AN ABSTRACT OF THE THESIS OF

Man W. Wong for the degree of Master of Science in

Electrical and Electronics Engineering presented on November 17, 1993.

Title: Low Noise FSCL Digital Circuits For Decimation Filter

Abstract approved: Redacted for privacy

Sayfe Kiaei

A new circuit technique called Folded Source Coupled Logic (FSCL) has been developed to implement the digital section of mixed-signal IC applications. This FSCL circuit technique offers the advantage of low overlap current spikes during the switching transitions of conventional CMOS gates. This overlap current spike has become one of the major obstacles in improving the accuracy and performance of mixed-signal IC applications. Using simple circuits, FSCL logic family can be interfaced with the existing CMOS family. Thus it can nearly eliminate the power noise issue in the mixed-signal IC design.

In this thesis, design of a $sinc^3$ decimation filter using the FSCL technique for a 2^{nd} order delta-sigma modulator has been presented. Simulation results show that this particular decimation filter, using the newly developed FSCL technique, improves the performance of the mixed-signal system.

Low Noise FSCL Digital Circuits For Decimation Filter

by

Man W. Wong

THESIS

Submitted to

Oregon State University

in partial fulfillment of

the requirements for the

degree of

Master of Science

Completed November 17, 1993

Commencement June 1994

APPROVED:

Redacted for privacy

Associate Professor of Electrical and Computer Engineering in Charge of Major

Redacted for privacy

Head of Department of Electrical and Computer Engineering

Redacted for privacy

Dean of Graduate School

Date thesis is presented November 17, 1993

Typed by researcher for Man W. Wong

ACKNOWLEDGEMENTS

I would like to thank my major professor, Dr. S. Kiaei for allowing me to work on this great project. I also thank him for all the encouragement and guidance through this entire project.

I would also like to thank my colleagues, Manu Shrivastava, PVS, Sridhar Jasti, Charlie Chairatanatrai, Rita Wells, for their help and hours of interesting discussions.

I thank Dr. S. Lu, Dr. J. Herzog and Dr. M. Koretsky, the member of my graduate committee, for their time and effort to preside my defense.

Further I would like thank my personal friends, Bernard Ficq, David Wong, Simon Lam and all other friends in Oregon State University, for their constant support. They made my life more interesting during my stay in Corvallis.

Most importantly, I'm grateful to my family Mrs. and Mr. Wong, my sister Isabel and my beloved fiancee Megan. Their total faith and endless support in giving me the sense of confidence that I needed. Their love and affection have been the most valuable source to my effort. Therefore, I dedicate this thesis to my family.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LOW NOISE FSCL DIGITAL CIRCUITS FOR DECIMATION FILTER

Chapter 1. Introduction

With the advent of VLSI technology, oversampling delta-sigma ($\Delta\Sigma$) modulators are becoming more and more popular for analog-to-digital data conversion. Due to the rapid development of mixed-signal ICs, design of analog and digital circuitry on a same IC chip have emerged for complex applications. The delta-sigma data converter consists of two parts: a relatively small analog modulator and a large digital decimation filter. In this case, due to the large transient current spike noise generated by the digital section, the operation of the sensitive analog components is hampered. For example, Figure 1.1 shows the noise spikes generated by the switching of MOSFETs in conventional static CMOS digital circuits. This noise is propagated to the analog side via the supply lines or common substrate which may degrade the performance of the analog circuitry[1]. Due to this digital noise, there is presently a limitation on the accuracy of mixed-signal integrated systems.

Though, static CMOS logic offers many advantages such as low static power, simplicity of design, and high packing density, it typically generates current noise spikes as large as 1 mA/gate. On a single chip where many thousands of transistor may switch synchronously, very large transient current will affect the analog section.

To circumvent the problem of noise propagation in mixed-signal ICs, several techniques have been developed. Good layout techniques such as using separate analog and digital power supply lines, and the minimization of their common impedance are marginally effective. The use of diffused guard rings to shield circuit noise may have some improvement.

The circuit techniques termed Folded Source Coupled Logic (FSCL) has recently been developed and has proved to be very effective in reducing the digital switching noise in CMOS mixed-signal ICs. Using FSCL technique for mixed-signal applications such as delta-sigma data converter, high accuracy and low noise can be achieved for the digital decimation filter.

The objective of this thesis is to design and implement a high performance decimation filter using the low noise FSCL technique for a second-order delta-sigma modulator. The decimation filter is used for a multi-bit delta-sigma A/D converter. In Chapter 2, some general system aspects of the delta-sigma modulator and the decimation filter are described. Chapter 3 includes a detailed design of the decimation filter employing the FSCL technique. Chapter 4 presents the chip layout of the system and the simulation results and noise issue were discussed as well.

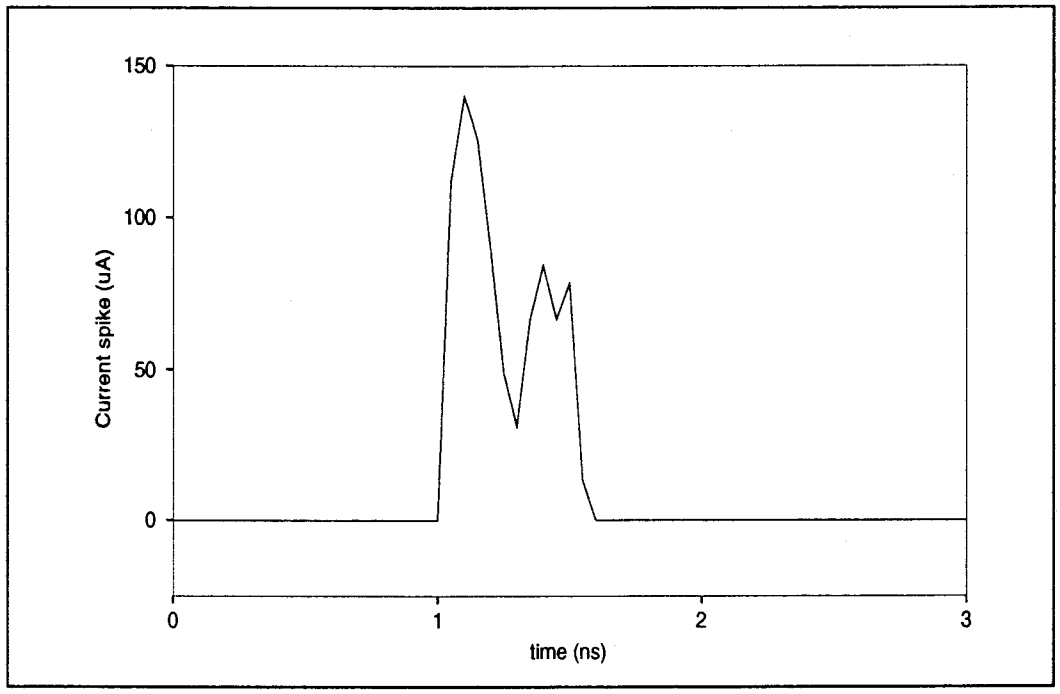


Figure 1.1 Current spike for a CMOS inverter

Chapter 2. Oversampling Delta-Sigma Modulator

2.1 Oversampling Modulation

The main advantage of the delta-sigma modulation data converter is that it incorporates oversampling and quantization noise shaping to provide high resolution without using highly accurate analog components.

Figure 2.1 shows a typical oversampling Pulse Code Modulation (PCM) encoder [3]. Delta-sigma modulator is an example of this type of modulator. In this figure, the output of the modulator consists of the input signal together with its out-of-band components, modulation noise (quantization noise) and circuit noise. A digital decimation filter is designed to remove the undesired noise and perform down sampling. Decimation is required to reduce the high sampling rate of modulated signals to a lower rate which is more suitable for processing and also the decimation process can improve the resolution. In order to improve the resolution, the decimation process will increase the word length of the digitally encoded signals. By doing so, the decimation is being performed by averaging (filtering and accumulating) the quantized signal.

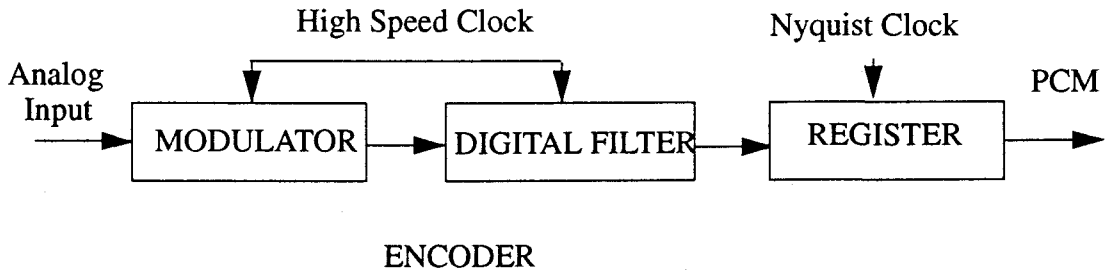


Figure 2.1 Oversampling Pulse Code Modulation Encoder

2.2. Delta-Sigma Modulator

The basic principle of the delta-sigma modulation is to shape the quantization noise in such a way that most of its energy is pushed into high frequencies. Therefore, after modulation, noise can be filtered out by the low-pass decimation filter.

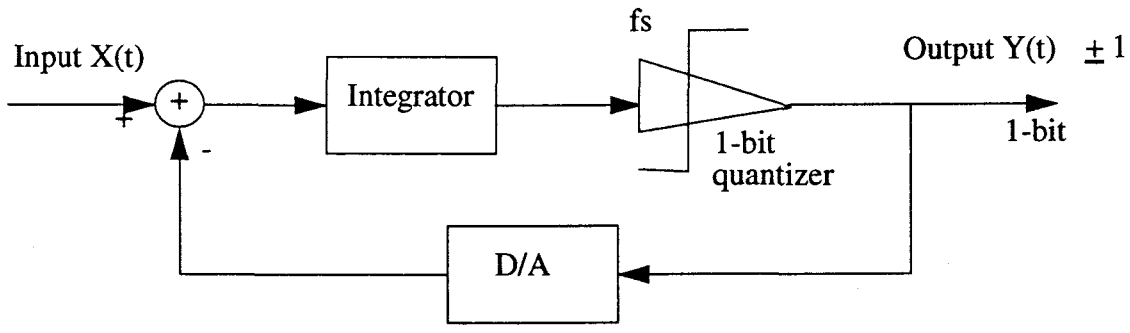


Figure 2.2 First Order Delta-Sigma Modulator

Figure 2.2 shows a basic fist order delta-sigma modulator[4]. The input signal is fed to the quantizer (A/D) via an integrator and the quantized output is fed back and subtracted from the input. This feedback forces the average value of the quantized signal to track the average value of the input signal. Any difference between quantized output and the input is integrated to minimize the error. If the input signal is sampled sufficiently, the quantization noise can be modelled as additive white noise uncorrelated with the input signal as shown in Figure 2.3.

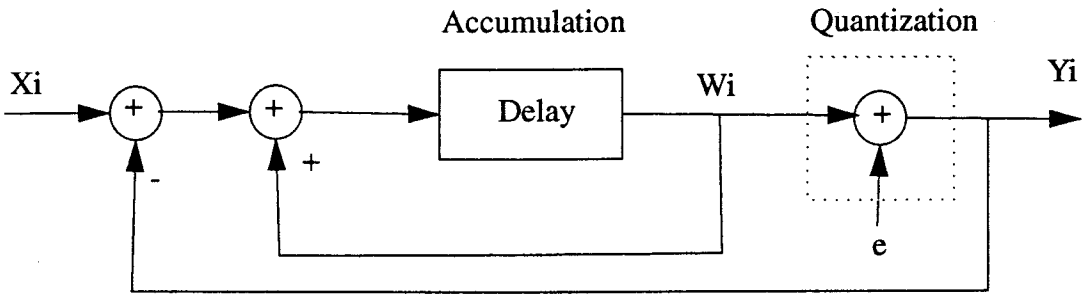


Figure 2.3 Quantization Noise Model of first Order Delta-Sigma Modulator

The rms value of the spectral density of the white noise E is

$$E(f) = \bar{e}\sqrt{2\tau} = \sigma\sqrt{\frac{\tau}{6}}$$

Where τ is the spacing of the sampling time, $\tau = \frac{1}{f_s}$, f_s is the sampling frequency, σ is the

quantization step, $\bar{e}$ is the rms value of the white noise, $\frac{\sigma}{\sqrt{12}}$.

Assuming linearity, the Z-transform of the modulated output is

$$Y(z) = Z^{-1}X(z) + (1 - Z^{-1})E(z)$$

The quantization noise has a transfer function of $(1\text{-}Z^{-1})$ E(z), where $(1\text{-}Z^{-1})$ is a high pass filter. So the frequency spectral density of the quantization noise is

$$N(f) = 2\bar{e}\sqrt{2\tau}\sin\left(\frac{\omega\tau}{2}\right)$$

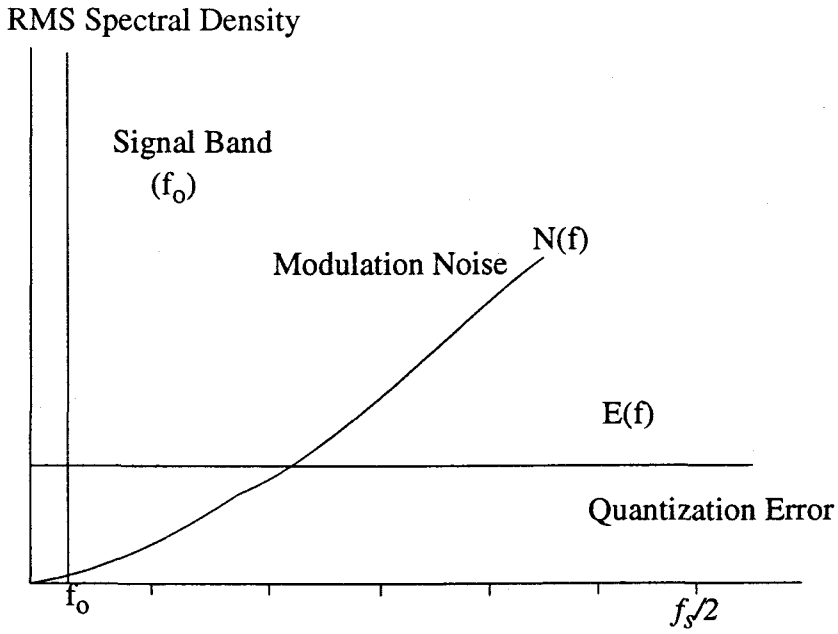


Figure 2.4 Spectral density of the noise N(f) from Delta-Sigma quantization compare with that of ordinary quantization E(f), OSR=16

Figure 2.4 shows that the feedback shapes the quantization noise and reduces the noise at lower frequencies, but increases the noise at high frequencies. The in band noise rms value can be expressed as $N_0 = \frac{\bar{e}}{\sqrt{3}}\pi(2f_0\tau)^{3/2}$.

