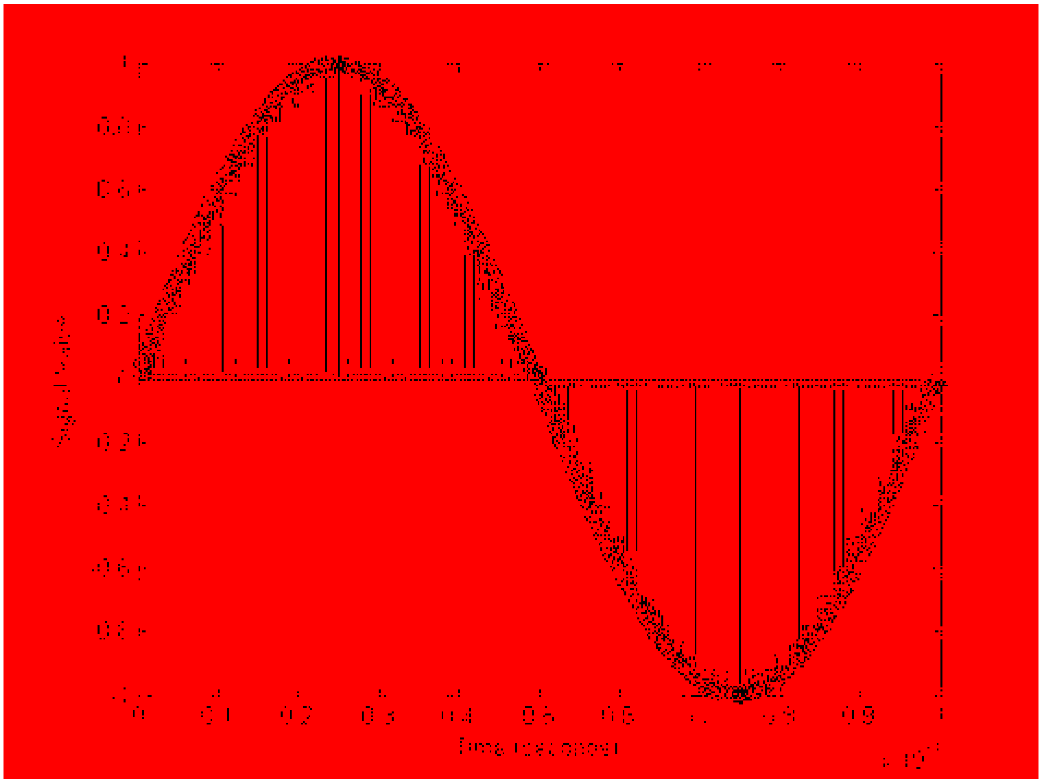
(a) For differential delay M=2.

Figure 3.11: Stem plot of the decimated signal, with 20 samples remaining after decimation for differential delay m=1 and m=2.

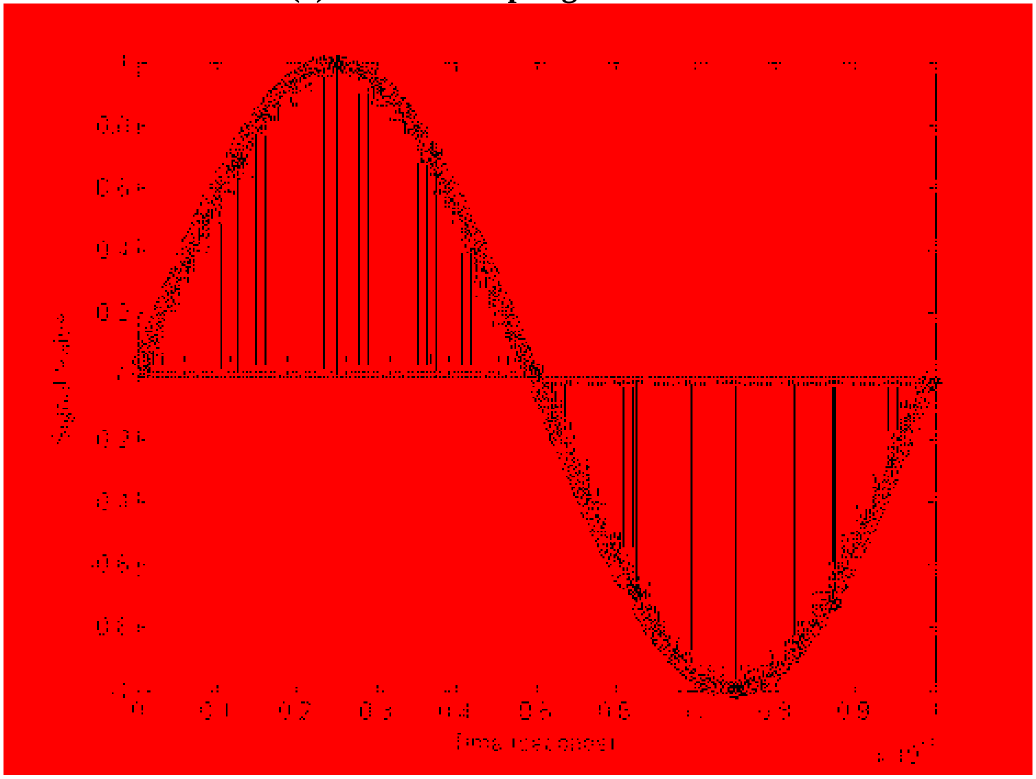
- **To demonstrate the decimation by 64 by a CIC filter, the following MATLAB code is used [21]**

```
K = 64;                 % Decimation factor.

Hm = mfilt.cicdecim(K); % Use default NumberOfSections,

%DifferentialDelay property values.

fs = 256e3;            % Original sampling frequency: 256kHz.

n = 0:10239;            % 10240 samples, 0.232 second long signal.

x  = sin(2*pi*1e3/fs*n);% Original signal, sinusoid at 1kHz.
y_fi = filter(Hm,x); % 5120 samples, still 0.232 seconds.

% Scale the output to overlay the stem plots.

x = double(x);
y = double(y_fi);
y = y/max(abs(y));

stem(n(1:256)/fs,x(2:257));

hold on;  % Plot original signal sampled at 256kHz.

stem(n(2:5)/(fs/K),y(3:6),'K','filled'); % Plot decimated signal (4kHz) in red.

xlabel('Time (seconds)');ylabel('Signal Value');
```

The decimation performed by the CIC filter is demonstrated by using the above code in MATLAB. Hm is the magnitude response of the CIC filter with the given oversampling ratio, K=64. x is the analog input sine wave of amplitude value ranging from -1 to +1 and a frequency of 1 kHz. The decimated output is scaled to get the discrete output values, y. x and y are plotted using stem plot function in MATLAB against time. The stem command plots the sine wave input as discrete values at each time instant instead of a continuous wave. The decimated output occurs only at fixed time instants depending on the oversampling ratio, K. Figure 3.12 shows the input and output waveforms of the analog input, x and discrete digital output, y.



(a) For oversampling ratio K=64.



(a) For oversampling ratio K=32.

Figure 3.12: CIC Decimator input and output for oversampling ratio K=64 and K=32.

3.3.3 USING “fdesign.decimator” in MATLAB

- **To realize the CIC filter using “fdesign.decimator” with a given passband frequency fp and Stopband Attenuation, Ast , the following MATLAB code is used [21]:**

```
M = 1;     % Differential delay
K = 64;    % Decimation factor 256 kHz --> 4 kHz
fs = 256e3;  % Sampling rate 256 kHz

f = fdesign.decimator(K,'CIC',M,'fp,Ast',0.55,55,Fs);
Hm=design(f);
Hm.InputWordLength = 20;

fvtool(Hm); % Plot Magnitude response
realizemdl(Hm);
```

A first order CIC filter can also be realized using the function “fdesign.decimator” in MATLAB. Figure 3.13 shows the block diagram realized using this method. The input parameters like pass-band frequency, fp or stop-band frequency fs and stop-band attenuation, Ast for the filter can be controlled using this function.

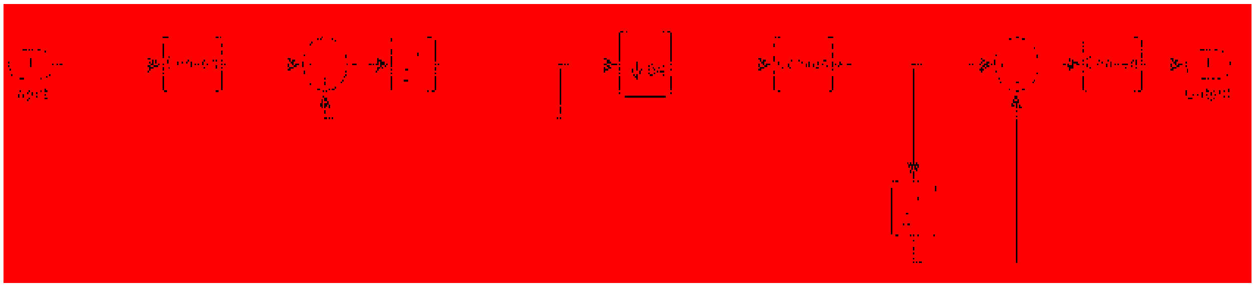


Figure 3.13: CIC filter realized using simulink model.

Chapter 4
Third Order Filter Design

4.1 Blocks of Decimation Filter Design

The block diagram of the decimator designed in this work is shown in Figure 3.8 of Chapter 3. It is also shown here as Figure 4.1 for completeness. The input to the decimator is a 1-bit pulse density modulated signal from a second order sigma-delta ADC modulator .The output from the decimator is a 20 bit digital output. The hardware blocks were designed to be fabricated in 0.5 μm n-well CMOS technology. The hardware required to build each block was designed using LEdit Version 13.0 layout editor and the net-lists extracted from layout have been simulated using PSpice AD. The following sections give a brief description of the individual circuits designed, their layouts and simulation results which are later integrated to make a complete third order CIC filter.

To start with an inverter is designed with minimum widths as shown in Figure 4.2. The layout of the inverter is shown in Figure 4.3. The W/L ratio of the PMOS to NMOS transistor has been taken as 2:1 to take into account the mobility variation. Hence the physical width of the NMOS is 1.5μm and that of PMOS is 3μm. The transfer characteristics of the designed inverter is plotted as shown in Figure 4.4. It can be observed from the plot that the switching point of inverter is at 2.5V. Also, Figure 4.5 shows the simulated transient response output of the designed inverter.

4.1.1 Level Shifter Circuit

A level shifter circuit acts as an interface between the already designed modulator and the digital decimator and is used at the input end of the decimator. The available modulator operates within a voltage range of ± 2.5V where as the decimator is designed to work in the 0V to 5V voltage range. To take into account the different voltage ranges, it becomes necessary to have a level shifter circuit at the input stage of the decimator. This level shifter shifts the modulator output from -2.5V to +2.5V voltage range to 0V to 5V voltage range, that can be used for the CIC filter. The level shifter circuit shown in Figure 4.6 has been designed and integrated for this purpose. The design used is a simple buffer circuit with the widths of the transistors adjusted to get the required level shift [15].

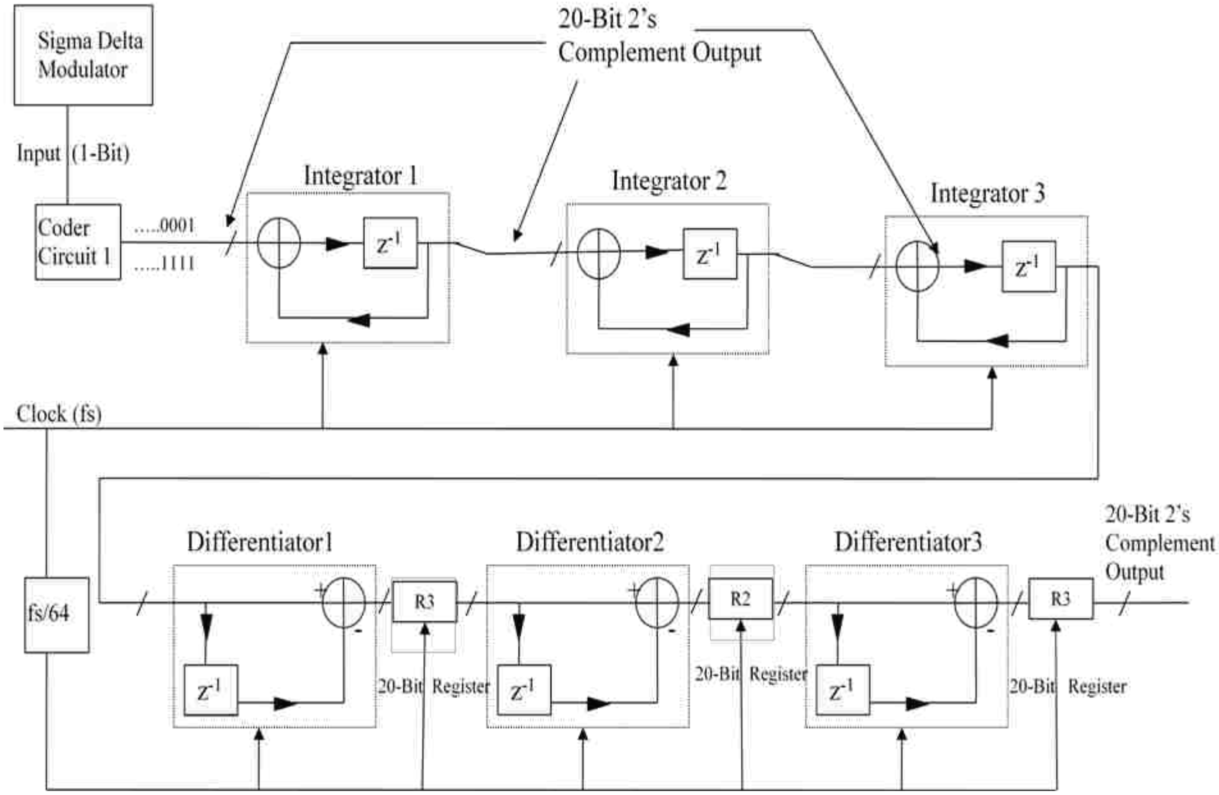


Figure 4.1: Block diagram of a third order CIC filter, K=64.

The input inverter has been modeled such that the output of that inverter is 0V when the input is + 2.5V and the output is 5V when the input is – 2.5V. The first inverter output is given as the input to the second inverter, so whenever the input to the level shifter circuit is + 2.5V the second inverter output is 5V and whenever the input is – 2.5V the second inverter output is 0V..

To achieve the above condition, the W/L ratio of the NMOS transistor, M1 has been increased such that the output goes to 0V when the input is +2.5V (Under normal condition when the W/L ratio of the NMOS transistor, M1 is 1.5µm/0.6µm and the input is +2.5V, the output goes to an intermediate value between 0V and 5V). By increasing the W/L ratio of M1 the pull down strength of the NMOS transistor is increased. When the input voltage is -2.5V, NMOS transistor is OFF and PMOS transistor turns ON strongly pulling up the output node to 5V.

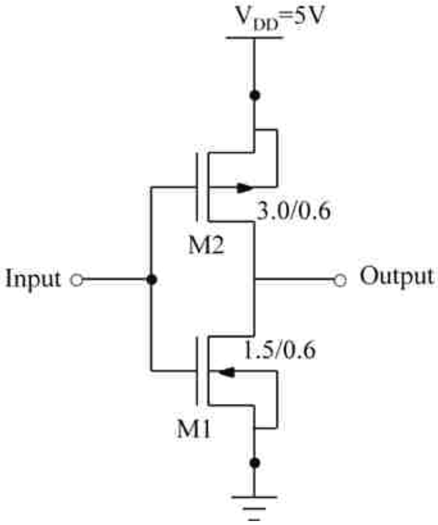


Figure 4.2: Inverter schematic.

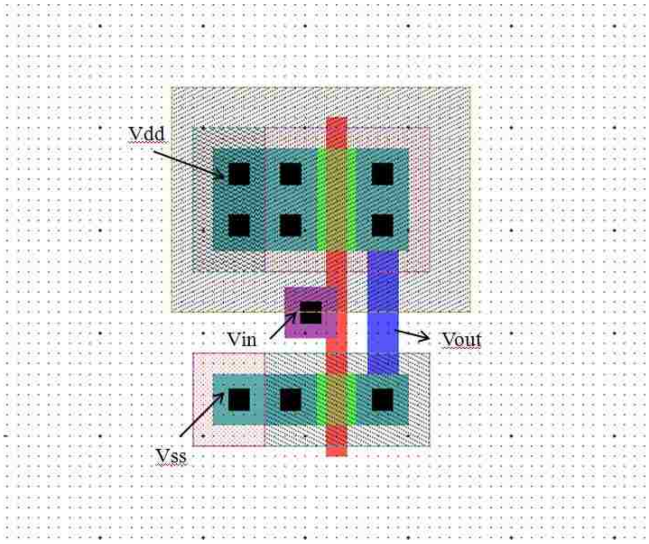


Figure 4.3: Layout of an inverter.

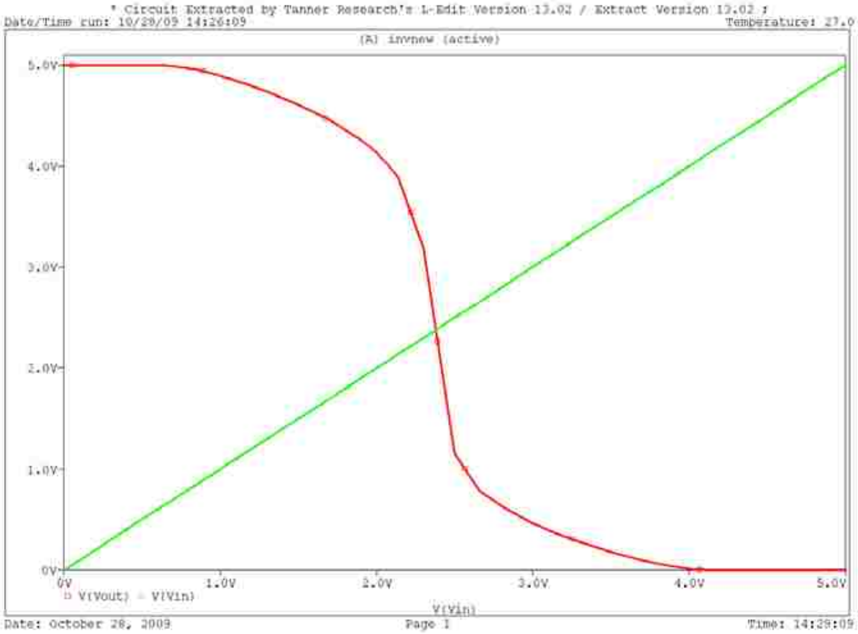


Figure 4.4: Inverter transfer characteristics.

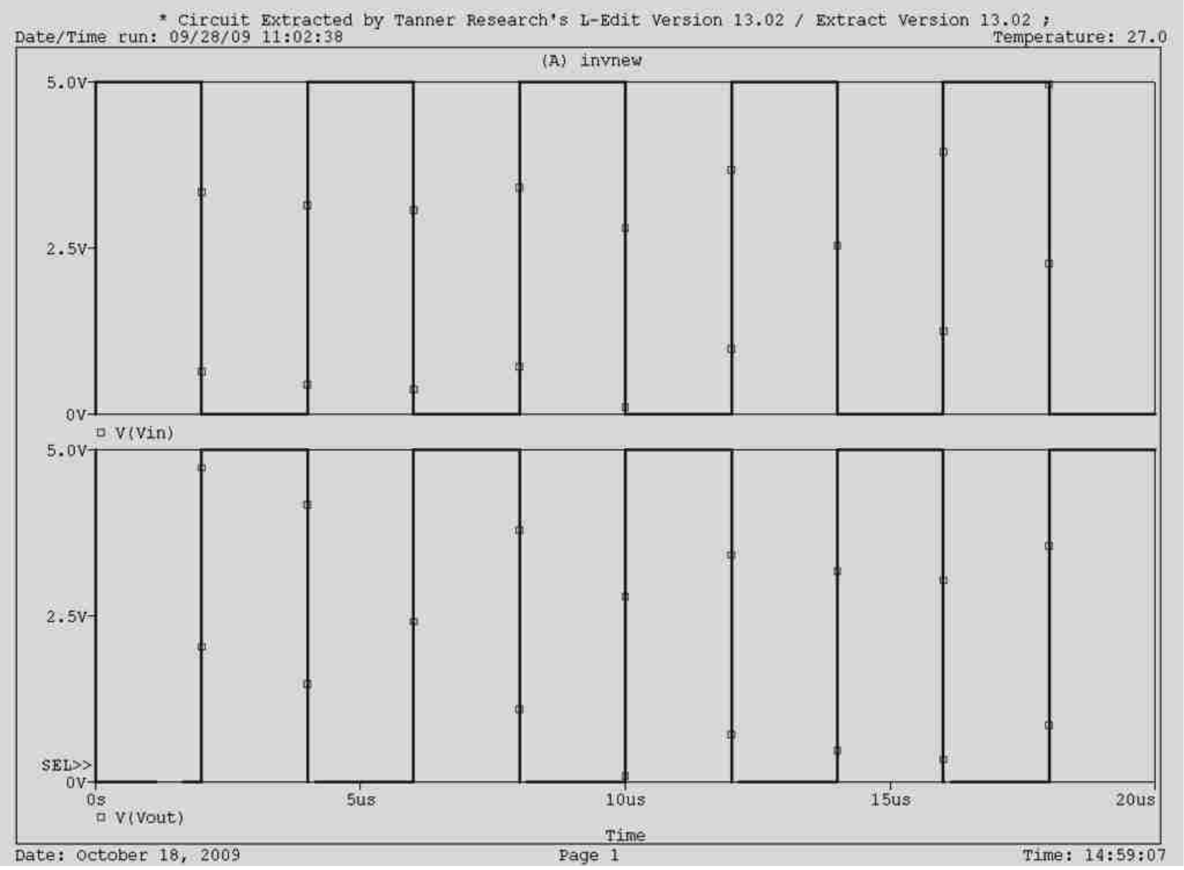


Figure 4.5: PSPICE simulated input and output of inverter.

Figure 4.7 shows the layout of the level shifter circuit designed and and Figure 4.8 shows the SPICE simulation results. From the layout, it can be observed that the W/L ratio of the NMOS transistor, M1 is comparatively large with respect to the other transistors. From the simulation results, the circuit achieves the output voltage range of 0V-5V for a ± 2.5V input.

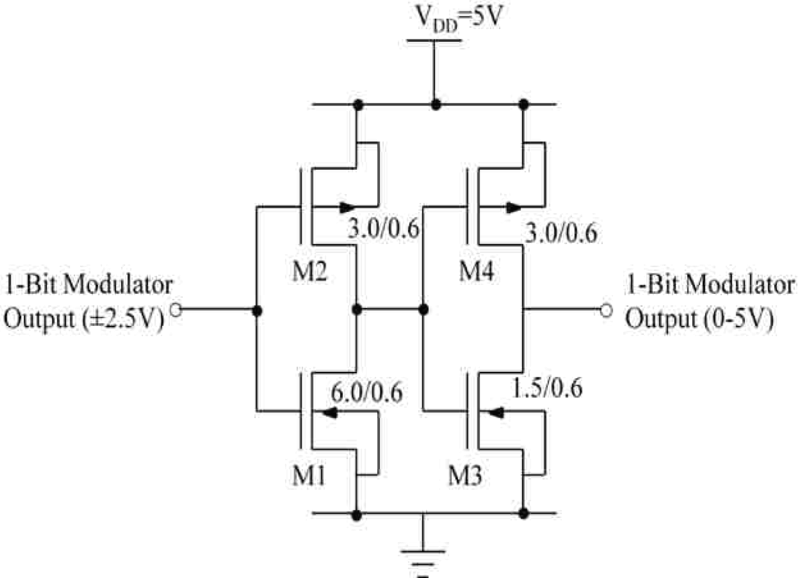


Figure 4.6: Schematic of the level shifter circuit.

4.1.2 Clock Divider Circuit

The function of the clock divider circuit is to divide the clock frequency by the oversampling ratio, K=64. The input to the clock divider circuit is the oversampling clock, f_s, which is also used as the clock for the modulator. Hence, the output of the clock divider is f_s/64 and is applied to the differentiator circuit of the CIC filter.

