**The Report Committee for Mostafa Elsayed Bayoumy**
**Certifies that this is the approved version of the following report:**

# A Study of 10-bit, 100Msps Pipeline ADC and the Implementation of 1.5-bit Stage

**APPROVED BY**
**SUPERVISING COMMITTEE:**

**Supervisor:**

Nan Sun

Deji Akinwande

# A Study of 10-bit, 100Msps Pipeline ADC and the Implementation of 1.5-bit Stage

**By**

**Mostafa Elsayed Bayoumy B.E.**

**Report**

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

**Master of Science in Engineering**

**The University of Texas at Austin**

**December 2013**

Dedicated to God and my family

# Acknowledgements

I would like to thank my wife Fatima for her endless love, support and encouragement throughout doing my master degree. My siblings, Mona, Yasser and Karim, nieces and nephews deserve my wholehearted thanks as well.

I would like to sincerely thank my supervisor, Prof. Nan Sun, for his guidance and support throughout this work and for his pushing for excellence. I'd like also to thank Prof. Deji Akinwande and Prof. Ranjit Gharpurey for making studying analog circuit design such an interesting classes.

Finally, I'd like to thank all my classmates and teaching assistances whom made from this learning experience such a memorable time.

# Abstract

## A Study of 10-bit, 100Msps Pipeline ADC and the Implementation of 1.5-bit Stage

Mostafa Elsayed Bayoumy, M.S.E.

The University of Texas at Austin, 2013

Supervisor: Nan Sun

The demand on high resolution and high speed analog-to-digital converters (ADC's) has been growing in today's market. The pipeline ADC's present advantages compared to flash or successive approximation ADC techniques. The high-resolution, high-speed requirements can relatively easier be achieved using pipelined architecture ADC's than other implementations of ADC's of the same requirements. Because the stages work simultaneously, the number of stages needed to obtain a certain resolution is not constrained by the required throughput rate. Latency is a result of a multistage concurrent operation of any pipelined system. But luckily enough, latency isn't considered to be a problem in many ADC applications. In this work, a 1.5-bit stage in the pipeline ADC is completely implemented including its two voltage comparators, a DAC with three possible output voltages, and a multiplying digital to analog (MDAC) blocks. Only ideal components were used for clocking operation. At the end of design, a total harmonic distortion (THD) of less than -70 dB was achieved.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1: Introduction

The use of pipeline analog-to-digital converters (ADCs) continues to grow, both as standalone parts and as embedded functional blocks in system-on-a-chip (SoC) Integrated Circuits (ICs). They boast acceptable resolution at high-speed operation and can be integrated onto relatively small die area. Many commodity Complementary Metal Oxide Semiconductors (CMOS) technology SoCs now include embedded pipeline ADCs because of their cheap cost and high performance.

Pipelined converters attain their final resolution through a series of cascaded lower resolution stages. For example, a 12-bit converter could be a cascade of four 3-bit stages. Most engineers are comfortable with the basic concepts of a 3-bit pipeline ADC stage. However, the 1.5-bit stage also is used extensively, despite the fact that its characteristics and advantages are less widely known.

Pipeline ADCs consist of a series of stages that are isolated by sample-and-hold (S/H) buffers. The stages work concurrently. The first stage operates on the most recent sample, while the following stages operate on the remainder analog voltages, called residues from previous samples. A consequence of multistage concurrent operation is latency, meaning delay. The output code for a given sample isn't obtained until a number of clock cycles later. Latency isn't considered to be a problem in many applications that rely on data converters.

The goal of this report was to produce a 10 bit ADC with a sampling rate, $f_s$, of 100MHz while maintaining the total harmonic distortion (THD) less -70 dB. Since all it takes to design a multistage pipe line ADC is to design a single stage and cascade it multiple times depending on the required resolution of the whole architecture, we'll initially build

the first stage with ideal components and verify its operation before we start replacing each component with its real transistor-level design equivalent.

This chapter provides an introduction to the concepts discussed in the rest of the report. The first sections of this chapter will provide an overview of pipeline ADC basics and operations. The following sections will be a more in-depth discussion of the benefits of pipeline ADC topology, as well as a discussion of the advantages of using a pipeline architecture versus a flash architecture. This chapter will conclude with an overview of the organization of the rest of the report.

## 1.1 PIPELINE ADC BASICS

This section begins with an introduction to the operation of a simple pipeline ADC. After the introduction, the operation of the Multiplying DAC (MDAC), an important block in pipeline ADC design, is presented in detail.

### 1.1.1 Pipeline ADC Operation

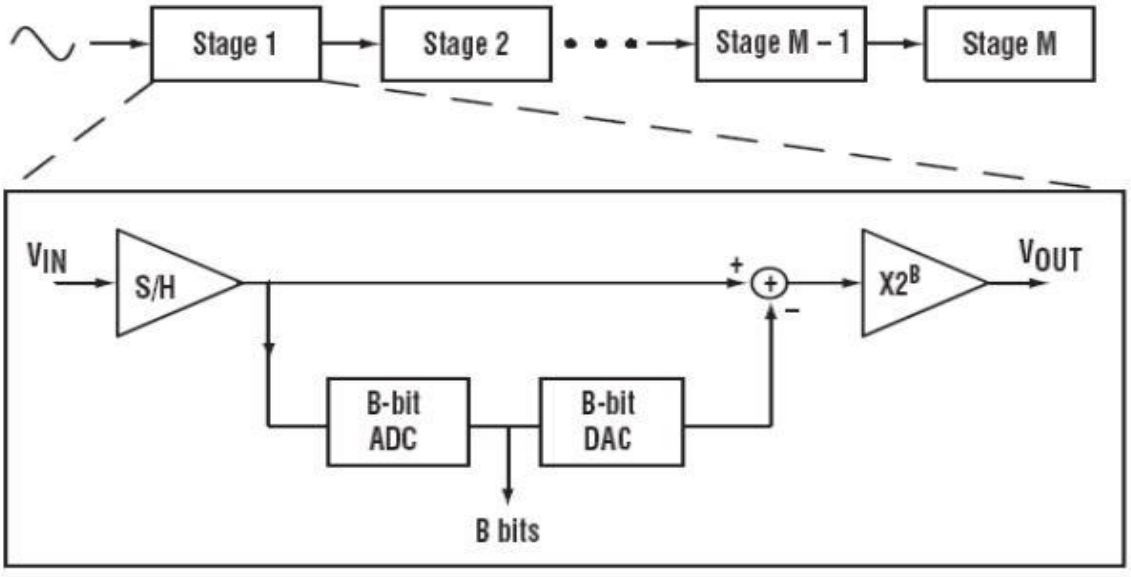


Fig.1.1 A general stage in a pipeline ADC

Figure 1 shows a block diagram of a general pipelined ADC with M stages. For every stage, there is a S/H, a low-resolution ADC, a low-resolution digital-to-analog converter (DAC), a subtracter, and a controlled gain amplifier. Each stage samples and holds the output from the previous stage. The held input is converted into a low-resolution digital code by the ADC, and then back to analog voltage by a DAC. The DAC output is then subtracted from the held input, and the difference is amplified to produce an output residue voltage that is passed to the next stage. In general, the amplification is corresponding to the resolution of the stage such that the full voltage range available is utilized. For example, a 3-bit stage would have an amplification of 8. The amplifier can be placed either in front of, or following, the subtracter.

### 1.1.2 1.5-Bit Stages

For high-speed converters, there's an advantage to minimum stage resolution. It minimizes the required inter-stage gain, which in turn maximizes bandwidth, since gain-bandwidth is a constant for a given technology. This factor is particularly important for parts made using a cost-effective commodity CMOS wafer-fabrication process, as maximum achievable speed is required from critical circuit blocks such as operational amplifiers. Assume symmetrical reference voltages of ±VREF. The minimum possible stage resolution (and maximum bandwidth) of 1 bit would place an analog decision level midway between the reference voltages—that is, at ground. The amplifier would have a gain of 2.

A 1.5-bit stage is a 1-bit stage into which some redundancy is built to provide a large tolerance for components' imperfections. A digital correction algorithm later eliminates the redundancy. A 1.5-bit stage is actually a stage that represents approximately

1.5 bits. The 1.5-bit stage uses two symmetrical analog comparison voltage levels, VH (high value) and VL (low value), instead of one (Fig. 1.2). The amplifier has a gain of 2. The choice of voltage levels VH and VL isn't critical, but because of the following gain block, they must lie within the range of -VREF/2 and +VREF/2.

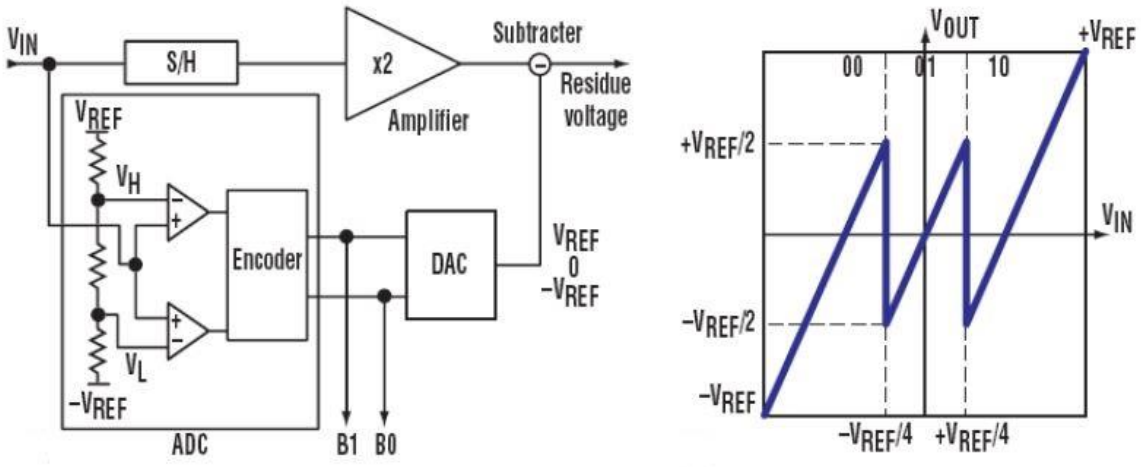


Fig 1.2 A 1.5-bit pipeline ADC stage

A common choice is VH = 0.25×VREF and VL = -0.25×VREF. The 1.5-bit configuration holds an advantage in that there's no analog decision level or trip point at mid-range, which benefits low-signal-level operation. The operating voltage range is divided into three sections: High (H) above VH, Mid (M) between VH and VL, and Low (L) below VL. This system is known as Redundant Signed Digit (RDS) because the High range was originally tagged as +1, Mid-range as 0, and Low range as -1. Table 1 lists summary information about the 1.5-bit stage configuration.

| $\mathbf{V_{IN}}$ | Range | B1 | B0 | DAC O/P | Residue |
|---|---|---|---|---|---|
| $\mathbf{V_{IN} > V_H}$ | H | 1 | 0 | $\mathbf{+V_{REF}}$ | $\mathbf{2V_{IN} - V_{REF}}$ |
| $\mathbf{V_L < V_{IN} < V_H}$ | M | 0 | 1 | 0 | $2V_{IN}$ |
| $\mathbf{V_H < V_L}$ | L | 0 | 0 | $\mathbf{-V_{REF}}$ | $2V_{IN} + V_{REF}$ |

Table 1.1: Summary Information for Fig 1.2

The stage low-resolution ADC comprises two voltage comparators plus some simple encoding. The ADC stage output consists of two bits—B1 and B0. This is the initial digital output, before code conversion and error correction. The output codes are 00, 01, and 10 for $V_{IN}$ (input voltage signal) in the L, M, and H input ranges, respectively. The DAC outputs are -VREF, 0, and +VREF for VIN in the L, M, and H input ranges. The stage transfer function ($V_{OUT}$ / $V_{IN}$), which is highly nonlinear, is shown in Fig 1.2.

### 1.1.3 Circuit Implementation

The 2-bit ADC block in each 1.5-bit stage in the pipelined ADC can be implemented in a flash ADC architecture which is most engineers are familiar with. This architecture is preferred since it is a low resolution coarse flash ADC. It uses only 2 voltage comparators and relatively faster than other type of ADC's architectures which makes it ideal to use in a pipeline operation to improve the overall latency of the system. For the DAC implementation, a basic voltage-dividing resistor string typically creates the reference voltages. For CMOS circuitry, this is the most critical use of resistors. All other

high-accuracy operations, such as the ×2 amplification, are performed using capacitor ratios. The S/H, DAC, subtrater, and ×2 amplifier generally can be combined in a multiplying DAC functional block (MDAC). MDAC operation is based on charge transfer using ratioed capacitors array. An illustration of principles of charge-domain operation is proposed in coming sections of the report. Though these descriptions have been in terms of single-ended circuitry for clarity, the implementations of the pipelined ADC with 1.5-bit stages are fully differential systems. Another important issue, not addressed here, is clocking and the synchronization of the data output circuitry.

### 1.2 ADVANTAGES OF PIPELINING TOPOLOGY

Pipeline ADCs provide an optimum balance of size, speed, resolution, and power consumption. Comparing to the flash ADC architecture, pipeline ADC utilizes much less number of comparators for the same resolution which results in significant reduction in power and size. In the flash ADC the number of comparators increases exponentially with the resolution, while on the other hand, it increases linearly in the pipeline architecture. For example, to build a 10-bit ADC in a flash architecture we would need $2^{10}$ comparators, while we would need $18, (10-1)\times 2,$ comparators only in a pipelining architecture (assuming 9 stages, each stage includes 2 comparators). This huge reduction of in design area and power makes pipeline ADC an ideal for mid to high resolution applications. Another advantage of using fewer comparators is to eliminate sparkle codes and thermometer bubbles.

The use of separate track-and-hold (T/H) circuitry for each stage release each previous T/H to process the next incoming sample, enabling conversion of multiple samples simultaneously in different stages of the pipeline.

## 1.3 ORGANIZATION OF THE REPORT

The rest of the report will focus on the design details of this ADC. Chapter 2 discusses architectural decisions and improvements that were applied to the general circuit described in Chapter 1. Chapter 3 discusses the full design of the ADC using ideal components. Chapter 4 covers the design and integration of the OTA, as well as final ADC performance numbers with the integrated OTA. Chapter 5 provides closing remarks, as well as a discussion of the challenges encountered during the implementation of the design.

# Chapter 2: Pipeline ADC Concept

Pipeline analog-to-digital converters have been extensively used in high-speed medium-accuracy systems due to their partitioned nature. The accuracy partitioning of pipeline ADCs significantly reduces the power and area requirements of a given ADC design, thus allowing designers to push pipeline ADCs to very high speeds.

This chapter will cover pipeline ADCs in their most common implementations, as well as established circuits and systems design practices for improving the performance of a pipeline ADC. There will also be some discussion on pipeline ADC error sources.

## 2.1. BASIC A/D CONCEPTS

The basics of the analog-to-digital conversion should be covered before any attempts are made to discuss Pipeline ADCs in detail. The major function of any ADC is to convert a continuous-time, continuous-value signal into a discrete-time, discrete-value signal. This implies quantization in both the time and signal-level (or “value") domains, where value is most often voltage. These two quantization steps are functionally independent, but in some cases they are combined into a single step.

The first step is time-quantization, also known as sampling. A simple example of this function in circuit form is shown in Fig. 2.1. In this example, while the switch is closed, the voltage on the sampling capacitor closely matches the input voltage. When the switch is opened at time $t = t_o$, the value of the input signal at the switching instant $t_o$ is held on the sampling capacitor.

The next step in A/D conversion is signal-level-quantization, and is often given the general name quantization. There are many different methods of accomplishing this task, but they can all be reduced to two basic implementations: serial and parallel. Serial

quantization requires some kind of feedback system to identify what the previously resolved bit was, but parallel converts the signal all at once and does not need any information feedback. The parallel implementation is the basis of the flash ADC.

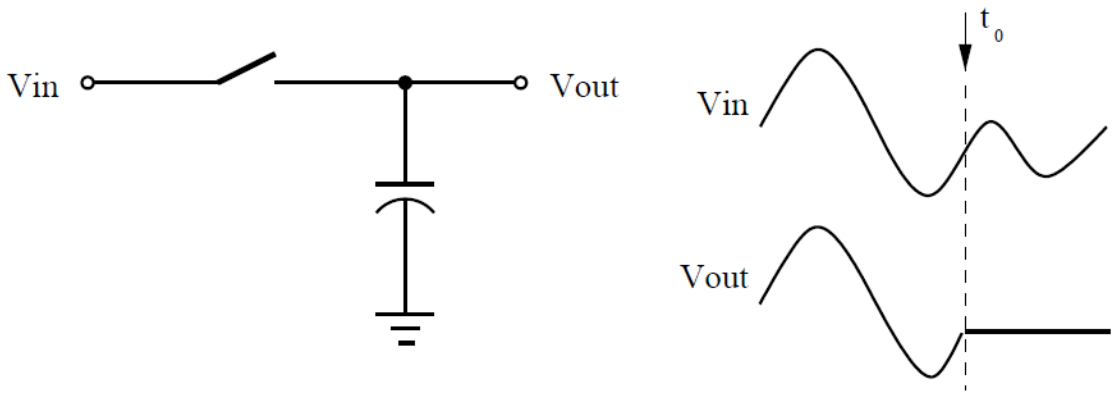


Fig. 2.1: Sampling in A/D Systems

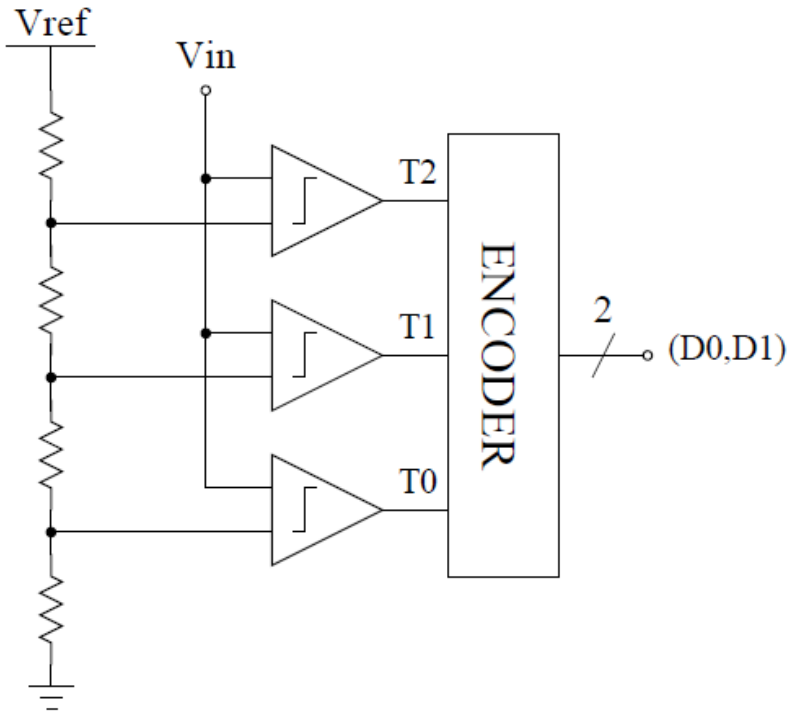


Fig. 2.2 Flash ADC

The flash ADC compares the input with the analog equivalents of all possible digital levels in parallel. In other words all levels comparisons occur at the same time. A simple example of a flash comparator (without the sampling function) is shown in figure 2.2. The output of this converter is a digital word that approximates the input signal with respect to an analog reference voltage ($V_{ref}$). The intermediate digital signals (T0, T1, and T2) that appear in figure 2.2, represent a *Thermometer Code* signal. Thermometer Code is defined as a grouping of bits (like binary code) where the total number of ones increases for each increase of one in value. The ones "fill up" like a thermometer where the value is determined by the location of the boundary between the ones and zeros. For example the value of '3' represented in an 8-bit thermometer code is '00000111'.