The above equation shows that each doubling of the oversampling ratio (i.e. $\tau' = \frac{1}{2}\tau$) will reduce the in band noise by 9dB and provide 1.5 bits of extra resolution [4], as $20log(2)^{3/2}$ =9dB. This improvement in the resolution requires that the modulated signal be decimated to the Nyquist rate as the signal is being processed in an oversampling rate by the modulator. This requires a sharp cut-off low pass filter to attenuate the high frequency noise.

2.3 Comb Decimation Filter

A simple method for the decimation filter is to use a comb filter. The comb filter performed simple averaging of the input signal as: $H(z) = \frac{1}{N}\sum_{i=0}^{N-1} Z^{-i}$.

The transfer function of the 1st order comb decimation filter is $H(z) = \frac{1}{N}\left(\frac{1-z^{-N}}{1-z^{-1}}\right)$, N is the decimation ratio $N = \frac{f_S}{f_D}$, f_D is the decimation frequency.

For $Z = e^{j\omega\tau}$ the frequency response of the comb filter is $H(e^{j\omega\tau}) = \frac{\sin c(\pi f N\tau)}{\sin c(\pi f\tau)}$

where $\sin c(f) = \frac{\sin(\pi f)}{\pi f}$.

The the above function is also called *sinc* function filter. The motivation of using this type of filter for the decimation is that these filters are easy to build and their action is easy to predict.

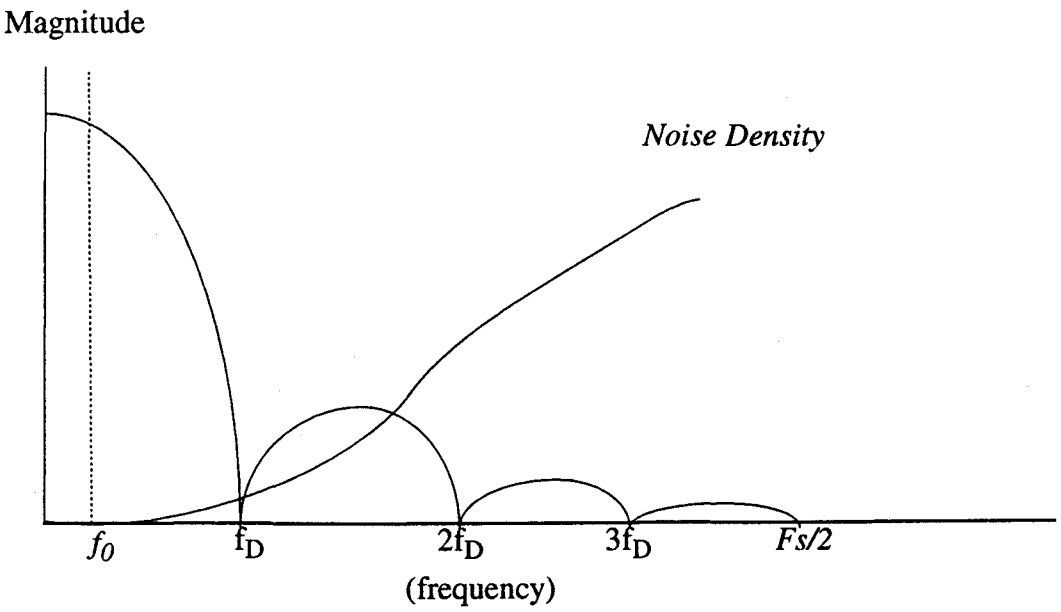


Figure 2.5 Frequency response of first order comb filter

Figure 2.5 shows that the function has zeros at nf_D in the range $f_O \leq f < f_s$.

2.4 Higher Order Decimation Filter for Delta-Sigma Modulator

The resolution of the delta-sigma modulation can be improved by using a cascaded second order modulator without increasing the sampling rate. The modulator as shown in Figure 2.6 is to place another feedback around the quantizer. In this case, in band quantization noise is shaped by a second order high-pass filter $(1-Z^{-1})^2E(z)$. The in band noise rms value is

$$N_0 = \bar{e}\left(\frac{\pi^2}{\sqrt{5}}\right)(2f_0\tau)^{5/2}$$

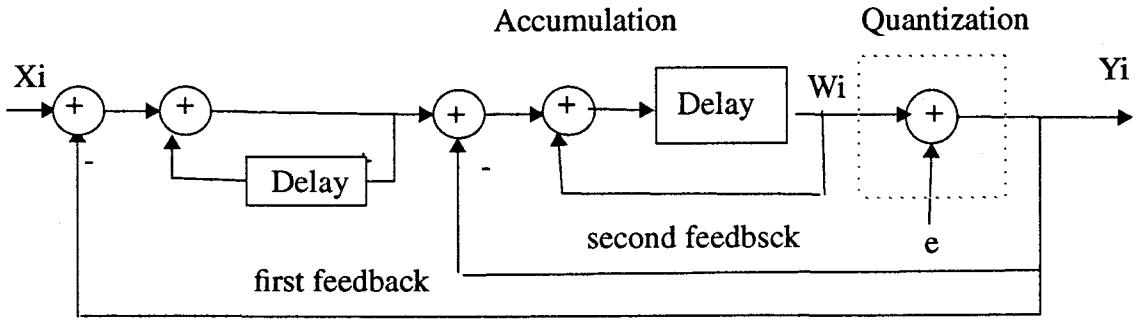


Figure 2.6 Quantization Noise Model of second order Delta-Sigma Modulator

Now by doubling of the sampling frequency, the quantization noise is reduced by 15 dB which provides 2.5 bits of additional resolution. This process results in a more stringent requirement for the decimation filter. It has been shown that in general, for a L-th order delta-sigma modulator, the optimum *sinc* function decimation filter has to be of order L+1 as:

$$H(e^{jw\tau}) = \left[\frac{\sin c\,(\pi f N\tau)}{\sin c\,(\pi f\tau)}\right]^{(L+1)}$$

This thesis, focuses on a design of a decimation filter for a 2nd order delta-sigma modulator (L=2). A 3rd order *sinc* function decimation filter with OSR, N=32 is used.

$$H(z) = \frac{1}{32}\left[\frac{1-Z^{-32}}{1-Z^{-1}}\right]^3$$

The frequency response of this *sinc* function decimation filter is

$$H(e^{j\omega\tau}) = \frac{\sin c^3(\pi f(32)\tau)}{\sin c^3(\pi f\tau)}$$

2.5 Block Diagram of a 3rd order *sinc* Function Decimation Filter

This 3rd order *sinc* function decimation filter can be implemented by the accumulate-and-dump topology[4], shown in Figure 2.7.

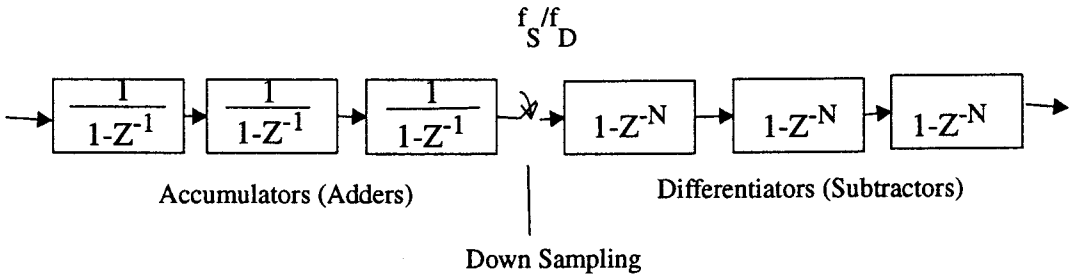


Figure 2.7 System diagram of the $sinc^3$ decimation filter

Figure 2.7 shows a time domain representation of the decimation filter. The system transfer function $H(z) = \frac{1}{32}\left[\frac{1-Z^{-32}}{1-Z^{-1}}\right]^3$, can be represented as the all poles (auto regressive) by $H_1(z)$ and all zeroes (moving average) by $H_2(z)$.

Then $H(z)=H_1(z)H_2(z)$

where: $H_1(z) = \frac{1}{(1-Z^{-1})^3}$, $H_2 = (1-Z^{-32})^3$.

The integrator $H_1(z)$ is three cascaded accumulators as the coefficients of the auto regressive accumulation can be generated by the system shown in Figure 2.8. The differentiators implement the transfer function $H_2(z)$ is also shown in the same figure.

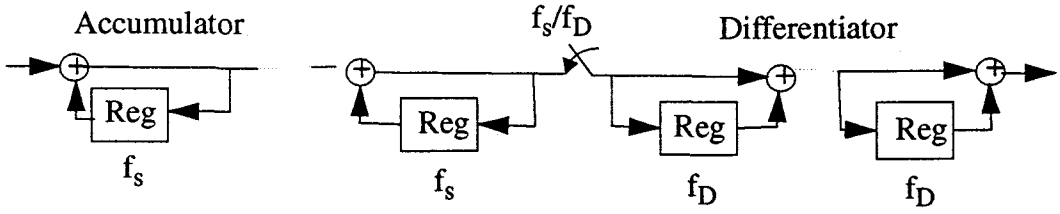


Figure 2.8. $Sinc^K$ decimating circuit that comprises K accumulators followed by downsampling and K differentiators

The overall system specifications are as follow. The quantizer is a 5-bit quantizer. For the accumulators the required word length is $Klog(N)+b$ bit [8]. With N=32 and K=3, the maximum number of bits of the output word is $3log(32)+5$=20 bits. The main purpose of using this formula is to provide sufficient number of bits for the hardware to generate enough data point for all coefficients required by the decimation filter.

A carry-skip adder is used for the accumulators(adder) and for the differentitors(subtractors), a bit-serial topology is adopted. A parallel-to-serial converter is designed to perform 20-bit to 1-bit data conversion. The parallel-to-serial converter is combined with the decimation register. This allows the converter to store the data during downsampling and per-

form the parallel-to-serial data conversion. The complete block diagram of the decimation filter is shown in Figure 2.9. A more detailed discussion of the system implemented by the low noise FSCL circuit technique will be described in Chapter 3.

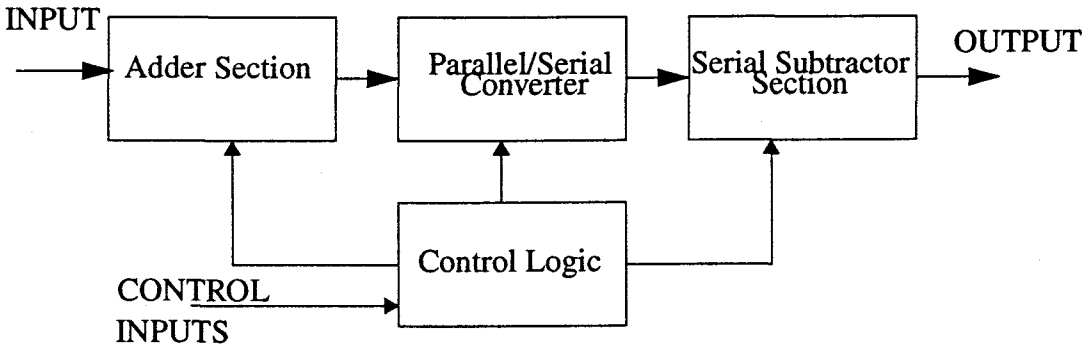


Figure 2.9 Block diagram of the $sinc^3$ decimation filter

Chapter 3. Decimation Filter Implementation Using FSCL

3.1 Architecture of the Decimation Filter

Figure 3.1 shows the schematic of the complete $sinc^3$ decimation filter in the form of accumulate-and-dump topology.

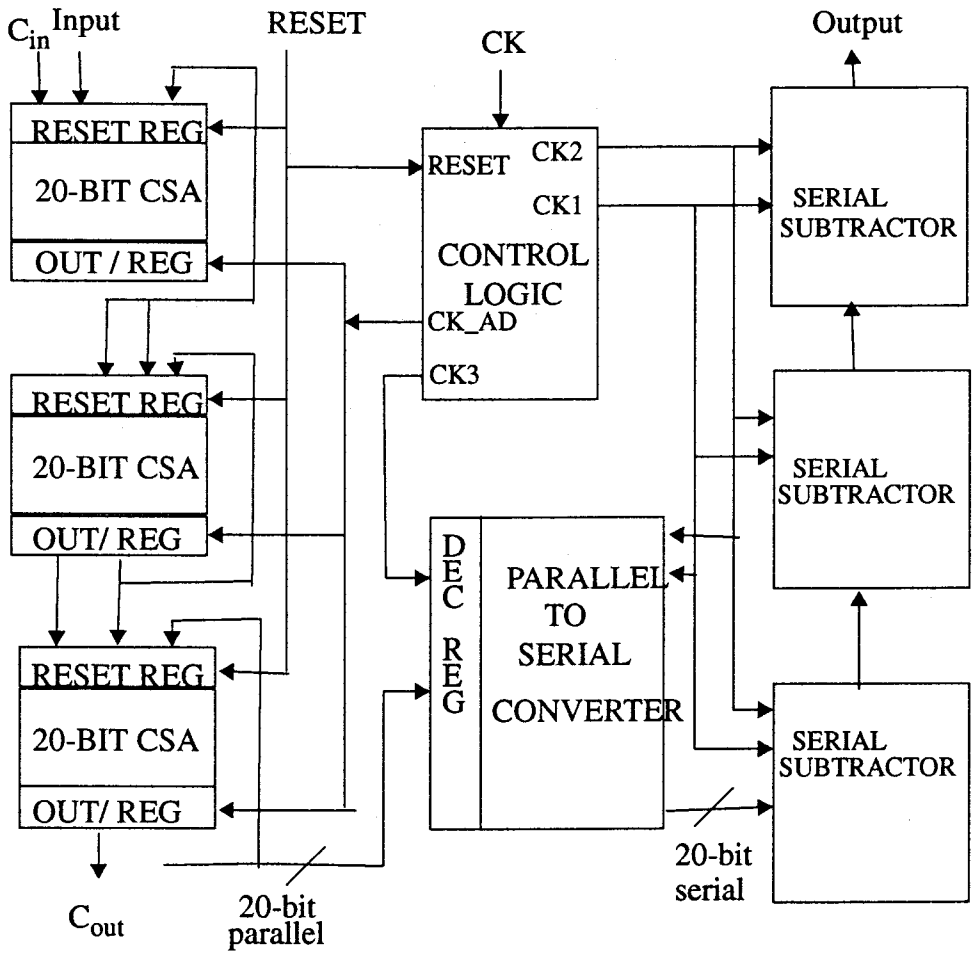


Figure 3.1 Schematic of the $sinc^3$ decimation filter

The input and output timing diagram of the decimation filter is also shown in Figure 3.2