The clock divider circuit is designed using negative edge triggered T-flip flops and AND logic gates. A T-flip flop or "toggle" flip flop changes its output on each negative clock edge. The output frequency of T-flip flop is half the frequency of the input clock signal. Hence, a single T-flip flop acts as a divide-by-two counter as two active transitions of the clock signal generate one active transition at the output. T flip-flops can be connected sequentially to form a "divide by N" counter, where N is usually a power of 2. These T-flip flops can be designed using a master-slave JK or a master slave D-flip flop. The design of T-flip flop using JK-flip flop occupies more area than the one using D-flip flop. Hence, in the present design, a D-flip flop was used to design a T-flip flop. To get the T-flip flop the output Q' of D-flip flop is connected back to the D input as shown in Figure 4.9. Figure 4.10 shows the schematic of an edge triggered D-flip flop using NAND gates.

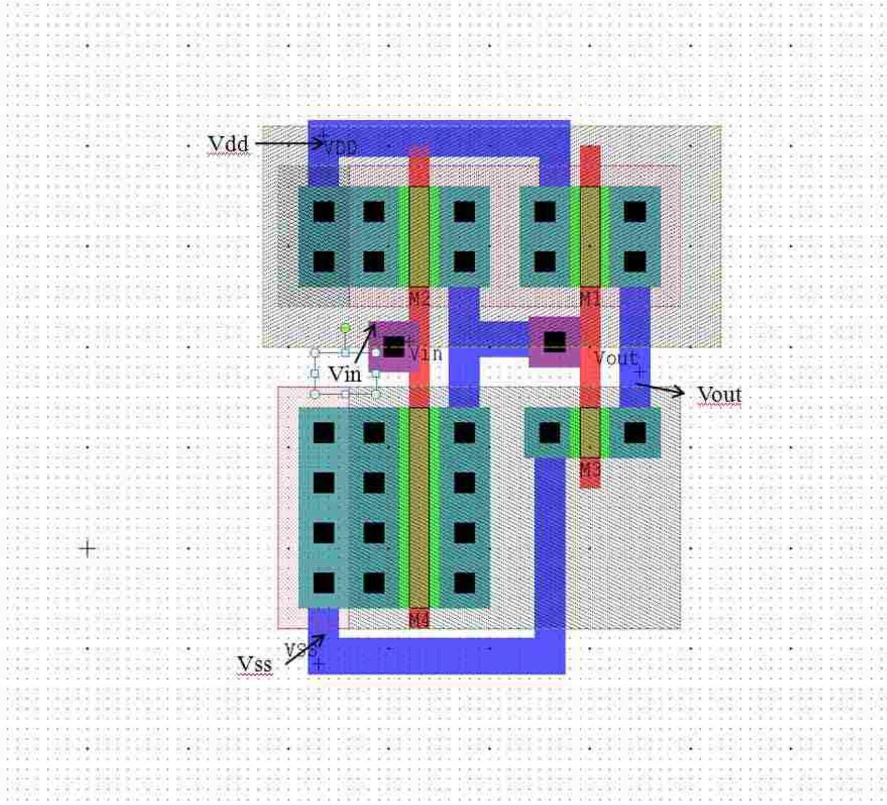


Figure 4.7: Layout of the level shifter circuit.

The output of a single stage T-flip flop for a clock input of frequency, f is a clock of frequency f/2. By cascading more number of T-flip flop stages; the clock frequency can be reduced in multiples of 1/2. Thus, N-stage cascaded T-flip flop provides with a frequency division by a value of 2^N. The final output of the cascaded design will have the same duty cycle as that of the input clock. In the present application, the differentiator circuit has to be operated with a clock such that the difference between two pulses should be K x T_s, where T_s is the time period of the oversampling clock signal. A representative output of the clock divider circuit that has to be applied to the differentiator circuit is shown in Figure 4.11 and illustrates the frequency division by K. The output has an ON time same as that of the oversampling clock ON time of $T_s/2$ sec. The output pulses are separated by K samples i.e., in between two pulses there exists K samples since each sample with a time period of T_s.

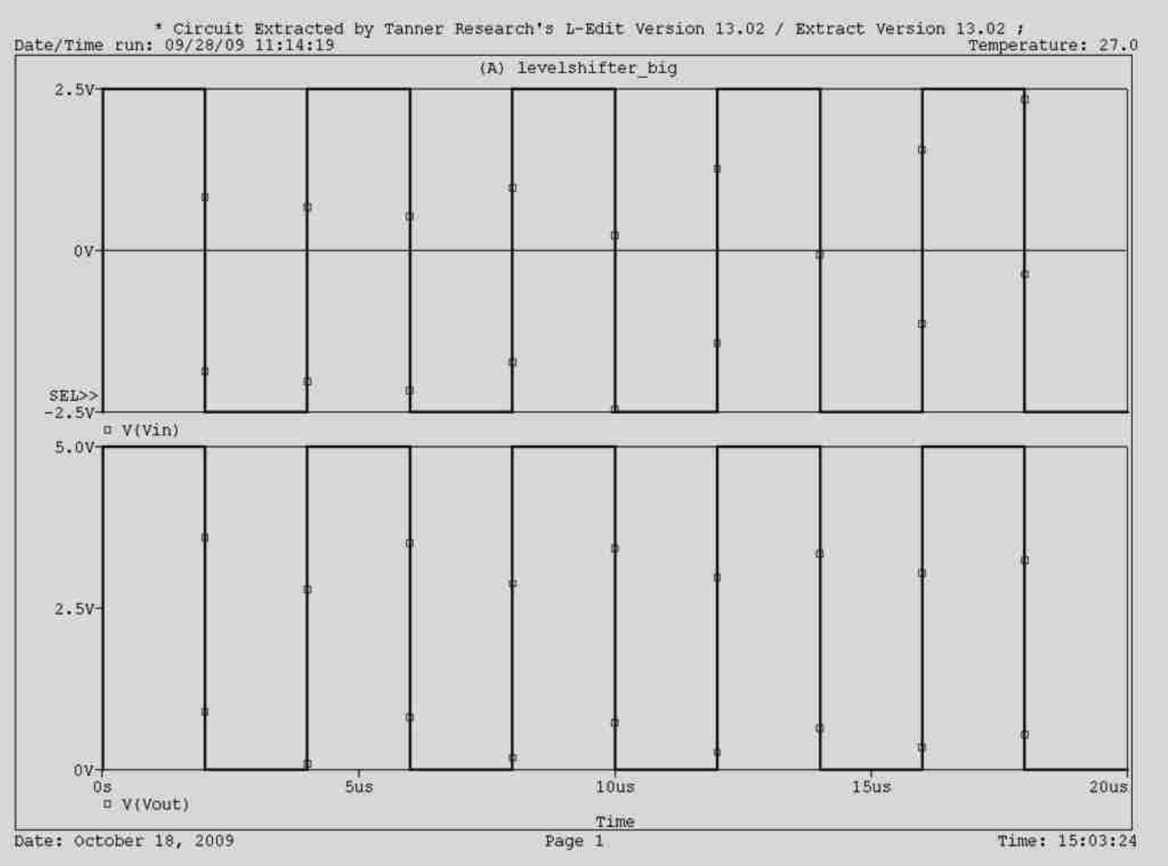


Figure 4.8: PSPICE simulated input and output of level shifter.

To generate the required clock output for the differentiator of the CIC filter, the circuit shown in Figure 4.12 is designed. It shows the gate level schematic of the clock divider circuit that is used to generate output pulses with a clock frequency division by 64. As 64 $=2^6$, N=6, we need 6-stage T-flip flops to achieve a frequency division by 64. Whenever the input and output of a T-flip flop are given as inputs to an AND gate, only the ON time of the input clock is transmitted to the output. The output of the AND gate remains at logic ‘1’ during this ON time only.

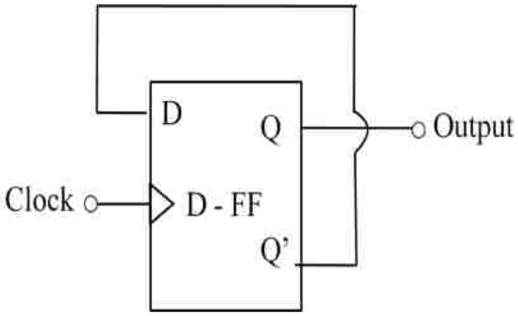


Figure 4.9: T-flip flop from D-flip flop.

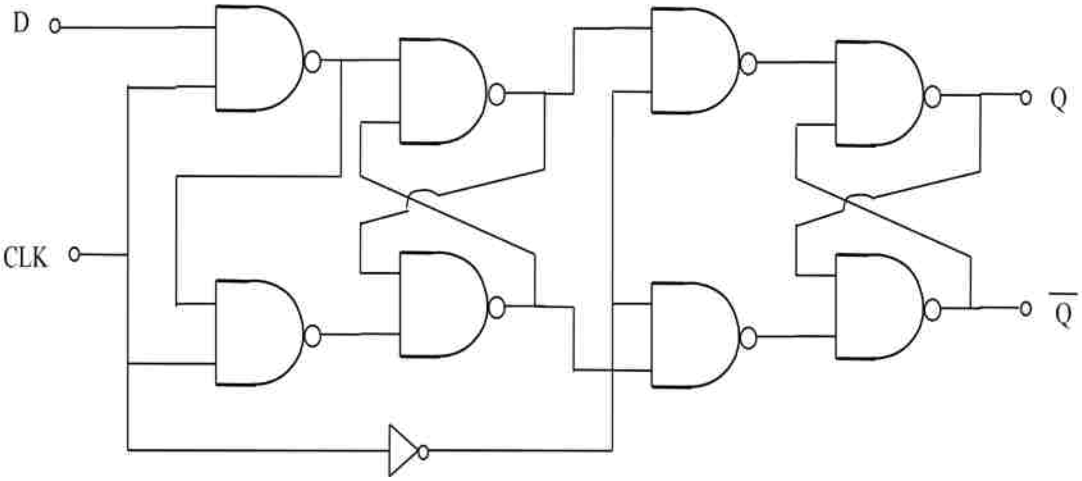


Figure 4.10: Schematic of the edge-triggered D- NAND flip-flop.

From Figure 4.12 it can be seen that for the divide by 64 case, the output of the AND 2 and the clock input is given as inputs to AND 1. The output of AND 6 is ON when outputs from T-FF 6 and T-FF 5 are ON. The output of AND 5 is ON when outputs from T-FF 6, T-FF 5 and T-FF 4 are ON. This is continued. The output of AND 1 is at logic 1 during the ON time of the clock input and T-FF 6. The operation is continued and the final output obtained from the AND 1 is a pulse waveform with the pulses separated by 64 x T_s.

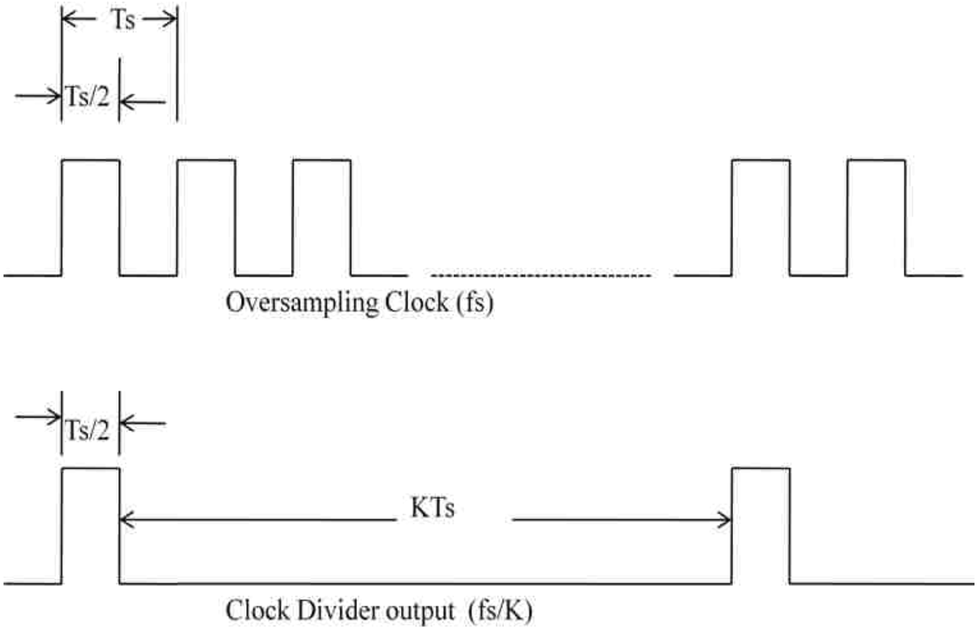


Figure 4.11: Representation of the output of the clock divider circuit.

The layout of the complete clock divider circuit for generating the divide by 64 clock output is as shown in Figure 4.13. The W/L ratios of the NMOS and PMOS used in the T-flip flop design are of values 3.0 µm/0.6 µm and 1.5 µm /0.6 µm, respectively. The AND gates are also designed using the same W/L ratios. Multiple buffers with high W/L ratios have been added to the output of the clock divider circuit at all the clock inputs for each differentiator stage to take care of the fan-out problems and to achieve sharp rise and fall times in the clock output waveform.

The detailed operation of the clock divider circuit for the divide by 64 case is clearly shown in Figure 4.14 (a), (b) and (c). The output of clock divider is a sequence of pulses that are separated by 64 samples, i.e., the time difference between two consecutive samples is 64 x T_s, where T_s is the time period of the input clock. The output pulse ON time is same as the pulse ON time of the input oversampling clock (The ON time of a pulse waveform is the time period when the pulse remains high or '1').

The simulated output for the clock frequency divider by the oversampling ratio of K=64 is taken from the output of AND1 logic gate in the gate level schematic of the clock divider circuit. Figure 4.14(c) shows the simulated results of the clock divider for the input clock whose time period is chosen to be 4μs. The obtained output pulses are separated by 256μs (64 x 4μs) illustrating clock frequency division by a factor 64.

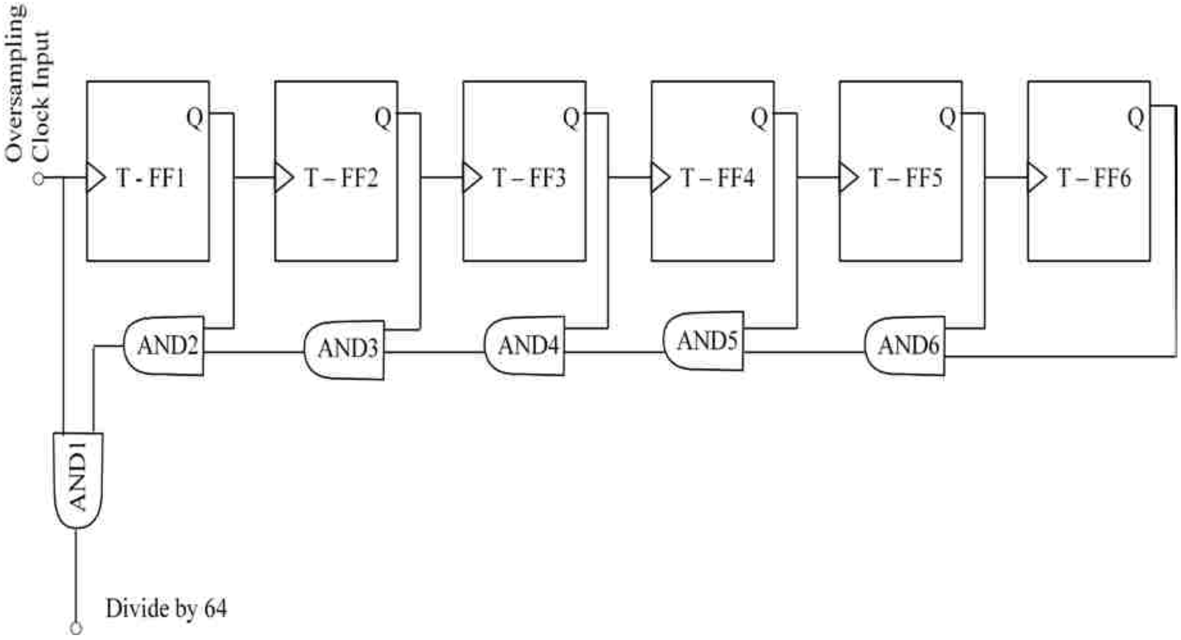


Figure 4.12: Gate level schematic of the clock divider circuit.

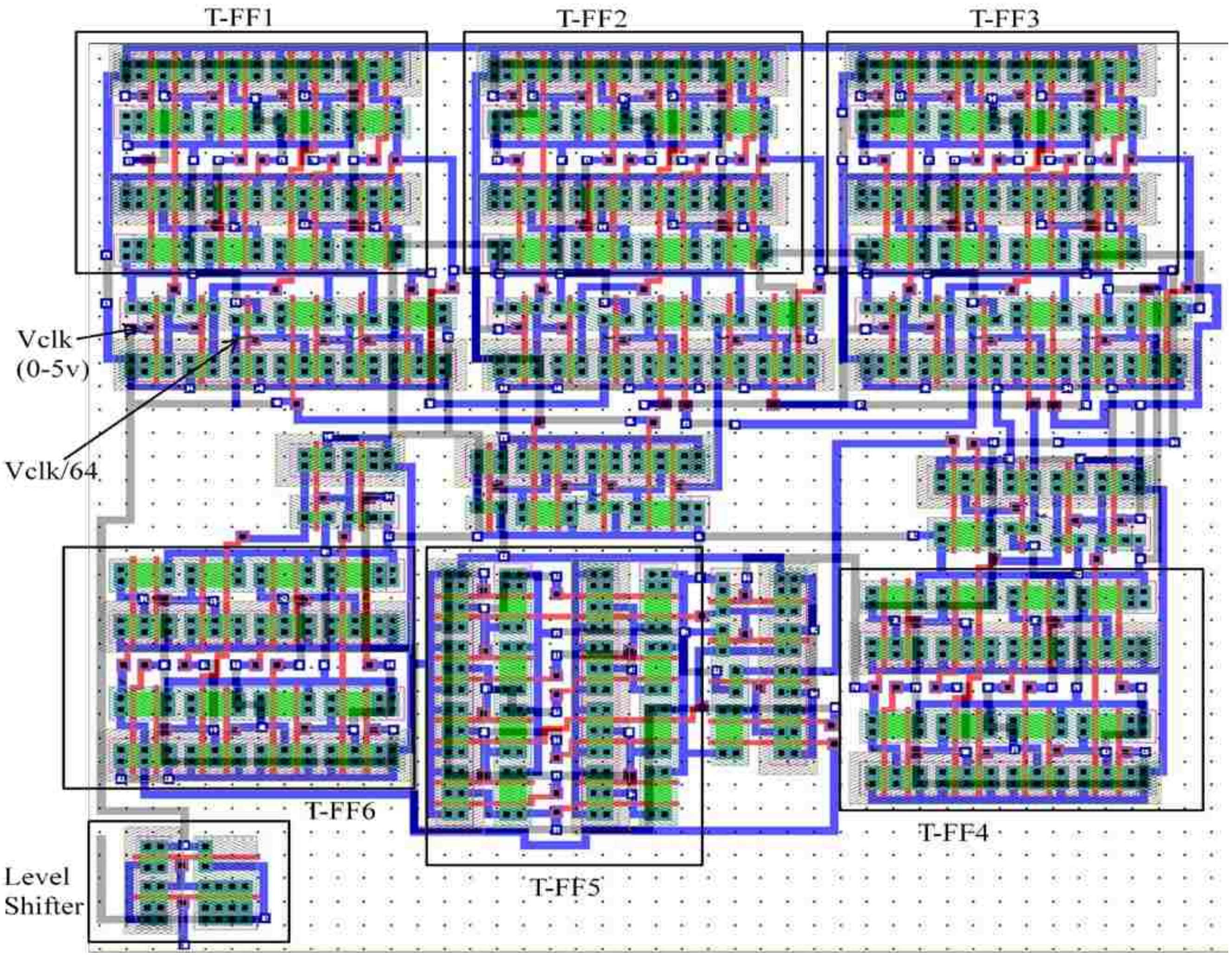
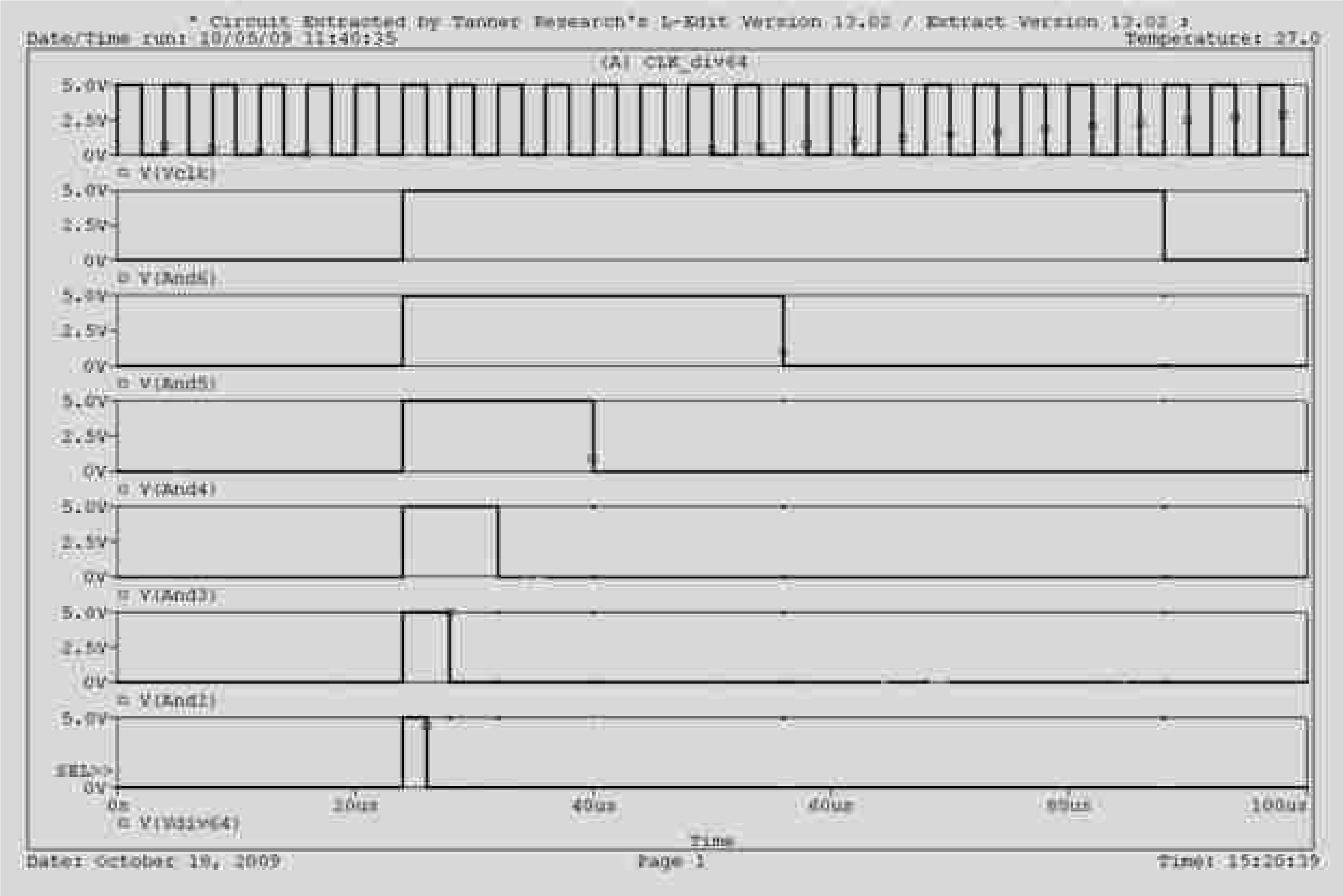


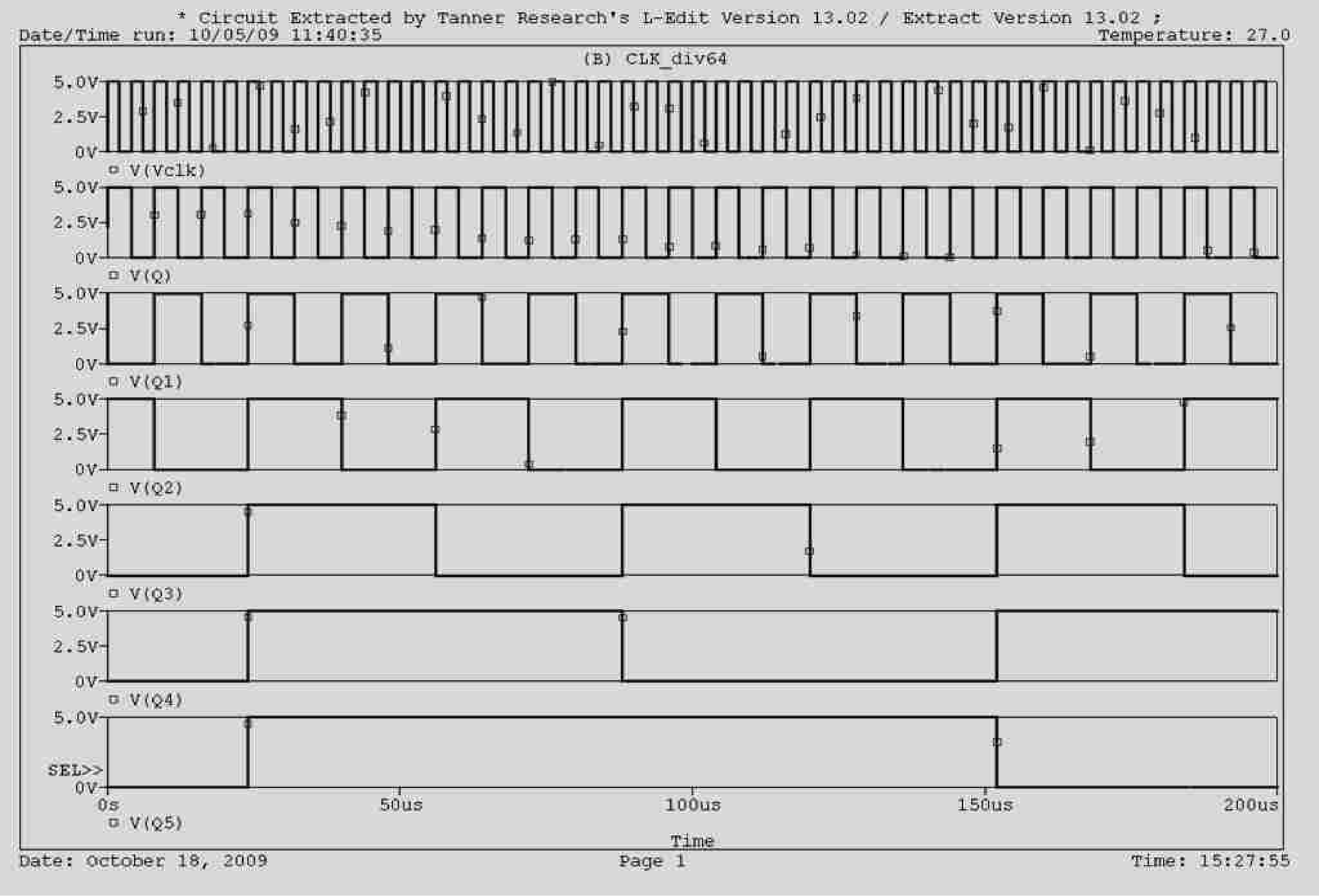
Figure 4.13: Layout of a clock divider circuit.