A common measurement reference in data-converter design is known as LSB (Least Significant Bit). This is defined as converter full-scale divided by the total number of levels converted. For example, if a flash A/D converter is designed for a 0V-1V input range and converts 4 bits, 1 LSB is equal to (1/16) V. Many specifications in data-converter designs are given in LSBs.

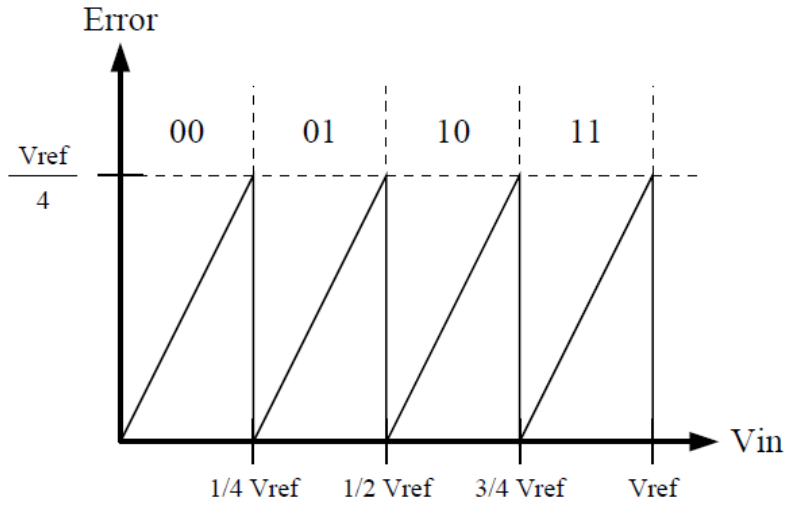


Fig. 2.3 Flash ADC Quantization Error

The quantization step introduces an error in the final value of the signal known as *quantization error* and can be shown to have an input/output relationship as is shown in figure 2.3. It is easy to show that the quantization error can be reduced by increasing the accuracy of the data conversion; however, for a flash ADC this can be quite costly. The reason for that is for each one-bit increase in ADC resolution, the number of comparisons required goes up by a factor of two.

A closer look at the quantization error reveals that the signal is still present in the error. If the quantization error of the conversion can be extracted and amplified, it would be possible to repeat the conversion step to resolve more bits. This structure is known as a Two-Step ADC and is often used to increase the resolution of high-speed flash ADC designs. The quantization error can be generated by subtracting the analog-equivalent of the resolved digital bits from the original input signal. This signal is known as the residue of the first conversion. An amplification block is then used to make the maximum possible residue value equal to the analog reference voltage (figure 2.4). The digital-to-analog conversion, subtraction and multiplication are often combined into one functional circuit block known as the MDAC (Multiplying-Digital-to-Analog Converter).

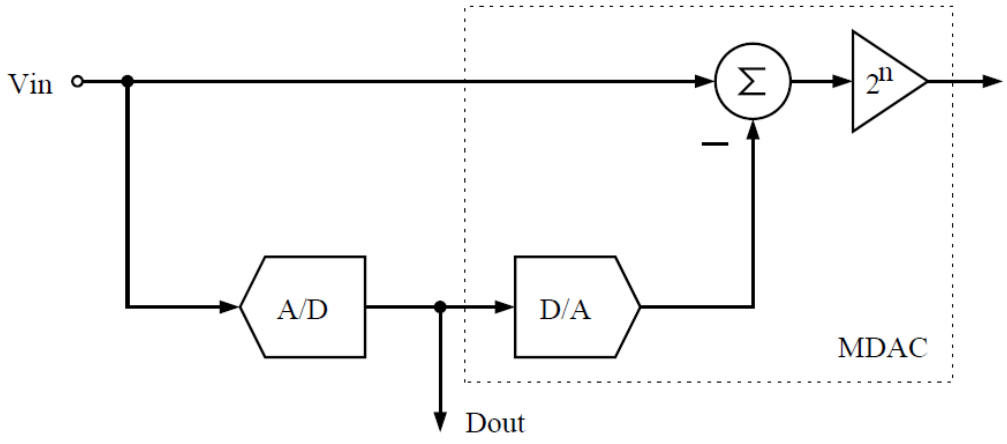


Fig. 2.4 Pipeline ADC Stage Building Blocks

The previous illustration can be looked at as if we have to two different back-to-back ADC blocks as shown in figure 2.5. The first one is the "front-end ADC" with a "coarse resolution" and introduce large quantization error to the converted signal. The second one is the "back-end ADC" and its function is to quantize the quantization error produced by its peer front-end ADC. The back-end ADC acts as fine-tune ADC with a "fine resolution". The digital output of the whole pipeline ADC is the combination of the coarse and fine resolutions of the front-end and the back-end ADC subsystems. The digital-to-analog (D/A) block in figure 2.4 is replaced with a pass-through block since the D/A converters don't introduce quantization error to the converted signal, unlike the A/D converters. As we can see from the equations on the signals paths in figure 2.5 that the accuracy of the final digital output depends only on the accuracy of the back-end ADC.

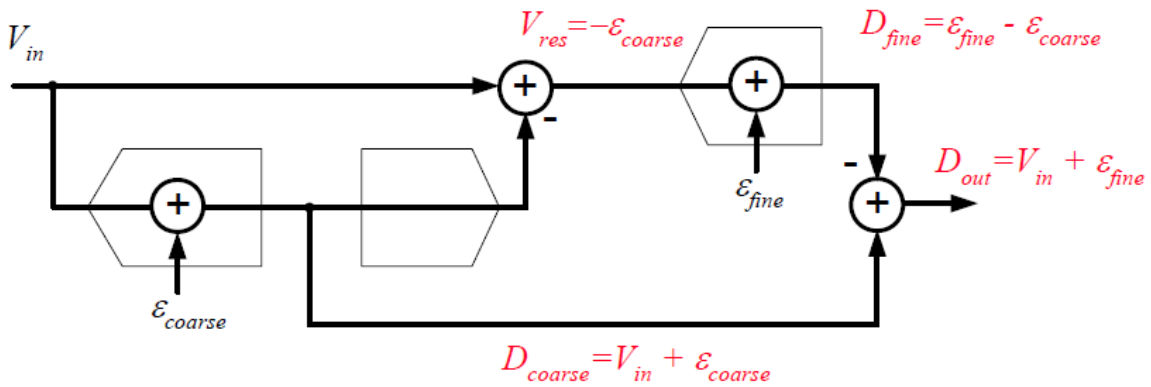


Fig. 2.5 Pipeline ADC Stage Modeling

## 2.2 PIPELINE ADC

The two-step ADC can be further extended by adding additional MDAC and comparator stages - this creates the basis for the pipelined ADC. In fact, the system shown in figure 2.4 is a simple representation of a pipeline ADC stage. This stage is repeated until the number of bits required can be resolved from the pipeline. Figure 2.6 shows the architecture of scaling a single stage into multiple stages till the required resolution is achieved. The last stage in the pipeline ADC is much less complex in design than the other stages, it is basically a flash ADC as there is no more conversion to be processed to produce the digital output.

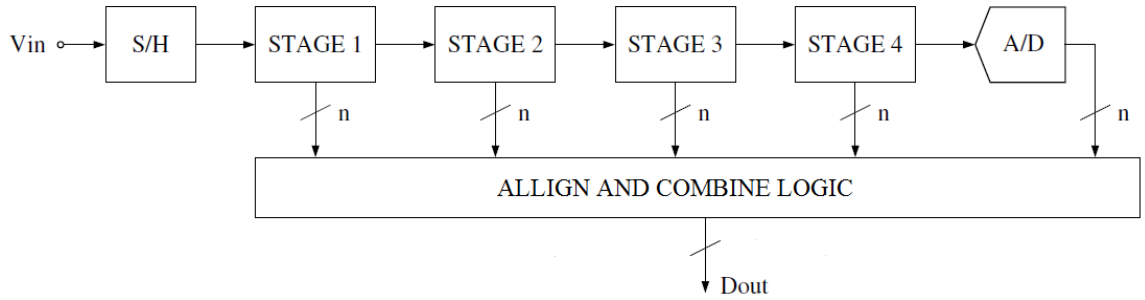


Fig. 2.6 Pipeline ADC

### 2.2.1 PER-STAGE RESOLUTION

The pipelined ADC architecture allows a designer flexibility in the choice of quantizer (or sub ADC) resolutions for each stage. As such, the most common implementation of the pipeline ADC is a 1.5-bit per stage structure (the partial-bit resolution nomenclature will be explained shortly). The reduction in stage resolution allows for very fast conversion times and small power consumptions. This is due to the relaxed requirements on the sub ADC portion of the pipeline stage, and is most effective in designs for low-to-medium resolutions. Most other pipeline ADCs fall into a class of

pipelines known as Multi-Bit Pipeline ADCs. These are pipelines with a greater than 1.5-bit per stage sub ADC resolution, and are most useful in design situations requiring large overall ADC resolutions. The advantage of a multi-bit pipeline is its large gain at the output of the first MDAC and hence noise power contributions of all following stages are reduced by the squared MDAC gain. High-resolution designs place stringent requirements on the ADC noise and, with the larger gain from a multi-bit stage, the size and power consumptions of the back-end stages can be aggressively scaled. The aggressive scaling leaves more budgeted power and area available for the critical initial stages of the pipeline.

### 2.2.2 DIGITAL REDUNDANCY

Another factor in the choice of sub ADC resolution is the achievable accuracy of the comparator block. This concern is not limited to the choice of stage resolution, but also affects the overall ADCs resolution and linearity. Without careful consideration, it is also possible for comparison errors to erroneously saturate lower bits in the ADC.

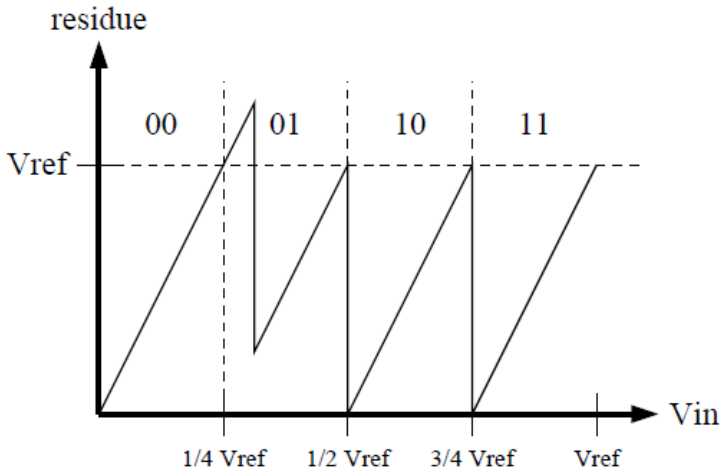


Fig. 2.7 Residue Error Example

Consider the example shown in figure 2.7. A comparison error at the '00' to '01' transition has allowed the residue voltage to extend beyond the ADC reference voltage. For all signals greater than $\frac{V_{ref}}{4}$, and less than the actual comparison level, the converter will produce all ones in the remaining pipeline stages and generate significant nonlinearities in the converted signal. An error in the opposite direction will have the same clipping effect, but instead all of the remaining conversions will be zero.

One solution to this problem would be designing a very accurate voltage comparators for all the stages in the pipeline architecture which will be at the cost of increased circuit complexity. Such solution is not practical since the designers will have to deal with different types of mismatch in the comparator blocks itself rather focusing on improving other aspect of design like area or power consumption, etc.

Another, more elegant, solution is to allow each pipeline stage's conversion to overlap with adjacent stages which is commonly referred to as *digital redundancy* or *digital correction.* This allows neighboring stages to weigh-in on the correctness of a given output code from neighboring stages. Not only does this correct mistakes in conversion, but it also keeps the residue output voltage within $V_{ref}$ as long as the comparison error never exceeds $\frac{1}{2}$ LSB.

This technique greatly reduces the accuracy requirement of the sub ADC. Figure 2.8 shows the transfer function of a 1.5-bit MDAC that applies *digital correction* technique. A 1.5-bit sub ADC quantizes the input to 3-levels and the residue is limited to half the full-scale range.

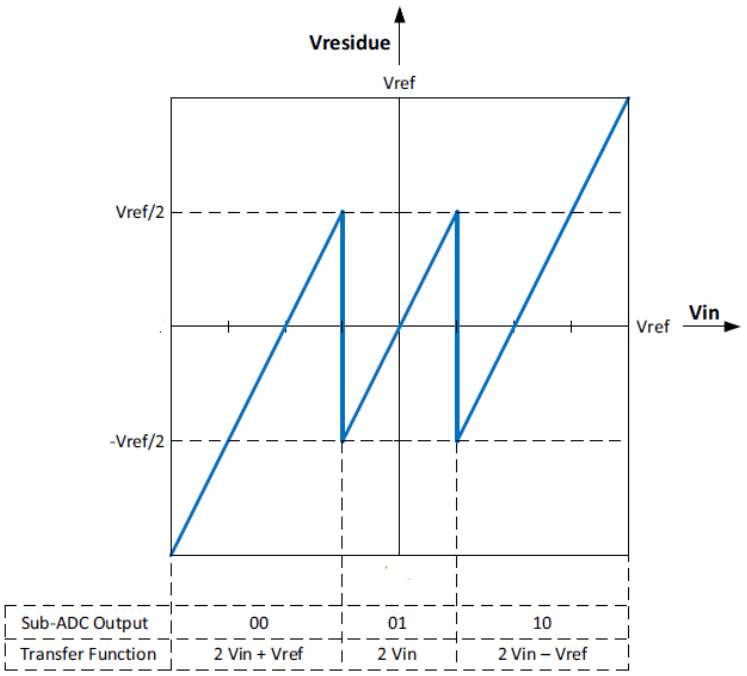


Fig. 2.8 Transfer function of a 1.5-bit MDAC

If comparator threshold offsets cause the residue signal to fall outside its normal range, the residue signal will be accurately passed onto the next stage provided the offset is within $\frac{\pm V_{ref}}{4}$. Figure 2.9 demonstrates this by showing the transfer function of a 1.5-bit MDAC when there are threshold offsets in the sub-ADC. The additional bit from each adjacent stage is overlapped to correct the over-range error. The fourth level is removed because the MDAC only needs to indicate whether the residue output is above or below $\frac{V_{ref}}{2}$ and therefore, there is technically only an overlap of half a bit.

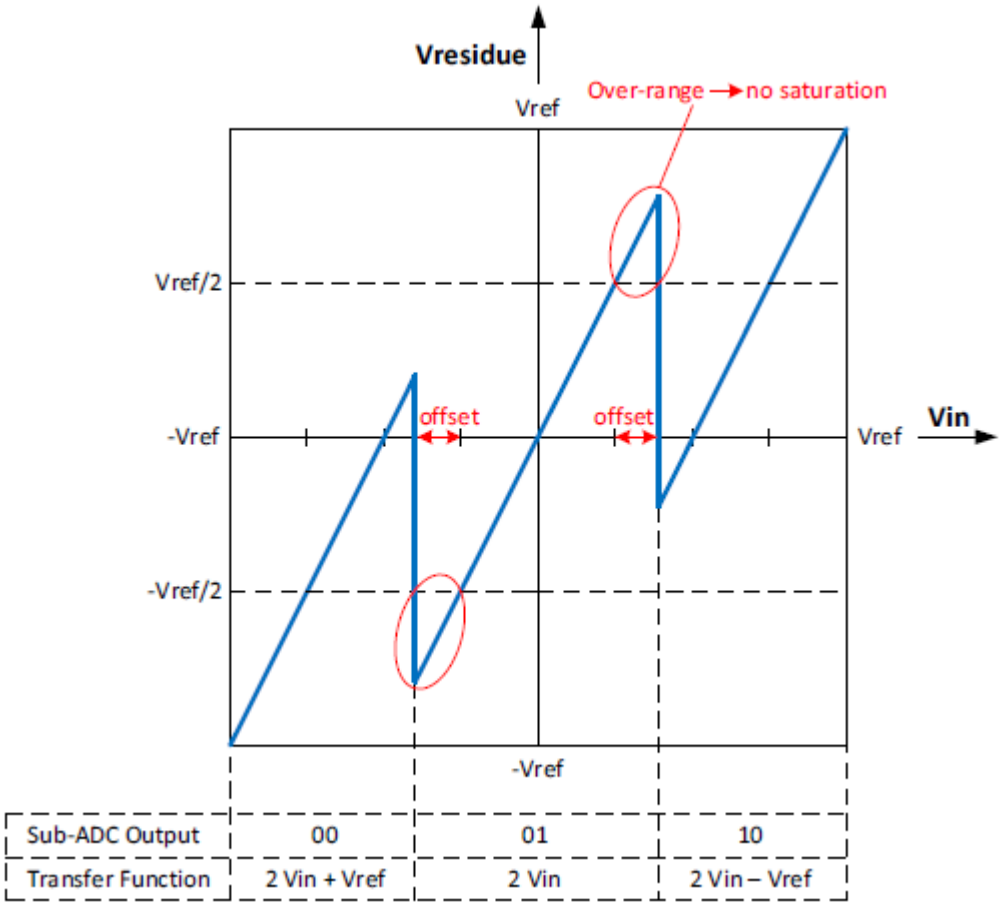


Fig. 2.9 Transfer function of a 1.5-bit MDAC with comparator offset

As a result, the accuracy of the 1.5-bit sub ADC is reduced from N to 2 bits with *digital correction* technique. In general, with *digital correction* technique, the accuracy of the sub ADC can be reduced if fewer bits are resolved in each stage. The seminal work on this concept was completed by S. Lewis, and also introduced the 1.5-bit per stage architecture. This effectively allows the removal of one comparison in a 2-bit sub ADC (making it a 1.5-bit stage). The concepts of digital redundancy and $\frac{\text{LSB}}{2}$ offset are now standard in most Pipeline ADC designs.