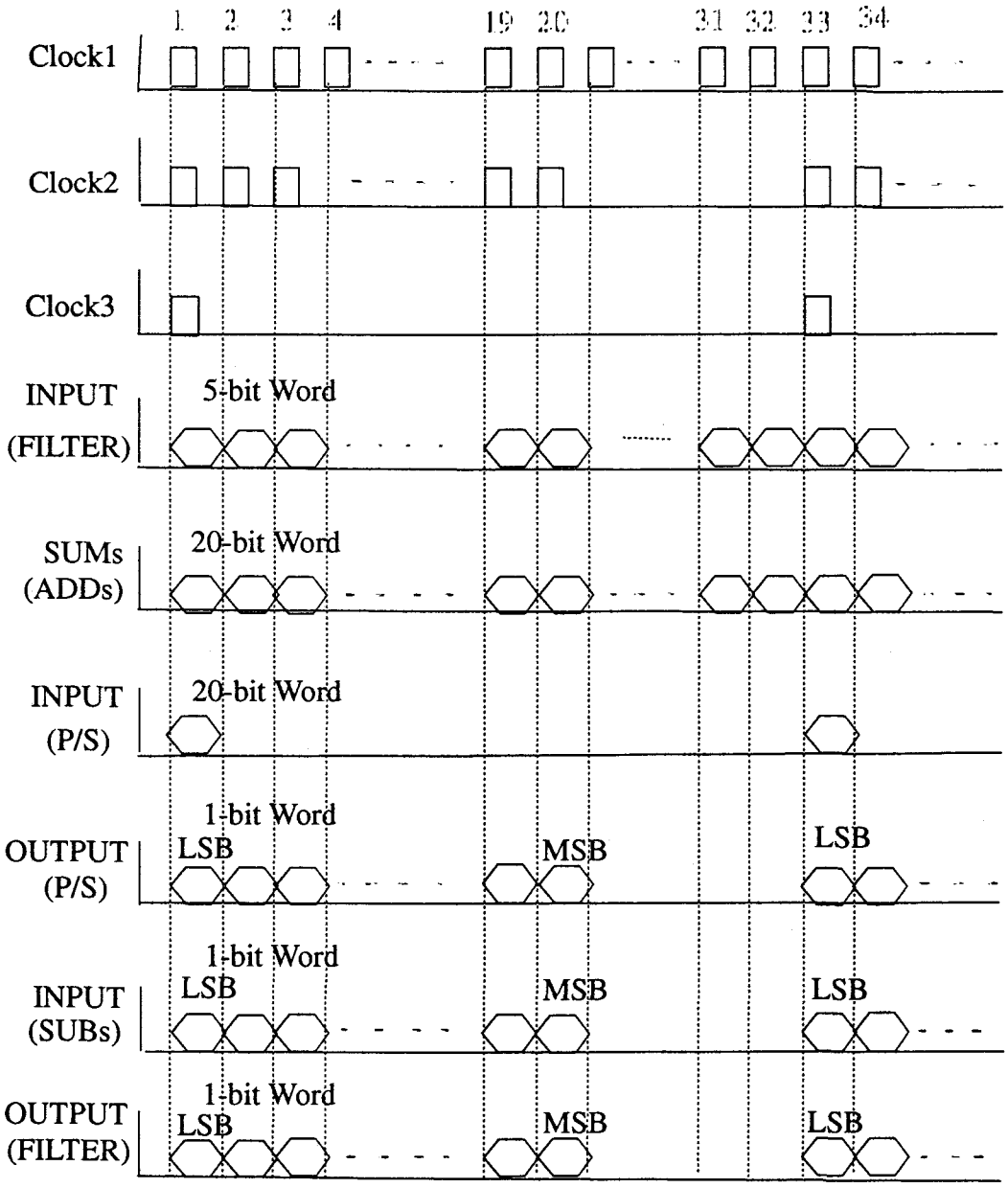


Figure 3.2 Input and output timing diagram of the decimation filter

Figure 3.2 shows that with a 5-bit input word of the modulated signal, a 20-bit serial output data is generated at each 32 clock cycles. There are inputs and outputs occur in each

specific time period for each individual blocks. Assuming in a steady state condition, that is all data are valid in each block. In the clock 1 cycle, a 5-bit (sign-extended to 20-bit) input is taken by the adder section, the adder will produced a 20-bit parallel data at each clock cycle, then after downsampling at each 32^{nd} clock cycle, a 20-bit parallel output data (SUMs) will be loaded into the parallel-to-serial converter. The converter then converse the 20-bit parallel data into a 20-bit serial data following CK2. The 20-bit serial data then propagate through the three cascaded serial subtractors, where each of the subtractor will perform a subtraction between its current input and its 32^{nd} predecessor. Therefore, a complete 20-bit serial output data will be generated only each 32 clock cycles.
The operating frequency of the decimation filter depends on the maximum speed of the logic circuits used to realized the filter. Form Figure 3.1, the critical delay is assumed in the adder and this will be verified by the simulation results. The adder is essential, as the adder operates with the original data sampling clock and normally it should have the longest data path.

3.2 Carry-Skip-Adder (CSA)

Figure 3.1 showed that three accumulators are required to implement the filtering function of the integrators. Note that the adder section of the decimation filter represents a trade-off between die area and speed. A conventional carry-ripple-adder (CRA) would be less complicated and more area efficient than a fully carry-lookahead-adder (CLA). However, the slow operation of CRA would not be appropriate for a high-speed digital system, such as the digital decimation filter. On the other hand, the excessive complexity and the large number of transistors required for a fully CLA would not be favorable in the application of FSCL techniques. Therefore, compromising between the merits and disadvantages of the

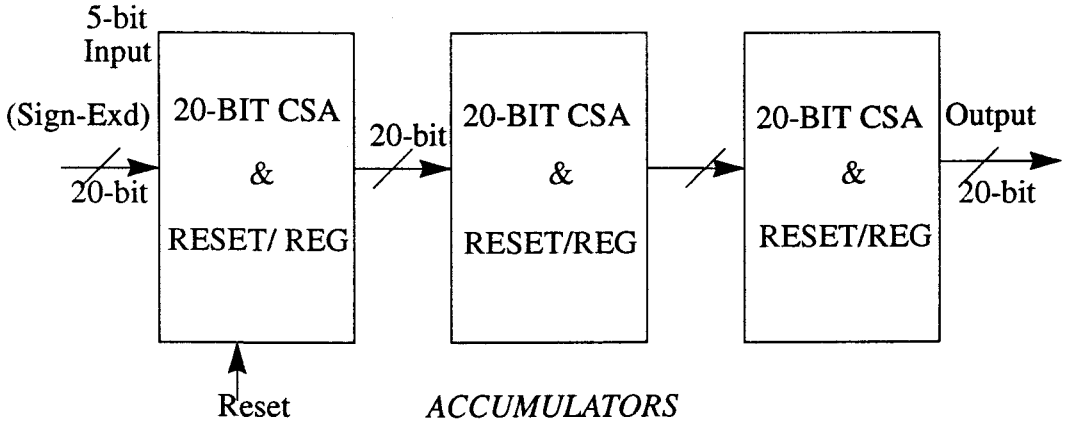


Figure 3.3 Block diagram of the adder section

It should be noted that the input signal is only a 5-bit word. It is necessary to sign-extended the input signal to a 20-bit word. As we mentioned early that 20-bit output data is required for the hardware to provide sufficient coefficients for the decimation filter. The sign-extension process can be done by connecting the 6th to 20th bits of the adder input to the most-significant-bit (MSB) of the input data.

For the CLA, the CARRY function consists of:

$$C_1 = G_1 + P_1 C_0$$

$$C_2 = G_2 + P_2 G_1 + P_2 P_1 C_0$$

$$C_3 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 C_0$$

$$C_4 = G_4 + P_4 G_3 + P_4 P_3 G_2 + P_4 P_3 P_2 P_1 C_0$$

and the SUM section consists of

$Si=A_i \Theta B_i \Theta C$

(Θ is exclusive-or logic)

Where P_i is the propagation signals and G_i is the generation signal as:

$$P_i = A_i + B_i \qquad G_i = A_i \times B_i$$

The carry-lookahead scheme, it is practical to compute up to 4th carry (C4), as the complexity involved with higher stages become impractical. Preliminary experiment results showed that the 4-bit carry-lookahead is not feasible using FSCL gates. Because of the irregularity of the carry computation in carry-lookahead scheme, and the speed of multi-stage FSCL gate seems to be lower than the one with simple stage. Table 1 shows the simulation results of a CLA and a CSA for the same 4-bit adder carry calculation.

Table 1: Comparison of CSA and CLA

Type of Adder	Number of Transistor To Generate Carry	Delay of Carry Calculation
4-Bit CLA	152	10ns
4-Bit CSA (2 x 2-bitCLA)	134	6.5ns

Based on the simulation results, it is decided to use only a 2-bit carry-lookahead scheme to implement the 20-bit CSA, as shown in Figure 3.4. In this figure, the CARRYs are calculated through the 10 rippled 2-stage carry-lookahead chain (carry-skip), and the corresponding SUMs are also calculated by the full adder FAs, which the CARRY-INs are computed in the carry-lookahead stage.

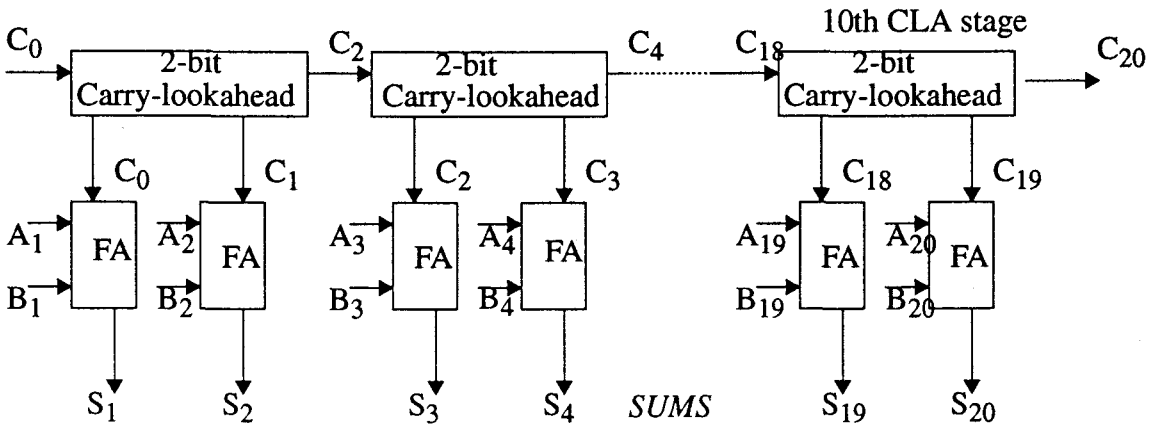


Figure 3.4 Schematic of 20-bit carry-skip-adder, FA is a full adder for sum

The different delays in the CARRY calculation resulted in varying delays in the SUM calculation Therefore, it is necessary to have latch the calculated data until all results are valid. Also, circuitry must be included to reset the system. This circuitry is combined with the down sampling storing registers as shown in Figure 3.5

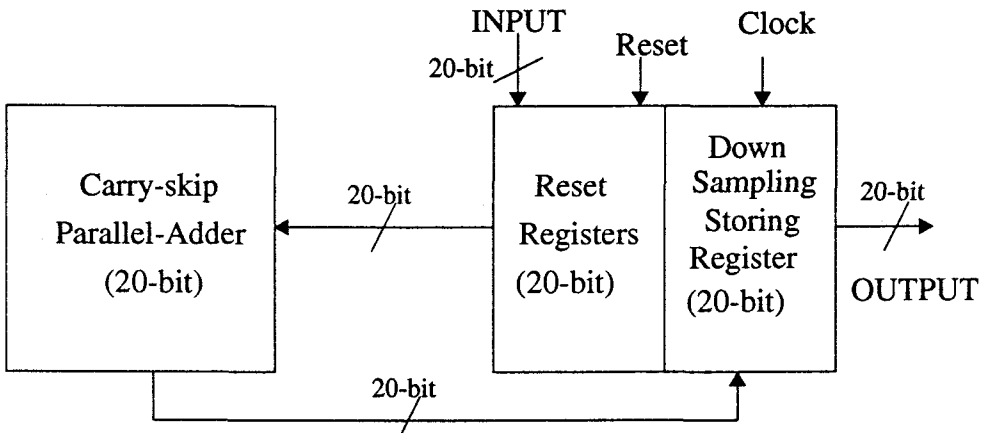


Figure 3.5 Block diagram of a 20-bit CSA with reset and downsampling storing registers

The reset register circuitry consists of a D-flip flop array clocked by a reset signal and an array of regular D-flip flops clocked by the control signal CK1 in Figure 3.1. The basic building blocks of the adder section of the decimation filter are FSCL gates, which include full-adder for sum calculation, multi-input nand/and gates and multi-input nor/or gates for carry calculation in the carry-lookahead stage. Also there are some other complex logic gates, such as D-flop flops with reset and regular D-flip flops. These FSCL gates will be discussed in a detailed manner in the latter section of this chapter.

3.3 Parallel to Serial Converter

The main task of the parallel to serial converter is to convert the 20-bit data of the parallel adder to a 1-bit serial data. The basic element of the converter is a D-flip flop. With the parallel input serial output shift register topology. The schematic of the converter is shown in Figure 3.6.