(a)The output waveforms from gates AND 6 to AND 1.

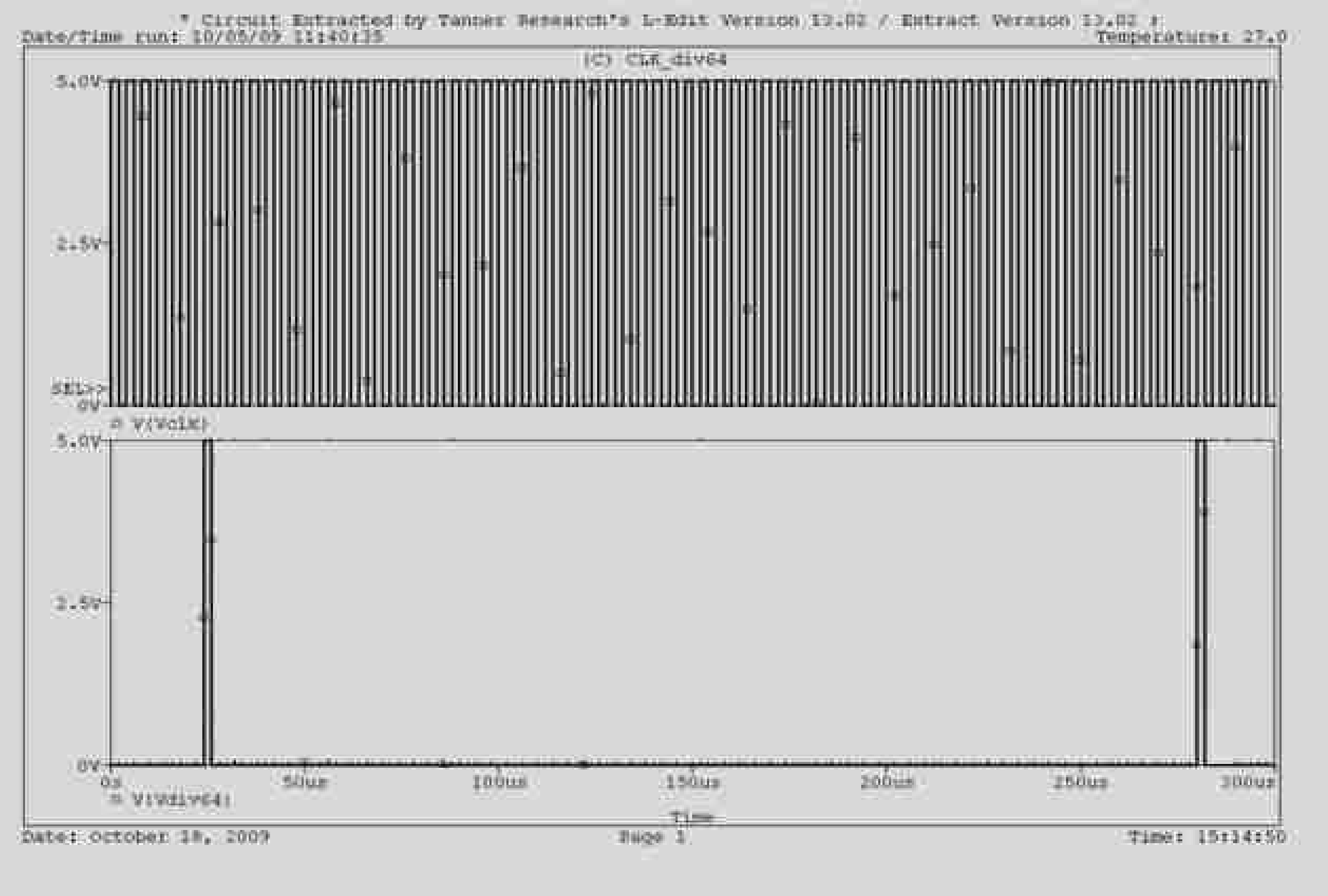
Figure 4.14: Simulated outputs of the clock divider circuit for divide by 64 case, the time period of the input clock is 4µs.

(Figure 4.14 continued)



(b)The output waveforms from T-FF1 to T-FF6.

(Figure 4.14 continued)



(c) Oversampling clock input and divide by 64 output.

4.1.3 Adder Circuit

As discussed in Chapter 3, both the 1-bit integrator and the 1-bit differentiator need a full adder circuit to perform their respective operations. An adder is a digital circuit that performs addition of numbers. A half adder circuit adds two bits and gives the outputs sum and carry. The sum is the XOR of two input bits and carry is the product of the two bits. A full adder adds three binary bits and produces sum and carry output. The truth table operation for a binary full adder is shown in Table 4.1. The table contains two input bits A and B. C_{in} is the carry from the earlier stage and the outputs of the adder are the sum bit, S and the carry bit, C_{out}. The Sum and C_{out} bits shown in the truth table are defined using equation (4.1). It can be observed from the truth table that a full adder can be constructed with two half adders. Since a full adder has both carry in and carry out capabilities, it is highly scalable and is used in many cascade circuit implementations, which add 8, 16, 32, etc. binary numbers. The carry input for the full-adder circuit is from the carry output from the circuit "above" itself in the cascade. The carry output from the full adder is fed to another full adder "below" itself in the cascade.

$$\left.\begin{aligned} &\text{Sum}=A \oplus B \oplus C_{in} \\ &\text{Cout} = AB + BCin + ACin \end{aligned}\right\} \tag{4.1}$$

Table 4.1: Truth table of binary full adder.

A	B	Cin	Sum(S)	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

The hardware implementation of the above equation can be done in different ways. Careful optimization of the adder circuit is required for optimum operation. The optimization can be done both at the logic level and at the circuit level. Logic level optimization is done by optimizing the Boolean equation defined above so that a faster circuit can be designed by reducing the min terms [22]. The full adder acts as a fundamental building-block to the larger circuit units of the decimator. Therefore, the timing and power consumption optimization efforts at the adder level can lead to improved circuit throughput ratings, enhanced speed performance, and lower power consumption requirements. Different full adder architectures have been proposed in literature. Our goal is to achieve an area efficient design by reducing the number of transistors used.

A transistor schematic of the binary full adder circuit used for the addition operation is shown in Figure 4.15. This circuit device used consists of 18 transistors in total (4 transistors used for the construction of two inverters). Transmission gate based design is chosen which has minimum number of transistors. Since the decimator uses adder circuit in every bit operation, optimizing the adder yields an area efficient design. Figure 4.15 also shows the W/L ratios of each transistor. Transistors with constant W/L ratio are used. Transistor sizes can be calculated by considering the worst-case path in which maximum delay can occur. Parasitic capacitances exist at each node and the transistor at that node should be sized such that the time it takes to charge or discharge that capacitor is minimized. In the adder circuit of Figure 4.15, the worst case path for maximum delay exists from the input to the C_{out} . The transistors M2, M4, M6 and M8 with their associated parasitic resistance and capacitors constitutes for the delay. Buffers with sufficient W/L ratios were added to drive the load better.

In the adder circuit shown in Figure 4.15, the transistor combination formed by transistors M1 through M6 performs the XOR operation taken at node Y. The operation of the XOR gate formed by using these six transistors is explained as follows. When signal A is high, node X is low disconnecting the transmission gate formed by transistors M5 and M6 and signal B appears at node Y as B’. When signal A is low, node X is high disconnecting the inverter formed by transistors M3 and M4 and signal B appears at node Y. The sum output in Figure 4.15 obtained from the two transmission gates formed by transistors M11 through M14 is given by $A \oplus B \oplus C_{in}$, where C_{in} is the carry input. The carry output C_{out} is equal to C_{in} when $A \oplus B$ is HIGH and is either A or B when ever $A \oplus B$ is LOW as shown in the adder truth table given in Table 4.1 [6]. This adder uses minimum number of transistors and has equal Sum and Carry delay times. The layout of the area efficient adder circuit designed is as shown in Figure 4.16. Here the output carry bit C_{out} of the binary full adder is applied as the input carry C_{in} to the next full adder.

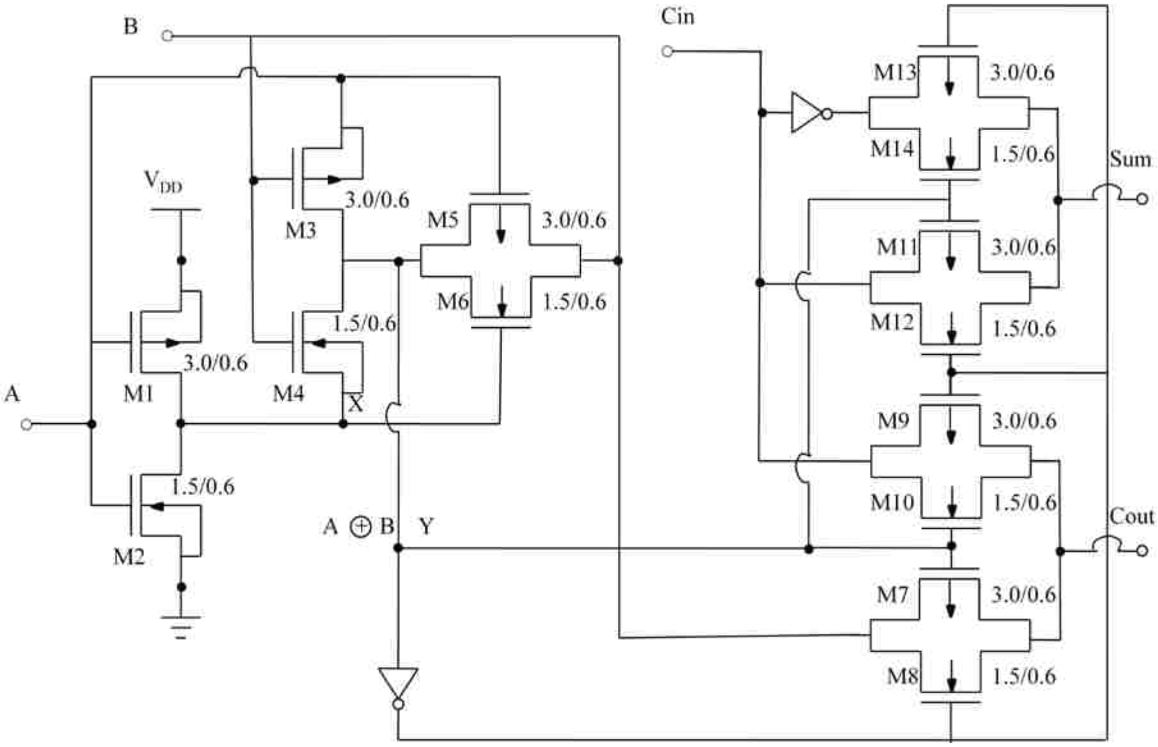


Figure 4.15: Circuit diagram of an 18 transistor binary full adder.

Glitches could appear in the output if all the three inputs are not appeared at the same time for a full adder. Glitches occur due to the delay in the signal lines. The metal line connecting the C_{out} of the first adder to the C_{in} of the second adder should be short to avoid these glitches, since long metal lines can cause a delay in the carry bit. In order to achieve this, the C_{out} of the adder is placed such that it is close to the C_{in} of the next adder. Since a decimator circuit requires large number of adders (120 adder circuits: 60 for integrator stage and 60 for differentiator stage); it is very crucial to have the adder outputs with minimum delays. The simulation results of the adder circuit are shown in Figure 4.17. The input and output bit signals shown in the simulation results is in accordance with the inputs and outputs shown in the adder truth table. The results for Sum and C_{out} match with the values for Sum and C_{out} in the adder truth table in Table 4.1.

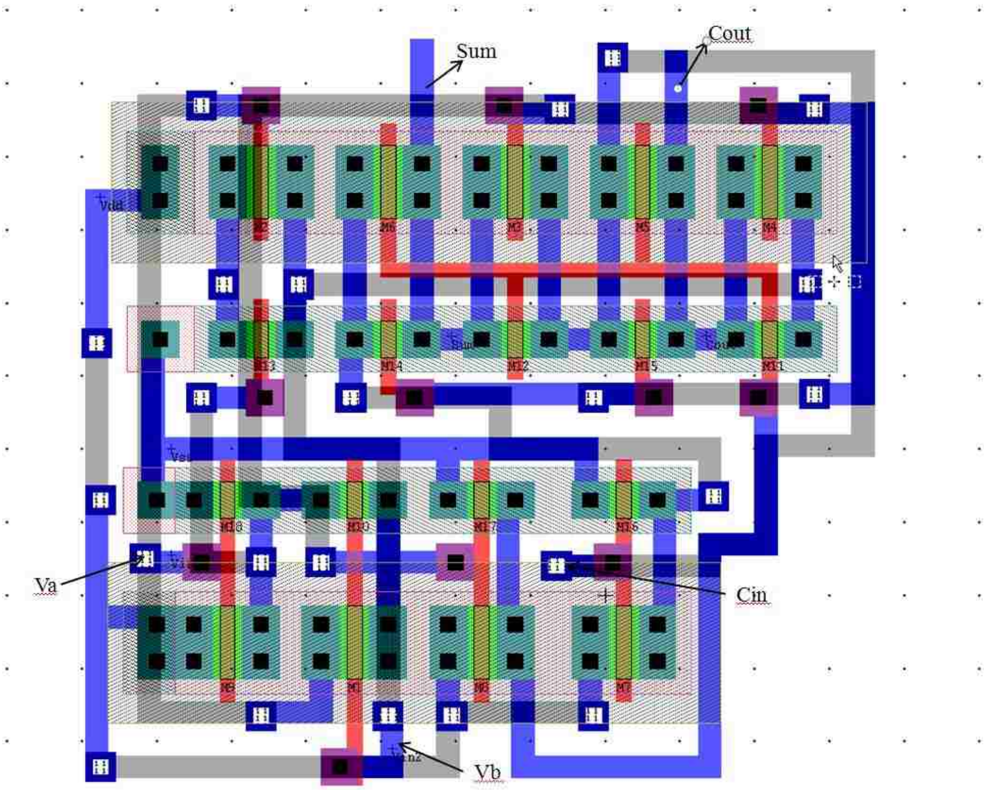


Figure 4.16: Layout of the binary full adder circuit.

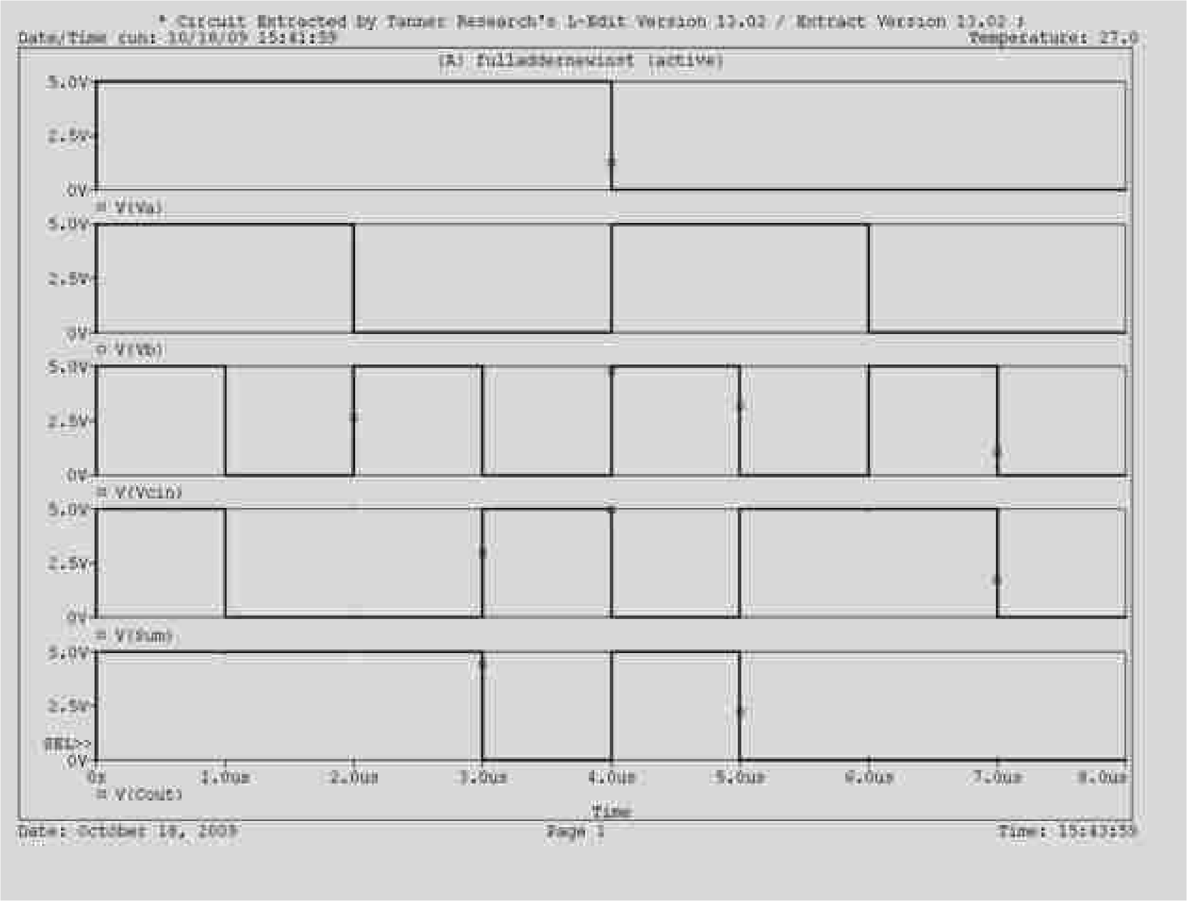


Figure 4.17: Simulation results showing the inputs and outputs of the adder circuit.

4.1.4 Delay Element for Integrator

As discussed in Chapter 3, each 1-bit integrator requires one delay element. A delay element is a register circuit, which is used to provide a delay by one clock period. The delay element can be implemented using a register structure, which can store data. In the present work, the integrator and differentiator circuits of a 3rd order CIC filter requires 120 delay elements. The integrator block alone requires 60 delay elements, 20 for each stage. The differentiator block requires 60 delay elements, which are divided equally between the three differentiator stages. Hence it is very important to design a delay element which is area efficient and has smaller rise and fall times. A delay element can be implemented using a combination of switches and inverters. The circuit used as a delay element in the 1-bit integrator to delay the input by an oversampling clock time period is as shown in Figure 4.18.

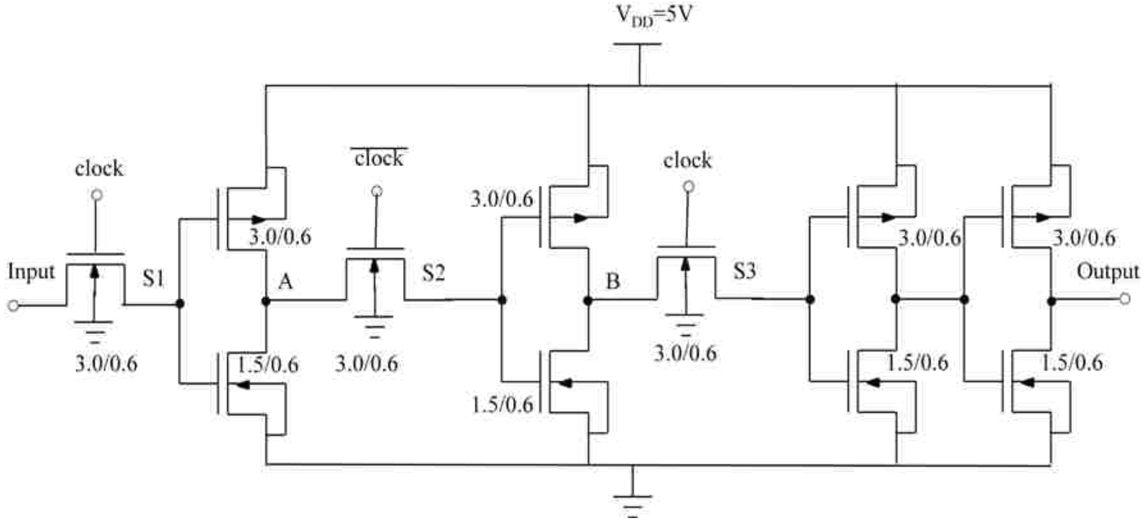


Figure 4.18: Transistor level schematic for achieving a delay by two clock cycles.

The circuit operates with an oversampling clock, which is used to alternatively transfer the data to the right. Simple NMOS switches are used for data transfer from left to right in alternate clock cycles. When clock signal is HIGH switch S1 is closed and the inverted input is available at node A. Since switch S2 is operated with $\overline{\text{clock}}$, it is open and no transfer occurs through this switch. Whenever the clock signal goes LOW, switch S1 is opened and the input is disconnected from the delay circuit. Now, $\overline{\text{clock}}$ is HIGH so switch S2 is closed and the previous inverted input stored on the node A is inverted and transferred towards the node B. During the next cycle when clock is HIGH, the value stored on node B is transferred to the output and fed to the stages following the delay element. It can be observed that an input bit takes one clock cycle to reach to the output. Hence the delay element works fine in providing a delay by one clock period. The time delay is achieved by data transfer using two non-overlapping clock signals. But in practice, the two non-overlapping clocks exhibit clock skew, if the clock skew is large then the delay circuit does not work as delay element. To avoid this condition an NMOS switch based design is used here instead of a transmission gate based design. Since the transmission gates requires two non-overlapping clocks to work as a switch, the presence of clock skew can degrade the signal. By using NMOS switches, non-overlapping clocks should be applied to two different switches and the problem with the clock skew is neutralized with the delay of the inverters. The other advantage of the NMOS based design is that the use of buffer circuit at the output will give the delay circuit better driving capability in driving the loads connected to the circuit [22].