### 2.2.3 TYPES OF ERRORS IN DATA CONVERTERS

Non-idealities in data converters result in errors in conversion that are measured in LSBs. These errors are commonly referred to as Differential Non-Linearity (DNL) and Integral Non-Linearity (INL). DNL errors are defined by measuring the steps between adjacent codes. If the converter is ideal, the step sizes should be exactly equal to 1 LSB. This corresponds to a DNL of 0 LSB. INL errors on the other hand as their name implies are defined as the integration of all DNL values from code 0 to code $2^N - 1$.

Large errors in DNL can result in missing codes. These can occur when the DNL is greater the 1.0 LSB. Generally, the maximum DNL on a part should be less than 0.5 LSB if missing codes could be an issue in the system. This is because two $\frac{\text{LSB}}{2}$ comparison level errors can create a missing code as shown in figure in 2.10.

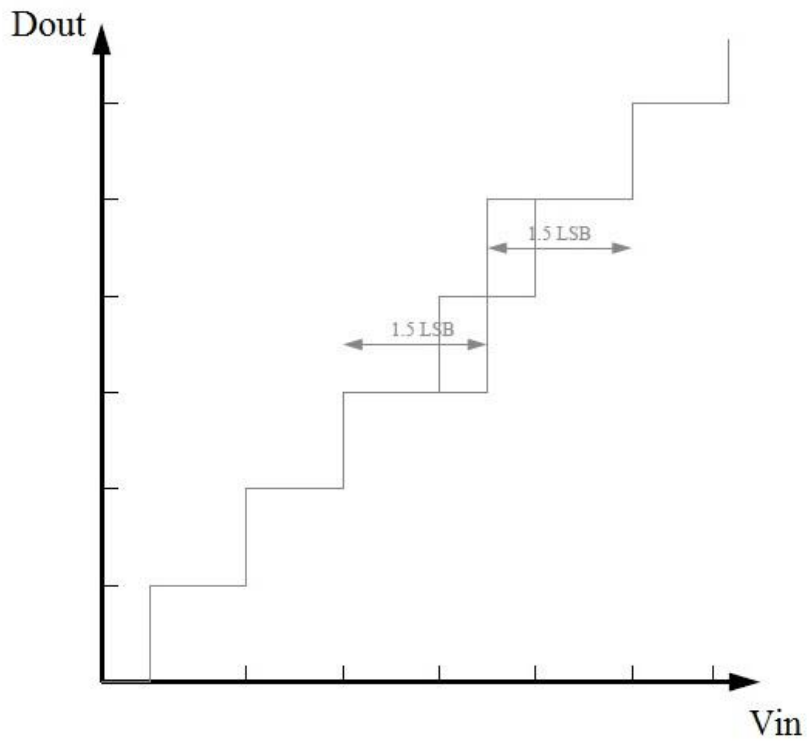


Fig. 2.10 Missing Code DNL Condition

These measurements (DNL and INL) are a "static" measurement of the non-linearity. A "dynamic" linearity measurement can be accomplished by applying a single-tone sine wave to the input of the ADC and measuring the spectra to determine harmonic components. Generally, the INL is a pretty good approximation of the value generated from this type of measurement; even so, both measurements are usually taken when characterizing a new ADC design.

## 2.3 DATA CONVERTERS SPECTRAL METRICS

This section defines some of the performance metrics that will be used throughout the rest of this report. We will start with knowing the effect of the quantization error (noise) on the overall linearity of the ADC giving that there is no other sources of noise. The quantization error of any ADC converter is directly related to its step size which is defined as the least significant bit (LSB). For ADC with a resolution of N bits, and a full-scale signal voltage ($V_{FS}$) , the expression of LSB can be written as:

$$\mathrm{LSB} = \frac{V_{FS}}{2^{N}} \tag{2.1}$$

Assuming that the transition level lies halfway between each digital code, the magnitude of the maximum error of an ideal ADC ($\epsilon_q$) can be defined as:

$$\epsilon_q = \frac{\mathrm{LSB}}{2} \tag{2.2}$$

Assuming a full-scale sinusoidal input and a uniform distribution of $\epsilon_q$ , the signal-to-quantization-noise ratio (SNQR) of an N bit ADC can be expressed as:

$$\mathrm{SNQR} = \frac{P_{sig}}{P_{qnoise}} \tag{2.3}$$

$$\mathrm{SQNR} = \frac{\frac{1}{2}\left(\frac{\mathrm{V_{FS}}}{2}\right)^2}{\frac{\mathrm{LSB}^2}{12}} = 6.02\mathrm{N} + 1.76\mathrm{dB} \tag{2.4}$$

Where $P_{sig}$ is the total power of the input voltage is signal, and $P_{qnoise}$ is the total quantization noise power.

The above expression for SQNR defines the signal-to-noise ratio (SNR) of an N-bit ADC with zero electronic noise and a perfectly linear transfer function. SQNR is considered to be an optimistic metric since, in reality, all real electronic circuits generate some electronic noise and have non-linear transfer functions.

A better metric of ADC performance is known as the signal-to-noise and distortion ratio (SNDR). The expression for SNDR is:

$$\mathrm{SNDR} = \frac{\mathrm{P_{sig}}}{\mathrm{P_{noise}} + \mathrm{P_{distortion}}} \tag{2.5}$$

Where $\mathrm{P_{noise}}$ is the total noise power including quantization and electronic noise and $\mathrm{P_{distortion}}$ is the distortion power. SNDR metric can also be used in calculating the effective resolution of an ADC (sometime the effective resolution is referred to as *Effective Number of Bits, ENOB*). Since SNDR is more realistic metric of the ratio of desired signal power to the un-desired signal power, we can substitute SNDR for SQNR in equation 2.4 and solving for N, the effective number of bits (ENOB) of an ADC can be expressed as:

$$\mathrm{ENOB} = \frac{\mathrm{SNDR(dB)} - 1.76\mathrm{dB}}{6.02\mathrm{dB}} \tag{2.6}$$

The last metric that we will discuss and will be used the most throughout the rest of this report is the total harmonic distortion (THD), which is defined as the ratio of the sum of the powers of all harmonic components (total distortion power) to the power of the fundamental frequency (signal power).

$$\mathrm{THD} = \frac{P_2 + P_3 + P_4 + \cdots + P_\infty}{P_1} = \frac{\sum_{i=2}^{\infty} P_i}{P_1} \tag{2.7}$$

where, $P_i$ is the power of the $i^{th}$ harmonic.

Empirical measurements show that the total distortion power consists of the 2nd through 7th harmonics.

# Chapter 3: 1.5-bit Stage Design Using Ideal Circuit Blocks

After defining all of the architectural features of the design, the next step was to use ideal circuit blocks to simulate the 1.5-bit stage design. Designing in this manner gave valuable insights into how different circuit block specifications affected overall ADC performance. This chapter begins with an explanation of the ideal circuit blocks that were used in this design. It follows with a discussion of the design and simulated results from a single-ended version of the ADC. This chapter concludes with a discussion of the design of the fully differential 1.5-bit stage and its simulated performance.

## 3.1 IDEAL CIRCUIT BLOCKS

A set of ideal circuit blocks were provided by Dr. Nan Sun for this design. These circuit blocks include a single-ended OTA, a differential OTA, a voltage comparator, a switch, a clock generator, a single-ended to differential signal converter and a differential-ended to single signal converter. This section discusses the operation of these blocks, and their input/output details as well as their settable parameters.

### 3.1.1 IDEAL OTA

The OTA model is used in the design of the first-stage MDAC. The single-ended and differential OTA models are essentially the same, the difference between the two is that the differential model has an additional output pin for the negative output voltage. The pins for the OTA models are given in Table 3.1. The differential model has an internal common-mode feedback (CMFB) circuit that ensures the common-mode output of the OTA matches the voltage on $V_{cmo}$. In this design, the power supply is 1.8V.

| Name | Pin Type | Description |
|---|---|---|
| $V_{dd}$ | Supply | DC Power Supply |
| $V_{in}$ | Input | Positive input voltage |
| $V_{inb}$ | Input | Negative input voltage |
| $V_{cmo}$ | Output | Common-mode output voltage |
| $V_{out}$ | Output | Positive output voltage |
| $V_{outb}$ | Output | Positive output voltage, differential model only |

Table 3.1 OTA Ideal Model Pins

In addition to the pins, the OTA models also have a number of parameters that govern its behavior. These parameters are given in Table 3.2 The device characteristics that are controlled by these parameters are the output voltage, and the input capacitance.

| Parameter | Description |
|---|---|
| Gain | OTA DC gain |
| $f_T$ | Transient frequency |
| $g_m / I_d$ | Current efficiency |
| $g_m$ | Transistor transconductance |

Table 3.2: OTA Ideal Model Parameters

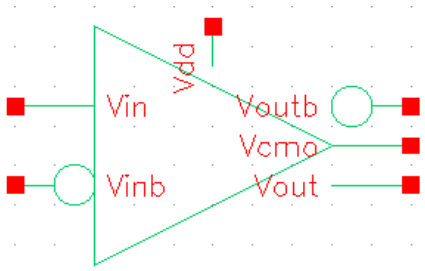


Fig. 3.1 Ideal OTA symbol

The Gain parameter is the ratio of the output voltage to the input voltage. The input capacitance of the OTA is:

$$C_{in} = \frac{g_m}{2\pi f_T} \tag{3.1}$$

Where $g_m$ and $f_T$ are OTA parameters. Equation (3.1) is simply representing the relationship between transient frequency, transconductance and input capacitance of any gain cell. The transient frequency is an important metric to know about since it limits the bandwidth of a device to the amount of $g_m$ for a given input capacitance.

### 3.1.2 IDEAL COMPARATOR

The pins for the ideal comparator model are given in table 3.3. When clock is low, the comparator is in tracking mode. During tracking phase, the input signals are sampled on the capacitors implemented inside the ideal comparator model. When the clock is high,

| Name | Pin Type | Description |
|---|---|---|
| $V_{in}$ | Input | Positive input voltage |
| $V_{inb}$ | Input | Negative input voltage |
| CLK | Input | Clock signal |
| d | Output | Positive output voltage |
| db | Output | Negative output voltage |

Table 3.3: Ideal Comparator Pins

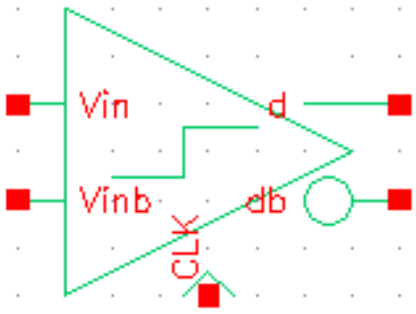


Fig. 3.2 Ideal Comparator symbol

the comparator does the comparison process. During this phase the decision is produced on the output pins of the model according the following relationship.

$$d = \begin{cases} \text{High} & \text{if } V_{in} > V_{inb} \\ \text{Low} & \text{if } V_{in} < V_{inb} \end{cases}$$

The output pin db is having an opposite value to output pin d when the comparison phase is finished.

### 3.1.3 IDEAL CLOCK GENERATOR

The clock generator is used to generate the non-overlapping clocks for clocking the design. The pins for this block are given in Table 3.4.

| Name | Pin Type | Description |
|---|---|---|
| f1 | Output | $\phi_1$ clock signal |
| f1e | Output | $\phi_1$ clock signal with early falling edge |
| f2 | Output | $\phi_2$ clock signal |
| f2e | Output | $\phi_2$ clock signal with early falling edge |
| f1b | Output | Inverted $\phi_1$ clock signal |
| f1eb | Output | Inverted $\phi_1$ clock signal with early falling edge |
| f2b | Output | Inverted $\phi_2$ clock signal |
| f2eb | Output | Inverted $\phi_2$ clock signal with early falling edge |

Table 3.4 Ideal Clock Generator Pins

The parameters that control the behavior of these outputs are given in table 3.5.

| Parameter | Description |
|---|---|
| Period | Clock period |
| t_early | Time difference between the falling edge of f1 and f1e |
| t_edge | The rise and fall times |
| t_nonoverlap | The non-overlapping time between f1 and f2 |

Table 3.5 Ideal Clock Generator Parameters

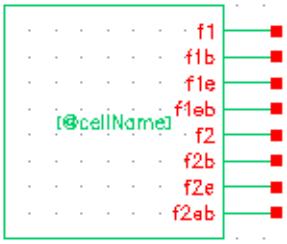


Fig. 3.3 Clock generator symbol

### 3.1.4 IDEAL SWITCH

The switch model has three pins, given in table 3.6. The switch is open when the voltage on the ctrl pin is less than 0.9V. When the voltage on ctrl is greater than 0.9V, the

| Name | Pin Type | Description |
|---|---|---|
| A | Input/Output | One side of switch |
| B | Input/Output | One side of switch |
| Ctrl | Input | Control voltage |

Table 3.6 Ideal Switch Pins

switch is closed and A is connected to B. The switch has only one parameter, given in table 3.7.

| Parameter | Description |
|---|---|
| ron | Resistance when switch is closed |

Table 3.7 Ideal Switch Parameters

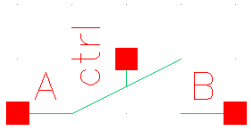


Fig. 3.4 Ideal switch symbol

### 3.1.5 SINGLE-TO-DIFFERENTIAL CONVERTER

This block takes in a single-ended signal along with a common-mode voltage and outputs a differential voltage. The pins for the single-to-differential converter are given in Table 3.8. The relationships between the output voltages and the input voltages are:

$$V_{out} = V_{cmi} + \frac{V_{in}}{2}$$
$$V_{outb} = V_{cmi} - \frac{V_{in}}{2} \quad (3.1)$$

| Name | Pin Type | Description |
|---|---|---|
| $V_{in}$ | Input | Single-ended input voltage |
| $V_{cmi}$ | Input | Common-mode input voltage |
| $V_{out}$ | Output | Positive output voltage |
| $V_{outb}$ | Output | Negative output voltage |

Table 3.8 Ideal Single-Ended-to-Differential Converter Pins


Fig. 3.5 Single-to-differential converter symbol

### 3.1.6 DIFFERENTIAL-TO-SINGLE CONVERTER

This block takes two voltage signals and outputs their differential voltage value along with their common mode voltage value. The pins for the differential-to-single converter are given in table 3.9. The relationships between the output voltages and input voltages are:

$$V_{dm} = V_{ip} - V_{im}$$
$$V_{cm} = \frac{V_{ip} + V_{im}}{2} \quad (3.2)$$

| Name | Pin Type | Description |
|---|---|---|
| $V_{ip}$ | Input | Input voltage |
| $V_{im}$ | Input | Input voltage |
| $V_{cm}$ | Output | Common output voltage |
| $V_{dm}$ | Output | Differential output voltage |

Table 3.9 Ideal Differential-to-Single Converter Pins

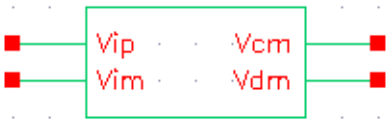


Fig. 3.6 Differential-to-Single converter symbol

## 3.2 Design Target Specifications and Implementation

The focus of the coming part of this chapter is to design the first stage of a fully-differential 1.5-bit per stage pipelined ADC using ideal circuit building blocks that we illustrated earlier, including operational transconductance amplifier (OTA), switch, voltage comparator, clock generator, single-to-differential converter, and differential-to-single converter.

### 3.2.1 Design Target

The specifications for the first pipelined stage are shown in the following table.

| | |
|---|---|
| Power supply | 1.8 V |
| Differential input signal range | 2 V peak-to-peak |
| Sampling rate | 100 MHz |
| Input referred noise | < 200 uV rms |
| THD at input (31/64)×100MHz | < -70 dB |
| Power consumption | As small as possible |

Table 3.10 1.5-bit stage specification

In our design for the 1.5-bit stage, we don't have a dedicated front-end sample-and-hold circuit. In other words, the first pipelined ADC stage directly connected to the continuous time input signal. To simplify the circuit analysis, we will study a single-ended circuit model for the first stage of the pipeline ADC as the calculations of the required design parameters are the same for both single-ended and fully differential circuits.

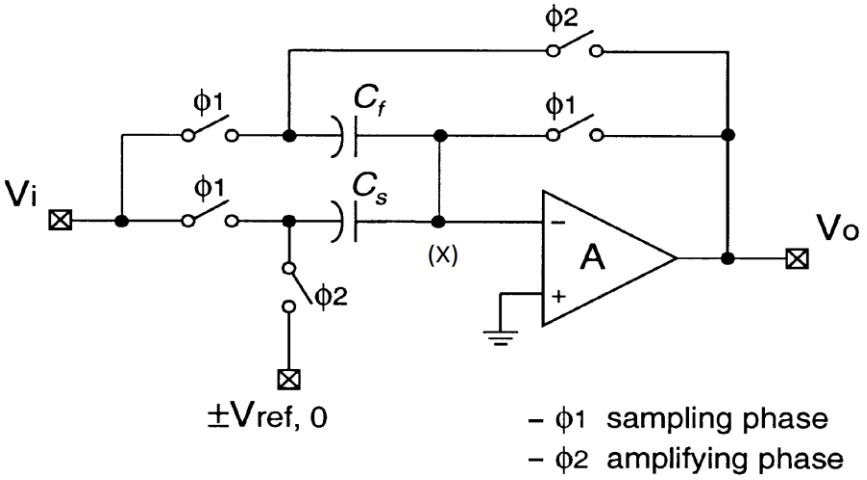


Fig 3.7 1.5-bit single-ended pipelined ADC with a capacitor-flip-over MDAC

A widely used "capacitor-flip-over" MDAC is shown in figure 3.7, where one of the capacitors pair used during sampling phase is utilized again during charge transfer (amplifying) phase by proper switching mechanism. This technique save on design area and reduce the effect of capacitors mismatch.

### 3.2.2 DESIGN FLOW

#### 3.2.2.1 Capacitor size analysis

Although capacitors are ideally noiseless elements, in a sampled system, sample and hold capacitors capture noise generated by noisy elements such as switch resistors, op-amps, etc. Given the max limit on the input referred noise, we can easily determine the capacitor sizes by calculating the noise generated from the different components of the system during sampling and charge transfer phases.