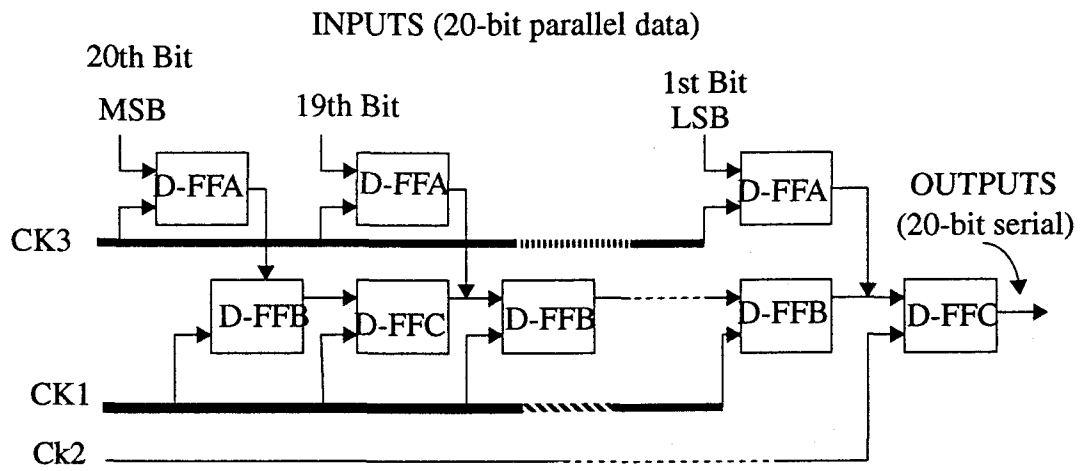


Figure 3.6 Schematic of the parallel to serial converter

The converter operates on 20-bit data from the adder section. The operation of this part of the decimation filter is as follows:

Step 1: The 20-bit parallel data is loaded to the input registers (D-FFAs) by the down-sampling control signal CK3 which is valid at every 32^{nd} clock cycle to perform decimation.

Step 2: The loaded data then are shifted serially through D-FFBs and DFFCs with LSB as the first output controlled by CK1 and CK2.

Step 3: Since there is only 20-bit data, but the decimation factor is 32, this means the

shifter will only shift data in the first 20 clock cycle, and it will stand still for the remaining 12 clock cycle before next set of data loaded into the input register. Ck2 has the exact required characteristic.

The clock scheme of the converter is shown in Figure 3.2. The control signals CK2 and Ck3 are produced by the control logic section, which will be described by the latter section of this chapter.

The basic circuit elements of the parallel to serial converter are D-flip flops and also will be discussed latter.

3.4 Bit Serial Subtractor

This block performs the filtering function of $(1\text{-}Z^{-32})^3$ as shown in Figure 2.7. It consists of three serial subtractors. The subtraction is performed by the 1's complement scheme (i.e. $A - B = A + \overline{B} + 1$). This operation employs a simple 1-bit adder with its carry in always at "1". Both the parallel-to-serial converter and the serial subtractor use the same control signals CK1, CK2 and CK3 for data transfer. There is also a 20-bit shift register for each serial subtractor. The circuit schematic is shown in Figure 3.7

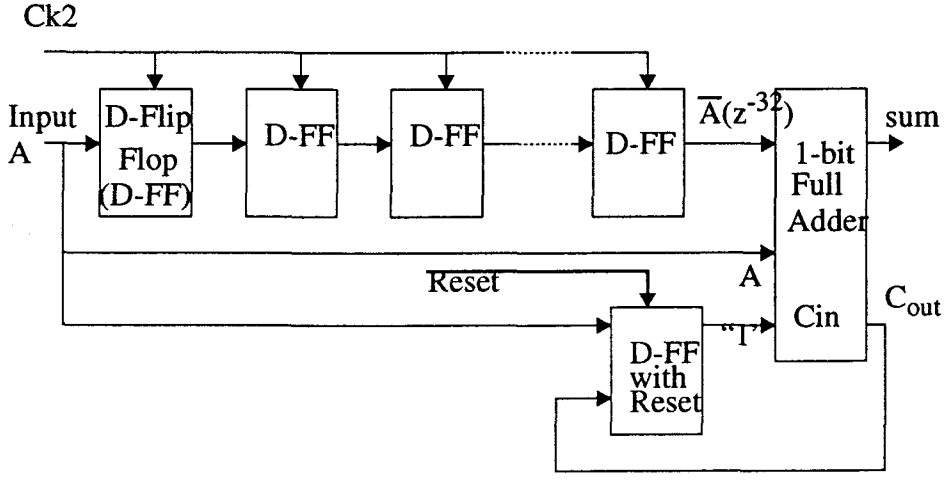


Figure 3.7 Circuit schematic of 20-bit serial subtractor

There are twenty FSLC D-flip flops and full adder for both carry and sum calculation required to implement each 20-bit serial subtractors. These FSCL gates will be discussed together with other FSCL gates latter.

3.5 The Control Logic

Control logic is the most important part of the decimation filter. It controls the operation of the other sections and it is designed to generate the control signals for the system. According to the basic timing diagram of the input and output as shown in Figure 3.1, the control logic section needs to produce a set of clock signals shown in Figure 3.1.

The state machine approach has been used to develop the design. It is a sequential counter shown in Figure 3.8 which is implemented by D-flip flops, and some other FSCL logic gates. The number of D-flip-flops required is determined by the number of states.The state diagram shown in Figure 3.9, has 32 states, and 5 bits (5 D-flip flops) are required. The Karnaugh-map technique has been applied to optimize the design of both the input logic and the output logic. The final circuit schematic of the control logic is shown in Figure 3.8. The outputs of the D-flip flops feed back to form the inputs of the input forming logic(-IFL).

$$D_A = A(\overline{B}+\overline{C}+\overline{D}) + BCD(A\Theta E)$$

$$D_B = B(\overline{C}+\overline{D}) + CD(B\Theta E)$$

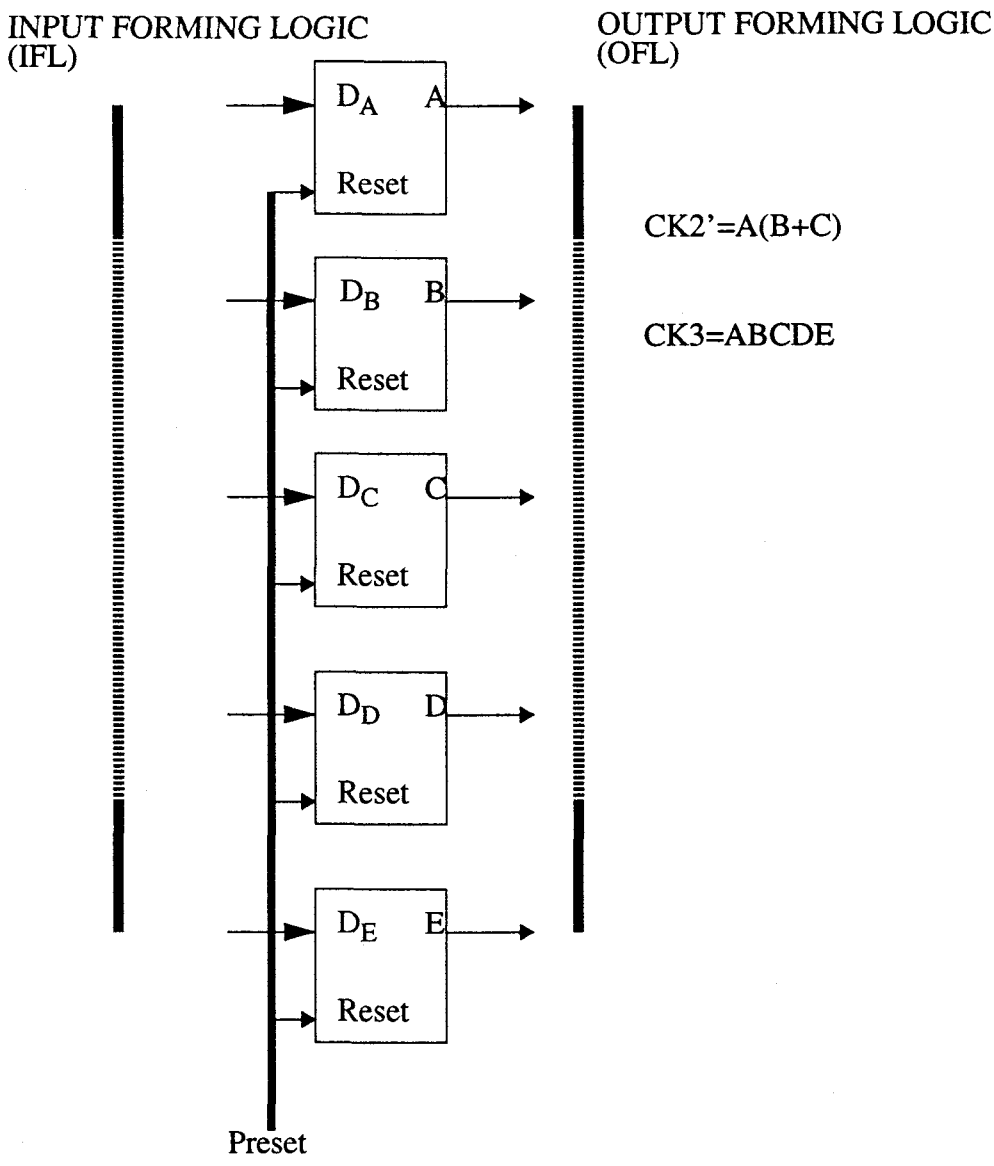
$$D_C = C\overline{D} + D(C\Theta E)$$

$$D_D = D\Theta E$$

$$D_E = \overline{E}$$

Output signals are:

CK2'=A(B+C); CK3=ABCDE



Note: The IFL D_A, D_B, D_C, D_D and D_E are listed in previous pages.

Figure 3.8 Schematic of the control logic

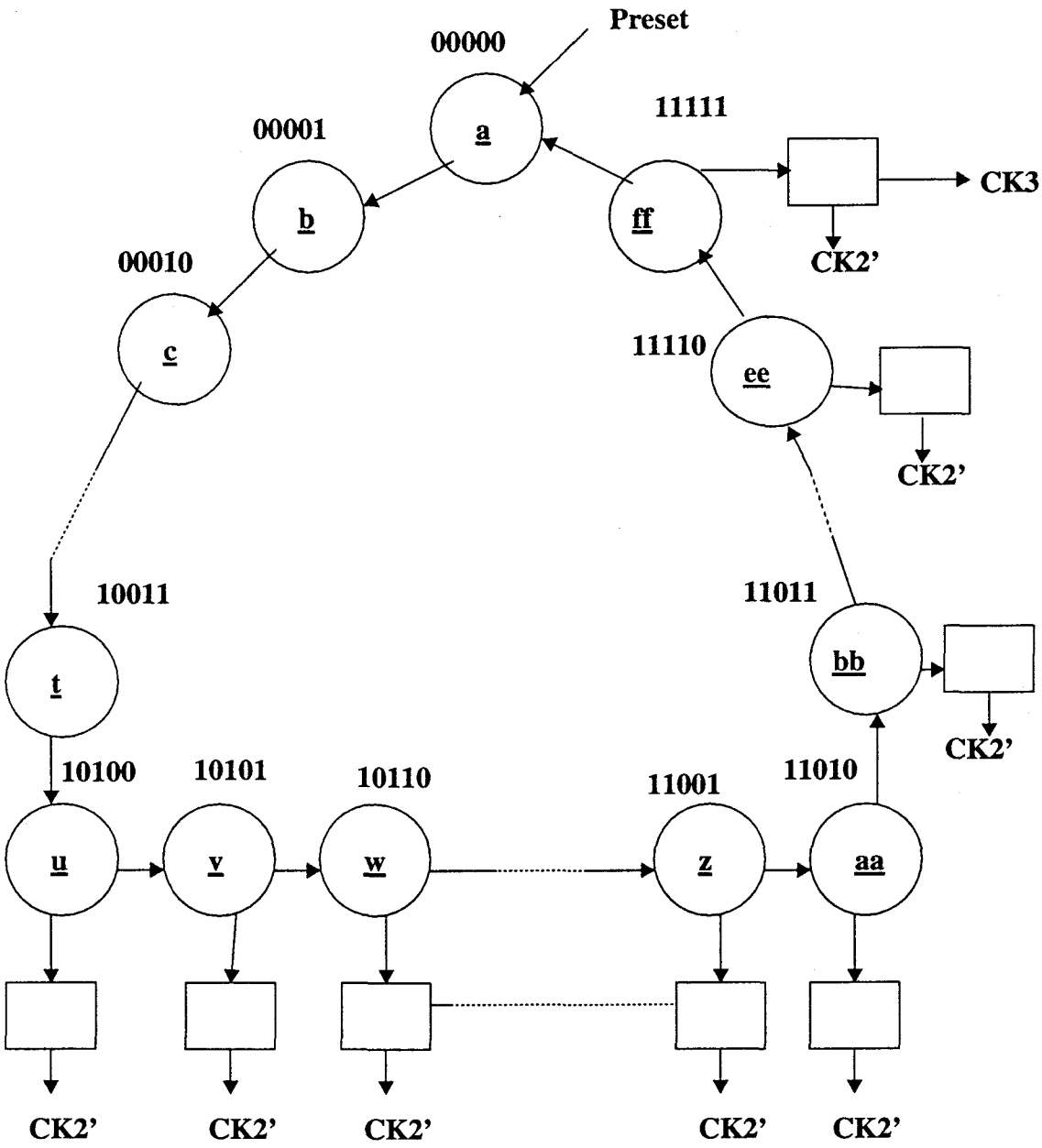


Figure 3.9 State diagram of a 32-state sequential counter

3.6 Folded Source Coupled Logic (FSCL)

FSCL has been developed to reduce the digital switching noise. This logic family employs current steering techniques similar to the bipolar ECL topology. The basic principle of this current-mode logic is to bias the circuit by a constant current supply. Because of the constant current sources connected to the supply lines, the large current spikes generated during the switching transitions are eliminated.

3.6.1 Basic Operation of FSCL gates

The fully differential FSCL inverter is shown in Figure 3.10. The input stage consists of the NMOS differential pair M_1 and M_2 biased by the constant current I_1. The output stages are diode-connected load devices M_3 and M_4 biased with the constant current I_2. If, for example, the input A is high, M_1 is on. Since $\overline{A}$ is low, M_2 is off, and the current I_1 flows through M_1, the current in M_3 is (I_1-I_2), and in M_4 is I_2. The two current difference in the load devices produces a voltage difference at Q and $\overline{Q}$.