The layout of the delay circuit is shown in Figure 4.19. The clock in the layout is represented with clk. The simulation results of the delay element are shown in Figure 4.20. The simulations are done with a clock input frequency of 256 kHz. From the simulation results in Figure 4.20 it can be observed that the output is same as the input except that the output is delayed by one clock time period (4μs).

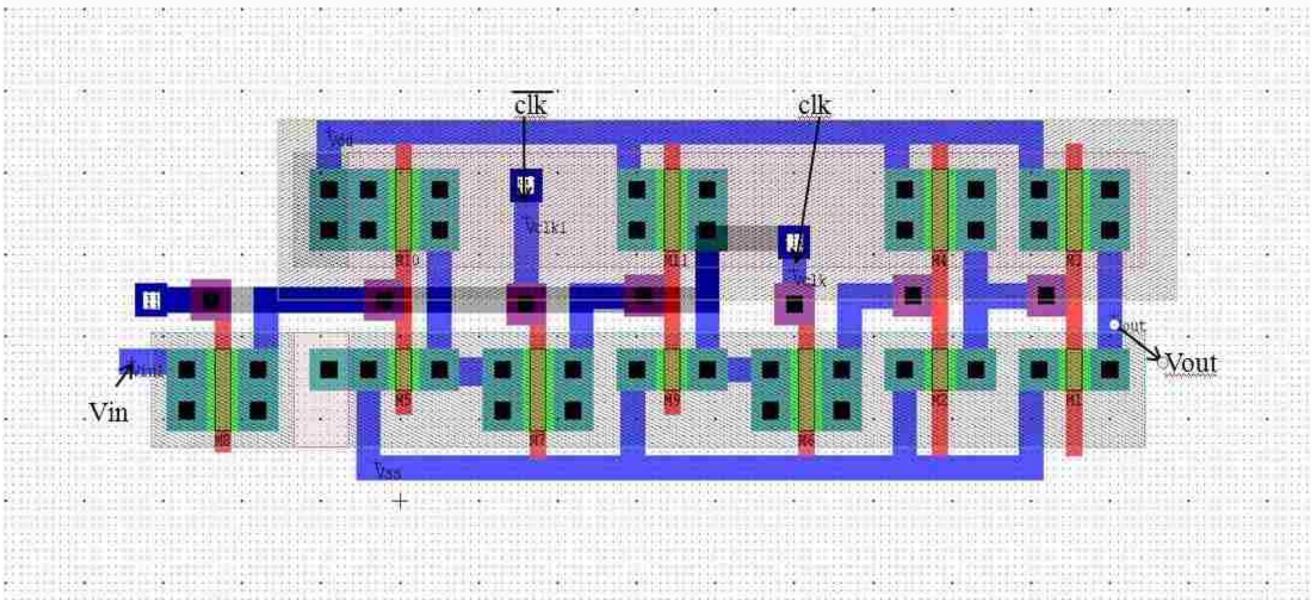


Figure 4.19: Layout of the delay circuit.

4.1.5 Delay Element for Differentiator

The delay element used for differentiator is a modified version from that used for integrator elements. In the differentiator circuit, the input has to be delayed by 64 clock cycles, which is equivalent to delay by one cycle of the clock divider output. The voltage level at the delay element output has to hold the high or low value until the 64th clock cycle without decay. To make this possible we add a register to the delay circuit as shown in Figure 4.21. The register used in this delay circuit is a D Latch. It helps to hold the delayed output value for one clock cycle of the clock divider output.

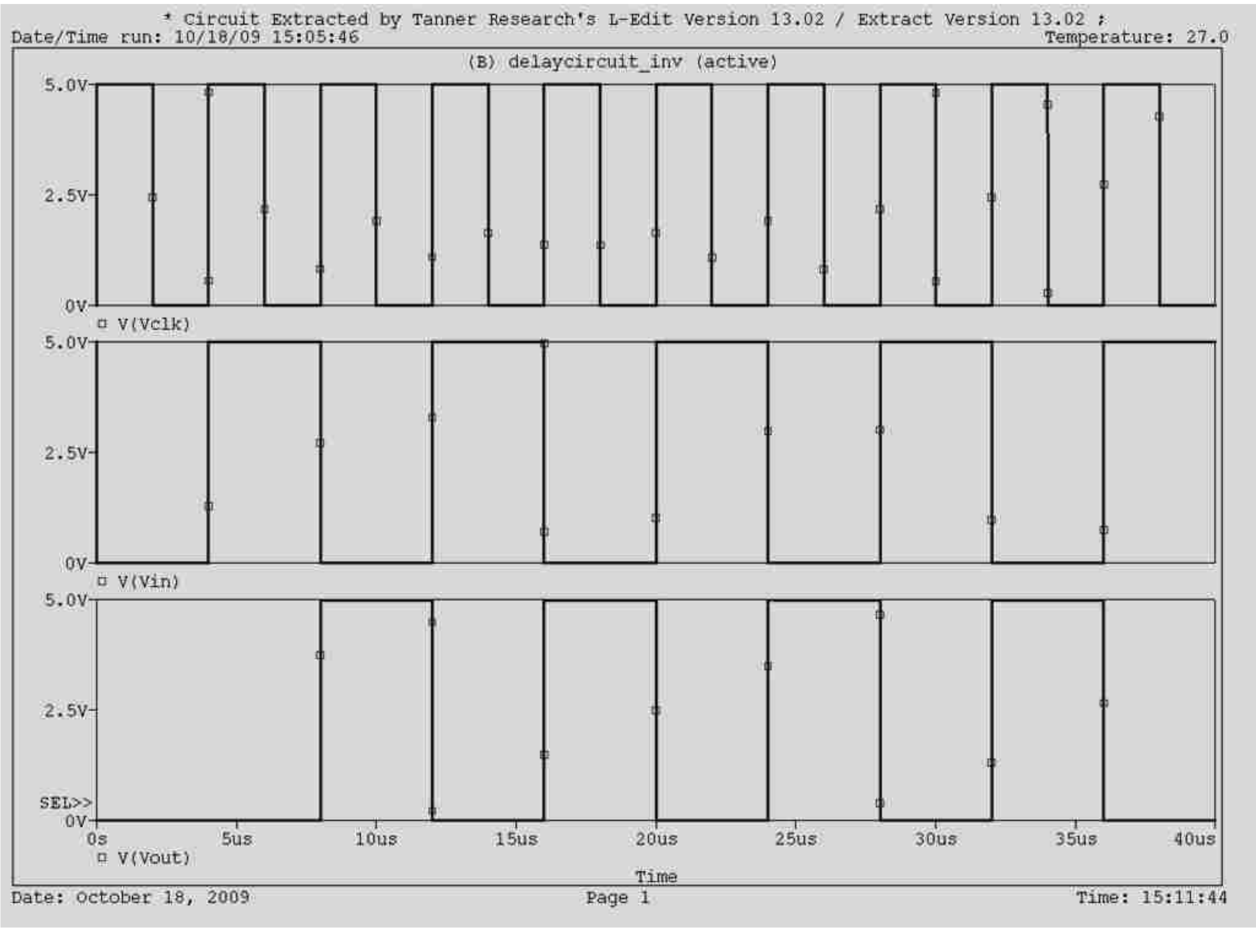


Figure 4.20: Simulated input and output of the delay circuit.

The D latch is also known as transparent latch. When the clock input or the enable is logic 1, the input D propagates directly through the circuit to the output Q. When the clock input falls to logic 0, the last state of the D input is trapped and held in the latch, for use by the further circuitry. The D-latch circuit also would not experience a "race" condition caused by all inputs being at logic 1 simultaneously, since the single D input is also inverted to provide the signal to reset the latch.

Figure 4.21 shows the modified schematic of the delay circuit used in the design of differentiator circuit. The layout of the delay circuit is as shown in Figure 4.22.

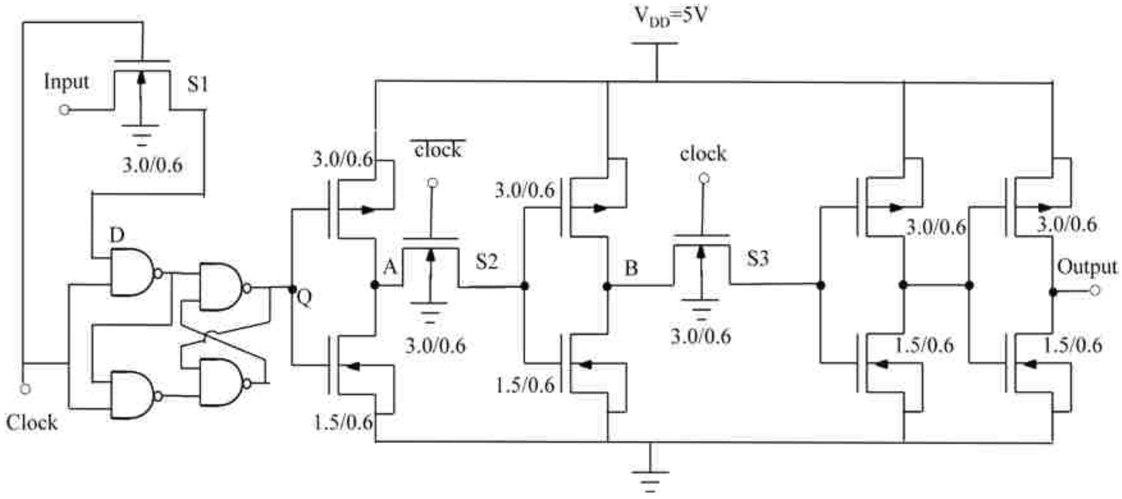


Figure 4.21: Transistor level schematic for achieving a delay by two clock cycles for fs/64.

4.1.6 Coder Circuit

The purpose of the coder circuit is to increase the resolution of the ADC. The output from the second order modulator is one bit binary, i.e. it is either 1 or 0. As discussed in Chapter 3, section 3.1.1, the register overflow might occur at the multiples of f_s due the infinite gain. This register overflow can be avoided if the register length of the integrator is chosen according to equation (3.5) and also by using the 2’s complement method of coding. The schematic of the coder circuit used in this design is shown in Figure 4.23. The two outputs of the coder circuit for binary 1 and 0 inputs are shown in Table 4.2. The circuit uses a basic AND and OR logic gates along with the inverter and buffers to achieve the required resolution and the 2's complement. For K = 64 case, this coder circuit should increase the resolution of the input by 19-bits such that the output of the coder is a 2's complement 20-bit data as given by equation (3.5).

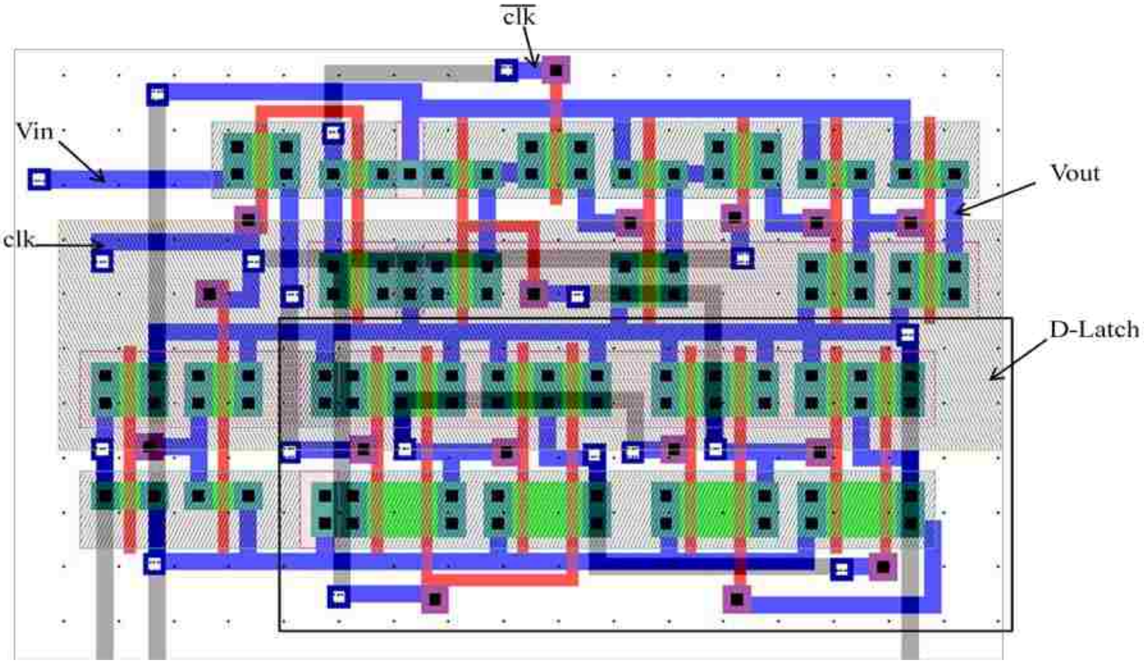


Figure 4.22: Layout of the delay circuit used for fs/64.

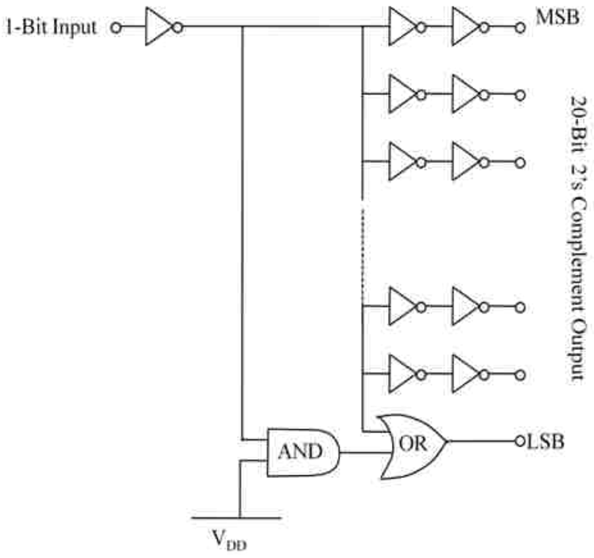


Figure 4.23: Gate level schematic of coder circuit.

Table 4.2: Tabular representation of input and output values of the coder circuit

Coder Circuit (K=64)	
INPUT (1-Bit)	**OUTPUT (20-Bits)**
1	00000……000001
0	11111……111111

4.6 Integrator Design

A 1- bit integrator is constructed using the one full adder followed by one delay element as shown in Figure 4.24. The design of the full adder and delay elements has been explained above in sections 4.1.3 and 4.1.4. Each 1-bit integrator has an oversampling clock input, f_s. The adder output is delayed by one oversampling clock period by the delay element and fed back as the second input to the adder. The other input to the adder is the coder output and the carry out from the preceding significant bit adder. The carry input for the least significant bit adder is taken as 0 and the carry out for the most significant bit adder is ignored. The integrator circuit shown in the Figure 4.24 is easier for implementation in hardware and is slightly different from the integrator shown in Figure 3.2 of Chapter 3. The magnitude response of both the circuits is same and the only difference is the change in the phase response of the circuit [22]. The layout of the 1-bit integrator with various blocks marked is shown in Figure 4.25.

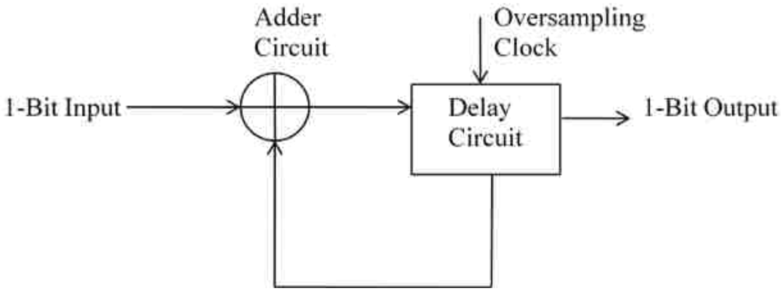


Figure 4.24: 1-bit integrator circuit used in Figure 4.25.

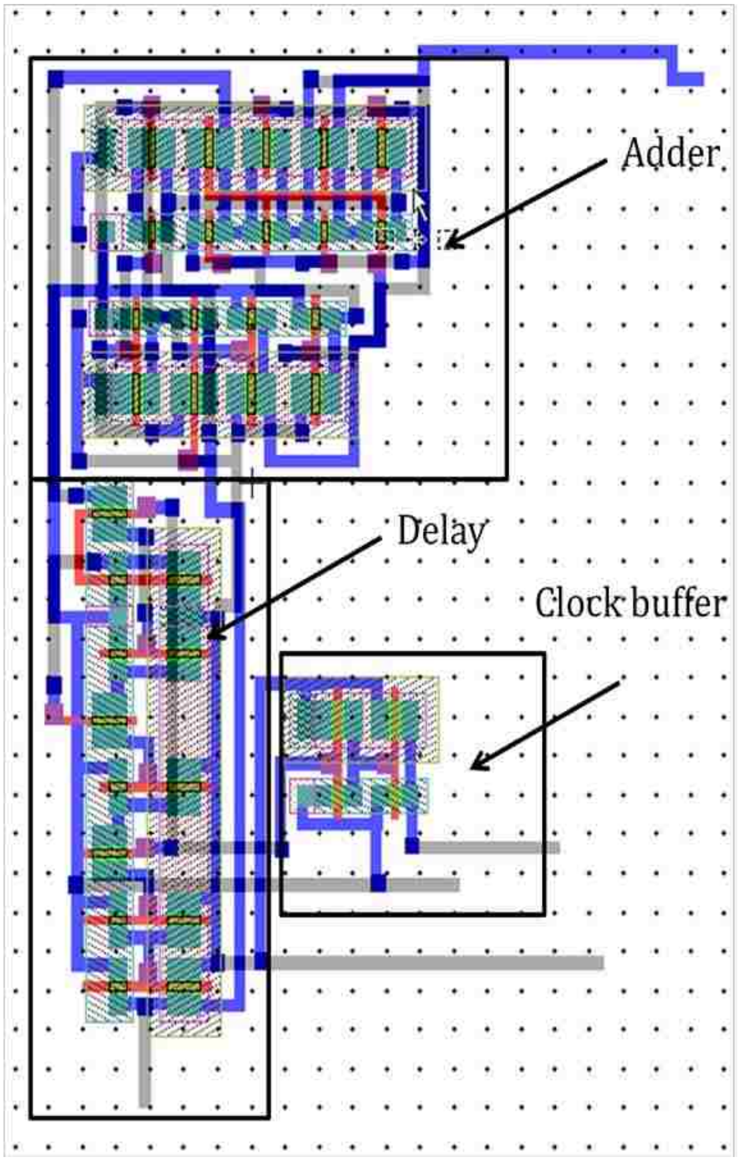


Figure 4.25: Layout of 1-bit integrator.

4.7 Differentiator Design

A 1- bit differentiator is constructed using the one full adder, one delay element and an inverter as shown in Figure 4.26. The design of the full adder and delay elements has been explained above in Sections 4.1.3 and 4.1.5. Unlike the integrator circuit, which uses the oversampling clock as clock input, the 1-bit differentiator circuit is triggered by a clock whose frequency is 1/K times the oversampling clock frequency. The circuit is used to calculate the difference between the one input bit and the other input bit delayed by KT_s, where T_s is the time period of the oversampling clock.

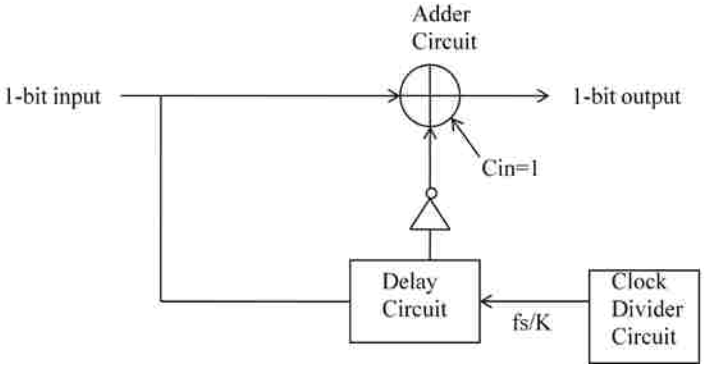


Figure 4.26: 1-bit differentiator circuit used in Figure 4.27.

In Boolean algebra, the difference operation of two inputs is nothing but the summation of the first input with the complement of the second input. To perform the difference operation, the second input is complemented and 2's complement method is used as the complementing scheme. To implement this in hardware, the second input, which is delayed by a time KT_s (sec) compared with the original input, is complemented using an inverter circuit. The carry input to the adder circuit is set at logic HIGH. By implementing these two conditions as illustrated in Figure 4.26, a 2's complement difference operation can be achieved [22]. The layout of the 1-bit differentiator with various blocks marked is shown in Figure 4.27.

4.8 Third Order CIC Filter Integration

A 3rd order digital integrator is used in designing the 3rd order CIC filter. The integrator is the main circuit for achieving the increase in resolution. The schematic block diagram of a 3rd order integrator circuit is as shown in Figure 4.28. As discussed in Chapter 3, the coder present before the integrator stage increases the input bit resolution to 20 bits. The 3rd order integrator block diagram is as shown in the Figure 4.29 and works for an oversampling ratio of K = 64. The layout of the third order integrator is as shown in Figure 4.29.

A 3^{rd} order digital differentiator circuit also called as a comb filter, acts as a low pass finite impulse response (FIR) filter. The 20-bit output from the 3^{rd} order integrator is applied as the input to the differentiator. The schematic block diagram of a 3^{rd} order differentiator circuit is as shown in Figure 4.30 and consists of three 20-bit differentiator stages. Here each 20-bit differentiator stage also includes a 20-bit down sampling register circuit, which is explained in detail in the following section. The layout of the 20-bit differentiator circuits including the 20-bit down sampling register circuit is as shown in Figure 4.31.

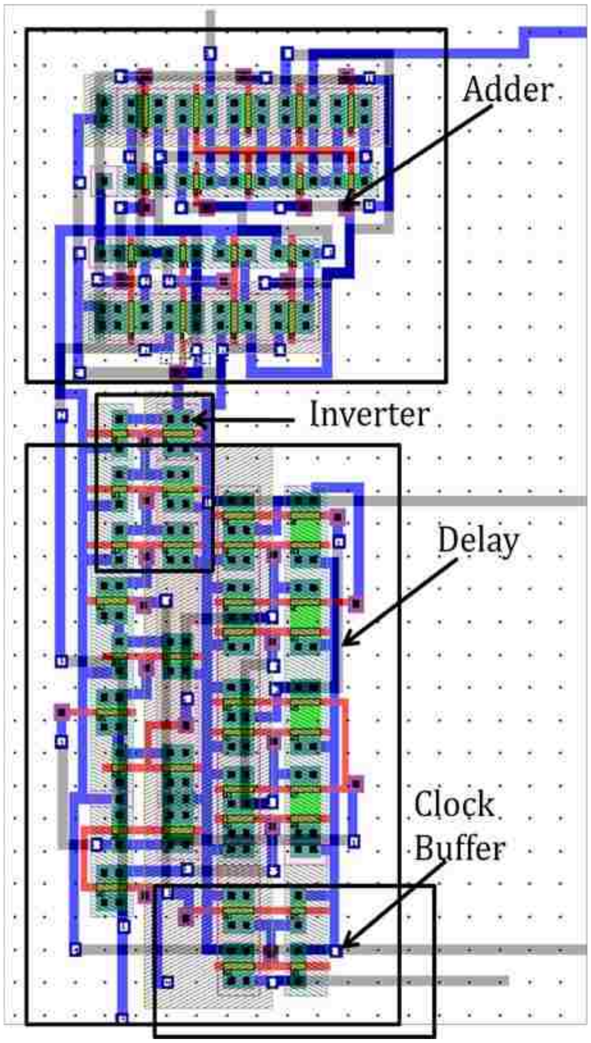


Figure 4.27: Layout of 1-bit differentiator.