Let's first go over some relationships that governs the capacitance, noise charge and noise voltage. Consider the following noise analysis of a capacitor sampling resistor noise as shown in figure 3.8, where $V_r$ is the equivalent voltage noise source of a resistor R per hertz of bandwidth.

$$\frac{V_r}{\sqrt{\Delta f}} = \sqrt{4KTR} \tag{3.3}$$

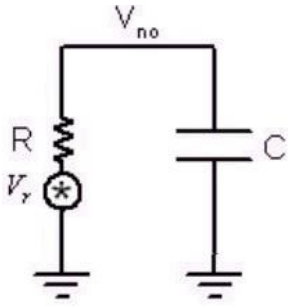


Fig. 3.8 RC noise model

Where K is the Boltzmann constant in joules per kelvin, T is the resistor's absolute temperature in kelvins.

The transfer function, H(s), of the system will shape the spectrum of the noise source. Under white noise (with equal power over all frequencies), it is convenient to replace the transfer function by a "brick wall" filter, in which the noise power is the same up to a certain frequency Δf and after that is zero.

For the 1st order low-pass filter, Δf, or Equivalent Noise Bandwidth (ENBW), can be proven to be equal to:

$$\Delta f = f_{-3dB}\frac{\pi}{2} \tag{3.4}$$

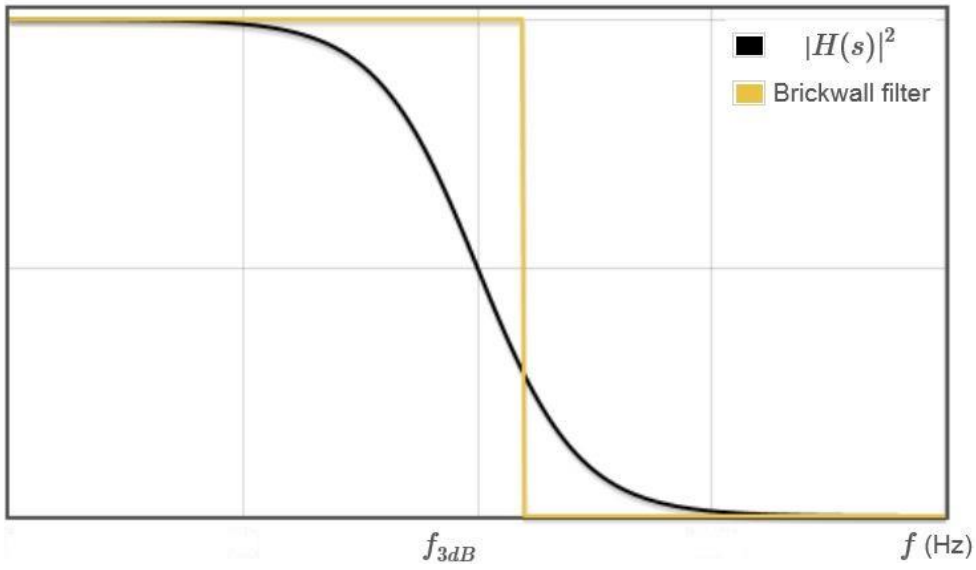


Fig. 3.9 Equivalent Noise Bandwidth

In figure 3.9, the area under this rectangular brick wall filter must be the same as the area under the original system. Also it is showing equivalent noise bandwidth is $f_{-3dB}\frac{\pi}{2}$. From equations 3.3 & 3.4, the mean-square noise voltage generated on the capacitor can be written as:

$$V_{no}^2 = (f_{-3dB}\frac{\pi}{2})(4\text{KTR}) \tag{3.5}$$

$$\because\ f_{-3dB} = \frac{1}{2\pi RC} \tag{3.6}$$

$$\therefore\ V_{no}^2 = \frac{KT}{C} \tag{3.7}$$

From the above analysis, it is clear that increasing the size of the sampling capacitor reduces the power of thermal noise. As thermal noise represents a dynamic noise source that reduces ADC SNR, a minimum capacitance (i.e.$C_s$, $C_f$) must be driven to ensure a sufficient accuracy – thus thermal noise imposes a tradeoff between power and accuracy. The total output noise of the 1.5-bit stage pipeline ADC can be calculated by analyzing the operation of the system into two phases as following:

**During the sampling phase $\phi_1$:**

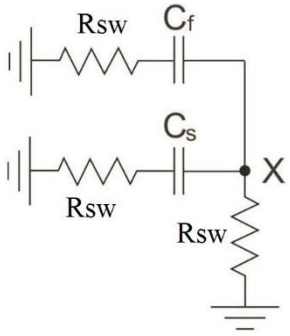


Fig. 3.10 Sampling phase equivalent circuit.

The thermal noise of the on-resistor of the sampling switches accounts for all KTC noise during the sampling phase.

During this phase, the OTA is not connected to the input and its output is shorted to ground, so it doesn't contribute to the total output noise during the sampling phase $\phi_1$.

Replacing each switch with its ON resistance during $\phi_1$, we can redraw the whole first stage pipeline ADC as it is show in figure 3.10.

Using equation 3.7 & the voltage-charge relationship of a capacitor in 3.8,

$$Q = C.V \tag{3.8}$$

then we can write the mean-square of the total noise charges accumulated at node X in figure 3.10 as:

$$\overline{Q_x^2} = KT\,(C_f + C_s) \tag{3.9}$$

Equation 3.9 didn't account for the parasitic capacitance at node X, which will increase the sampled noise charge when taken in account. The contribution of the sampled noise charge at node X during $\phi_1$ to the total mean-square output voltage noise can be written using equation 3.8 and equation 3.9 as:

$$\overline{V_{o,1}^2} = \frac{\overline{Q_x^2}}{C_f^2} \tag{3.10}$$

Where $C_f$ is the capacitor that the sampled charge during the sampling phase will transfer to during charge transfer phase $\phi_2$.

Substituting equation 3.9 in 3.10 results in:

$$V_{o,1}^2 = KT\frac{(C_f + C_s)}{C_f^2} = \frac{KT}{C_f}\left(1 + \frac{C_s}{C_f}\right) \qquad (3.11)$$

**During the charge transfer phase $\phi_2$:**

The equivalent circuit of the 1.5-bit stage pipeline ADC during the charge transfer phase, $\phi_2$, can be drawn as in figure 3.11. Where a capacitance load has been added at the output node, the ON switches during $\phi_2$ had been replaced with their equivalent resistors Rsw, and the noisy OTA has been replaced with a noiseless OP-AMP plus its equivalent current source noise model.

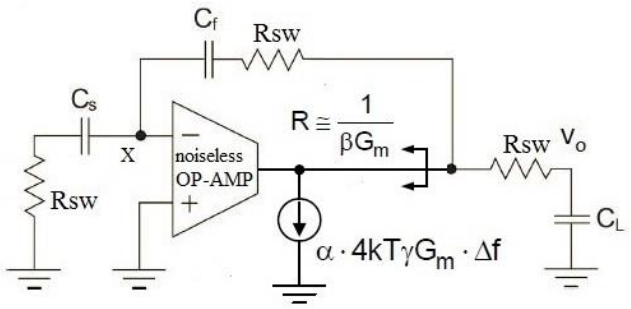

Fig. 3.11 charge transfer phase

In a proper design, the noise due to $R_{sw}$ will be much smaller than the noise from the OTA. We'll consider the noise during charge transfer is solely due to the OTA. The output resistance of the OTA in figure 3.11 can be written as:

$$R \cong \frac{1}{\beta\, g_m} \qquad (3.12)$$

Where $\beta$ is a return factor and equal to:

$$\beta = \frac{C_f}{C_f + C_s + C_x} \qquad (3.13)$$

Where $C_x$ is the parasitic capacitance at node X.

Equation 3.12 of the output resistance of the OTA can be derived by considering the analysis of the inverting amplifier shown in fig. 3.12, where its output resistance $R_o$ can be written as in equation 3.14.

$$R_o = \frac{R_f + R_{in}}{1 + g_m R_{in}} \tag{3.14}$$

For $g_m R_1 >> 1$, the expression in 3.14 can simplified as:

$$R_o \cong \frac{R_f + R_{in}}{g_m R_{in}} \tag{3.15}$$

Replacing $R_f$ & $R_{in}$ in fig. 3.12 with $C_f$ & $(C_s + C_x)$ respectively, then the expression in 3.12 can be obtained.

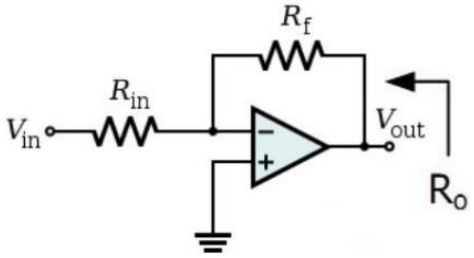


Fig.3.12 Inverting Amplifier

The total output capacitance at the output node is the equivalent of the load capacitance, $C_L$, in parallel with the capacitance looking back into the output node, $C_{look-back}$.

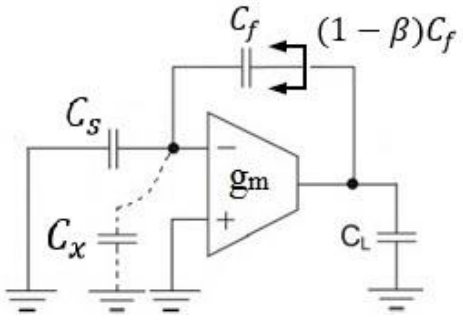


Fig. 3.13 Output node capacitance

As it is shown in fig. 3.13, $C_{look-back}$ can be written as:

$$\frac{1}{C_{look-back}} = \frac{1}{C_s + C_x} + \frac{1}{C_f} \tag{3.16}$$

Or,

$$C_{look-back} = \frac{(C_s + C_x).C_f}{C_s + C_x + C_f} \tag{3.17}$$

Using 3.13 in 3.16, $C_{look-back}$ can be simplified to:

$$C_{look-back} = (1 - \beta)\, C_f \tag{3.18}$$

The total output capacitance at the output node,$C_{Ltotal}$ is the parallel combination of the load capacitance $C_L$ and $C_{look-back}$.

$$C_{Ltotal} = C_L + C_{look-back} \tag{3.19}$$

Or,

$$C_{Ltotal} = C_L + (1 - \beta)\, C_f \tag{3.20}$$

By now, we can write the total impedance at the output node, $Z_o$, as the parallel combination of $R_o$ and $C_{Ltotal}$. The final step in calculating the mean-square voltage accumulated at the output node is to use the given current source noise model for the OTA, $i_{no}$, and $Z_o$to write and expression for the mean-square of the output voltage noise where:

$$\overline{i_{no}^2} = \alpha.KT.\gamma.\frac{1}{\beta R_o}.\Delta f \tag{3.21}$$

The mean-square of the output voltage can be written as:

$$\overline{V_{o,2}^2} = \alpha.KT.\gamma.\frac{1}{\beta R_o}.\Delta f.\left|R_o \,||\, \frac{1}{j\omega C_{Ltot}}\right|^2 \tag{3.22}$$

$$\overline{V_{o,2}^2} = \int_0^{\infty} \alpha . KT . \gamma . \frac{1}{\beta R_o} . \Delta f . \left| \frac{R_o}{1 + j\omega C_{Ltot}} \right|^2 df \qquad (3.23)$$

$$\overline{V_{o,2}^2} = \alpha\gamma \frac{1}{\beta} \frac{KT}{C_{Ltot}} \qquad (3.24)$$

**Adding up noise contributions:**

The previous noise analysis during sampling phase and charge transfer phase, $\phi_1$ & $\phi_2$ respectively is summarized in the figure 3.14

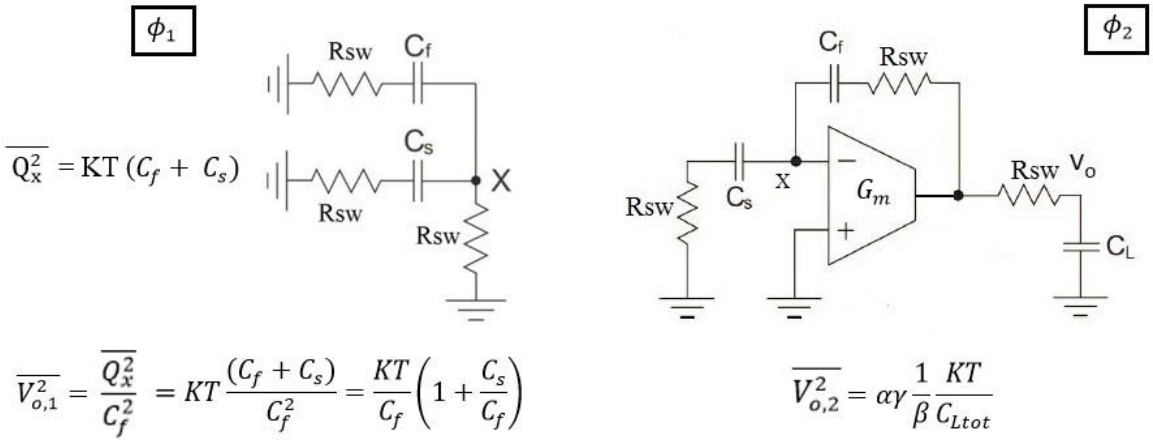


Fig. 3.14 Noise contribution from sampling and charge transfer phases

The total added up mean-square noise voltage can be written as:

$$\overline{V_{o,total}^2} = \frac{KT}{C_f}\left(1 + \frac{C_s}{C_f}\right) + \alpha\gamma \frac{1}{\beta} \frac{KT}{C_{Ltot}} \qquad (3.25)$$

Given the upper limit of the allowed input referred noise of the fully differential version of the 1.5-bit pipeline ADC as 200 μV rms, then we can write the maximum allowed value of the output noise voltage if the MDAC multiplier factor is determined. For our design, the MDAC has a gain of two, which can be achieved by making the ratio between the capacitors $C_s$ and $C_f$ equal to one.

That should comply with our understanding of the relation between the charge domain and the voltage domain. When $C_s$ and $C_f$ are equal, this means that we sample the input voltage using a capacitor that is twice the size of the capacitor that is used during the charge transfer (the flip over technique that is used in this design allows us to use one of the two sampling capacitors during the charge transfer phase). So in order to maintain the charge conservation law, the output voltage after the charge transfer phase is complete has to be twice the sampled input voltage.

Now we can easily set the limit on the maximum allowed value of the output noise voltage as:

$$\mathrm{V_{o,total,rms}} < 400\ \mu V \tag{3.26}$$

Where the total mean-square output noise voltage of the fully differential design can be written as:

$$\overline{\mathrm{V_o^2}} = \frac{2\,\mathrm{K\,T}}{\mathrm{C_f}}\left(1+\frac{\mathrm{C_s}}{\mathrm{C_f}}\right) + \alpha\,\gamma\,\frac{1}{\beta}\,\frac{\mathrm{K\,T}}{\mathrm{C_{Ltot}}} \tag{3.27}$$

Equation 3.27 is very similar to equation 3.25 except for equation 3.27 accounts for the difference in the sampled noise in the case of using fully differential design.

Solving equation 3.27 for proper value of $C_s$ and $C_f$ that satisfy inquality 3.26, assuming the following:

1- The transient frequency ($f_T$) = 10 GHz

2- $C_L = C_f = C_s = C$

3- Gm = 1mS ( as initial guess)

4- The product of $\gamma$ and $\alpha$ equals to 4/3.

Using the above assumptions, we can re-write equations 3.13 & 3.20 as:

$$\frac{1}{\beta} = \left(2 + \frac{C_x}{C}\right) \tag{3.28}$$

$$C_{L,total} = \frac{3C^2 + 2\,C\,C_x}{2C + C_x} \tag{3.29}$$

Substituting equations 3.28 and 3.29 in equation 3.27, then we can write the below inequality to determine the value of $C$ that satisfies the input referred noise condition:

$$400\ \mu V > \sqrt{\frac{4KT}{C} + \frac{4}{3}\left(2 + \frac{C_x}{C}\right).\frac{KT\left(2\,C + C_x\right)}{3C^2 + 2C\,C_x}} \tag{3.30}$$

Where,

$$C_x = \frac{G_m}{2\pi f_T} = \frac{1m}{2\pi\ 10G} = 16\ fF \tag{3.31}$$

The minimum value for *C* that satisfy inequality 3.30 is:

$$C = 147.7\ fF \tag{3.32}$$

#### 3.2.2.2 OTA gain analysis

Building an OTA with a reasonable gain is one the most important aspects of our design. For the ideal op-amp, where all error sources are ignored and the open-loop gain (A) is infinite, the output voltage residue in figure 1.2 can be written as:

$$V_{res} = 2\ V_{in} - V_{dac} \tag{3.33}$$

Where $V_{res}$ is the 1.5-bit stage output, and $V_{dac}$ is the output of the DAC block. In reality, where the gain of any operation amplifier is finite, the equality in 3.33 won't hold true, and thus the need to know the effect of finite gain on the precision output voltage has come up.

In our coming analysis to derive an expression for $V_{res}$ as a function of the finite gain of the OTA, we will employ the charge conservation principal at certain nodes of interest. The reader can use the same steps in the following analysis to derive the expression in equation 3.33, but I preferred to go through it once in the case of finite OTA gain.

As we did in the previous section to determine suitable values for the capacitors' sizes by breaking up the operation of the circuit into two different phases, this way still works for determining the value of the OTA gain.

By looking at the circuit during the sampling operation, where the OTA input and output terminals are connected to ground, the circuit can be simplified and re-drawn as in figure 3.15.