In this case, voltages at $\overline{Q}$ and Q are: $V_{\overline{Q}} = V_T + \sqrt{\dfrac{2(I_2 - I_1)}{K_N(\frac{W}{L})M_3}}$

$$V_Q = V_T + \sqrt{\frac{2I_2}{K_N(\frac{W}{L})M_4}}$$

Since $I_2 > I_1$, the output Q is high and output $\overline{Q}$ is low. The output voltage swing is

$$\Delta V_o = V_Q - V_{\overline{Q}} = \sqrt{\frac{2I_2}{K_N(\frac{W}{L})M_4}} - \sqrt{\frac{2(I_2 - I_1)}{K_N(\frac{W}{L})M_4}}$$

Assume M_3 and M_4 are matched transistors, and also $I_2 = \alpha I_1, \alpha > 1$

$$\Delta V_O = \sqrt{\frac{2I_2}{K_N(\frac{W}{L})M_3}}(1-\sqrt{1-\frac{1}{\alpha}})$$

In order to minimize the power consumption, it is necessary to minimize the difference of I_1 and I_2. Meanwhile a reasonable output voltage swing (0.3v-1v) needs to be maintained to sustain a valid logic. By using minimum size devices and $\Delta V_O \approx 0.7V$, the following parameters were obtained. $(\frac{W}{L})_{M_3} = (\frac{2um}{1.2um})$

$K_N = 62.5\frac{uA}{V^2}$

I_1=68 uA, I_2=82 uA, α=1.2

$V_{OL} = V_{\overline{Q}} = 1.18v$ $V_{OH} = V_O = 1.88v$

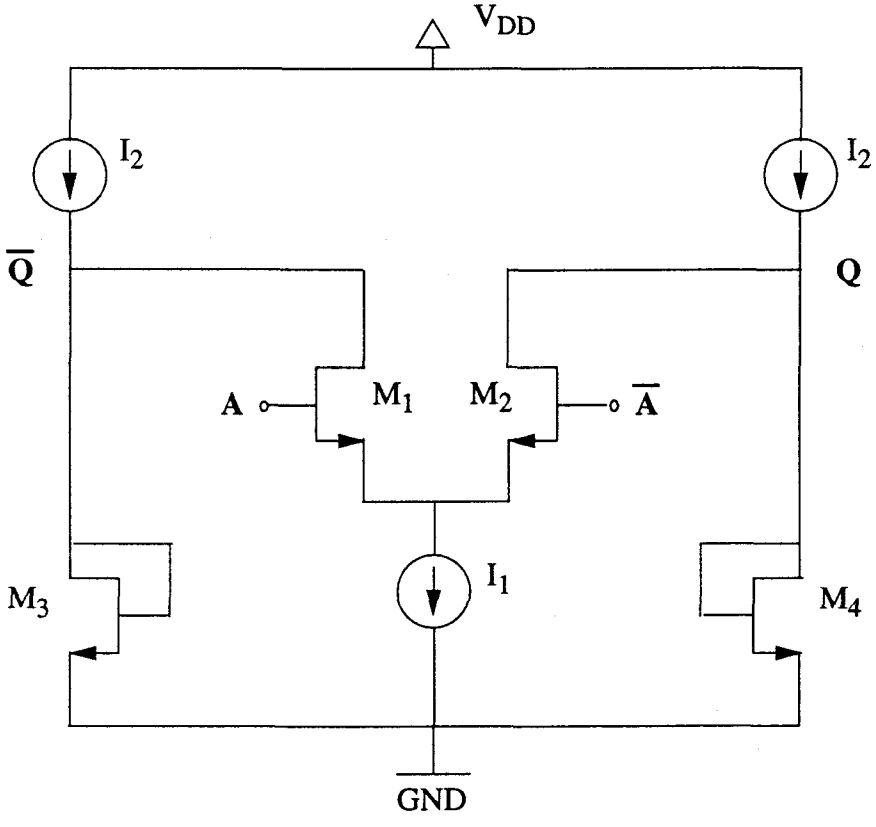


Figure 3.10 Schematic of a FSCL inverter

3.6.2 FSCL Logic Family

In order to implement the decimation filter described previously, a number of FSCL gates are required as building blocks of the system. These high level FSCL gates were developed by replacing the input transistors M_1 and M_2 by a stack of NMOS transistors. This technique is called "series-gating". These FSCL gates have differential inputs, and the gates have the same power consumption as a FSCL inverter and the same output voltage swing.

The FSCL logic family offers many advantages over conventional static CMOS as:

1. Low switching noise.
2. High operating frequency.
3. Reduced voltage swing which implies reduced dynamic power dissipation at high frequencies.
4. High noise immunity due to the differential input topology.
5. Availability of complementary outputs.

Some disadvantages of FSCL gates. Include high static power consumption and increased die area as compared to static CMOS logic due to the greater number of transistors for simple gate. However, FSCL actually uses fewer transistors for some complex gates. Therefore, FSCL is still very effective to implement the digital system in mixed-signal applications.

Due to the limitations of design parameters, the fan-in ability of a typical FSCL gate is limited to 3 inputs. As the number of stacked NMOS differential pairs of the input increases, the limited driving capability of the input voltage source causes the input voltage of the differential pair unable to steer the bias current flow through one branch only.

A list of circuit schematics of FSCL gates which will be used to implement the decimation filter described in previous section of this chapter are shown in Figures 3.11-3.16.