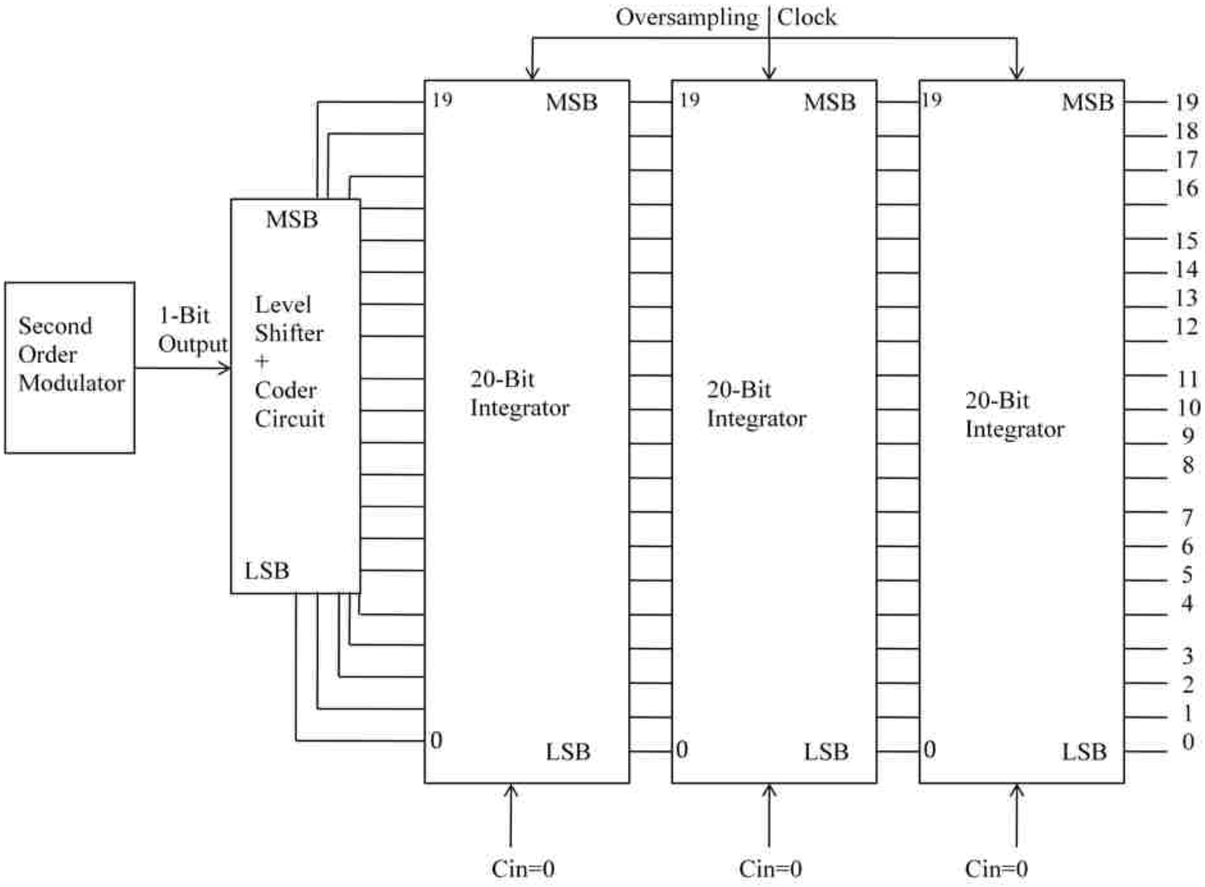


Figure 4.28: Hardware implemental block diagram of the 3rd order integrator circuit.

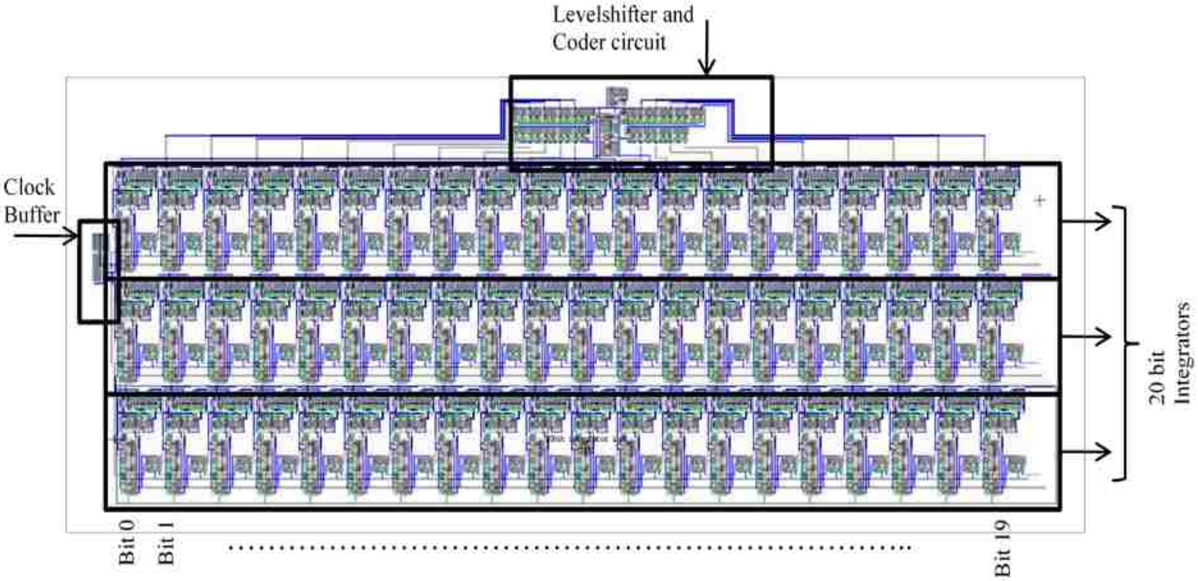


Figure 4.29: Layout of the 3rd order integrator circuit.

4.8.1 20-bit Downsampling Register

The 20-bit output from the differentiator stages has a continuous differentiated output (As the integrator stages operate at a sampling frequency fs, whereas the differentiator stages operate at a sampling frequency, fs/64). We need the output of the differentiator at the instances where the clock divider output (f_s/64) is high. This is achieved by using a simple down sampling register at the output bit of each differentiator circuit. This circuit uses two buffers and a pass transistor gate (PMOS gate), which is clocked such that it allows the output only when clock is high or '1' and grounds the output whenever the clock is low or '0'. The layout of the 20-bit down sampling circuit is as shown in Figure 4.32. The clock used for the register circuit is the same as the clock used for differentiator circuit.

In Figure 4.32, the layout shows the registers used for first 5-bits of the 20-bit input. The minimum transistor W/L ratios (PMOS – 3.0µm/0.6 µm and NMOS – 6.0 µm/0.6 µm) are used for the inverters in the buffer. The buffers present at the input and output of the register help to increase the resistance of the circuit and also helps to drive the load better at the output bits at each differentiator stage. They also help in solving the fan-out problems.

Since the delay element delays the input by KT_s seconds, the first output of the first differentiator appears only after KT_s seconds, which is a valid output. Similarly the second stage and the third stage differentiators also require KT_s seconds to have a valid output. So, the output should be considered only after this initial no-valid output time period has elapsed which would be $3xKT_s$ seconds. The differentiator circuit operates with 2's complement 20-bit data that is supplied from the integrator stage. The output of the 3rd order differentiator is also a 2's complement 20-bit data.

For the second order oversampled sigma-delta modulator and the third order CIC filter used in the design, the desired output resolution is given by the equation (4.2).

$$N_{inc} = \frac{50\log K - 12.9}{6.02} \qquad (4.2)$$

In equation (4.2), N_{inc} is the increase in resolution and K is the oversampling ratio. So, for K=64, the output resolution achieved is 14 bits. Hence, we select 15-Bits (1-sign bit +14 resolution bits) from the 20-bit output of the differentiator and drop the lower 5 bits. Bit 19 or MSB is taken as sign bit and bit 18 to bit 5 (14-bits) are considered to calculate the discrete value of the output.

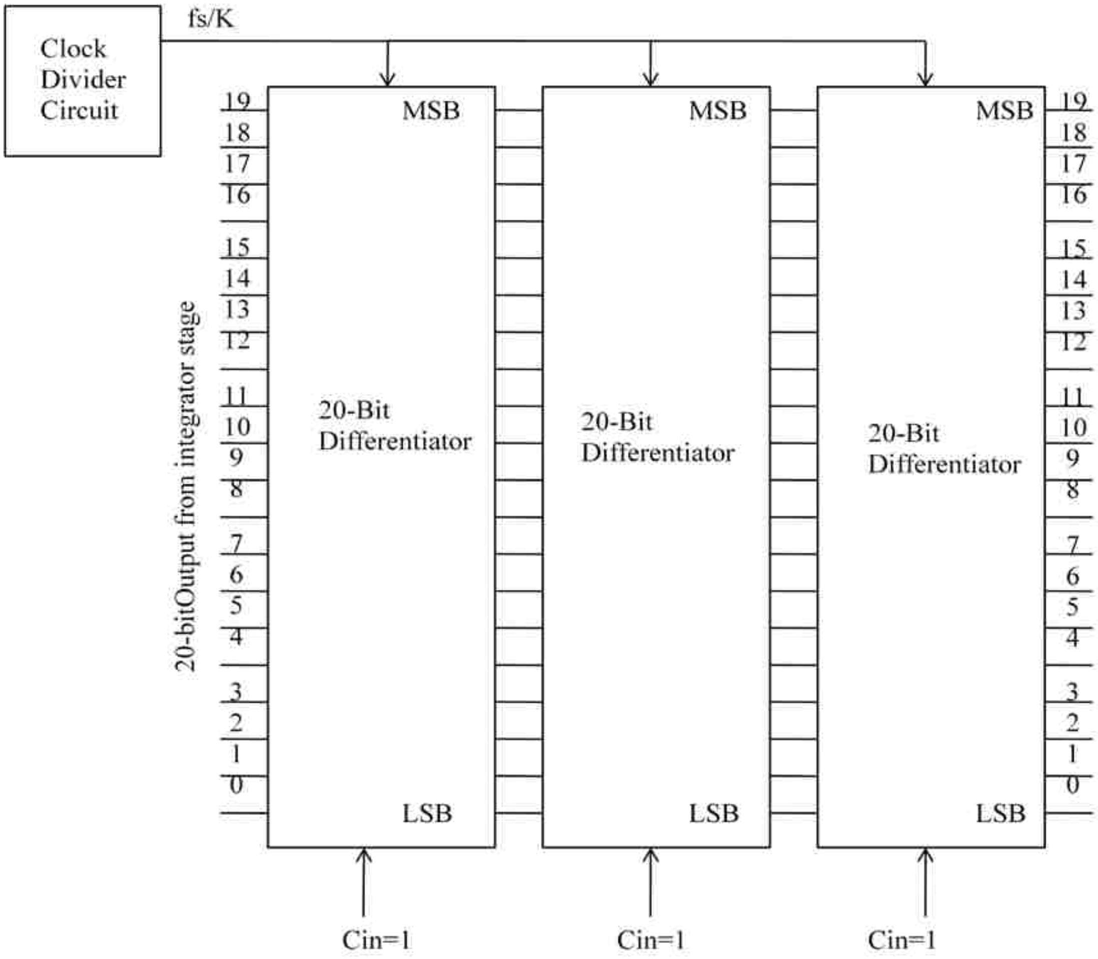


Figure 4.30: Hardware implementation of a 3rd order differentiator circuit.

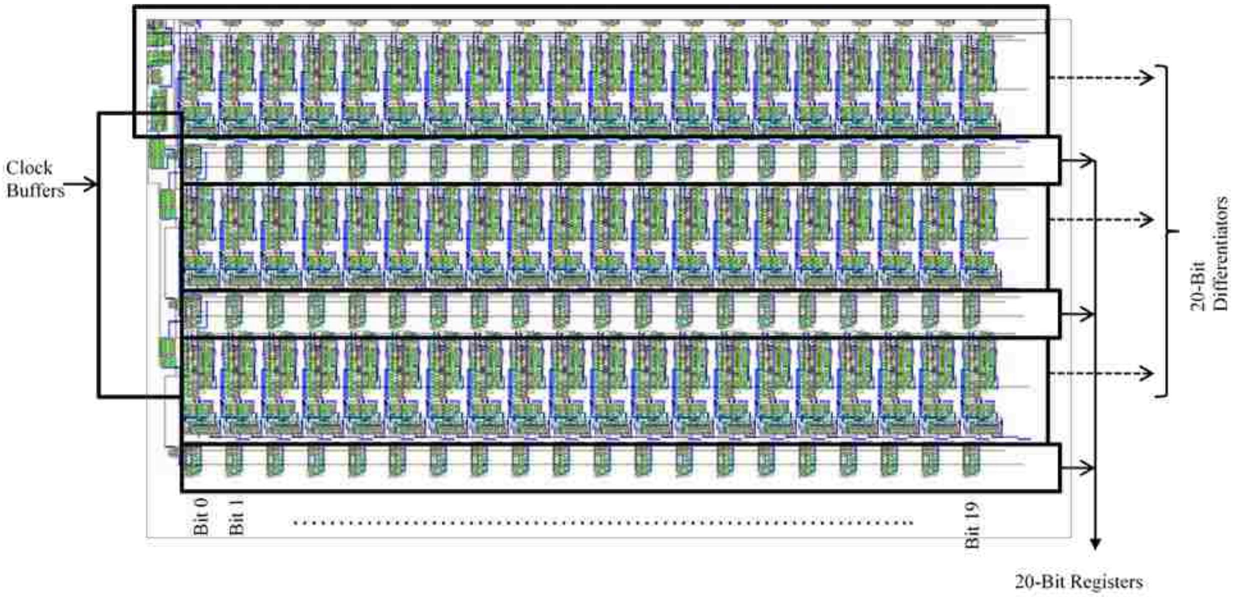


Figure 4.31: Layout of the 3rd order digital differentiator circuit.

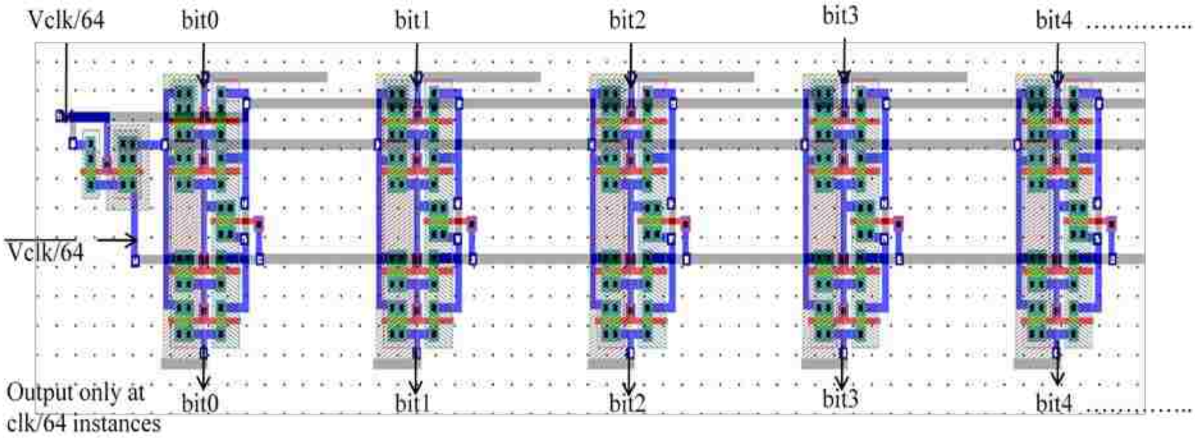
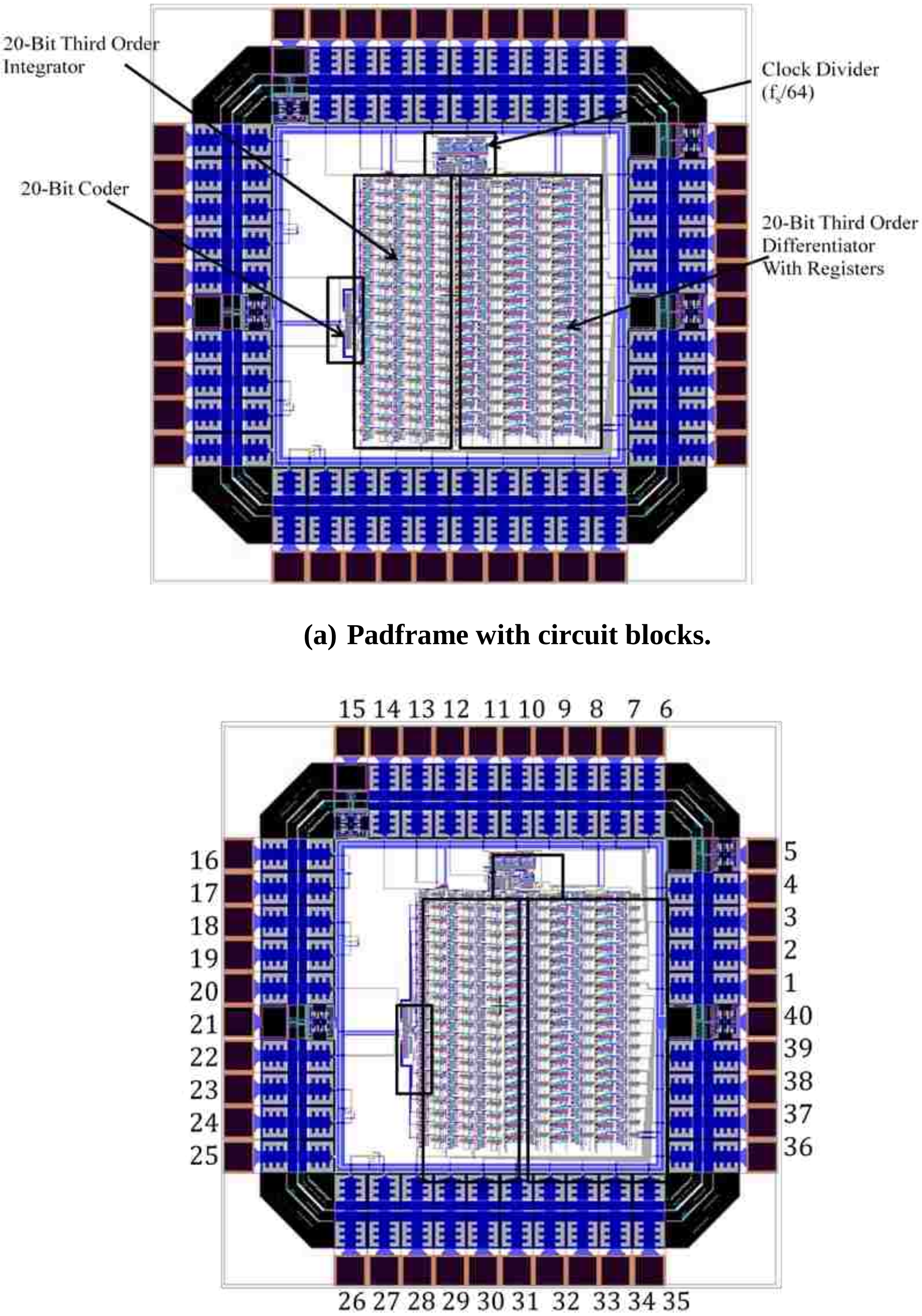


Figure 4.32: Layout of a 20-bit down sampling register.

Table 4.3 gives the different circuits used and their count. It also gives the total transistor count required to design the third order CIC filter. The number of full adder circuits and the number of delay elements required are very high and hence, we have considered an optimized design which minimizes the area and decreases the delay. The complete layout of the CIC filter designed in 0.5µm N-well CMOS process with the 40-pin padframe connections is shown in Figure 4.33 (a) and (b). The pin description for the fabricated CIC filter is given in Appendix B. The experimental results of the CIC filter are presented in Chapter 5.

Table 4.3: Tabular representation of the different circuits used and the number of transistors designed for a 3rd order CIC filter.

Circuit	Circuit Count (M)	No. of Transistors (N)	Total No. of Transistors/Circuit (M*N)
Full Adder	120	18	2160
Shift Register	120	12	1440
D-FlipFlop	6	34	204
D-Latch	60	16	960
AND gates	7	6	42
MUX and Buffers			~200
		TOTAL TRANSISTORS	5006



(a) Padframe with circuit blocks.

(b) Padframe with pin numbers.

Figure 4.33: Layout of a cascaded integrator comb (CIC) filter in a 40-pin padframe.

Chapter 5
Experimental Results

The filter has been tested with input from a second order oversampled sigma-delta modulator. The filter is tested to meet the specifications of the design and functionality desired. The CIC filter includes decimation and acts as an averaging filter.

5.1 Experimental Setup

The experimental setup used to test the CIC filter is as shown in Figure 5.1. Figure 5.2 shows the microphotograph of the fabricated chip. A function generator is used to provide clock input to the oversampled sigma-delta modulator and the CIC filter. Analog (Sine Wave) input is also generated using the function generator. The output bits of the CIC filter are observed using a logic analyzer. An analog oscilloscope has also been used to check the inputs and the output waveforms.

The input specifications that are applied to the ADC are shown below:

- **Modulator Specifications:**
  - Resolution, 14 bits
  - Baseband signal bandwidth, 800 kHz
  - Oversampling ratio, 64
  - Maximum sampling frequency, 25 MHz
  - Supply voltage, +2.5 V and -2.5 V
- **Decimator Specifications:**
  - Resolution, 14 bits
  - Clock divider ratio or oversampling ratio, 64
  - Maximum sampling frequency, 1MHz
  - Supply voltage, 5V

The input digital data to the decimator is given from the output pin of the modulator. The same oversampling clock is applied to the modulator and decimator chip. The digital output data and the clock divider output are analyzed using a logic analyzer.

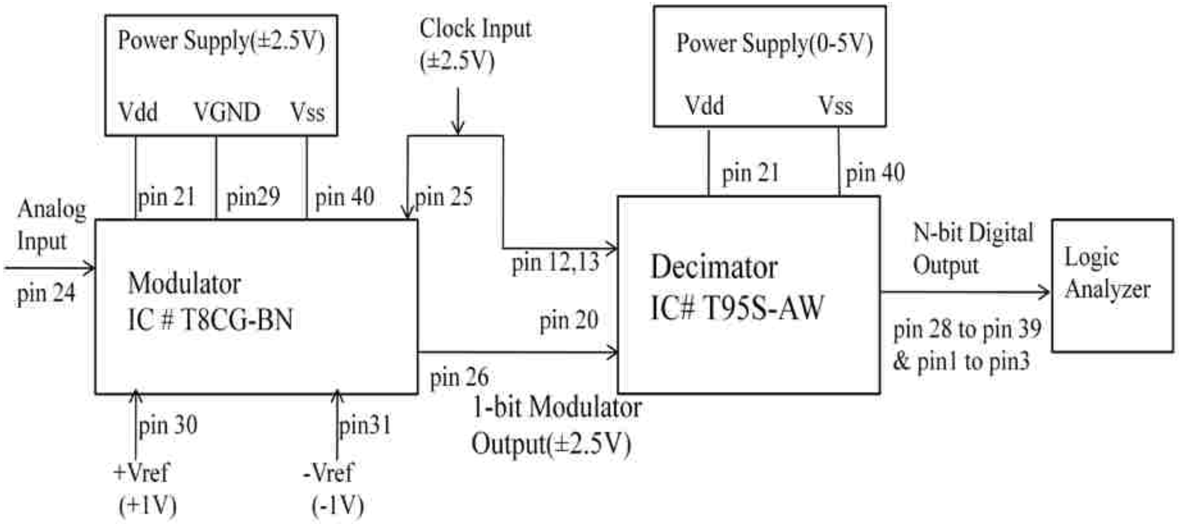


Figure 5.1: Experimental setup showing the modulator and decimator connections.