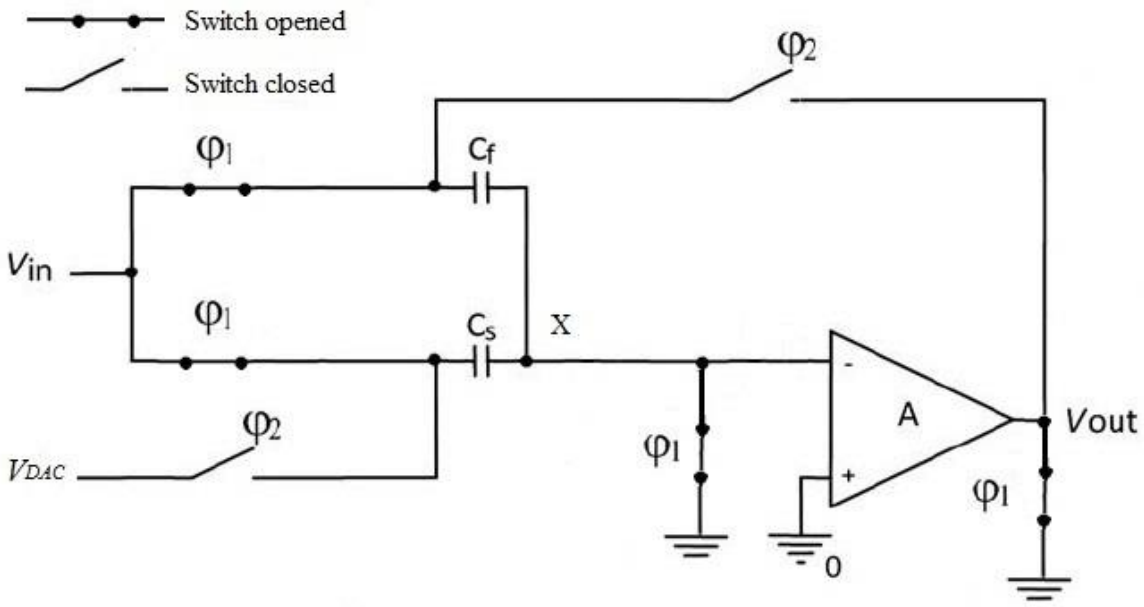


Fig. 3.15 1.5-bit MDAC sampling operation (single-ended representation)

The charge at node X after sampling the input single and before the starting of the charge transfer phase:

$$Q_1 = -V_{in}\,(C_f + C_s) \tag{3.34}$$

By the end of sampling phase, clock $\phi_1$ goes low and the input signal disconnect from the sampling capacitors. In the same time the input and output terminal of the OTA are not connected to ground, and the 1.5-bit stage is ready to perform the multiplication-subtraction task. Once clock $\phi_2$, goes high the out of the DAC gets connected and one of the sampling capacitors gets flipped over to transfer the charge and the amplification happens. During the amplification phase (charge transfer phase), the 1.5-bit MDAC stage can be drawn as in figure 3.16

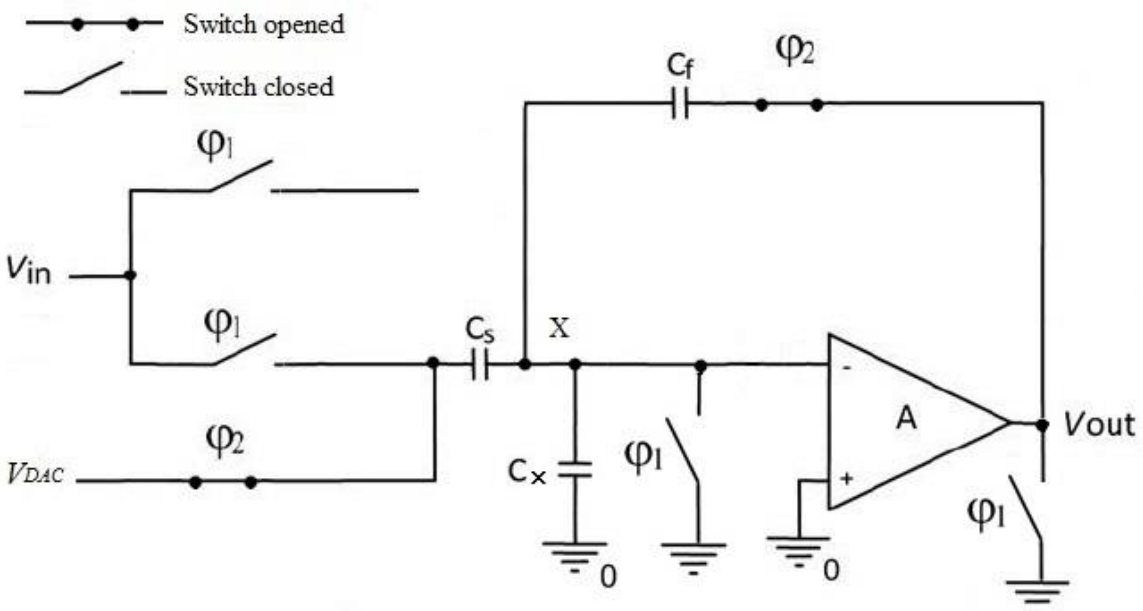


Fig. 3.16 1.5-bit MDAC amplification operation

The charge sampled at node X during clock $\phi_1$ is high and $\phi_2$ is low will remain the same during $\phi_2$ is high and $\phi_1$ is low. Because in the actual implantation of the MDAC circuit, the input signal get disconnected from the circuit after the switch connected to the inverted input terminal of the OTA goes off first. This way, we prevented any change in the amount of the sampled charge. And what actually happens to the sampled charge during $\phi_2$ is charge re-distribution, which causes change in the voltages across different nodes in the circuit.

The charge at node X after the charge transfer phase is complete can be written as:

$$Q_2 = -\frac{V_{res}}{A} C_x + \left(\frac{-V_{res}}{A} - V_{DAC}\right) C_s + \left(\frac{-V_{res}}{A} - V_{res}\right) C_f \qquad (3.35)$$

$Q_1$ and $Q_2$ in equations 3.34 and 3.25 are equal due to conservation of charge principals, which will allow us to write the equality in 3.36

$$-V_{in}\,(C_f + C_s) = -\frac{V_{res}}{A}\,C_x + \left(\frac{-V_{res}}{A} - V_{DAC}\right)C_s + \left(\frac{-V_{res}}{A} - V_{res}\right)C_f \quad (3.36)$$

And as we chose for our capacitors' design in the previous section to have equal sizes for the sampling capacitors and the flip-over capacitor, *C*, we can write the residue output voltage of the 1.5-bit stage as function in op-amp finite gain, A, as the following :

$$V_{res} = \frac{2\,C\,V_{in}}{\frac{C_x}{A} + C(1 + \frac{2}{A})} - \frac{C\,V_{DAC}}{\frac{C_x}{A} + C\left(1 + \frac{2}{A}\right)} \quad (3.37)$$

Now we can see the effect of the finite gain on the residue output voltage of the pipeline ADC stage. As the finite gain, A, in equation 3.37 goes higher, the closer the $V_{res}$ expression approaches its ideal value in equation 3.33. And in ideal case of infinite gain, equation 3.37 and equation 3.33 are exactly the same. Given the upper limit on the total harmonic distortion (THD) in the design target in section 3.2.1 sets the minimum requirements on the OTA gain. Since designing an OTA with not high enough gain will cause the residue output voltage to deviate by significant values from its acceptable limits.

We'll use Fast Fourier Transform (FFT) analysis to determine the proper value of the OTA gain that should satisfy the design requirements.

Equation 3.33 tells us if we use an ideal OTA, and we want to reconstruct the input signal,$V_{in}$, from the residue output voltage, $V_{res}$, and the DAC block's

output, $V_{DAC}$, we can write a reconstructed expression for the input signal as the following:

$$V_{in,reconstructed} = \frac{V_{res} + V_{DAC}}{2} \tag{3.38}$$

Substituting 3.37 in 3.38 results in:

$$V_{in,reconstructed} = \frac{C\,V_{in} + V_{DAC}\left(\frac{C_x}{A} + \frac{2C}{A}\right)}{\frac{C_x}{A} + C\left(1 + \frac{2}{A}\right)} \tag{3.39}$$

By examining equation 3.39, it agrees with our analysis that as the OTA gain approaches infinite, the constructed input signal will be exactly equal to the original input signal.

In the coming section we'll calculate and draw the THD of the constructed input signal in equation 3.39 using a Matlab script (see Appendix A) assuming that the original input signal is a pure sinusoidal wave.

The script is simulating the operation of 1.5-bit stage pipeline ADC with a sampling frequency of 100MHz and input signal frequency equal to 48.44MHz. The rest of the voltages values in the script will be discussed in details in the section demonstrating the building of the 2-bit flash ADC and the DAC blocks. The idea of the script is calculating 64 point Fast Fourier Transform (FFT) of the constructed signal for different values of the finite gain of the OTA starting with a gain of a hundred and ending with a gain of four thousands with incremental step of a hundred. Once the FFT result for each iteration is generated, the THD is estimated and recorded. Finally it plots all these recorded values of the THD's against their equivalent gain values.

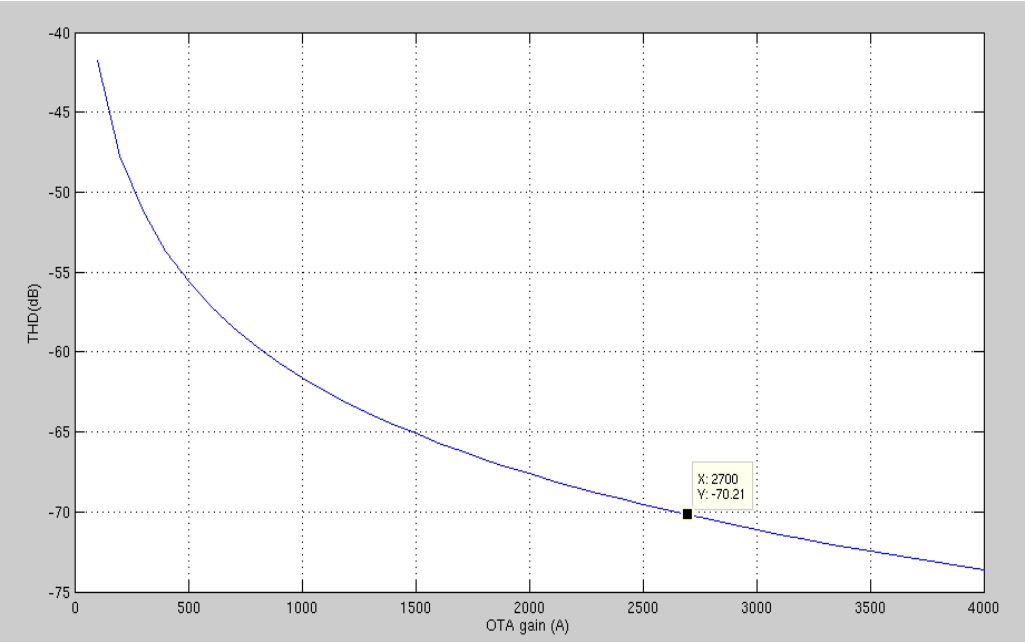


Fig. 3.17 THD vs. finite OTA gain (A)

The plot in figure 3.17 shows the lower limit on OTA gain, 2700, in order to achieve a THD less than -70dB. This plot was generated for a value of sampling capacitor equal to 200fF instead of 147.7fF as calculated in section 3.2.2.1. The reason of increasing the capacitor size is to relax the requirements on the OTA gain as larger capacitor would help in suppress the noise and hence improve the THD.

### 3.2.2.3 DAC design

For the digital-to-analog (DAC) block used in the implementation of the 1.5-bit stage pipeline ADC, we are using a resistor ladder DAC topology. The idea of operation of this kind of DACs is simple, where the resistors connected between two reference voltages is a resistor string ladder network. The resistors act as voltage dividers between the referenced voltages. Each tap of the string

generates a different voltage that can be produced on the output of the DAC if all the switched between a specific tap and the DAC's output are turned ON. The schematic of the used resistor ladder DAC is shown in figure 3.18, where the tap voltage nodes are shown on the resistors string, and the binary switches controlling voltages are B0 and B1, and their complements B0b and B1b allowing no more one tap voltage to the output node at the same time.

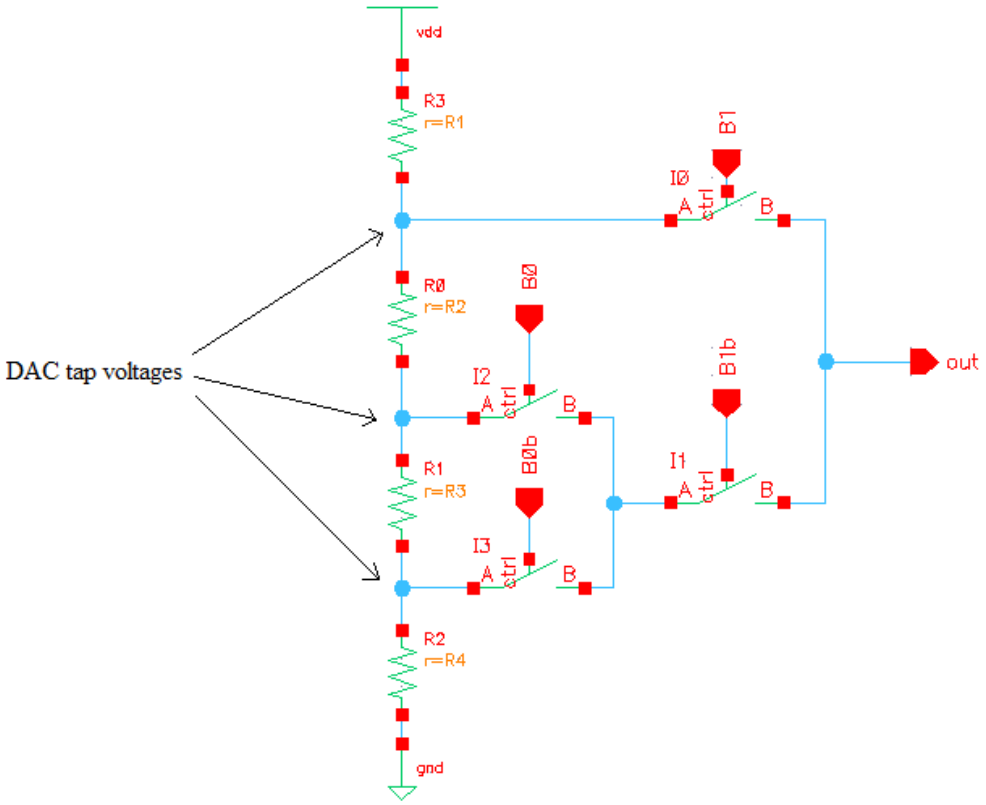


Fig. 3.18 DAC implementation

Since the binary voltages that control the operation of the switches used in the DAC design are produced by the flash ADC block, they could have any of the following possibilities: 00, 01, and 11. Also, since the input signal ranges from 0.4V to 1.4V, it would be wise to choose the DAC tap voltage as 0.4V, 0.9V

and 1.4V. The values of the resistors that achieve these tap voltage values can be calculated for VDD =1.8V as in the following table:

| Resistor Name | Value in Ohm(s) |
|---|---|
| $R_0$ | 500 |
| $R_1$ | 500 |
| $R_2$ | 400 |
| $R_3$ | 400 |

Table 3.11 DAC string resistors values

**3.2.2.4 Flash ADC Design:**

The selection of the reference voltages for the flash ADC can be determined by studying the residue output voltage plot in fig. 1.2. The output of the flash ADC makes transitions when the input voltages crosses $\frac{-V_{ref}}{4}$ and $\frac{+V_{ref}}{4}$.

Since the value of the measured analog voltage of any node is bounded by a lower limit of zero voltage (ground) and maximum limit of the voltage supply (1.8V), we can map the range from $-V_{ref} \rightarrow +V_{ref}$ to 0V → 1.8V.
That yields to the flash ADC's transitions should happens at $\frac{3\,VDD}{8}$ and $\frac{5\,VDD}{8}$; i.e. 0.675V and 1.125V. The schematic in figure 3.19 is showing a basic architecture of the implemented flash ADC in our design. The values of the resistors R0, R1, and R2, that achieves $V_{ref0}$ equals to 0.675V and $V_{ref1}$ equals to 1.125V, are given in table 3.12. The symbol of the flash ADC block is shown in fig. 3.20 and the pins description is given in table 3.13.

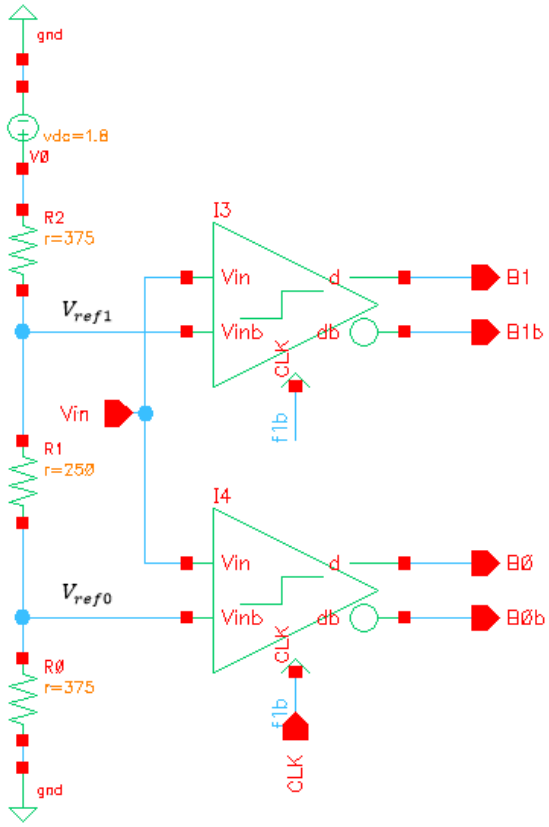


Fig. 3.19 Flash ADC schematic

| Resistor Name | Value in Ohm(s) |
|---|---|
| $R_0$ | 375 |
| $R_1$ | 250 |
| $R_2$ | 375 |

Table 3.12 Flash ADC string resistors values


Fig. 3.20 Flash ADC symbol

| Name | Pin Type | Description |
|---|---|---|
| CLK | Input | Input clock signal |
| Vin | Input | Input voltage signal |
| B0 | Output | LSB of two outputs bits |
| B0b | Output | Complement output of LSB bit |
| B1 | Output | MSB of two output bits |
| B1b | Output | Complement output of MSB bit |

Table 3.13 Flash ADC pins

### 3.3 Simulation of 1.5-Bit Stage Using Ideal Components

After obtaining the parameter of the ideal OTA, capacitors' sizes and other reference voltages for the comparator and the DAC blocks, we'll put together all these blocks to build a single ended version of the 1.5-bit stage pipeline ADC and verify the residue output voltage for a ramp input signal. The next step is to simulate the fully differential design with a sinusoidal input signal and verify that it is meeting the design requirements.