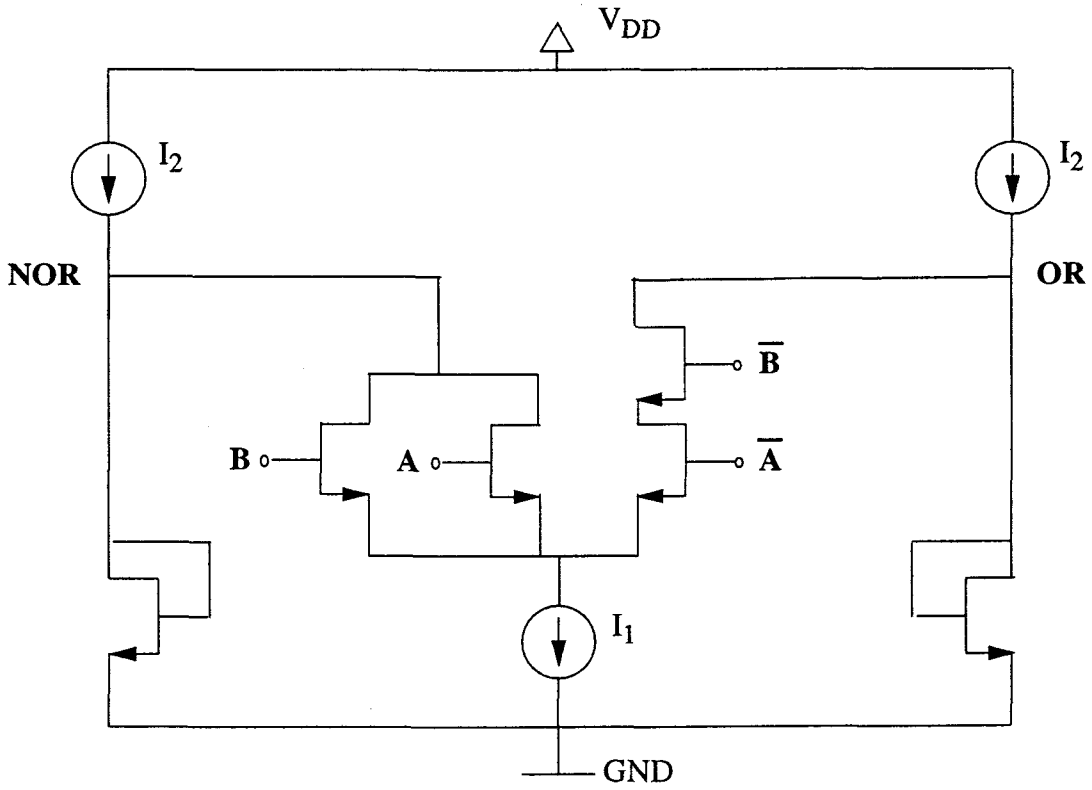


Figure 3.11 Schematic of a 2-input FSCL NOR/OR gate

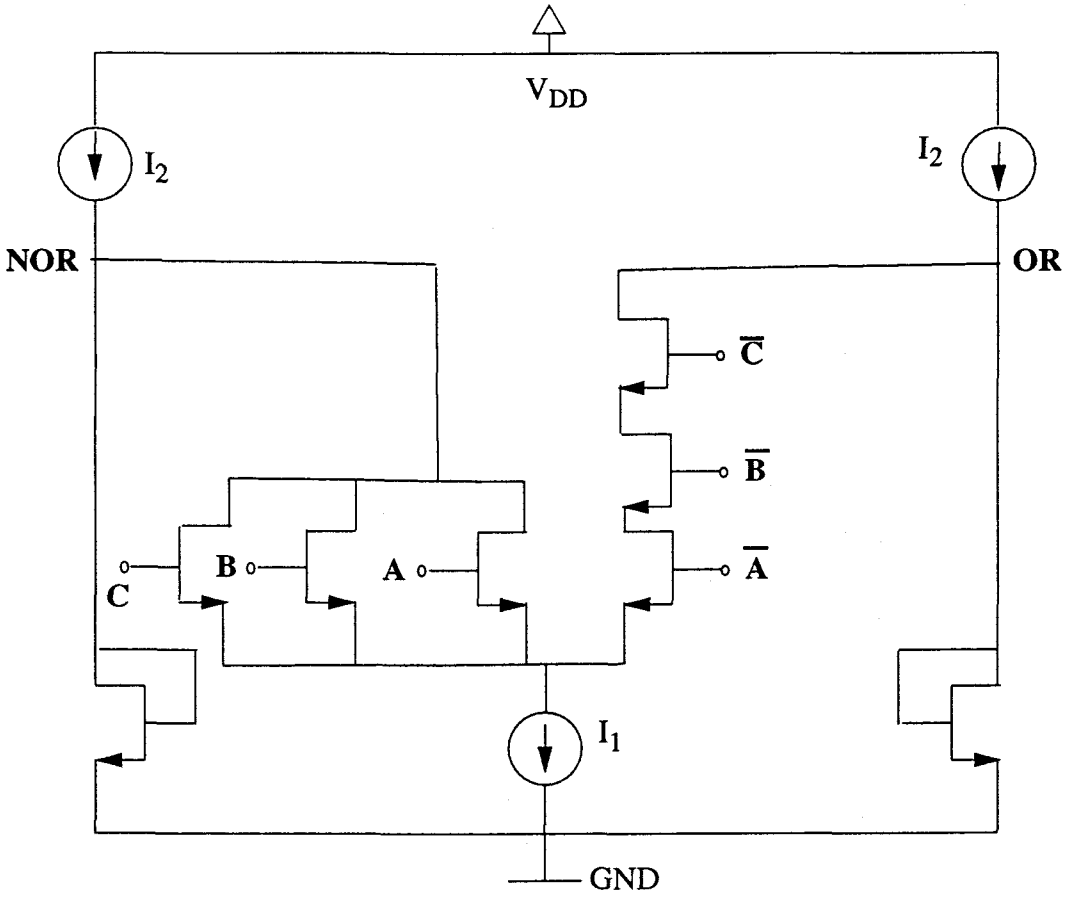


Figure 3.12 Schematic of a 3-input FSCL NOR/OR gate

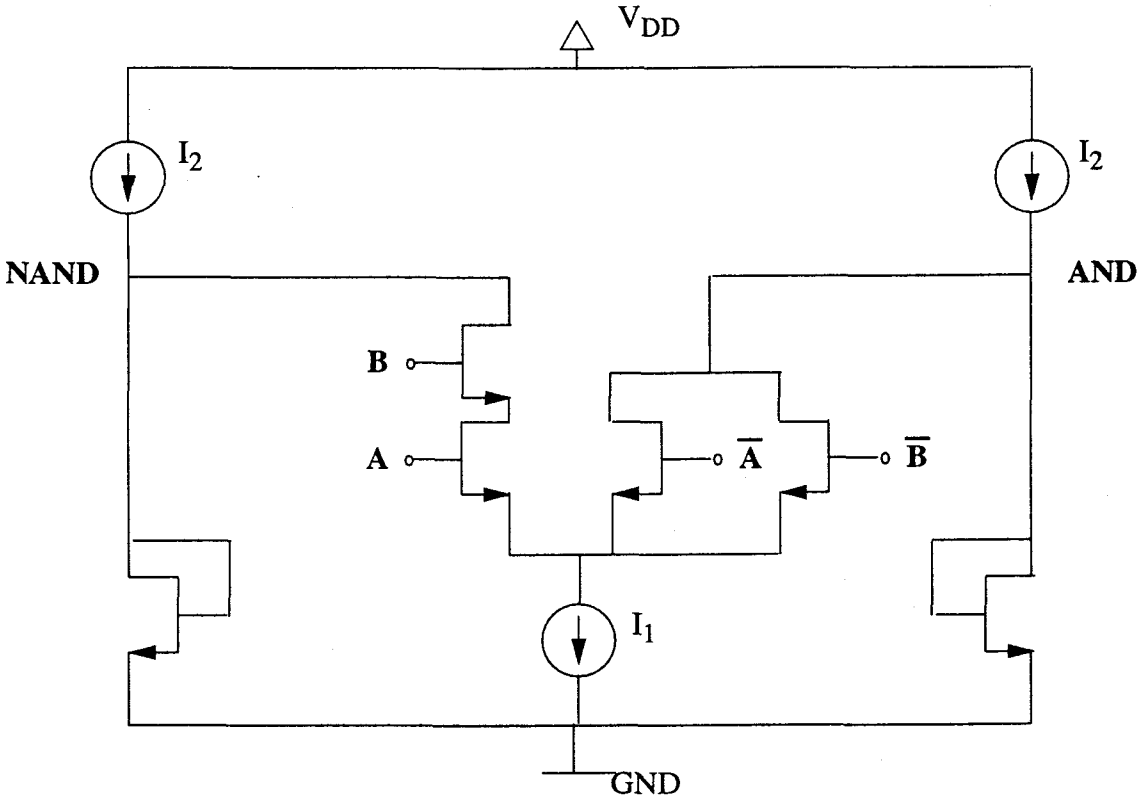


Figure 3.13 Schematic of a FSCL 2-input NAND/AND gate

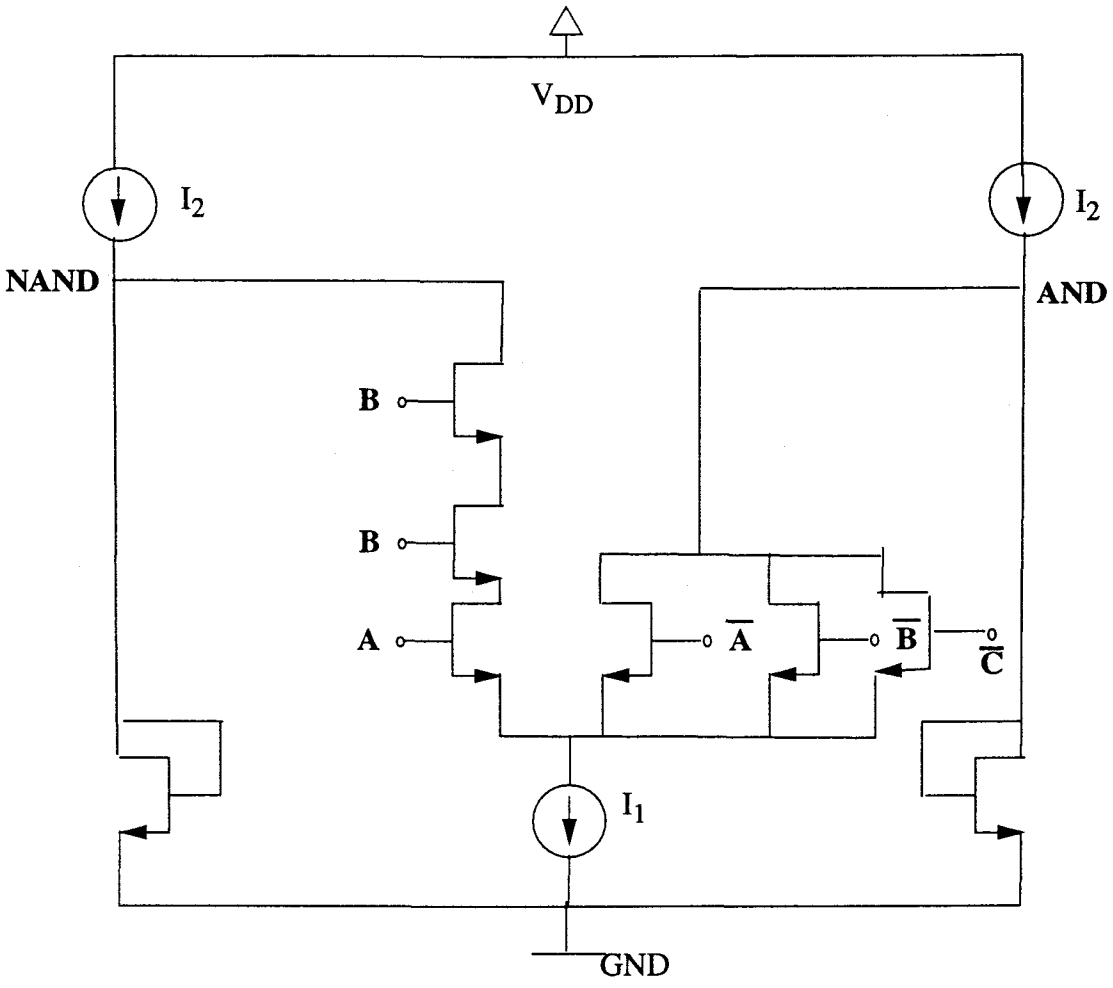


Figure 3.14 Schematic of a FSCL 3-input NAND/AND gate

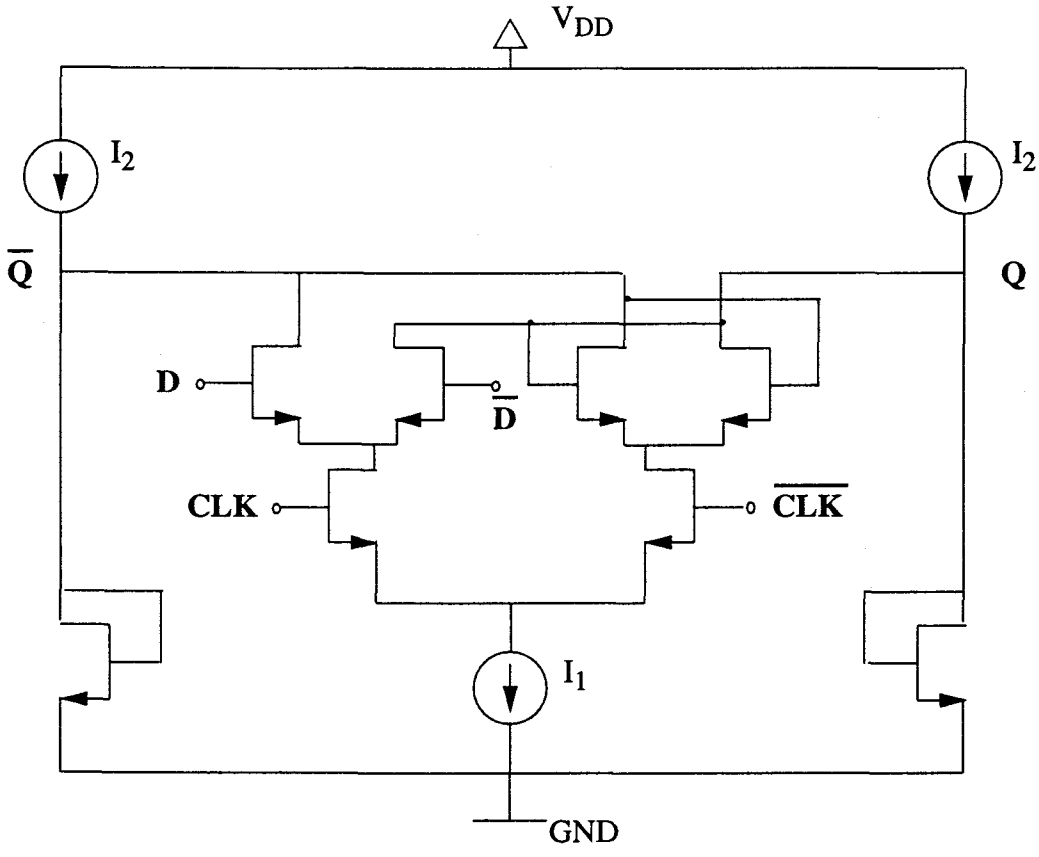
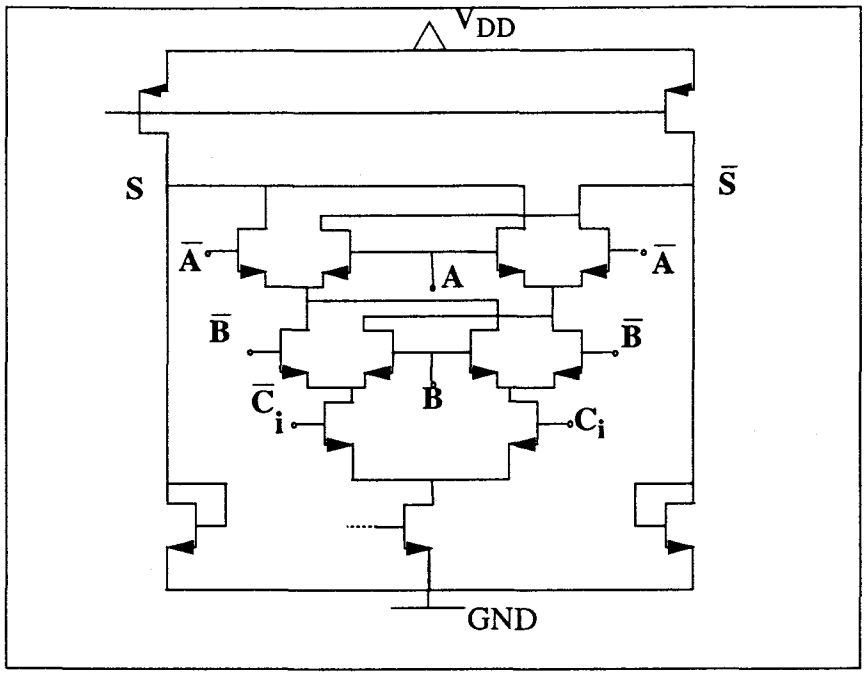
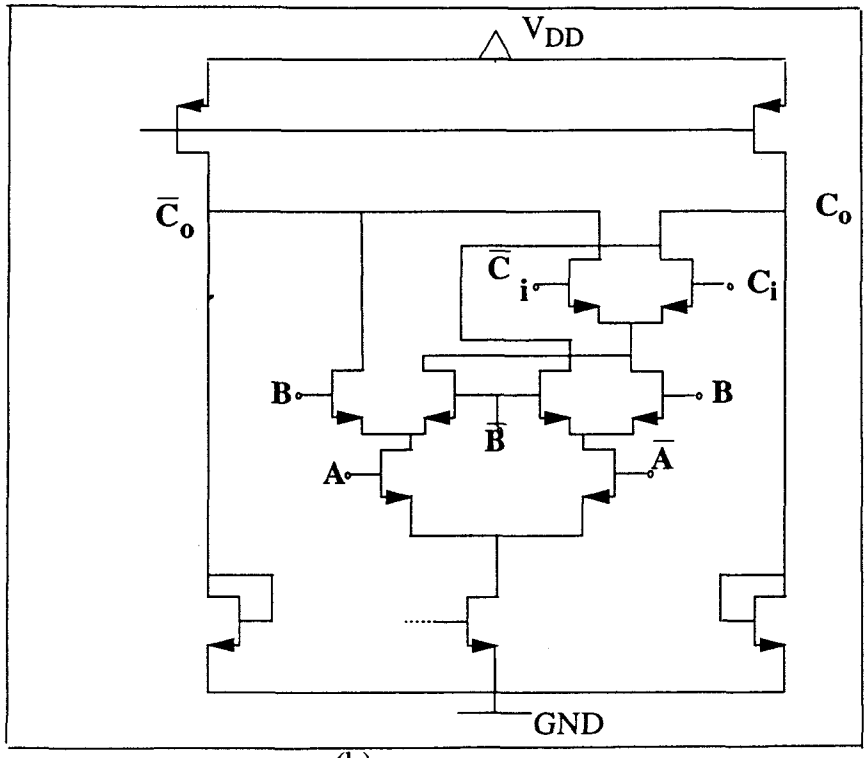


Figure 3.15 Schematic of a FSCL D-flip flop



(a)



(b)

Figure 3.16 Schematics of (a) SUM and (b) CARRY for a 1-bit FSCL full adder

Table 2 summarizes the simulation result of the FSCL gates used to implement the $sinc^3$ decimation filter.

Table 2: Design Parameters of FSCL Gates

FSCL Gates	Delay (ns)	Number of Tansistors	Size of Input transistor (W/L)(um)	Power Consumed (mW)
inverter	1.3	7	7.2/1.2	0.449
2-input Nand/And	2	9	14.4/1.2	0.449
2-input Nor/Or	2	9	14.4/1.2	0.449
3-input Nand/And	4	11	21.6/1.2	0.449
3-input Nor/Or	4	11	21.6/1.2	0.449
D-flip-flop	2	11	14.4/1.2	0.449
D-flip-flop with Set/Reset	3.5	13	21.6/1.2	0.449
Sum (FullAdder)	4.5	15	21.6/1.2	0.449
Carry (FullAdder)	4.5	13	21.6/1.2	0.449

Chapter 4. Simulation Results and Noise Discussion of The Decimation Filter

The main objective of this chapter is to investigate the power supply noise issue of the system and to evaluate the functionality of the system by HSPICE simulations also the chip layout of the decimation filter will be presented. In order to demonstrate the low noise property of the FSCL technique, a comparison is made between the FSCL structure and the standard CMOS implementation.

4.1 Carry-Skip-Adder (CSA)

As mentioned in Chapter 3, the adder section of the decimation filter represents a trade-off between die area and speed. By selecting the carry-skip-adder scheme, the simulation results in Figures 4.1-4.5 show the adder performs as expected. Figures 4.1-4.2 show the action of a 2-bit carry-lookahead-adder(CLA), which has been used as the fundamental building block of the adder section. Both the carry and sum calculations demonstrated the speed of the CLA. The longest delay of the data calculation is measured as 2.5ns for the second sum, and the second carry is measured having a 2ns of delay.

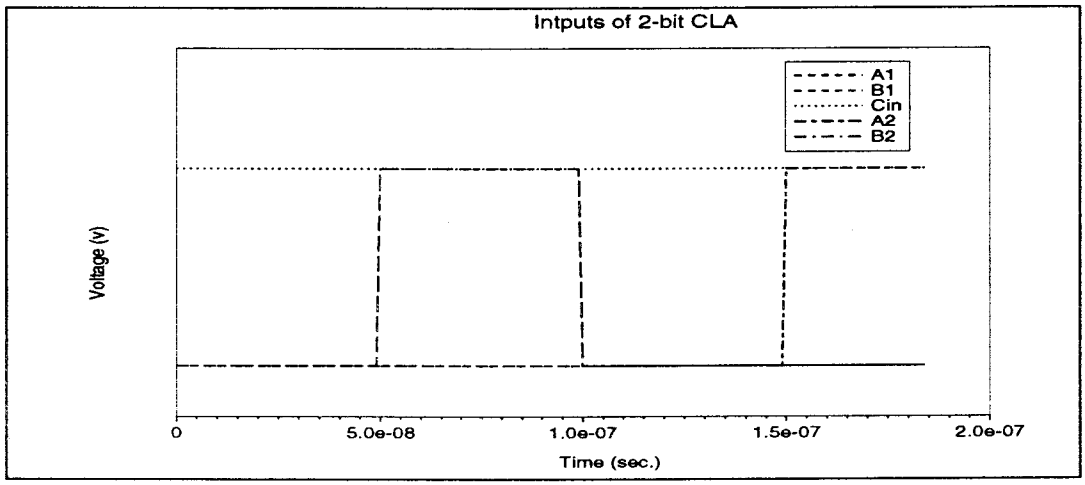


Figure 4.1 Inputs of a 2-bit carry-lookahead adder

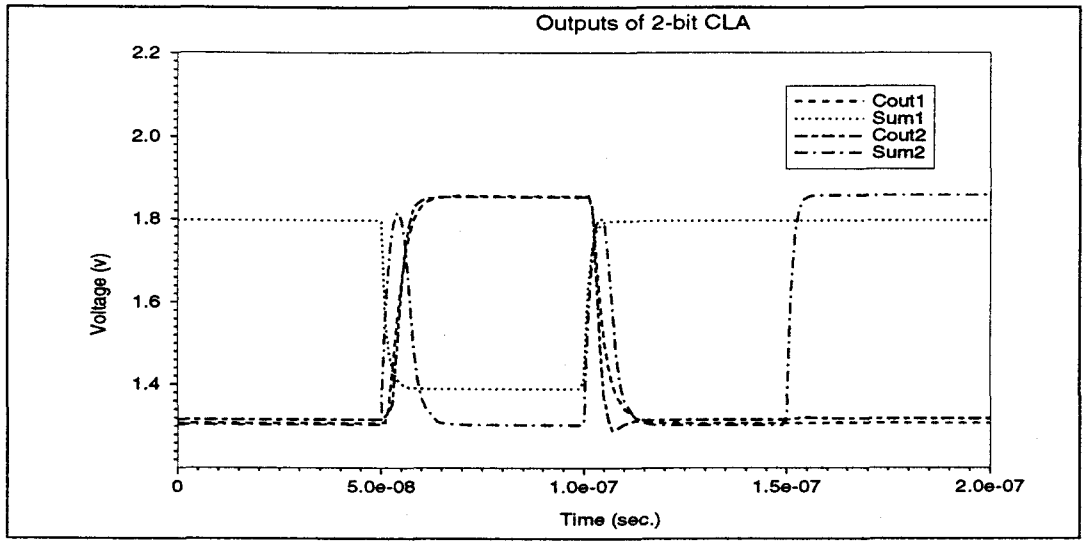


Figure 4.2 Outputs of a 2-bit carry-lookahead adder

When this 2-bit CLA is used to implement a 4-bit carry-skip-adder (CSA), the simulation result shown in Figure 4.4 indicates that the longest delay of data calculation is for the fourth sum, which has a delay of 16ns. The longest delay of the 20-bit CSA data calculation is 55ns for the twentieth sum calculation. Figure 4.5 shows the HSPICE simulation result of the twentieth sum-and-carry calculation.