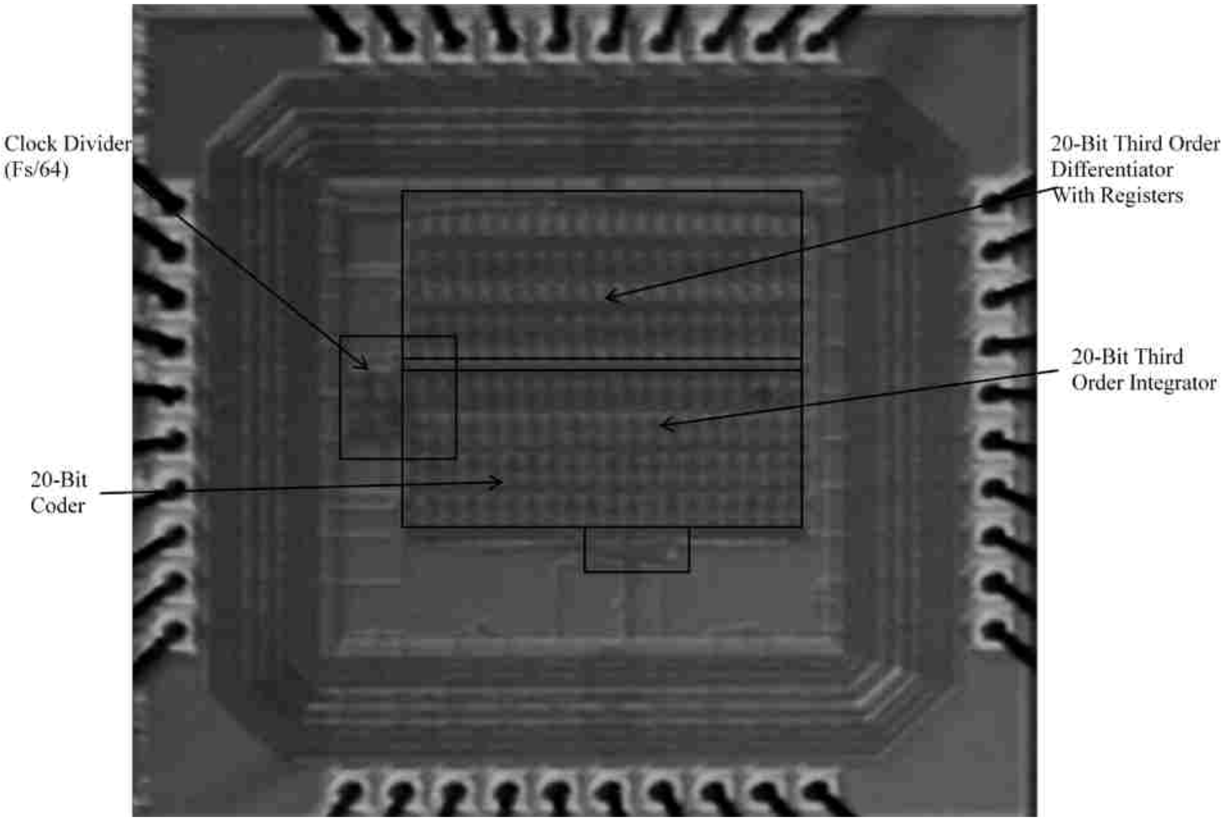


Figure 5.2: Microphotograph of the fabricated decimation filter.

5.2 Modulator Output

As stated earlier the oversampling clock required is supplied from an external functional generator to both modulator and decimator. The input to the modulator is a sine wave with a frequency of 1 kHz frequency and a bandwidth of 2 kHz. The oversampling clock applied to the chip has a frequency of 256 kHz (K x Nyquist rate), where K is oversampling ratio and Nyquist rate is twice the bandwidth of the input sine wave. The input and output of the oversampled sigma-delta modulator along with clock for different input amplitudes are obtained as shown in Figure 5.3 through Figure 5.5. The output of the modulator is a 1-bit pulse density modulated signal occurring at oversampling clock rate. The 1-bit output of the modulator swings between -2.5V to +2.5V, which is the supply voltage of the modulator. The 1-bit output from the modulator is applied as the input to the decimator circuit.

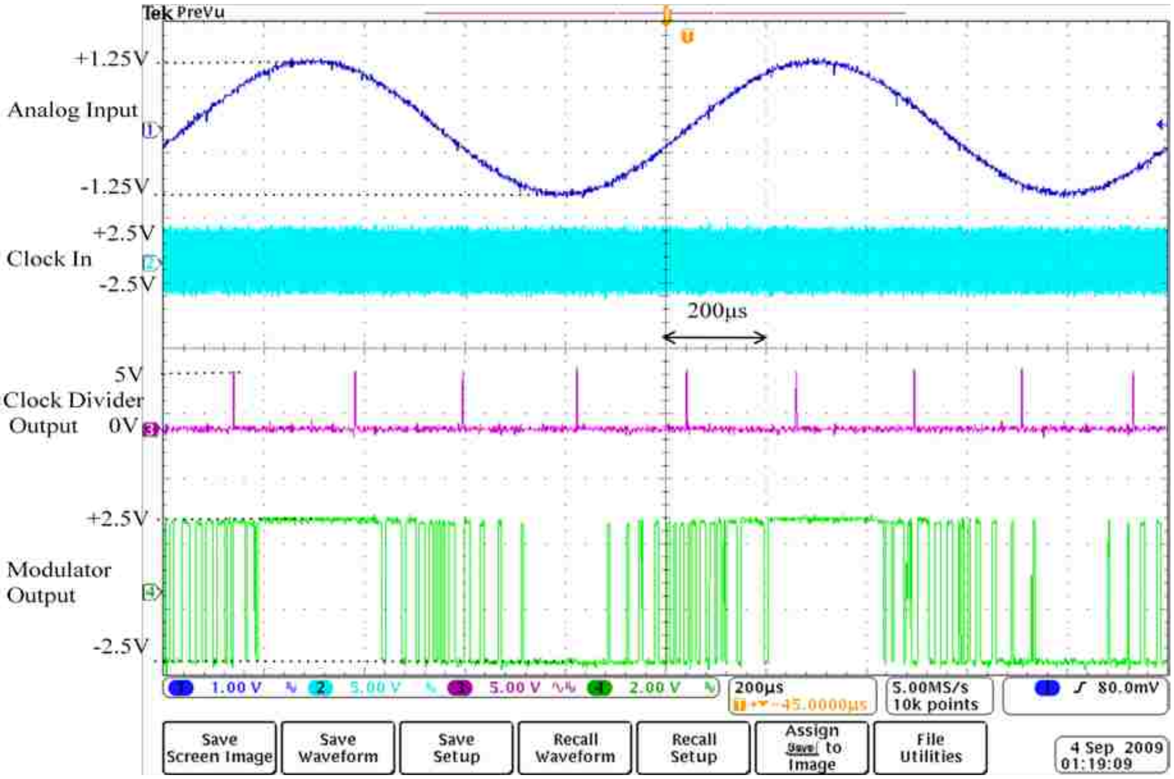
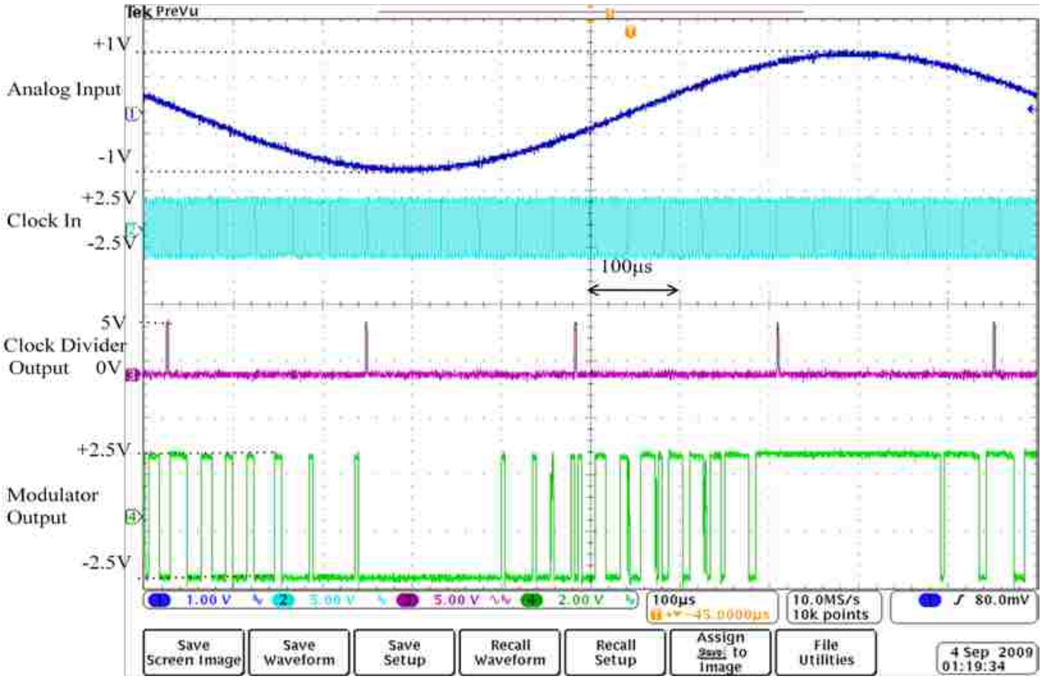


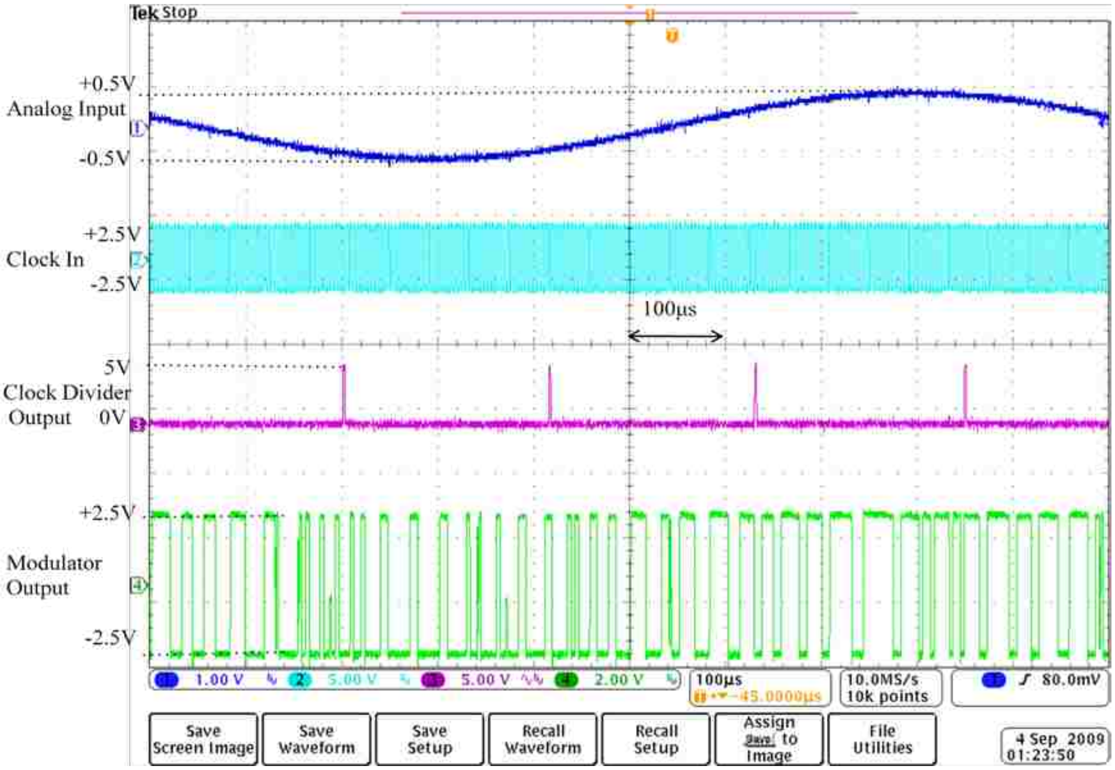
Figure 5.3: Modulator input, oversampling clock and modulator output for 1 kHz, 2.5 Vp-p input and 256 kHz clock.



1 1.00 V 2 5.00 V 3 5.00 V 4 2.00 V 100µs -45.0000µs 10.0MS/s 10k points 1 80.0mV

4 Sep 2009 01:19:34

Figure 5.4: Modulator input, oversampling clock and modulator output for 1 kHz, 2Vp-p input and 256 kHz clock.



1 1.00 V 2 5.00 V 3 5.00 V 4 2.00 V 100µs -45.0000µs 10.0MS/s 10k points 1 80.0mV

4 Sep 2009 01:23:50

Figure 5.5: Modulator input, oversampling clock and modulator output for 1 kHz, 1Vp-p input and 256 kHz clock.

5.3 Clock Divider Circuit Output

The oversampling clock (± 2.5 V, 256 kHz) is applied to the clock divider input of the designed CIC decimation filter. The clock is level shifted using the internal level shifter. The input and output of the clock divider are as shown in Figure 5.6. The output frequency of the clock divider is 4 kHz (256/64). The time spacing between two pulses is 250µs (KT_s), where T_s is the oversampling clock time period. In this case T_s=3.9µs. The output of the decimator occurs at the same time instants as the output pulses of the clock divider, i.e. we get one discrete output value for every 250µs. Hence the output signal frequency is at Nyquist rate. The clock divider circuit plays a crucial role in generating the high-resolution digital output.

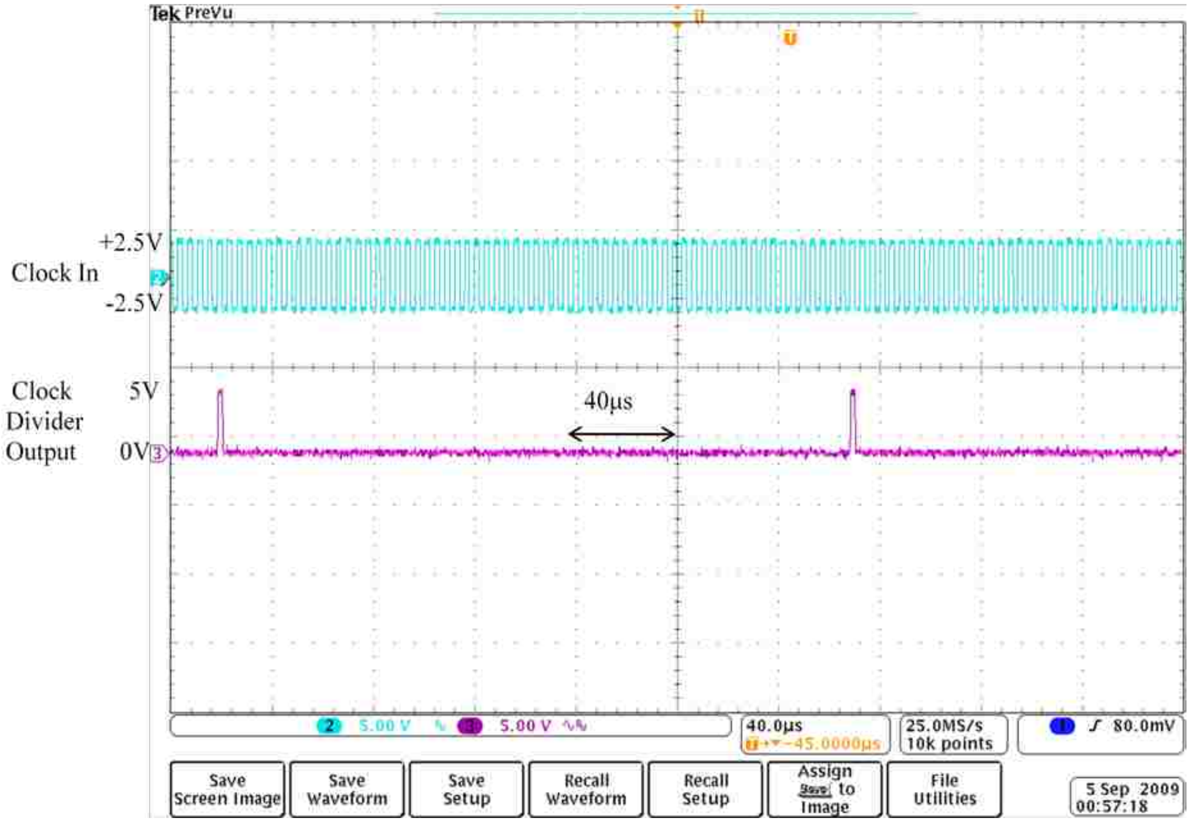


Figure 5.6: Clock divider input, output for 256 kHz input.

5.4 CIC Filter Output

The 1-bit ± 2.5V output from the modulator is applied to the input of the CIC filter. The in-built level shifter circuit shifts modulated signal from ± 2.5V voltage range to 0V to 5V voltage range. The built-in coder circuit boosts the resolution from 1-bit to a higher resolution of 20-bit. The 20-bit 2's complement input is then applied to the 3rd order integrator. The output from integrator stages is then applied to the 3rd order differentiator stage. Each differentiator stage output is passed through the down sampling registers. The output of the differentiator, which is also the output of the decimator, is the final output of the sigma-delta analog-to-digital converter.

The output of the CIC filter is in 2's complement form occurring at twice the input signal bandwidth. As discussed earlier the desired 15 bit output is converted from 2’s complement to equivalent binary form. The applied sine wave input to the ADC is 2.5Vp-p, 1 kHz with a bandwidth of 2 kHz and the applied oversampling clock frequency is 256 kHz and hence the output occurs at 4 kHz. For an ADC 1 LSB is defined as [4],

$$1LSB = \frac{V_{FSR}}{2^N} \qquad (5.1)$$

In equation (5.1), V_{FSR} is the full-scale voltage range of the input signal and N is the number of output bits. In this particular case, V_{FSR} for positive half cycle of the sine wave is 1.25V and number of bits, N=14. The experimental results are illustrated in Table 5.1, which shows the 15-bit 2's complement output, binary output, its decimal equivalent, and the actual analog voltage. In Table 5.1 a set of 4 data samples are shown together. Based on the equation (5.1), for the case of K=64 and 2.5Vp-p, the value of 1 LSB for a 14-bit digital output is $1.25/2^{14}$ = 0.0762 mV. The LSB value is used in finding the analog equivalent of the digital output. The left most bit is the sign bit and the bit next to it is the most significant bit (MSB) and the right most bit is the least significant bit (LSB). The 2's complement output is converted to the binary form when the sign bit is '1'. This can be done by complementing all bits from MSB to LSB except sign bit and adding 1 to it. The decimal equivalent of the binary output is calculated by multiplying the bit coefficients with the respective powers of 2. The sign bit is not included to calculate the value.

The decimal equivalent value for the first data stream is obtained as shown below:

$0*2^{13} + 0*2^{12} + 0*2^{11} + 1*2^{10} + 1*2^{9} + 1*2^{8} +1*2^{7} + 1*2^{6} + 0*2^{5} + 0*2^{4} + 1*2^{3} + 1*2^{2} + 1*2^{1} + 1*2^{0} = 1999$.

The actual analog value of the decimal equivalent is obtained by multiplying the decimal equivalent value with the value for 1 LSB. For the decimal value of 1999, its analog equivalent is given by 1999 x 0.0762 mV = 0.152V. The ADC is operated with a peak-peak voltage of 2.5V (± 1.25V). The entire range of 0-1.25V is represented by the 15-bit 2's complement form. Hence the analog values are appended with a '+' sign if the sign bit is 0 and '-' if the sign bit is 1. The experimental output results calculated for 1 cycle of sine wave input are given in Table 5.1. Similarly, for the case of K= 64 and 3Vp-p, the value of 1 LSB for a 14-bit digital output is $1.5V/2^{14} = 0.0915mV$.

The waveforms for the 15-bit digital output data for different sine wave and clock inputs are shown in Figure 5.7, Figure 5.8, Figure 5.9 using the labels Lab1 1 through Lab1 15 where Lab1 1 represents the MSB and Lab1 15 represents the LSB. The output waveform of the clock divider circuit is shown using the label Lab1 0. The output of the decimator exists only at the output of the clock divider circuit, i.e. the output is at the Nyquist rate. As discussed earlier, the decimal and analog voltage value for the other digital outputs are calculated for samples as shown in Table 5.1. Table 5.1 lists several sets of digital outputs read for different input amplitudes and frequencies of the analog input. The analog equivalent values from Table 5.1 are plotted as discrete values as shown in Figure 5.10 (a) and (b), Figure 5.11 (a) and (b), Figure 5.12 (a) and (b) and Figure 5.13 (a) and (b) using MATLAB. Here, each output is superimposed with a respective analog sine wave.

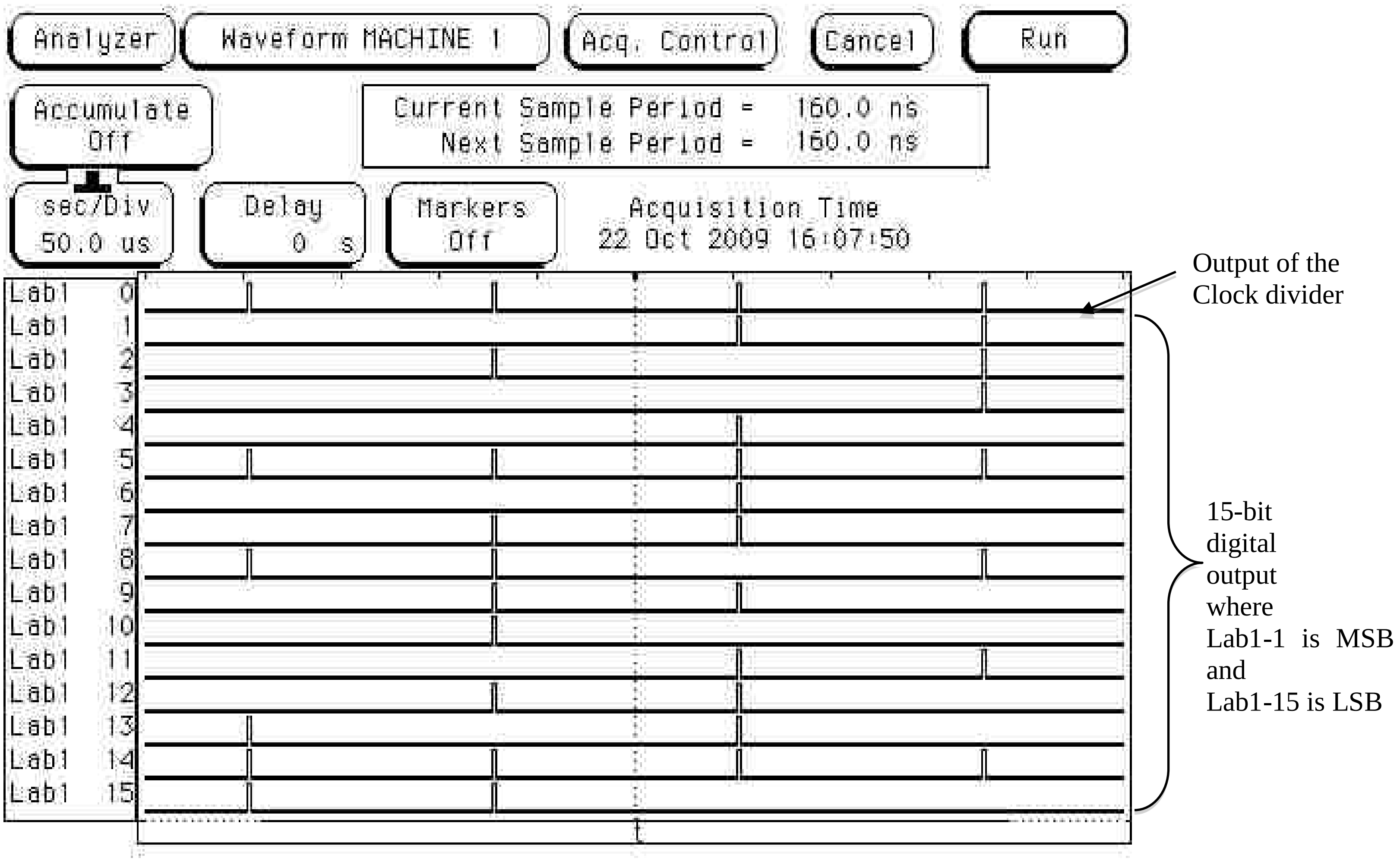
In the first case, since the output frequency is at 4 kHz and the input signal is at 1 kHz, there exist four output data words in one clock cycle of the input signal. It can be observed that alternative outputs have approximately same magnitude. Due to the noise and buffering problems, the discrete output value does not exactly coincide with the sine wave at all points. It is either less or more than the exact sine wave value. This problem can be eliminated by adding a low-pass filtering block at the output of the decimator stage, which is usually the case in most of the commercial CIC filters.