### 3.3.1 TRANSIENT SIMULATION OF SINGLE ENDED CIRCUIT

The schematic of the single ended stage pipeline is shown in fig. 3.21. The OTA parameters' values used in the simulation are given in table 3.14.

| Parameter | Value |
|---|---|
| Gain | 2700 |
| $f_T$ | 10GHz |
| $g_m / I_d$ | 10 |
| $g_m$ | 1mS |

Table 3.14 OTA Ideal Model Parameters' values

The sampling frequency and other attributes of the clocking mechanism is set by clock generator block's parameters in table 3.15.

| Parameter | Value |
|---|---|
| Period | 1/(100M) |
| t_early | 200p |
| t_edge | 60p |
| t_nonoverlap | 200p |

Table 3.15 Ideal Clock Generator Parameters

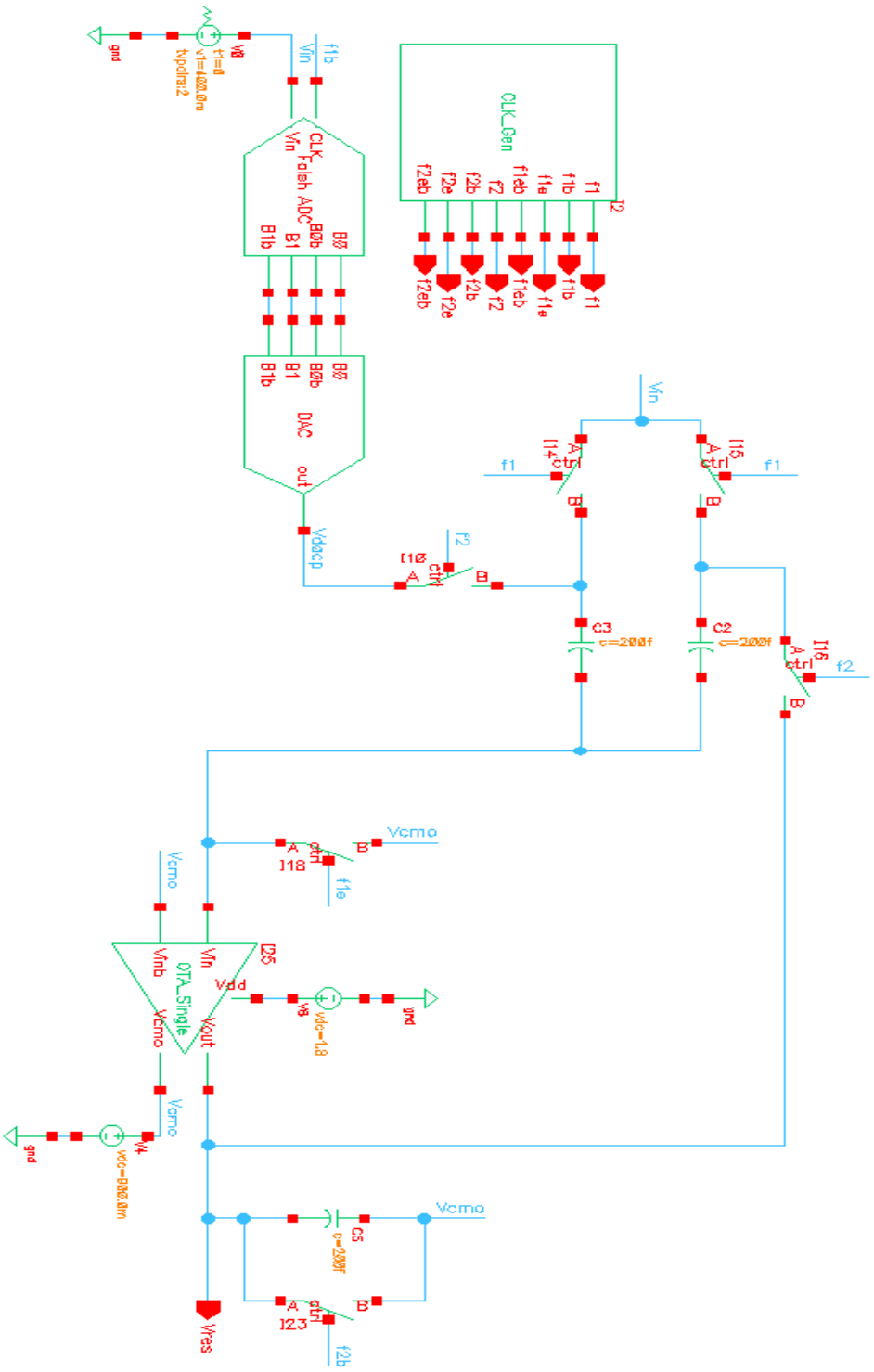


Fig. 3.21 Single-ended 1.5-bit pipelined ADC

The residue output voltage as a response of a ramp input function rise from 0.4V to 1.4V in a duration of 1µs is shown in figure 3.22. We can see from the residue plot that the transitioning is happening when the input ramp signal crosses 0.675V and 1.125V.

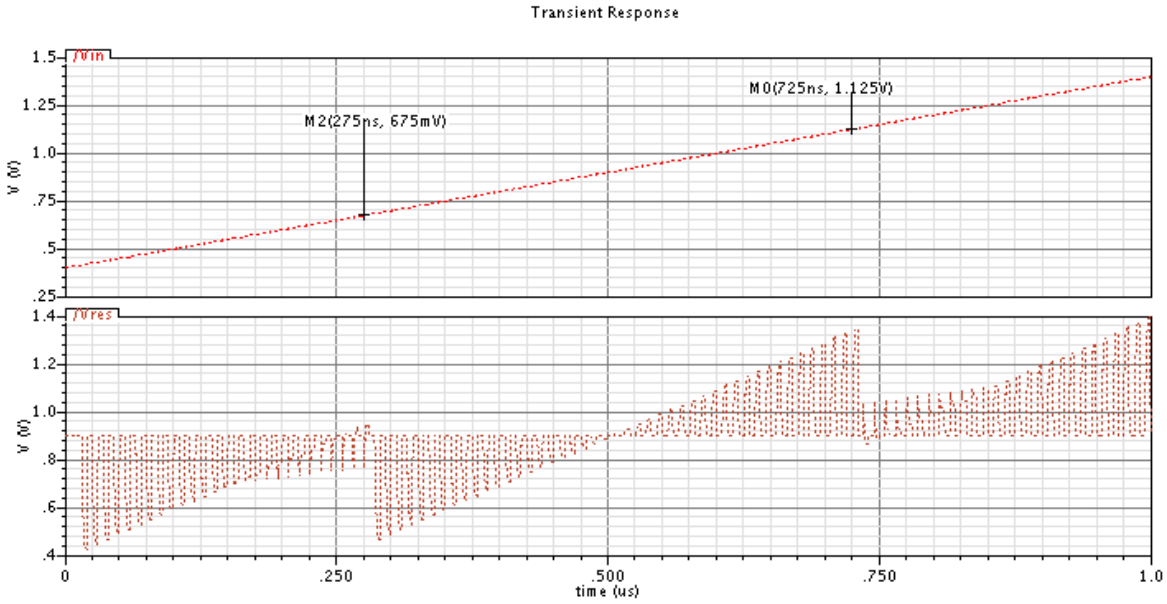


Fig. 3.22 Input Ramp Signal and its Residue Output Voltage Response

### 3.3.2 TRANSIENT SIMULATION OF THE FULLY DIFFERENTIAL CIRCUIT

The schematic of the fully differential stage pipeline is shown in fig. 3.23, where the OTA and clock generator blocks' parameters are having the same values as in tables 3.15 and 3.16 respectively. The input a sinusoidal signal of amplitude 1V and frequency of (1/64) × 100MHz. The differential output voltage, which is equivalent to $V_{res} - V_{resb}$, is showing in figure 3.24, and a close view of the output is showing in figure 3.25. We can see that the complete settling of the output is happening each cycle during the charge transfer phase.

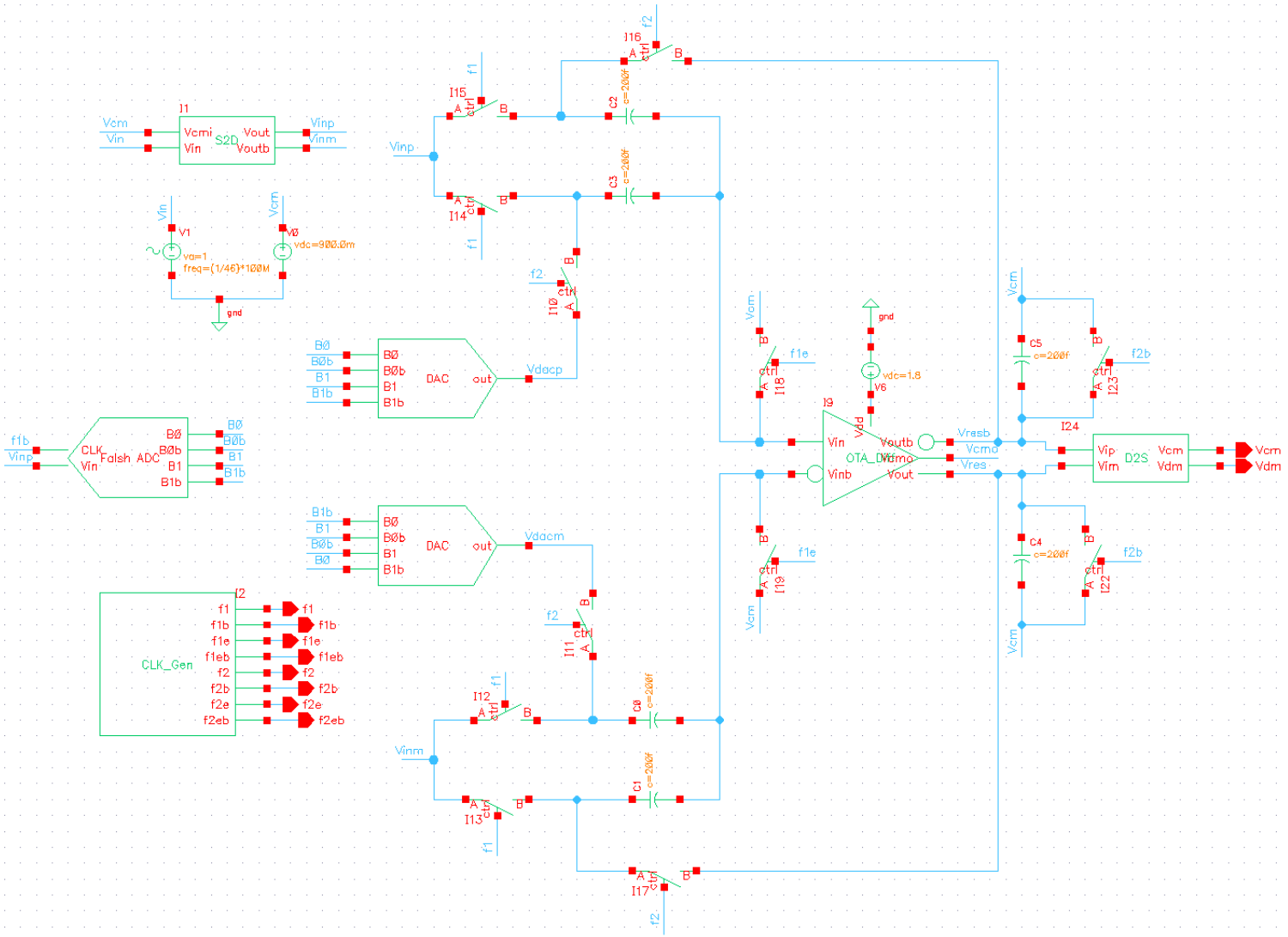


Fig. 3.23 Fully Differential 1.5-bit stage pipeline ADC

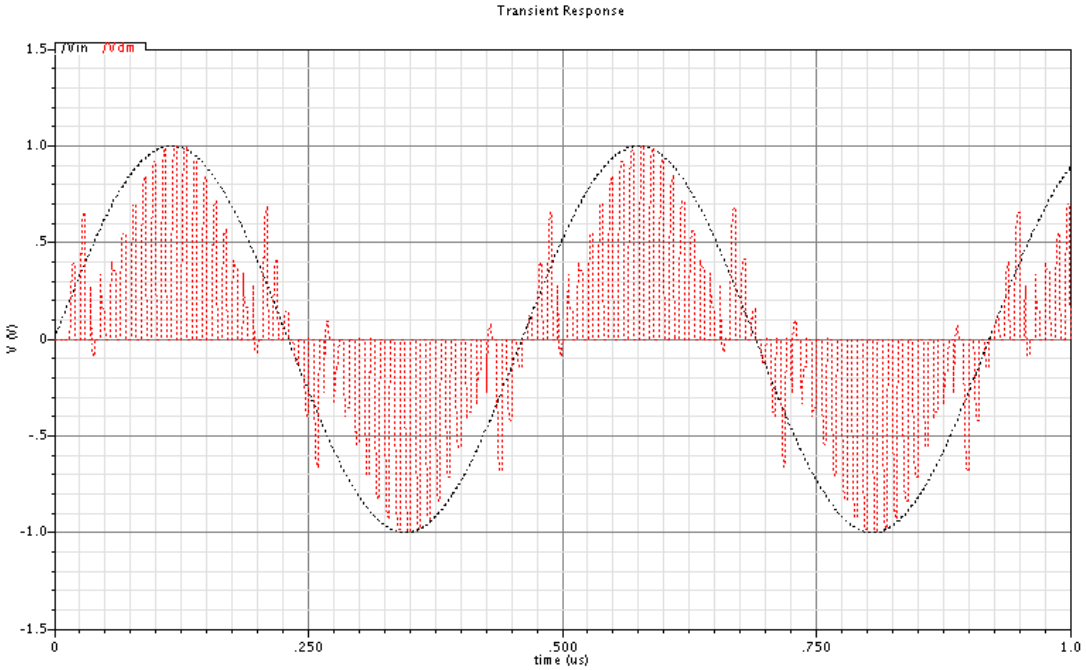


Fig. 3.24 Differential Output Voltage of the Fully Differential 1.5-bit stage

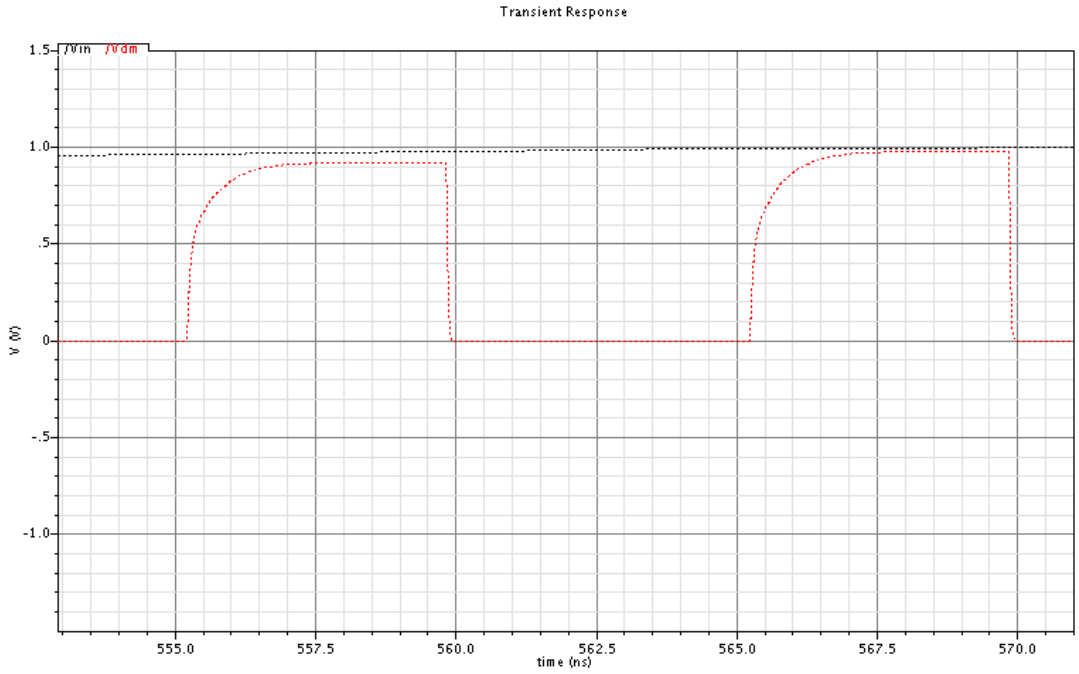


Fig. 3.25 A close view of the output to show a complete settling is happening

The time constant (given the design parameters we obtained from the previous analysis) can also be calculated using the equation in 3.40.

$$\tau = \frac{1}{\beta}\frac{C_{Ltot}}{G_m} \tag{3.40}$$

Where, $C_{L,total} = 304$ fF and $1/\beta = 2.07958$, using equations 3.13 and 3.20.

For $G_m$ equals 1mS, $\tau \approx 0.7$ ns.

The dynamic settling error, $\varepsilon_{dynamic}(t)$ can be calculated as:

$$\varepsilon_{dynamic}(t) = \frac{V_o(t) - V_{o,final}}{V_{o,final}} = \frac{V_{o,final}\left(1 - e^{-\frac{t_s}{\tau}}\right) - V_{o,final}}{V_{o,final}} \tag{3.41}$$

$$\varepsilon_{dynamic}(t) = e^{-\frac{t_s}{\tau}} \tag{3.42}$$

Where $V_{o,final}$ is the value of $V_o(t)$ when t → ∞.

It is clear form equation 3.42 that the dynamic settling error, $\varepsilon_{dynamic}(t)$ is inversely proportional with time, i.e. the more time we give for the output voltage signal before sampling it, the more we grantee that the dynamic settling error is sufficiently small.

Sine this time is certainly bounded by the half cycle time (5ns), so if we chose the sampling time, $t_s$ to be ≈ 4.8 ns as it is shown in figure 3.33, and $\tau$ = 0.7 ns, the dynamic settling error ($\varepsilon_{dynamic}$) would be in the range of 0.1%.

### 3.3.3 SAMPLING PHASE NOISE SIMULATION (.noise)

The noise during the sampling phase can be calculated by simulating the voltage noise accumulated on all the capacitors. To do so, we can use voltage controlled voltage source (VCVS) to add up the different noise components. The circuit schematic for the simulated circuit is shown in fig. 3.26 and the .noise simulation result is showing in fig. 3.27. The total integrated noise can be calculated using the rmsNoise function in Cadence which gives a value of 290.4 μV rms.

The charge transfer phase noise can be calculated by doing same kind of simulation analysis, but rather sensing the voltage difference between the two terminals of the OTA. The schematic for the simulated OTA along with the flip-over capacitors and the turn-on switch resistors is showing in fig. 3.28. The simulation result of the noise at the output nodes during the second phase is showing in fig. 3.29. The total integrated noise during the charge transfer phase is 267.2 μV rms.