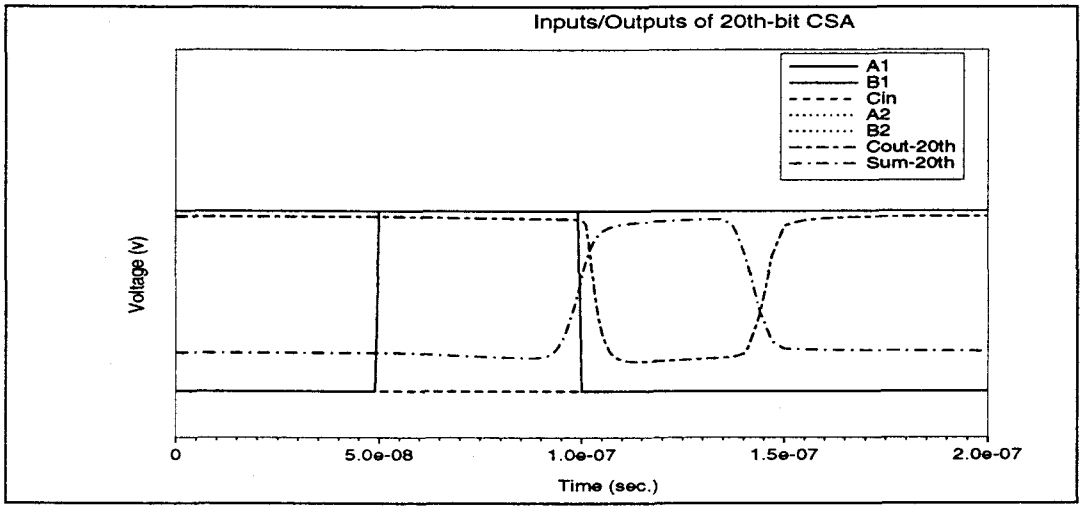


Figure 4.5 Inputs and outputs of a 20-bit carry-skip adder

Figures 4.1-4.5 have shown the valid functionality of the adder. Since the longest data delay of the adder is 55ns, the operating frequency of the adder can be 18 MHz. This implies that the input signal of the decimation filter can have a data rate of 18MHz.

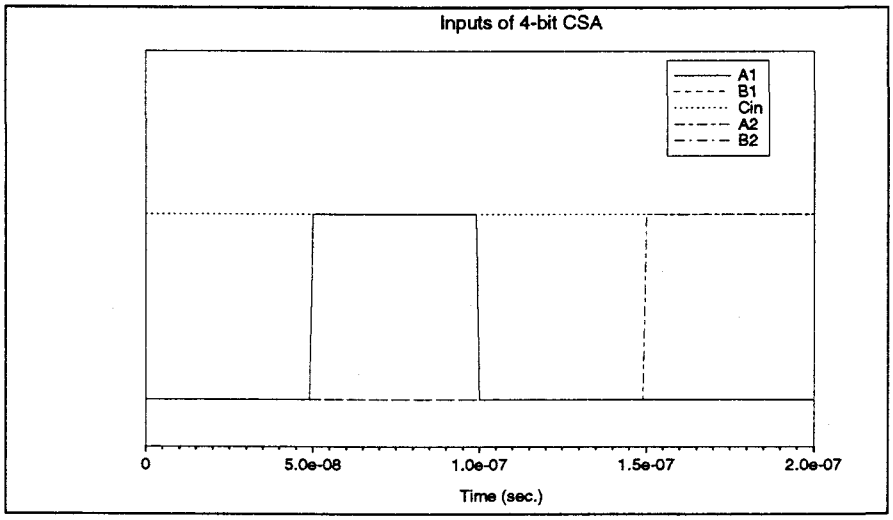


Figure 4.3 Inputs of a 4-bit carry-skip adder

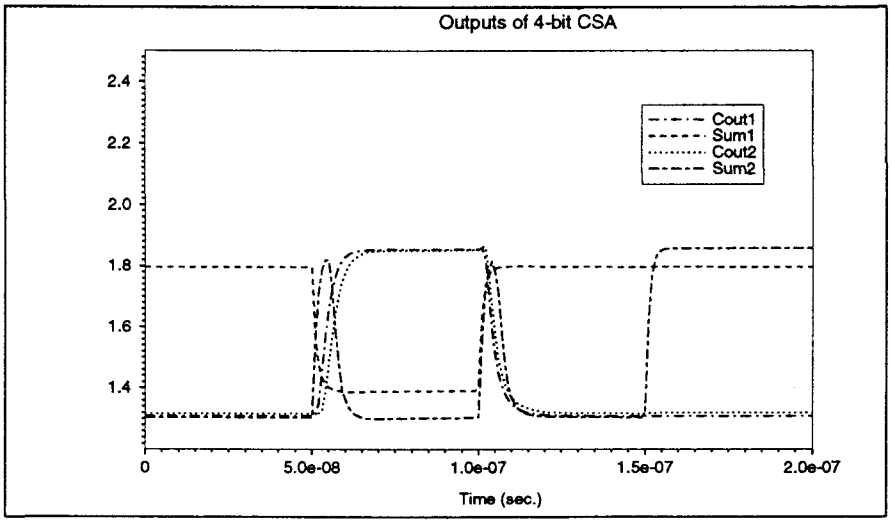


Figure 4.4 Outputs of a 4-bit carry-skip adder

To investigate the power supply noise issue of the decimation filter, some simulations were performed to study the switching noise of the system. Figure 4.7 shows the noise spike of a FSCL inverter gate. When we compare this noise of a standard CMOS inverter gate shown in Figure 1.1, we can see that there is a significant reduction in switching noise of the FSCL gate. Note that the CMOS gate has a noise spike as high as 120uA. and FSCL gate has a noise spike of 6uA.

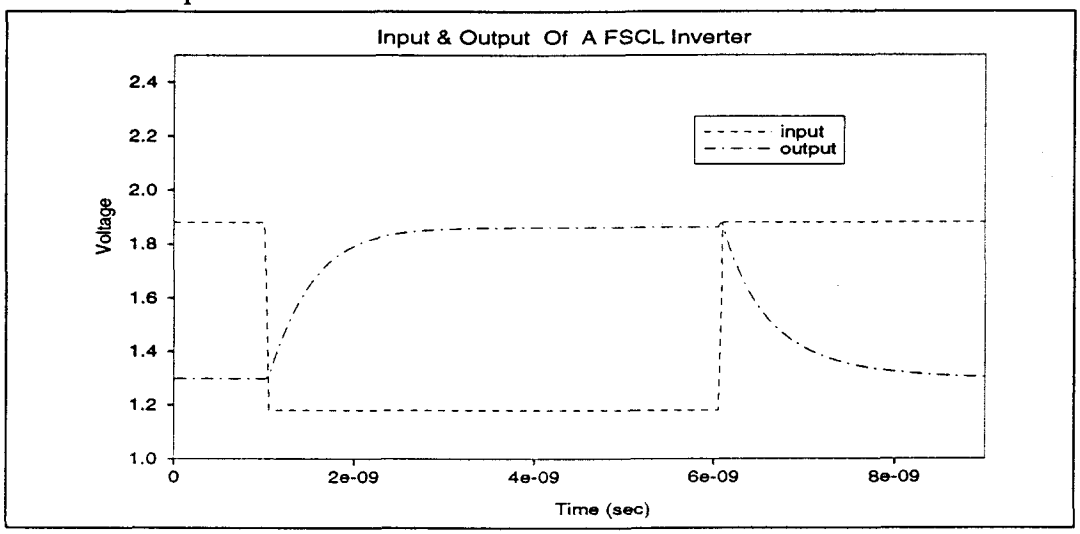


Figure 4.6 Hspice simulation of a FSCL inverter

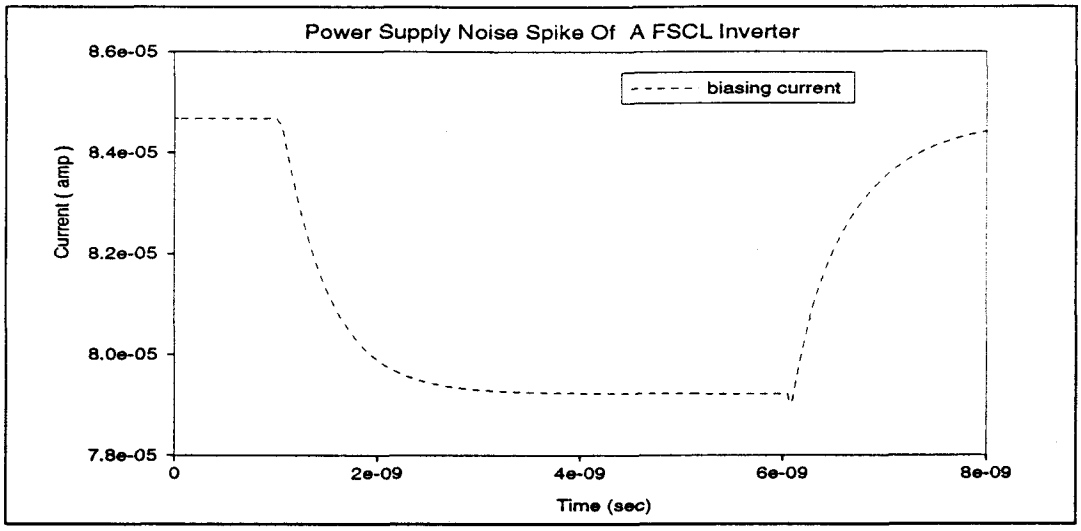


Figure 4.7 Hspice simulation of power supply noise spike of a FSCL inverter

In this design, the total number of gates used to implement the adder section is 570. Assuming that 50% of the gates switch simultaneously during each operation, the noise spike generated by the CMOS gates will be as large as 34mA. However, in the case of FSCL implementation, the noise spike will be only 3.42mA. This large reduction in the noise spikes will provide a great advantage in improving the performance of the mixed-signal system.

4.2 Parallel to Serial Converter

To verify the functionality of the parallel to serial converter, simulations were performed with large number of inputs. Figure 4.8 shows a 10-bit parallel to serial simulation result, which has been used as a key parameter to implement the 20-bit parallel to serial converter. Figure 4.8 shows this converter performed a valid function of converting a 10-bit parallel input to a 10-bit serial output. The frequency of the input data of the converter is the same as the frequency of the output data of the adder, which is 18MHz.

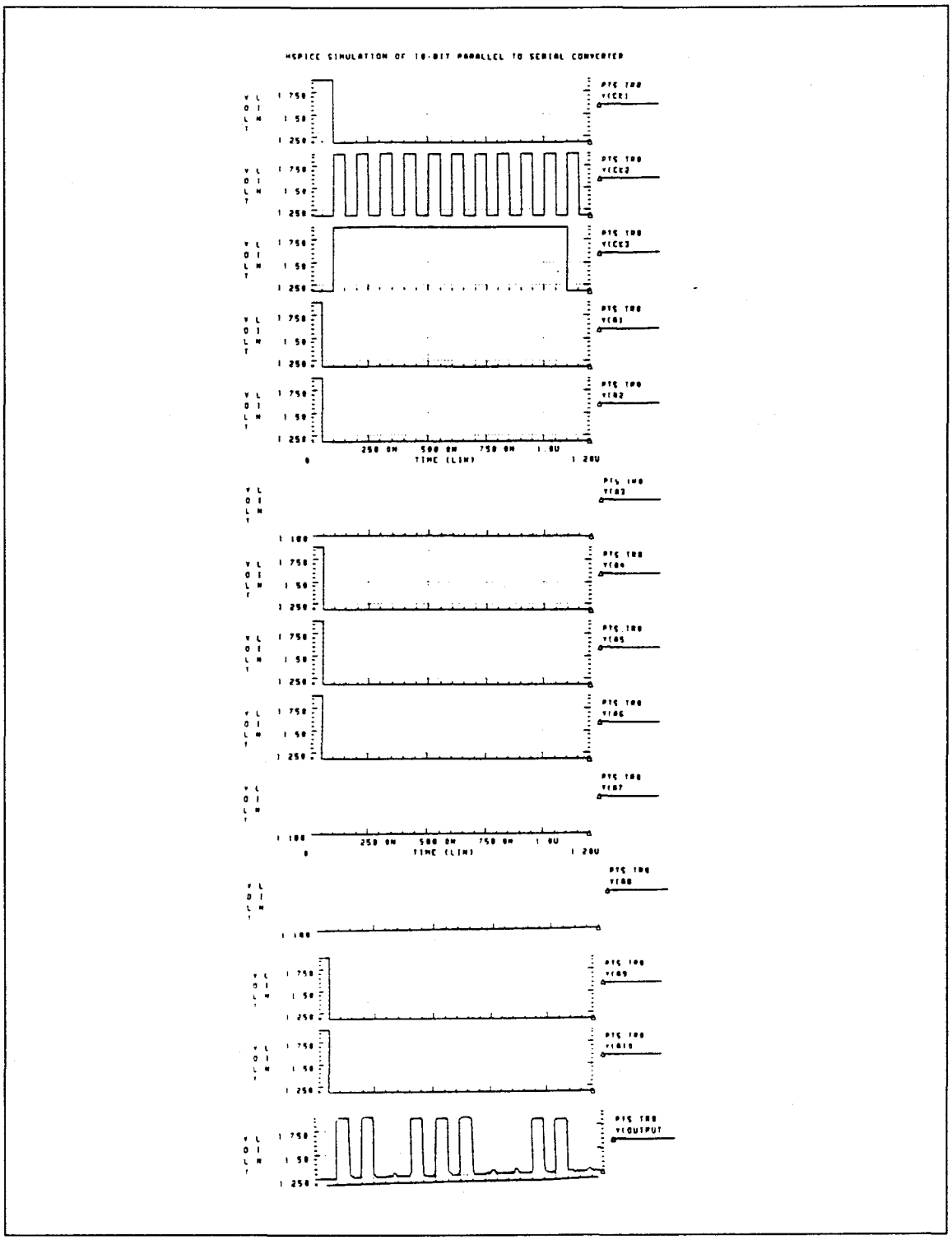


Figure 4.8 Hspice simulation of a 10-bit parallel-to-serial converter

The power supply noise characteristic of this block of the system has also been studied. The total number of gates in the converter is 60, by applying the same method used in 4.1 to calculate the noise spikes generated during each operation. It has been found that the noise spike is 0.36mA for a FSCL structure. The noise spike generated by a CMOS implementation for the same converter is 3.6mA. Thus the FSCL technique shows a great advantage over the CMOS structure regarding the power supply noise issue.