Table 5.1: Tabular data representation of the 14-bit decimator output for K = 64 case.

Input parameters	Digital Code (2's complement)	Digital Code (Binary form)	Decimal Equivalent	Analog Equivalent (A)V
2.5Vp-p, 1 kHz, 256 kHz	000011111001111	000011111001111	1999	0.152
	011011111001101	011011111001101	14285	1.089
	111100010111011	000011101000101	1861	-0.142
	100010000101111	011101111010001	15313	-1.166
2.5Vp-p, 1 kHz, 256 kHz	001010110101011	001010110101011	5547	0.422
	010101110111111	010101110111111	11199	0.853
	110010111001111	001101000110001	6705	-0.510
	101011010100101	010100101011011	10587	-0.806
2.5Vp-p, 4 kHz, 1024 kHz	000110100110100	000110100110100	3380	0.257
	011010100110101	011010100110101	13621	1.307
	110111111001101	001000000110011	4147	-0.316
	100010010101010	011101101010110	15190	-1.157
2.5Vp-p, 4 kHz, 1024 kHz	000111110101100	000111110101100	4012	0.305
	101011001111010	010100110000110	10630	-0.810
	111001100111010	000110011000110	3270	-0.249
	010100110101011	010100110101011	10667	0.812

(Table 5.1 continued)

Input parameters	**Digital Code (2's complement)**	**Digital Code (Binary form)**	**Decimal Equivalent**	**Analog Equivalent (A)V**
3Vp-p, 1 kHz, 256 kHz	110001011111010	001110100000110	7430	-0.679
	101011011100111	010100100011001	10521	-0.962
	001100110111011	001100110111011	6587	0.602
	010011010001000	010011010001000	9864	0.902
3Vp-p, 1 kHz, 256 kHz	111101011101000	000010100011000	1304	-0.119
	100001110000011	011110001111101	15485	-1.416
	000001110000011	000001110000011	899	0.082
	011110111110101	011110111110101	15861	1.451
3Vp-p, 2 kHz, 512 kHz	101101011110111	010010100001001	9481	-0.867
	110100000110010	001011111001110	6094	-0.557
	001100111111100	001100111111100	6652	0.608
	010101110011001	010101110011001	11161	1.021
3Vp-p, 2 kHz, 512 kHz	011001111110000	011001111110000	13296	1.216
	000101001101010	000101001101010	2666	0.243
	100111101101100	011000010010100	12436	-1.138
	111001101010010	000110010101110	3246	-0.297



Accumulate Off

Current Sample Period = 160.0 ns
Next Sample Period = 160.0 ns

sec/Div 50.0 us

Delay 0 s

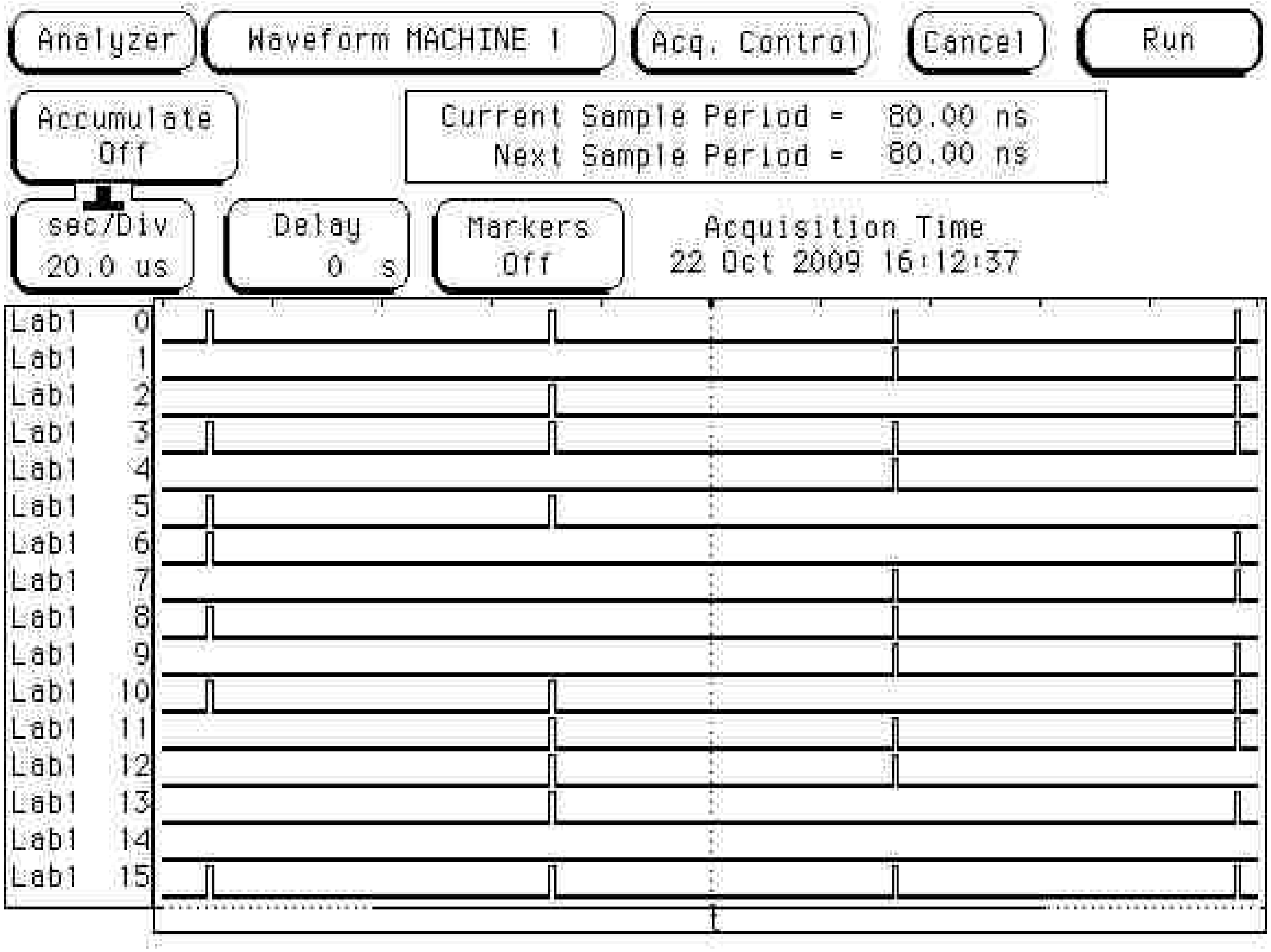
Markers Off

Acquisition Time
22 Oct 2009 16:07:50

Output of the Clock divider

15-bit digital output where Lab1-1 is MSB and Lab1-15 is LSB

Figure 5.7: Experimental results showing the waveforms for four digital output codes for 2.5Vp-p, 1 kHz, 256 kHz.



```
Current Sample Period =  80.00 ns
   Next Sample Period =  80.00 ns
```

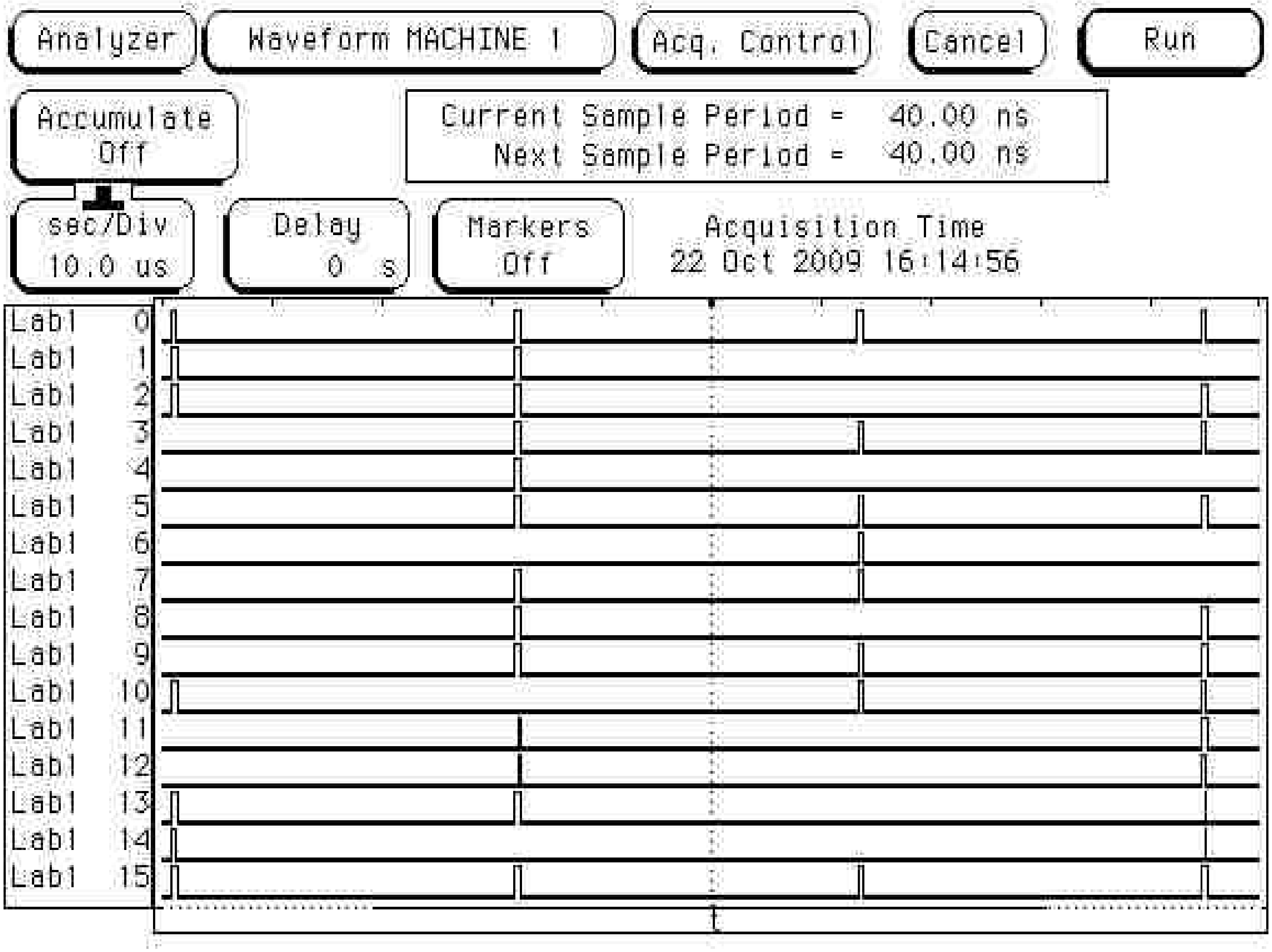
sec/Div 20.0 us

Delay 0 s

Markers Off

Acquisition Time 22 Oct 2009 16:12:37

Figure 5.8: Experimental results showing the waveforms for four digital output codes for 3Vp-p, 2 kHz, 512 kHz.



Accumulate Off

Current Sample Period = 40.00 ns
Next Sample Period = 40.00 ns

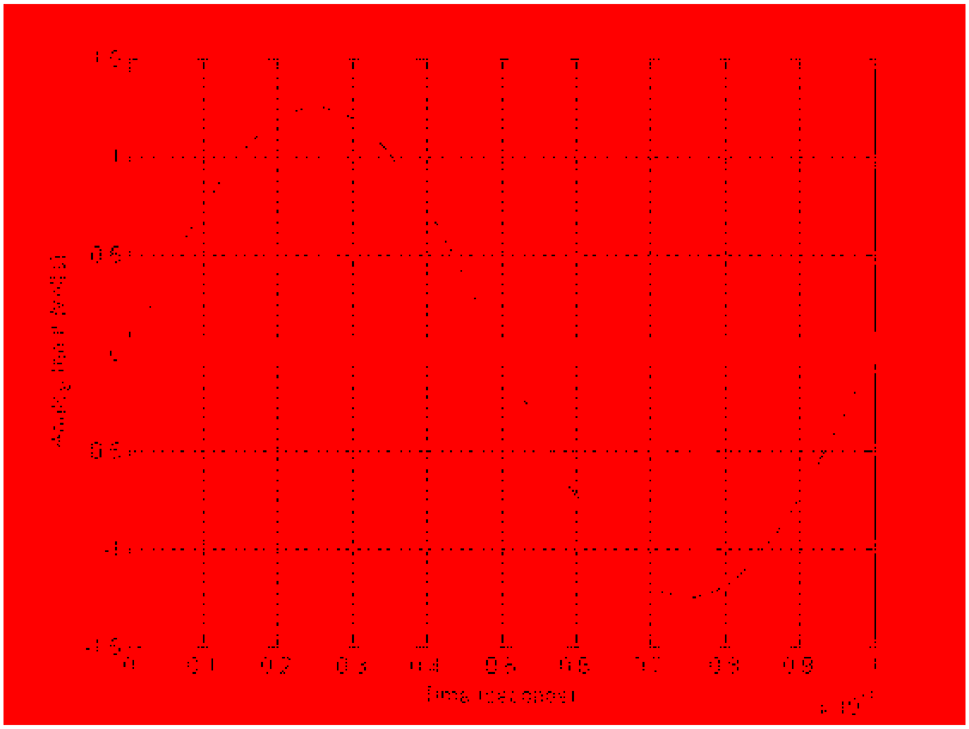
sec/Div 10.0 us

Delay 0 s

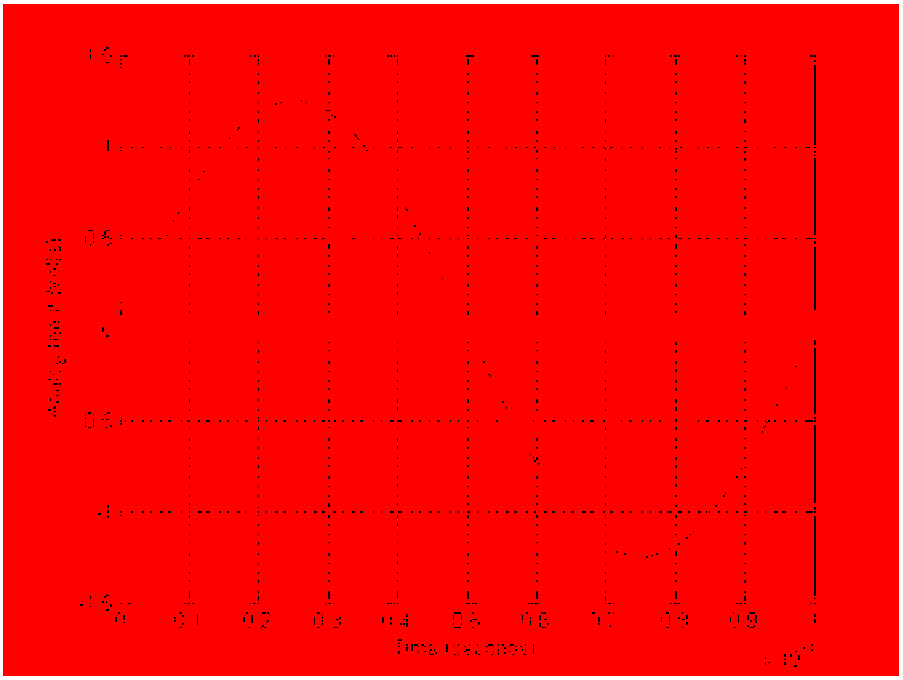
Markers Off

Acquisition Time 22 Oct 2009 16:14:56

Figure 5.9: Experimental results showing the waveforms for four digital output codes for 2.5Vp-p, 4 kHz, 1.024 MHz.

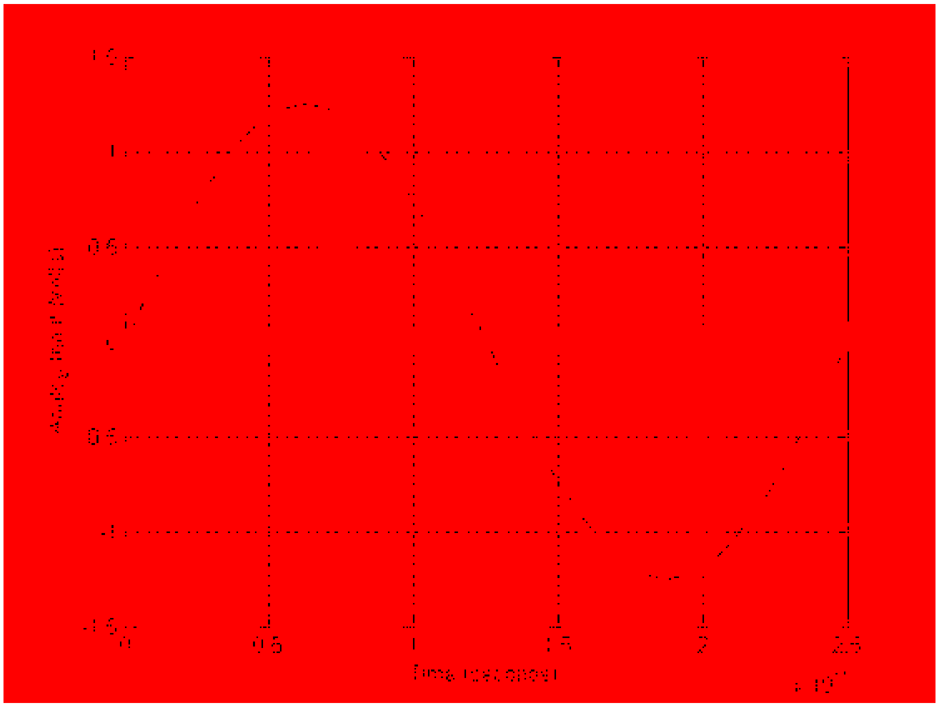


(a)

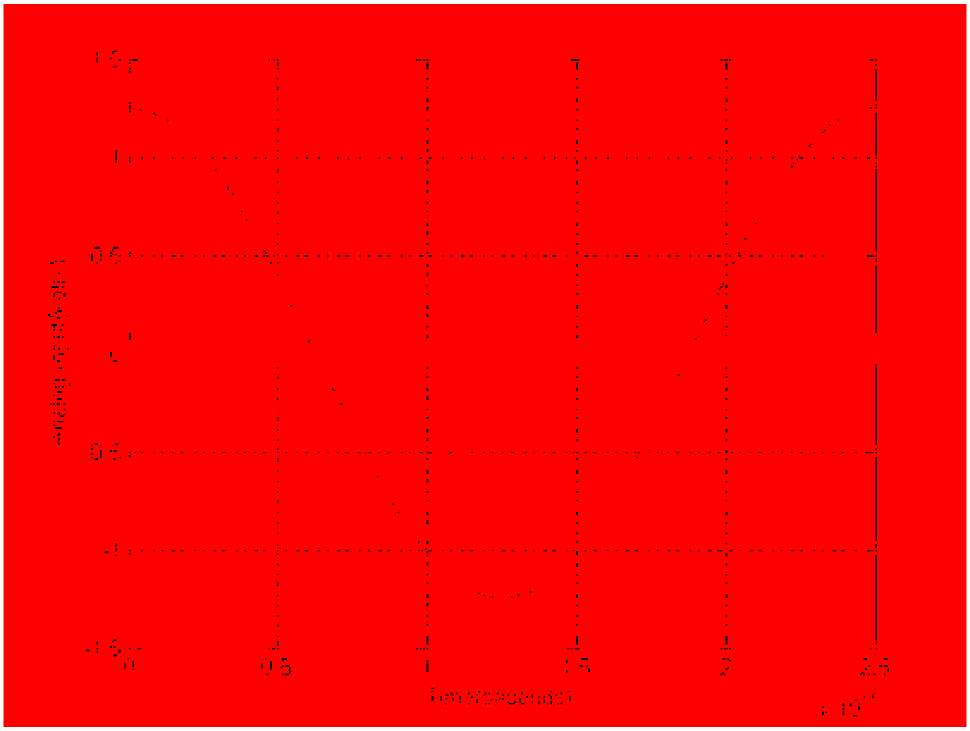


(b)

Figure 5.10: Plot of four digital output codes for 2.5Vp-p, 1 kHz, 256 kHz.

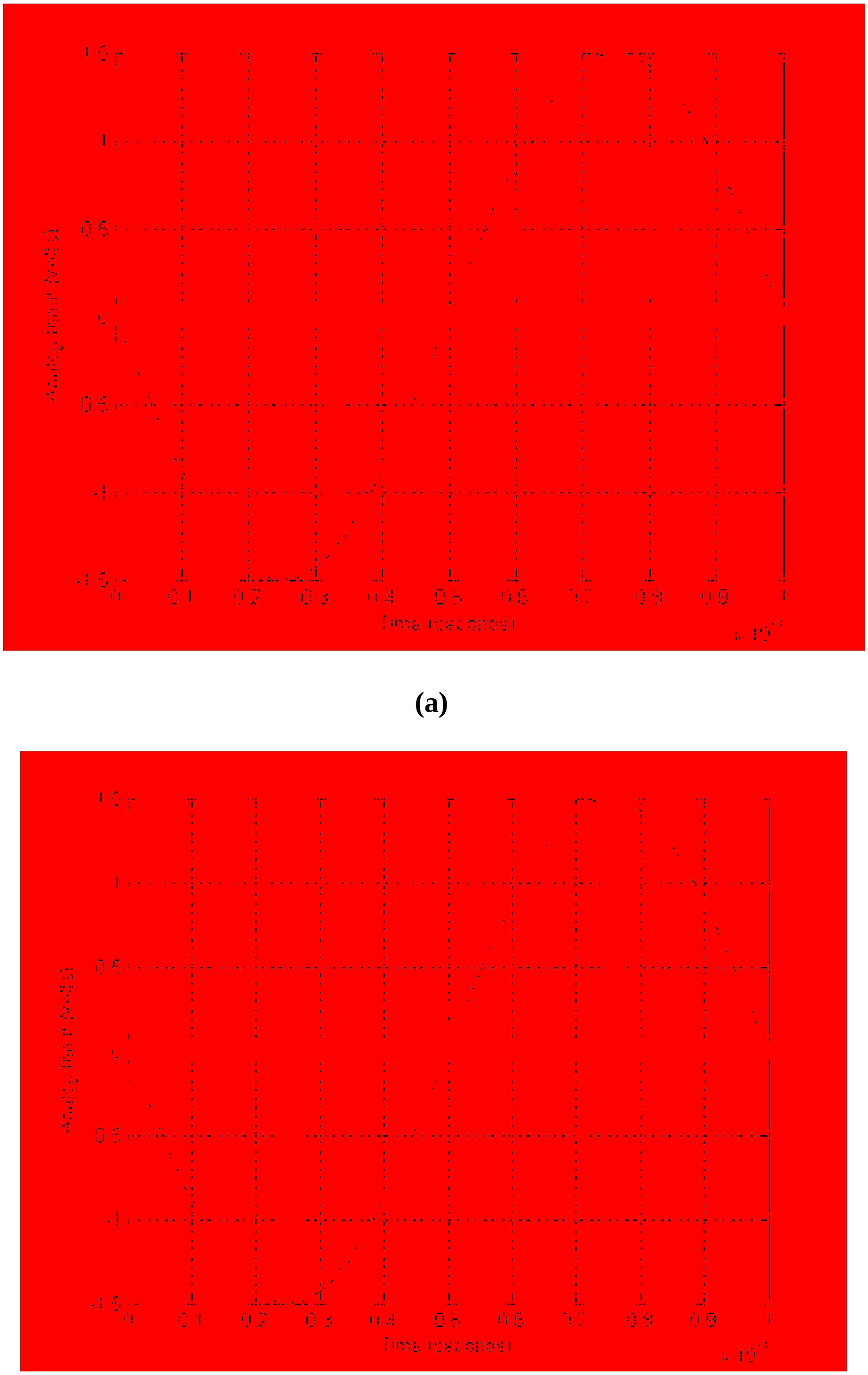


(a)



(b)

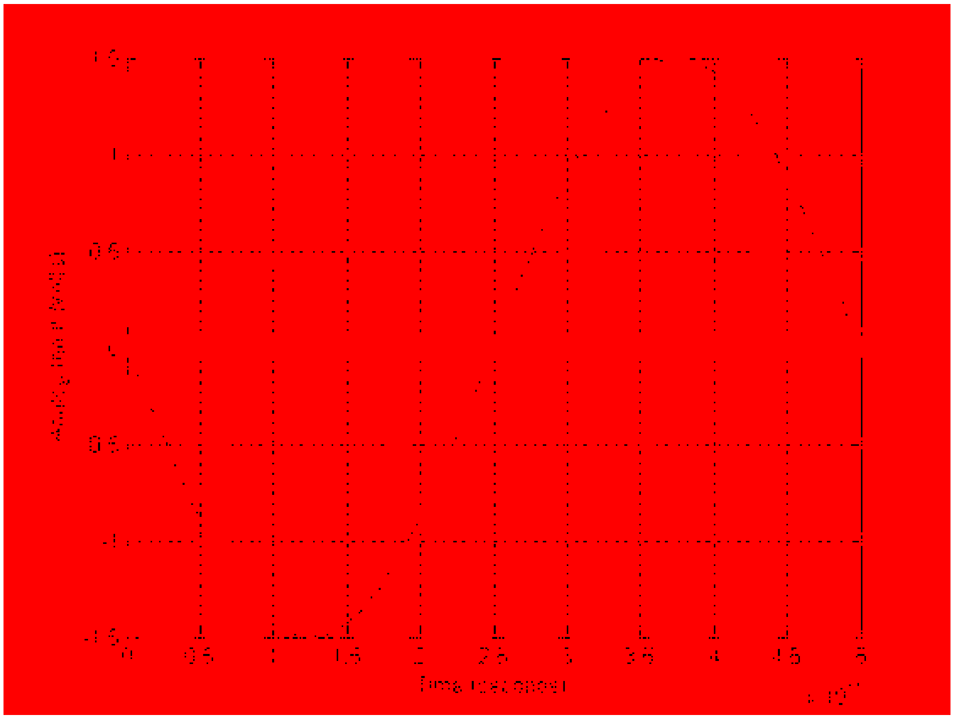
Figure 5.11: Plot of four digital output codes for 2.5Vp-p, 4 kHz, 1024 kHz.