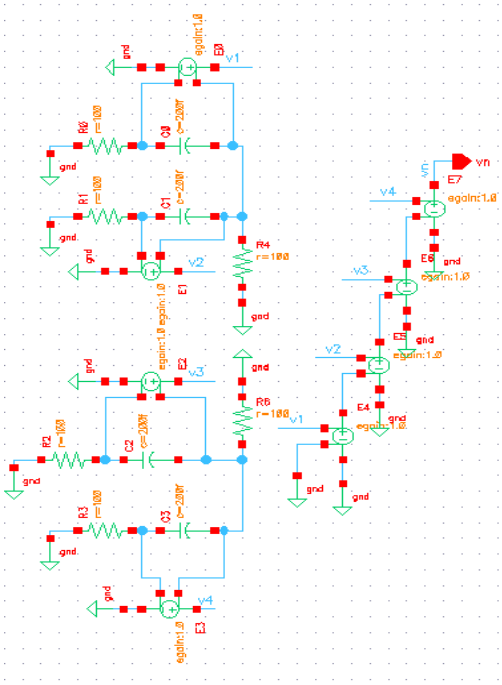

Fig. 3.26 Circuit Schematic of the AC noise during sampling phase

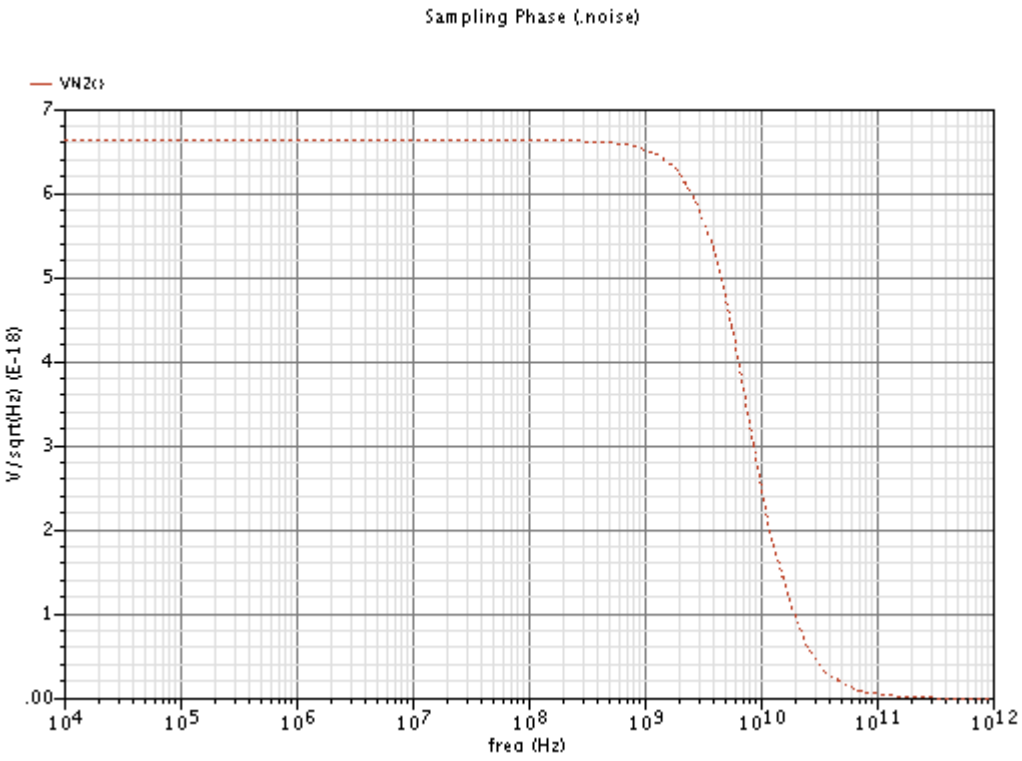


Fig. 3.27 AC Noise during Sampling Phase of Differential ADC

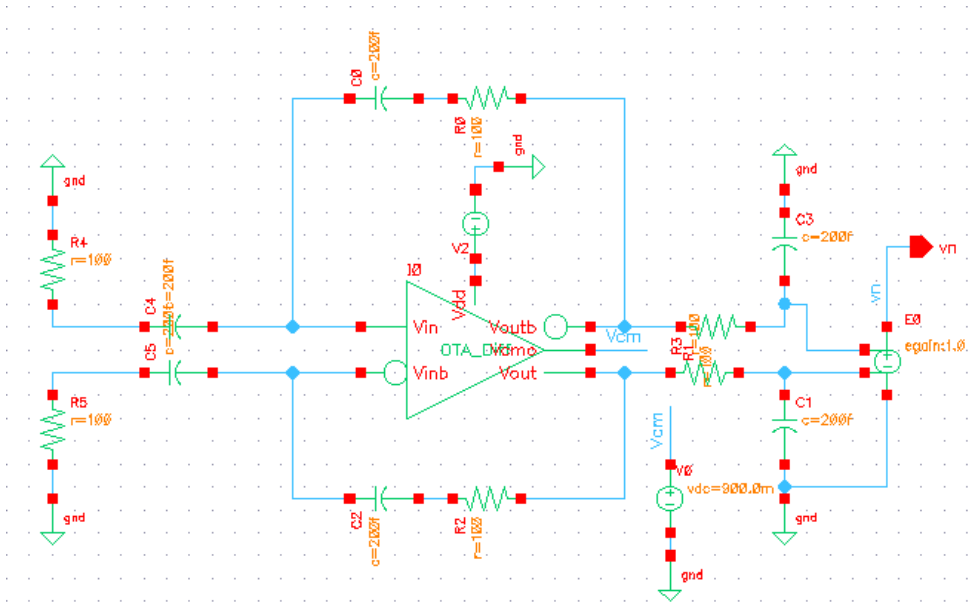


Fig. 3.28 Circuit Schematic of the AC noise during charge transfer phase

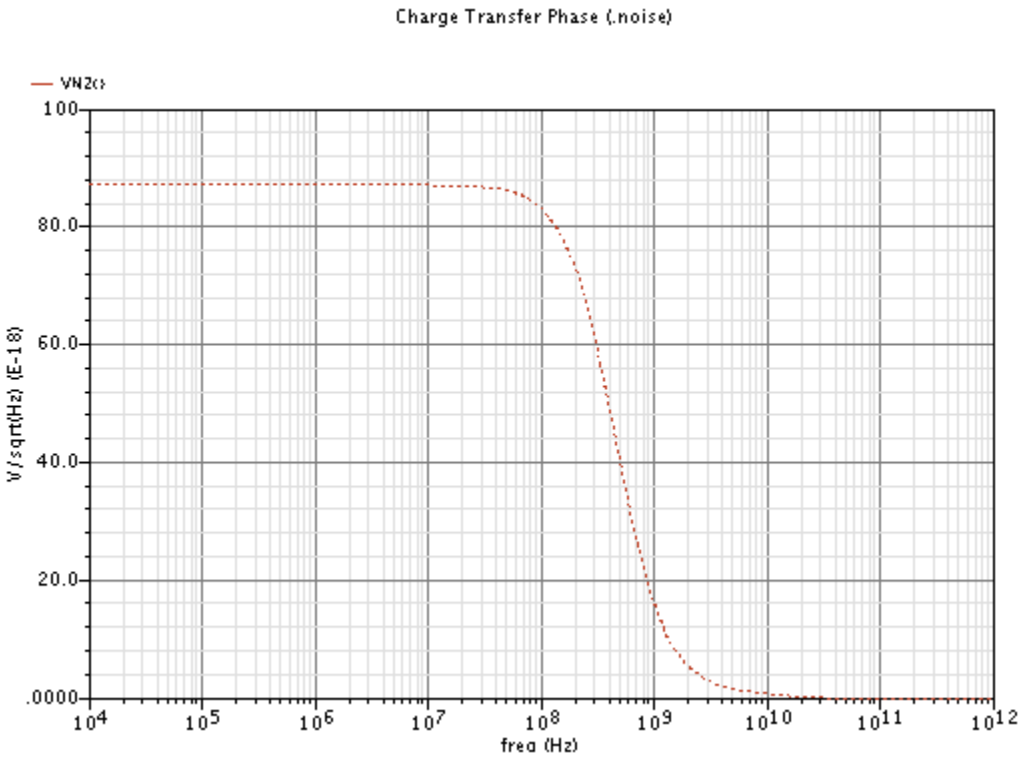


Fig. 3.29 AC Noise during Charge Transfer Phase of Differential ADC

### 3.3.4 PERIODIC NOISE ANALYSIS

The Periodic Noise analysis (Pnoise) is similar to the conventional noise analysis except that it models frequency conversion effects. Hence Pnoise analysis is useful for predicting the noise behavior of mixers, switched-capacitor filters and other periodically driven circuits. First, the circuit is linearized using the PSS operating point. The periodically time-varying qualities of the linearized circuit create the frequency conversion. Then the Pnoise analysis simulates the effect of a periodically time-varying bias point on component-generated noise.

Initially, PSS computes the response to a large periodic signal such as a clock. The subsequent Pnoise analysis computes the resulting noise performance. PNOISE analysis computes the total noise at the output, which includes contributions from the input source and the output load. The amount of the output noise that is attributable to each noise source in the circuit is also computed and output individually. The analysis includes the effects of noise folding, aliasing and frequency conversion effects. It is useful in computing input-referred noise, noise figure, the phase noise of oscillators, and the noise behavior of mixers, switched-capacitor filters and similar circuits. One example of the aliasing might be the noise at the image frequency of a mixer getting into the IF along with that of the desired mixing product. This is why in a high quality, high price, multiple conversion receiver the mixers after an IF amplifier chain is preceded by another band pass filter to keep the noise from the IF amplifiers at the mixer's image frequency out of the next stage of the signal path. The results of the pnoise analysis is showing in figure 3.30, where the OTA parameter used in the simulation is as given is table 3.14.

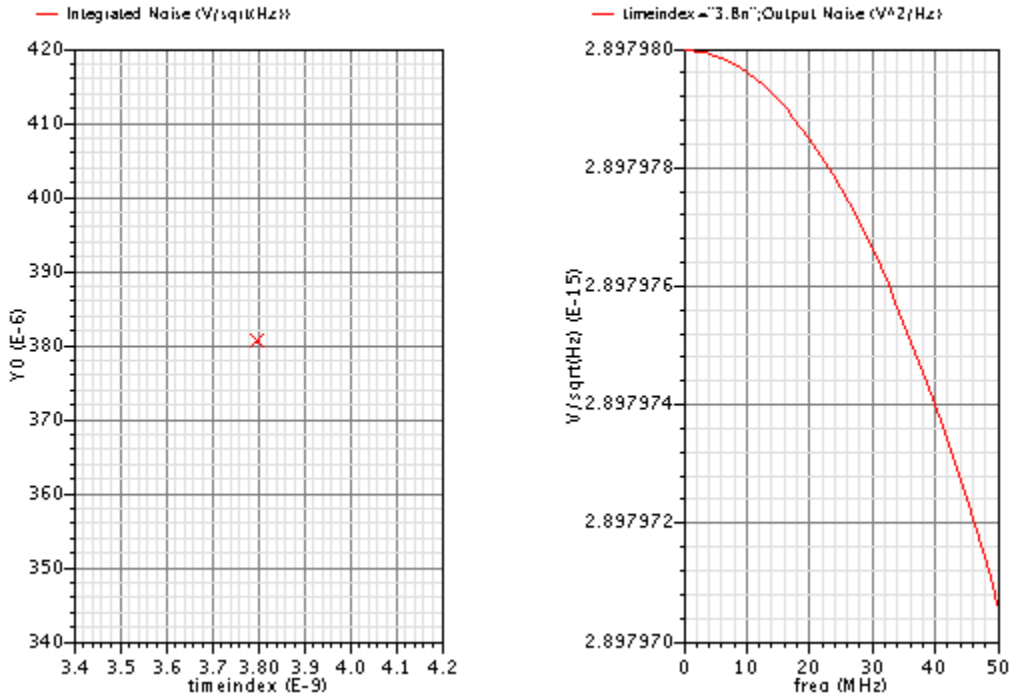


Fig. 3.30 Differential Pnoise Simulation Results

### 3.3.5 TRANSIENT NOISE ANALYSIS

This kind of transient analysis gives us the opportunity to examine the impact of noise in the time domain on various circuit types without requiring access to the SpectreRF analysis. One critical parameter to setup is “noisefmax” value, which can be translated to the bandwidth of pseudorandom noise sources. The larger value of “noisefmax”, the accurate results, but the longer simulation time. The result of the transient noise analysis is showing in figure 3.31, where noisefmax is set to 20GHz in the simulation setup, and the resulted output voltage noise is 387 μV rms.

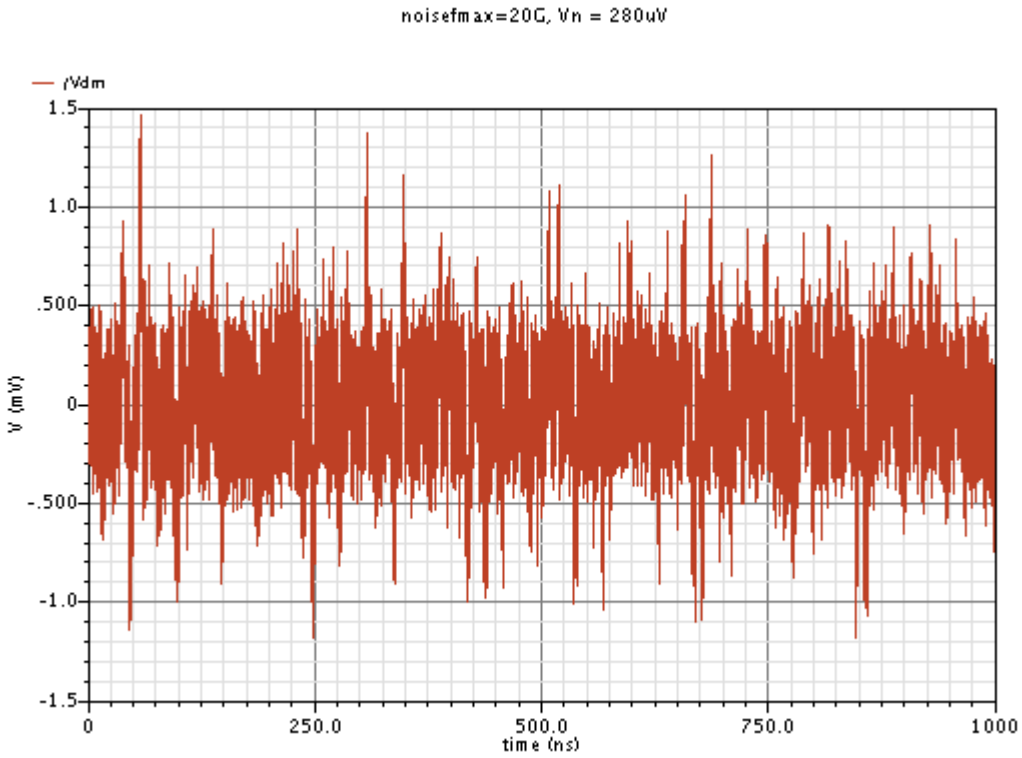


Fig. 3.31 Transient Noise Simulation Results

### 3.3.6 NOISE ANALYSIS COMPARISON

The following table summarizes the results obtained by running different kind of noise analysis.

| Method | $\phi_1$ [µV rms] | $\phi_2$ [µV rms] | Total [µV rms] |
|---|---|---|---|
| Hand Calculation | 288 | 234 | 371 |
| .noise | 290.4 | 267.2 | 395 |
| PSS/Pnoise | NA | NA | 381 |
| Transient | NA | NA | 387 |

Table 3.16 Noise Analysis Results Summary

### 3.3.7 THD MEASUREMENTS

The total harmonic distortion, or THD, is measured for the constructed input signal which is calculated from the MDAC and DAC differential outputs as shown in equation 3.43, which is similar to equation 3.38 in the case of single ended circuit. And in order to get accurate results of the THD, the frequency of the input signal used is set to (31/64)×100MHz. The reason for that is to avoid taking the sampling points at the same values of the signal of interest every cycle. Figure 3.32 is showing the constructed input signal from the MDAC output signals ($V_{res}$ and $V_{resb}$), and the DAC outputs ($V_{dacp}$ and $V_{dacm}$).

$$V_{in,constructed} = \frac{V_{res} - V_{resb} + V_{dacm} - V_{dacp}}{2} \tag{3.43}$$

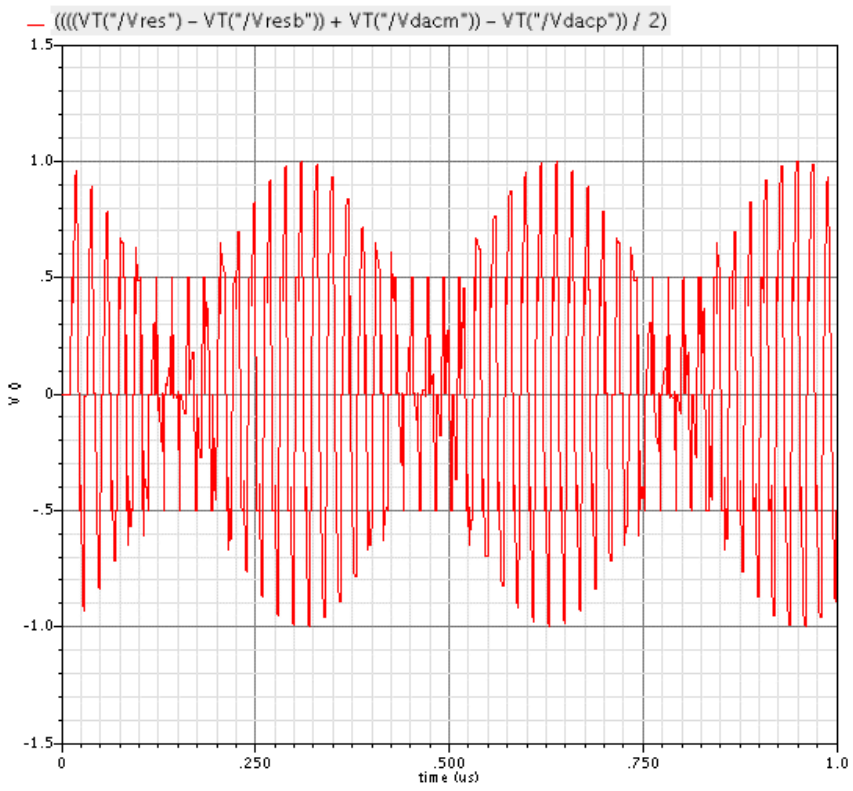


Fig.3.32. The constructed signal used in THD Calculation

One critical thing to consider when selecting the points required for doing FFT for the constructed signal, is to take those points after the signal level is completely settled as in figure. 3.33. It was experimented through multiple iterations of calculating the THD of the pipeline ADC stage that taking those FFT points before reaching the complete settling level is resulting in significant deviation of the THD from the right value, which might mislead us to choose for example an OTA with higher gain than actually needed and make the realization of the design in transistors a big challenge.