4.3 Serial Subtractor

The subtractor consists of a 20-bit serial shift register and a 1-bit full adder for both sum and carry calculation. The speed of this subtractor section does not have a significant impact on the overall decimation filter performance. Since the decimation process and the parallel to serial converter already convert the high speed parallel data into a lower rate serial data, the subtractor only needs to perform a single 1-bit addition during each clock period. Three serial operations are performed to implement the filtering function of $H(z) = (1 - Z^{-32})^3$. The total delay of the output data calculation is measured as 7ns.

The noise spikes resulting from the operation of this section of the system is 0.4mA, as the total number of FSCL gates required to implement the subtractor is 69. If the same subtractor were implemented by the standard CMOS gates, the noise spike would be as high as 4.14mA. Obviously, the CMOS implementation would be more detrimental to the mixed-signal system performance when we consider the system's power supply noise.

4.4 Control Logic

As described in Chapter 3, the control logic is a very crucial part of the system. A number of simulations were performed to verify the functionality of this part of the decimation filter. The result in Figure 4.9 shows the control logic has been designed properly.

To calculate the noise spikes generated by the control logic, we use the same approach as described previously. The control logic requires 33 gates. For the FSCL structure, the noise spike was calculated 0.198mA, and the noise spike for the standard CMOS implementation was calculated at 1.98mA. The FSCL technique once again proved to be very effective in reducing the power supply noise.

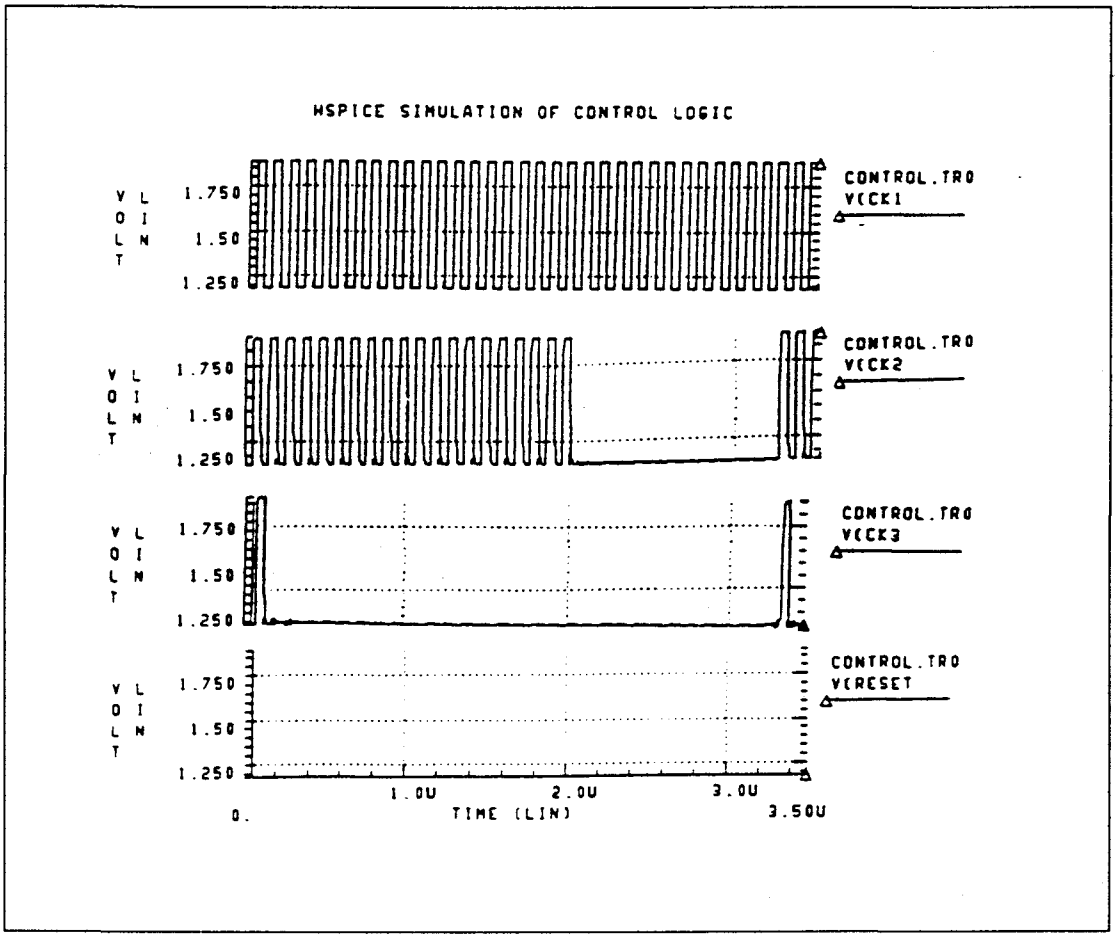


Figure 4.9 Hspice simulation of the control logic

4.5 Overall Discussion of the Decimation Filter

After the investigation of each individual block of the system, it becomes clear that the FSCL technique offers many advantages over the standard CMOS technique. For example, the low noise characteristic of FSCL has been shown explicitly. The overall performance of the decimation filter has been improved by using the low noise FSCL technique.

The filter is capable of operating with an input signal frequency of 18 MHz.

The complete characterization of the decimation filter is summarized in Table 4 and Table 5.

Table 3: Delay and Power Consumption of the Decimation Filter

	Longest Data Delay (ns)	Power Consumption (watt)
Adder Section	55	0.3
Parallel To Serial Converter	7.5	0.032
Serial Subtractor	7	0.037
Control Logic	19	0.017
Overall Decimation Filter	N/A	0.386

Table 4: Noise Spikes Comparison Between FSCL & CMOS Implementation of the Decimation Filter

	Number of Gates	Total Noise Spike (FSCL)	Total Noise Spike (CMOS)
Adder Section	570	1.73mA	34mA
Parallel To Serial Converter	60	0.18mA	3.6mA
Serial Subtractor	69	0.2mA	4.14mA
Control Logic	33	0.1mA	1.98mA
Overall Decimation Filter	732	2.21mA	43.72mA

Meanwhile, after all the functionality verification and noise analysis simulation, the decimation filter has been proven functional. And this decimation filter has also been layout with the 1.2um n-well scalable CMOS process, it is being fabricated by the MOSIS. The floor plan of the chip is shown in Figure 4.10.

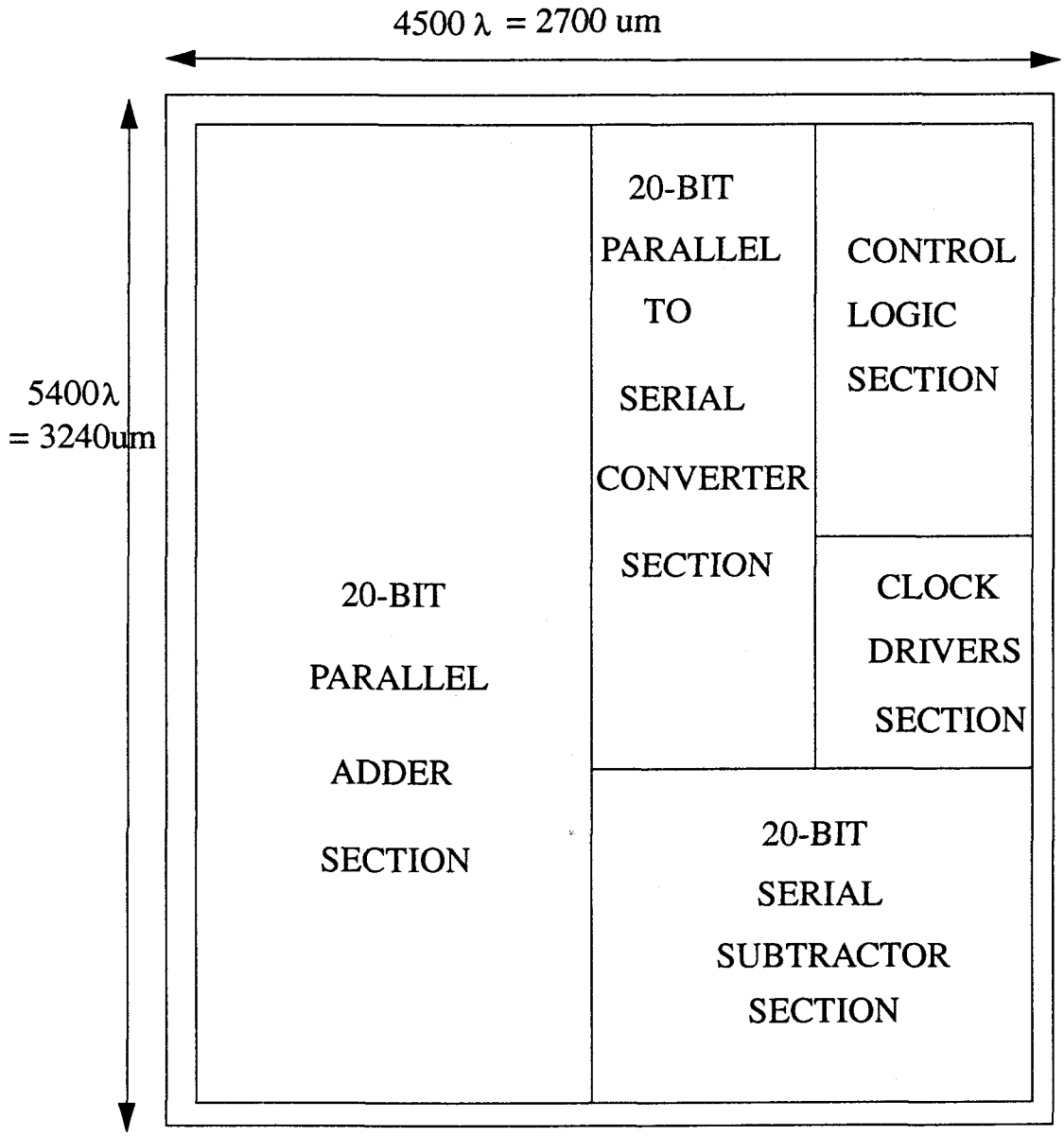


Figure 4.10 Chip floor plan diagram of the $sinc^3$ decimation filter

Chapter 5. Conclusion

Decimation plays an important role in producing good signal resolution in the oversampling modulation applications. The decimation process reduces the high sampling rate of a modulated signal to a lower rate which is more suitable for processing. In this thesis it has been shown that a $sinc^3$ decimation filter, which employs the accumulate-and-dump topology, can match the structure of the quantization noise from a 2^{nd} order delta-sigma modulator to provide decimation and maintain high resolution. This type of filter also has been proved that they have a simple structure and can be easily imlemented in circuit level.

On the other hand, conventional static CMOS logic is unsuitable for mixed-signal ICs due to the large overlap current spikes generated during the switching transition of the MOSFETs. This digital switching noise severely degrades the performance of a mixed-signal system.

The CMOS Folded-Source-Coupled-Logic (FSCL) is aimed at reducing the power supply line noise of the mixed-signal IC applications. The use of this low noise circuit technique in implementing the decimation filter showed that FSCL is very effective in reducing the digital switching noise. Simulation results showed that the swtching noise can be reduced almost by a factor of 10 in this filter design. During the study of this thesis, author found that there are many other evidence which strongly support that the low noise FSCL technique is very feasible for the mixed-signal IC application.

Bibliography

[1] J.A. Olmstead, "Noise Problems in Mixed Analog-Digital Integrated Circuits," Proc. of the IEEE Customs Integrated Circuits Conference, 1987, pp659-662.

[2] Max. W. Hauser, "Principle of Oversampling A/D Conversion," J. Audio Eng.-Soc. vol.39 pp3-26, Jan.-Feb.1991.

[3] G.Temes, J.C. Candy, " Oversampling methods for A/D and D/A conversiom," Oversampling Delta-Sigma Data Converters, J. C. Candy and G. C. Temes, Eds. New York: IEEE Press 1992, pp1-3.

[4] G.Temes, J.C. Candy, " Oversampling methods for A/D and D/A conversiom," Oversampling Delta-Sigma Data Converter, J. C. Candy and G. C. Temes, Eds. New York: IEEE Press 1992, pp19-24.

[5] R.M. Gray. "Spectral Analysis of Quantization Noise in Sigma-Delta Modulator with dc Input," IEEE Trans. Commun. vol. COM-37, pp599-599, June 1989.

[6] J. C. Candy, Y. C. Ching and D. S. Alexander, "Using Triangular Weighted Interpolation to Get 13-bit PCM from a Sigma-Delta Modulator," IEEE Trans. Commun. vol. COM-24, pp1268-1275, Nov. 1976.

[7] J. C. Candy, "Decimation for Sigma-Delta Modulation," IEEE Trans. Commun. vol.COM-34 pp72-76, Jan. 1986.

[8] E. Dijkstra, O. Nys, C. Piguet and M. Degrauwe, "On the Use of Modulo Arithmetic Comb Filters in Sigma-Delta Modulators," IEEE Proc. ICASSP'88, pp2001-2004, April 1988.

[9] S. Kiaei, S. Chee and D. Allstot, "CMOS Folded Source Coupled Logic for Mixed-Mode VLSI," ICASSP, May 1990.

[10] N. H. Weste and K . Eshraghian, "Principle of CMOS VLSI design, A System Perspective," Addison Wesley, pp320-321, June 1988.