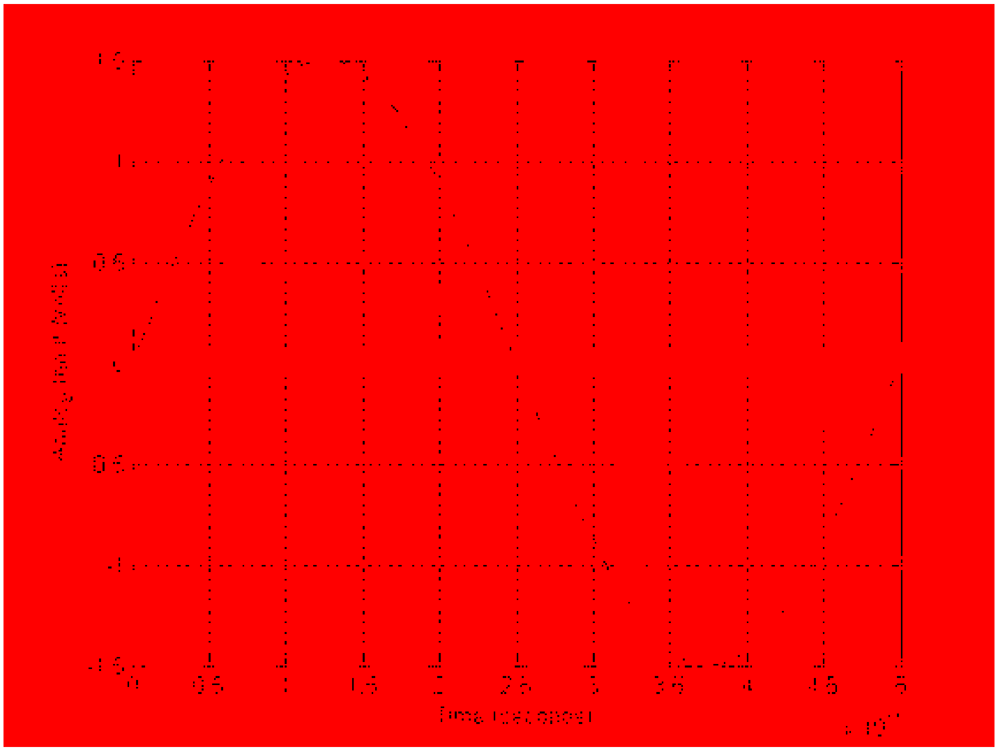
(a)

(b)

Figure 5.12: Plot of four digital output codes for 3Vp-p, 1 kHz, 256 kHz.



(a)



(b)

Figure 5.13: Plot of four digital output codes for 3Vp-p, 2 kHz, 512 kHz.

Chapter 6

Conclusion

A 3rd order cascaded integrated comb filter has been designed and fabricated in 0.5μm n-well CMOS technology. The complete CIC filter is implemented by integrating a three-stage integrator with a three-stage differentiator and a coder circuit. It also includes a clock divider circuit which has been designed to generate a divide by 64 clock signal. The designed decimator can be used to work with either a first order or a second order oversampled sigma-delta modulator with an oversampling ratio of 64. The output of the decimator is a 15-bit digital output which can be used to calculate discrete digital equivalent output voltage values.

The discrete digital values can then be traced using a smooth curve to get back the analog input. The experiments were performed with a clock frequency ranging from 256 kHz to 2.5 MHz and the corresponding input signal bandwidth ranging from 2 kHz to 16 kHz. The integrated ADC (2nd order sigma-delta modulator interfaced with 3rd order CIC filter) has been tested.

The second-order sigma-delta modulator used for testing the CIC decimator designed uses a switched capacitor architecture ADC modulator. Switched capacitor based circuits require boot-strapping techniques to drive the switches in order to extend the dynamic range and sampling rates of the converter. Continuous-time ADCs have been proposed for applications that require lower power requirements, higher signal bandwidths and better noise immunity. Despite the advantages of continuous-time sigma-delta ADCs, audio band ADC implementations still use the discrete time domain as they achieve relatively high linearity, they are very tolerant of clock jitter, and as high signal bandwidths are not required moderate sampling rates can be employed in sigma-delta based ADCs.

There is a droop in the passband of the CIC filter frequency response, which is dependent on the decimation factor, K. Several techniques exist for compensating for the droop in the passband of the CIC filter frequency response. CIC filters preceded by higher performance linear-phase low-pass tapped-delay-line FIR filters can be designed to correct the droop for a particular decimation factor. This compensation filter can be designed to provide frequency correction and spectrum shaping. However, to compensate for variable droop when there is wide variation in the decimation factor, this filter must be programmable, which can be significant in terms of hardware consumption [20]. Filter sharpening can be used to improve the response of a CIC filter. The sharpening of partially non-recursive CIC decimation filters is proposed in reference [20]. This can be taken up as future work on this project.

CIC filters are also used as anti-imaging filters or interpolators (sample rate increase) in designing a sigma-delta digital-to-analog converter. A cascaded integrator-comb (CIC) interpolating filter is a type of digital linear phase finite impulse response (FIR) filter. It can be obtained by interchanging the integrator and differentiator blocks of a CIC decimator. Here the signal is first differentiated and then up sampled by a factor K and then integrated to give the final output. As opposed to CIC decimator where the register widths can be different in each integrator and differentiator stages, in a CIC interpolator the word lengths of the filter sections must be non-decreasing. That is, the word length of each filter section must be the same size as, or greater than, the word length of the previous filter section [16]. A future project can be done using the integrator and differentiator blocks designed for CIC decimator and modify the circuitry accordingly to design a CIC interpolator.

References

1. P. Sutar Ga, D. D. Tang, J. ALtieri, L.E. Thon, G. Coleman, S. Setbbanna, J. Y. C. Sun, "55-mW 300-MHz analog-digital converters using digital VLSI technology," *IEEE Low Power Electronic Symposiums,* pp.68-69, October 1995.

2. Z. Shi, "Sigma-delta ADC and DAC for digital wireless communication," *IEEE Radio Frequency Integrated Circuits (RFIC) Symposium*, pp.57-62, June 1999.

3. S. P. Kommana, "First order sigma-delta modulator of an oversampling ADC design in CMOS using floating gate MOSFETS," *MS (EE) Thesis*, Louisiana State University, December 2004.

4. C. Chou, S. Mohanakrishnan and J. B. Evans , "FPGA implementation of digital filters," *In Proceedings of the Fourth International Conference on Signal Processing Applications and Technology*, pp. 80-88, September 1993.

5. A. Chamakura, "I_{DDQ} testing of a CMOS first order sigma-delta modulator of an 8-bit oversampling ADC," *MS (EE) Thesis*, Louisiana State University, August 2004.

6. B. R. Jacob, H. W. Li, D. E. Boyce, *CMOS Mixed Signal Circuit Design*, IEEE Press, 2002.

7. S. R. Norsworthy, R. Schreier and G. C. Temes, *Delta-sigma Data Converters: Theory, Design, and Simulation*, IEEE Press, 1996.

8. D. Jarman, "A brief introduction to sigma-delta conversion," *Application Note AN9504*, May 1995.

9. Xignal Technologies AG, "Putting continuous time delta-sigma modulators to work," *ADC_whitepaper*, 2005.

10. K. M. Daugherty, *Analog-to-Digital Conversion: A Practical Approach*, Mc Graw-Hill, 1995.

11. D. F. Hoeschele, *Analog-to-Digital and Digital-to-Analog Conversion Techniques,* John Wiley & Sons, 1994.

12. *Signal Processing Toolbox™ 6-user's guide*, The MathWorks, Inc. 3 Apple Hill Drive Natick, MA 01760-2098, September 2009.

13. M. D. Lutovac , D. V. Tosic and B. L. Evans, *Filter design for signal processing using MATLAB and mathematica,* Prentice Hall, 2001.

14. D. J. Goodman and M. J. Carey, “Nine digital filters for decimation and interpolation,” *IEEE Trans. Acoustics, Speech, Signal Processing,* Vol. ASSP-25, pp. 121-126, April 1977.

15. A. Srivastava, R. R. Anantha, “A programmable oversampling sigma-delta analog-to-digital Converter,” *IEEE proceedings of Midwest Symposium on Circuits and Systems*, Vol. 1, pp. 539-542, August 2005.

16. E. B. Hogenauer, “An economical class of digital filters for decimation and interpolation,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 29, pp.155-162, 1981.

17. M. P. Donadio, “CIC filter introduction,” *m.p.donadio@ieee.org, For Free Publication by Iowegian*, July 2000.

18. K. F. Chong, P. K. Gopalakrishnan, T. H. Teo, "Low power approach for decimation filter hardware realization,” *Proceedings of World Academy of Science, Engineering and Technology*, Vol. 32, pp. 2070-3740, August 2008.

19. Altera Corporation, “Understanding CIC compensation filters,” *Application Note 455*, Ver.1.0, April 2007.

20. G. Stephen, R. W. Stewart, "Sharpening of partially non-recursive CIC decimation filters," *Asilomar Conference on Signals Systems and Computers*, Vol. 2, pp. 2170-2173, November 2004.

21. *Filter Design Toolbox™ 4-user’s guide*, The MathWorks, Inc. 3 Apple Hill Drive Natick, MA 01760-2098, September 2009.

22. N. H. E. Weste and D. Harris, *CMOS VLSI Design- A Circuits and Systems Perspective*, 3rd edition, Addison Wisley, NY, 2005.

Appendix A: SPICE Parameters from MOSIS

T77X SPICE BSIM3 VERSION 3.1 PARAMETERS

SPICE 3f5 Level 8, Star-HSPICE Level 49, UTMOST Level 8

* DATE: Feb 6/08
* LOT: T77X   WAF: 8102
* Temperature_parameters=Default

```
.MODEL NMOS NMOS (                                 LEVEL   = 7
+VERSION = 3.1            TNOM    = 27             TOX     = 1.44E-8
+XJ      = 1.5E-7         NCH     = 1.7E17         VTH0    = 0.6194508
+K1      = 0.910282       K2      = -0.0951103     K3      = 29.8575293
+K3B     = -7.3433906     W0      = 1.058963E-8    NLX     = 1.26752E-9
+DVT0W   = 0              DVT1W   = 0              DVT2W   = 0
+DVT0    = 0.7864057      DVT1    = 0.3625457      DVT2    = -0.4435915
+U0      = 450.0046875    UA      = 1E-13          UB      = 1.361584E-18
+UC      = 2.153482E-12   VSAT    = 1.875507E5     A0      = 0.6755421
+AGS     = 0.1293751      B0      = 2.070569E-6    B1      = 5E-6
+KETA    = -3.501501E-3   A1      = 9.562038E-7    A2      = 0.3
+RDSW    = 1.311074E3     PRWG    = 0.0781539      PRWB    = -4.807104E-3
+WR      = 1              WINT    = 1.479507E-7    LINT    = 9.799411E-8
+XL      = 1E-7           XW      = 0              DWG     = 9.644874E-9
+DWB     = 5.604831E-8    VOFF    = -1.46619E-4    NFACTOR = 0.9863252
+CIT     = 0              CDSC    = 2.4E-4         CDSCD   = 0
+CDSCB   = 0              ETA0    = 2.547952E-3    ETAB    = 4.039323E-5
+DSUB    = 0.0819778      PCLM    = 2.9016236      PDIBLC1 = 1.064237E-3
+PDIBLC2 = 2.559761E-3    PDIBLCB = 1.956214E-3    DROUT   = 1.3673462
+PSCBE1  = 7.117852E8     PSCBE2  = 8.459665E-4    PVAG    = 0
+DELTA   = 0.01           RSH     = 78.6           MOBMOD  = 1
+PRT     = 0              UTE     = -1.5           KT1     = -0.11
+KT1L    = 0              KT2     = 0.022          UA1     = 4.31E-9
+UB1     = -7.61E-18      UC1     = -5.6E-11       AT      = 3.3E4
+WL      = 0              WLN     = 1              WW      = 0
+WWN     = 1              WWL     = 0              LL      = 0
+LLN     = 1              LW      = 0              LWN     = 1
+LWL     = 0              CAPMOD  = 2              XPART   = 0.5
+CGDO    = 2.26E-10       CGSO    = 2.26E-10       CGBO    = 1E-9
+CJ      = 4.279824E-4    PB      = 0.9363364      MJ      = 0.4427379
+CJSW    = 3.150575E-10   PBSW    = 0.8            MJSW    = 0.1798474
+CJSWG   = 1.64E-10       PBSWG   = 0.8            MJSWG   = 0.1798474
+CF      = 0              PVTH0   = -0.0410371     PRDSW   = 448.1330927
+PK2     = -0.0870937     WKETA   = -0.0121986     LKETA   = -2.071996E-3 )
```

```
.MODEL PMOS PMOS (                                 LEVEL   = 7
+VERSION = 3.1            TNOM    = 27             TOX     = 1.44E-8
+XJ      = 1.5E-7         NCH     = 1.7E17         VTH0    = -0.9152268
+K1      = 0.553472       K2      = 7.871921E-3    K3      = 42.252426
+K3B     = -3.0480702     W0      = 8.766078E-6    NLX     = 2.036336E-7
+DVT0W   = 0              DVT1W   = 0              DVT2W   = 0
+DVT0    = 1.2296676      DVT1    = 0.2771302      DVT2    = -0.0564871
+U0      = 201.3603195    UA      = 2.408572E-9    UB      = 1E-21
+UC      = -1E-10         VSAT    = 1.082183E5     A0      = 0.8423996
+AGS     = 0.0746069      B0      = 4.26808E-7     B1      = 5E-6
+KETA    = -4.865785E-3   A1      = 1.575105E-4    A2      = 0.4934288
+RDSW    = 3E3            PRWG    = -0.030364      PRWB    = -0.0443579
+WR      = 1              WINT    = 2.460037E-7    LINT    = 1.186871E-7
+XL      = 1E-7           XW      = 0              DWG     = -1.877936E-9
+DWB     = -2.747281E-9   VOFF    = -0.0728194     NFACTOR = 0.7347278
+CIT     = 0              CDSC    = 2.4E-4         CDSCD   = 0
+CDSCB   = 0              ETA0    = 1.000997E-3    ETAB    = -0.2
+DSUB    = 1              PCLM    = 2.132793       PDIBLC1 = 0.0445882
+PDIBLC2 = 3.351617E-3    PDIBLCB = -0.0294204     DROUT   = 0.2352764
+PSCBE1  = 8E10           PSCBE2  = 7.747623E-9    PVAG    = 5.428584E-3
+DELTA   = 0.01           RSH     = 107.3          MOBMOD  = 1
+PRT     = 0              UTE     = -1.5           KT1     = -0.11
+KT1L    = 0              KT2     = 0.022          UA1     = 4.31E-9
+UB1     = -7.61E-18      UC1     = -5.6E-11       AT      = 3.3E4
+WL      = 0              WLN     = 1              WW      = 0
+WWN     = 1              WWL     = 0              LL      = 0
+LLN     = 1              LW      = 0              LWN     = 1
+LWL     = 0              CAPMOD  = 2              XPART   = 0.5
+CGDO    = 3.22E-10       CGSO    = 3.22E-10       CGBO    = 1E-9
+CJ      = 7.383453E-4    PB      = 0.9639839      MJ      = 0.4986769
+CJSW    = 2.582508E-10   PBSW    = 0.99           MJSW    = 0.3288273
+CJSWG   = 6.4E-11        PBSWG   = 0.99           MJSWG   = 0.3288273
+CF      = 0              PVTH0   = 5.98016E-3     PRDSW   = 14.8598424
 +PK2     = 3.73981E-3     WKETA   = 0.0120179      LKETA   = -8.42708E-3    )
```

Appendix B: Pin Diagram

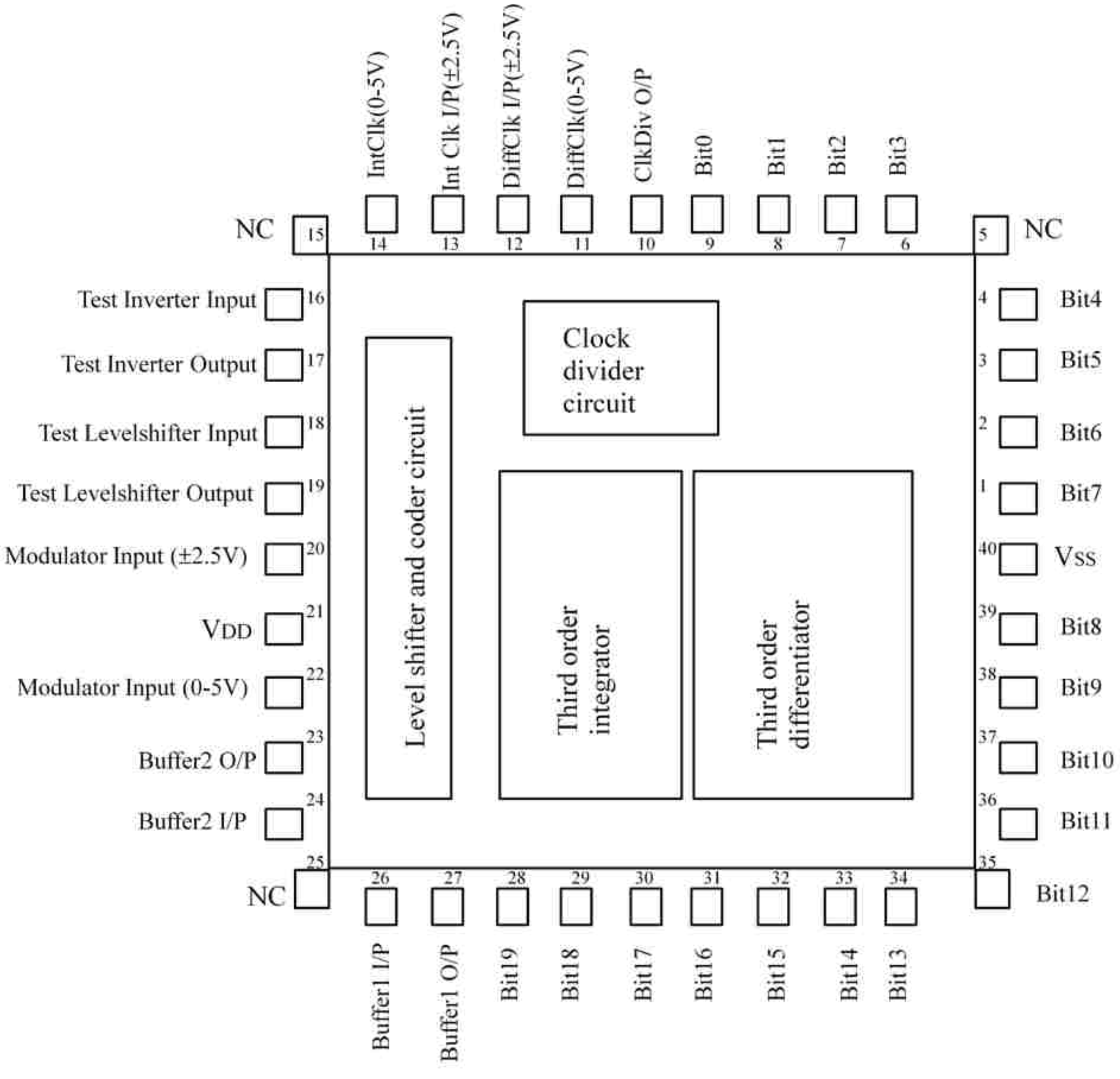


Figure B.1: Pin diagram of the decimator IC (T95S-AW).

Pin Description:

Pin No.	Description
1	Bit 7
2	Bit 6
3	Bit 5
4	Bit 4(discarded)
5	NC
6	Bit 3(discarded)
7	Bit 2(discarded)
8	Bit 1(discarded)
9	Bit 0(discarded)
10	Clk Div Output
11	Differentiator Clock1 (0-5V)
12	Differentiator Clock1 Input(±2.5V)
13	Integrator Clock Input(±2.5V)
14	Integrator Clock(0-5V)
15	NC
16	Test Inverter Input
17	Test Inverter Output
18	Test Levelshifter Input
19	Test Levelshifter Output
20	Modulator Input (±2.5V)

Pin No.	Description
21	VDD(5V)
22	Modulator Input (0-5V)
23	Buffer2 Output
24	Buffer2 Input
25	NC
26	Buffer1 Input
27	Buffer1 Output
28	Bit 19
29	Bit 18
30	Bit 17
31	Bit 16
32	Bit 15
33	Bit 14
34	Bit 13
35	Bit 12
36	Bit 11
37	Bit 10
38	Bit 9
39	Bit 8
40	VSS(0V)

Appendix C: MATLAB Code

C1: MATLAB CODE used for Figure 2.9(a) [12]

Decimate a signal by a factor of four:

```
t = 0:.00025:1;                  % Time vector
x = sin(2*pi*30*t) + sin(2*pi*60*t);

y = decimate(x,4);

stem(x(1:120)), axis([0 120 -2 2])   % Original signal
title('Original Signal')

stem(y(1:30))                    % Decimated signal
title('Decimated Signal')
```

- y = decimate(x,r) reduces the sample rate of x by a factor r. The decimated vector y is r times shorter in length than the input vector x. By default, decimate employs an eighth-order lowpass Chebyshev Type I filter with a cutoff frequency of 0.8*(Fs/2)/r. It filters the input sequence in both the forward and reverse directions to remove all phase distortion, effectively doubling the filter order.
- y = decimate(x,r,n) uses an order n Chebyshev filter.
- y = decimate(x,r,'fir') uses an order 30 FIR filter, instead of the Chebyshev IIR filter. Here decimate filters the input sequence in only one direction. This technique conserves memory and is useful for working with long sequences.
- y = decimate(x,r,n,'fir') uses an order n FIR filter.

Vita

Hemalatha Mekala was born in February, 1985, in Chittoor, Andrapradesh, India. She received her Bachelor of Technology in Electrical and Electronics Engineering from Jawaharlal Nehru Technological University, Hyderabad, Andrapradesh, India, in May 2006. She enrolled in the Department of Electrical and Computer Engineering at Louisiana State University, Baton Rouge, Louisiana, in August 2006 to attend graduate school. Her research interests are in VLSI/Circuit design, Mixed signal circuits and Digital signal processing.