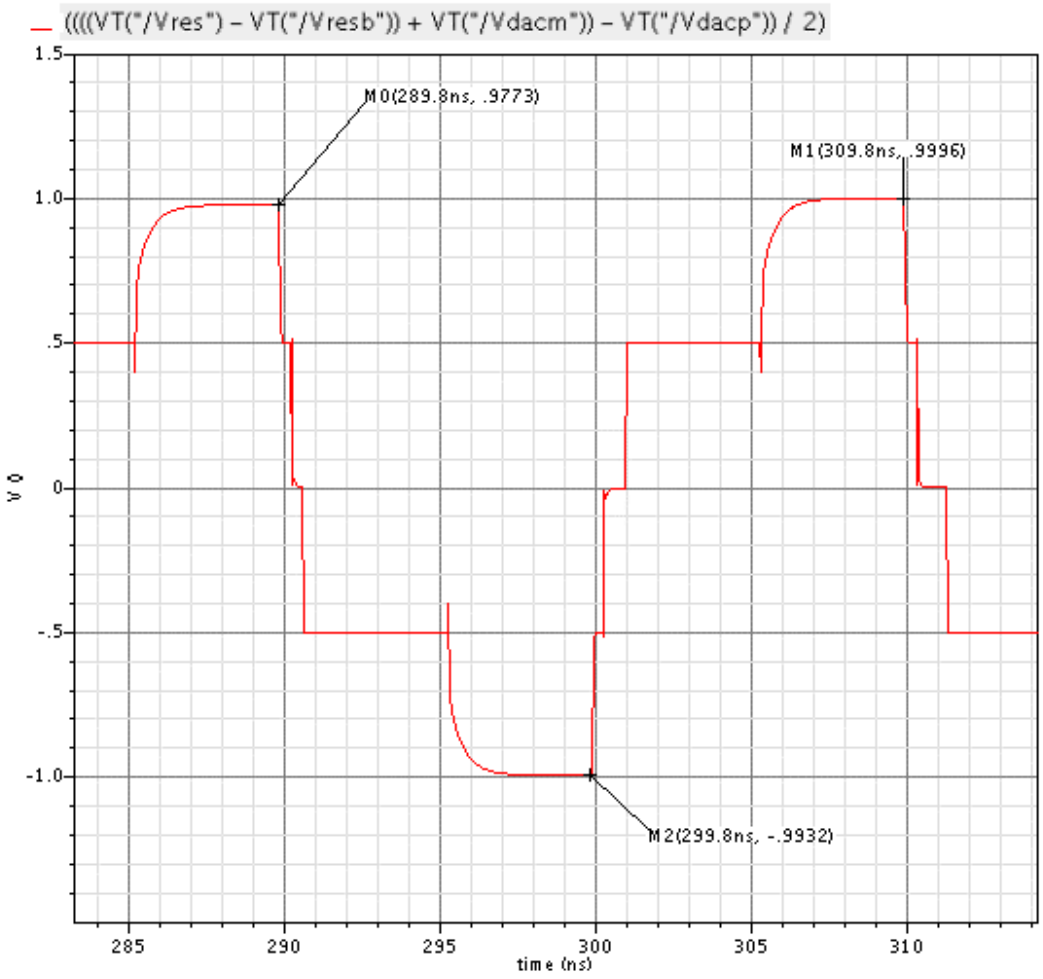


Fig. 3.33 The selection of the FFT points after complete settling is reached

The FFT plot for the constructed signal is shown in figure 3.34, where frequency (f) is normalized to the sampling frequency ($f_s$) along X axes.

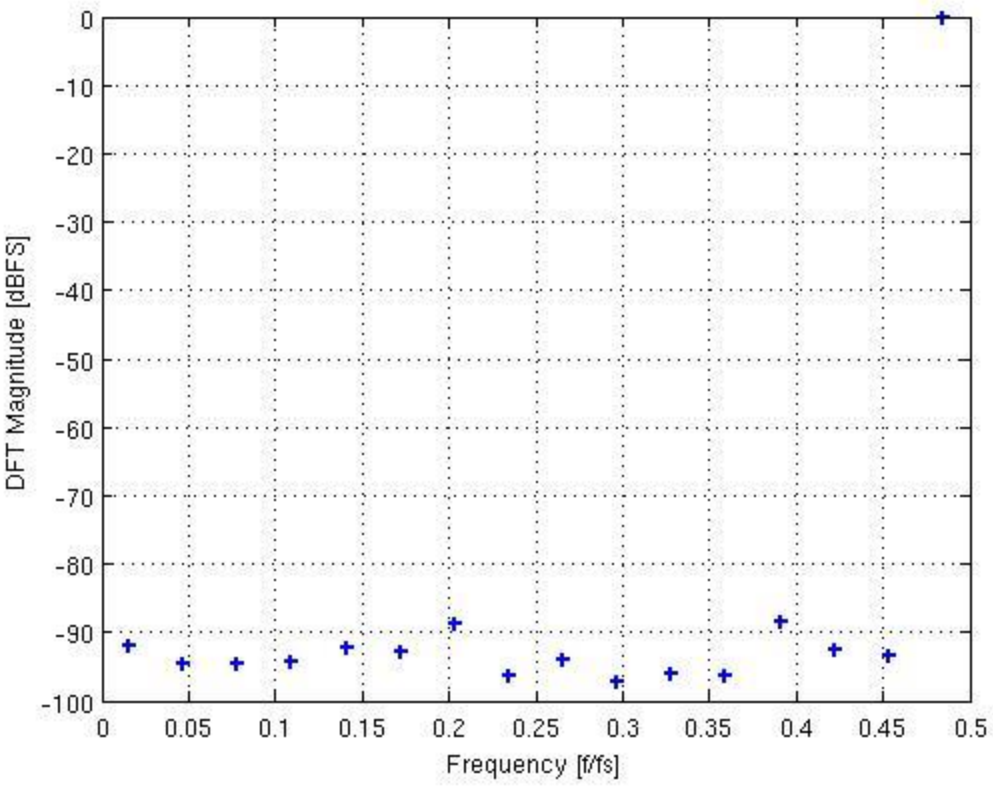


Fig. 3.34 The FFT plot of the constructed signal.

The THD can be calculated as the ratio of the sum of the powers of all harmonic components to the power of the fundamental frequency. The THD for pipeline ADC stage using the obtained design parameters is -80.9917 dB

### 3.3.8 SUMMARY

In the closing of this chapter, a summary of the design requirements, the obtained design parameters in order to achieve these requirements, and the achieved results is presented in table 3.17 and table 3.18

| | THD (dB)<br>for input at (31/64)×100MHz | Input referred noise<br>(μV rms) |
|---|---|---|
| Specifications | -70 | 200 |
| Results | -80.9 | < 200 |

Table 3.17 Design Specifications and Results

| Design Parameter | |
|---|---|
| C | 200f F |
| Gm | 1mS |
| A | 2700 |

Table 3.18 Design Parameters

# Chapter 4: Design and Integration of 1.5-bit Stage Pipeline ADC

After successfully simulating the design with ideal circuit blocks, the next step was to perform the transistor level design of the MDAC and the comparator. We will start off with the comparator design and integration, then will follow that with the implementation of the bootstrap switch design. We will end this chapter with the design of the MDAC block, and simulating the whole stage with transistor level components and verify that the design still meeting its requirements.

## 4.1 Voltage Comparator Design

Fortunately the design of a voltage comparator to be placed in a pipeline ADC stage is an easy task for a circuit designer. The reason for that is the low-resolution stages in the pipeline architecture allows comparators for these stages to be built very cheaply and to consume very little power as well. One of the most popular voltage comparators used in the pipeline ADC architecture is the "dynamic latch" design. This particular design of voltage comparators differentiates itself from other designs by its fast speed, low power consumption, high input impedance, full-swing output, and no DC current.

Figure 4.1 shows the actual example of dynamic latched comparator used our design. This design is also known as "Strong-Arm latch". M6 and M7 are input transistors. M2, M3 are reset transistors. During the rest phase, when clock is low, reset transistors are on, and the output nodes, Voutm and Voutp, get charged to the supply voltage, VDD. In the rest phase, the NMOS tail transistor, M8, is turned off, thus no supply current flows through the differential amplifier during the reset phase.

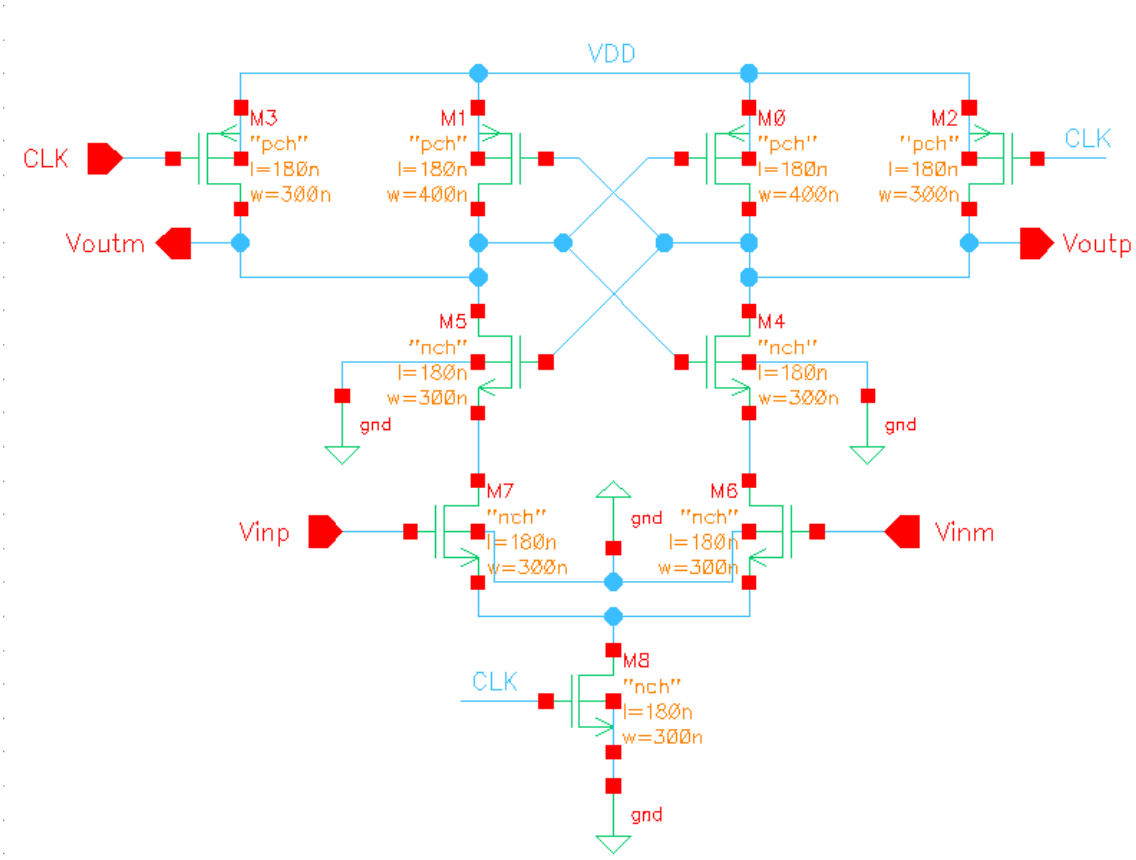


Fig. 4.1 Strong-Arm Latch Voltage Comparator

In regeneration phase, when clock is high, reset transistors are off, and the tail transistor is turned on. The back-to-back inverters receive different amount of current dependent on the input voltage and start to re-generate output. In other words, the drain voltages of input transistors are getting discharged (from VDD to ground) with different slew rates depending on their gate voltages. Once the drain voltage (either of M6 or M7) drops below (VDD – $V_{th}$), the NMOS transistor of the inverter connected to that drain node is turned on and the appropriate output node starts to discharge. Since the output node is connected to the input of another inverter, the PMOS transistor of this inverter is turned on. Consequently, the output is regenerated – one node is digital 1 while the other is 0.

The sizing of the different transistors of the "strong-arm latch" is showing in the table 4.1, where the length of all transistors are the minimum length in the 0.18µm CMOS technology.

| Transistor | Type | Width (in nm) |
|---|---|---|
| M0 | PMOS | 400 |
| M1 | PMOS | 400 |
| M2 | PMOS | 300 |
| M3 | PMOS | 300 |
| M4 | NMOS | 300 |
| M5 | NMOS | 300 |
| M6 | NMOS | 300 |
| M7 | NMOS | 300 |
| M8 | NMOS | 300 |

Table 4.1. Strong-Arm Latch transistors sizes (all transistors' lengths are 180nm)

The testing of the transistor-level design of the voltage comparator showed excellent results. It surprisingly outperformed the ideal comparator model when both comparators were tested in the 1.5-bit stage. The THD of the reconstructed signal in figure 3.31 using transistor-level design was less than that of the case when the ideal comparator model was used. Figure 4.2 shows the FFT plot of the reconstructed signal using the "strong-arm latch" voltage comparator in the 1.5-bit stage design. The THD of the FFT plot in figure 4.2 was calculated as -86.72dB.

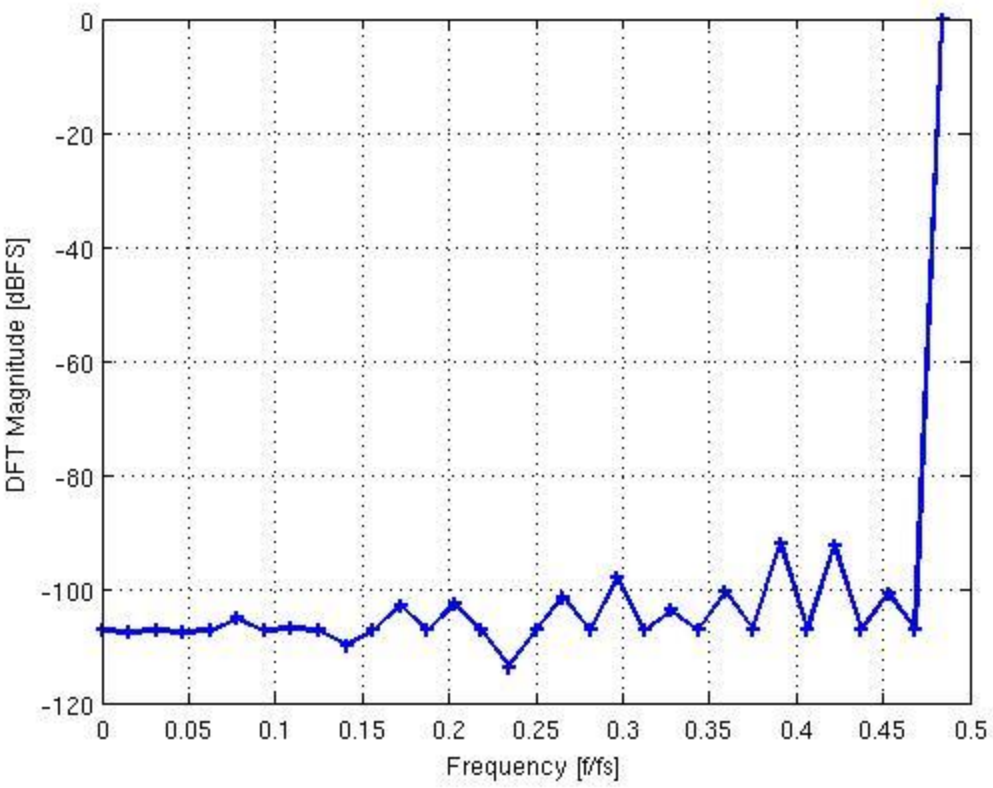


Fig. 4.2 The FFT plot of the constructed signal using

Transistor-level voltage comparators

## 4.2 SWITCHES DESIGN

Designing a high-performance sampling switch is very critical to achieve high linearity of the overall ADC design. Sampling switches non-linearity is mainly attributed to their non-linear on resistance and associated parasitic capacitance which produce harmonic distortion when sampling high-frequency signals. This limits not only SNAD (Signal to Noise and Distortion Ratio) but also SFDR (Spurious Free Dynamic Range) and THD (Total Harmonic Distortion) which are fundamental metrics to measure linearity.

Using transmission gates switches can alleviate the correlation problem between the time-varying input signal, $V_{in}(t)$, and the ON resistance of the transistors.

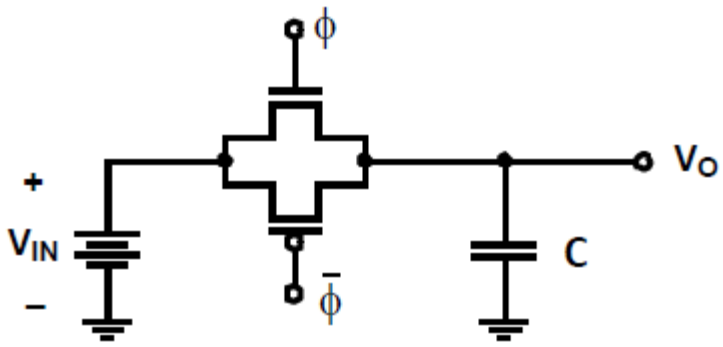


Fig. 4.3 Transmission Gate Switch

The equivalent ON resistance of the transmission gate shown in figure 4.3 can be written as the parallel combination of the ON resistance of the NMOS and PMOS devices as in equation 4.1.

$$R \approx \frac{1}{\mu_n C_{ox}\left[\frac{W}{L}\right]_n \left(V_{GS_n} - V_{tn}\right)} \Bigg\| \frac{1}{\mu_p C_{ox}\left[\frac{W}{L}\right]_p \left(V_{GS_p} - V_{tp}\right)} \tag{4.1}$$

The above equality can be shown to be independent on the input signal if the condition in equation 4.2 is satisfied.

$$\mu_n \left[\frac{W}{L}\right]_n = \mu_p \left[\frac{W}{L}\right]_p \tag{4.2}$$

Such requirement is hard to realize in real fabrication processes and hence the idea of using transmission gates as sampling switch and achieve high linearity vanishes.

### 4.2.1 BASIC BOOTSTRAP CIRCUIT CONCEPT

Another idea for implementing the sampling switches is the bootstrap switch. The idea beyond the bootstrap switch is to fix the voltage difference between the gate and source nodes of the sampling transistor. This way the ON resistance of the switch doesn't vary as the input signal varies.

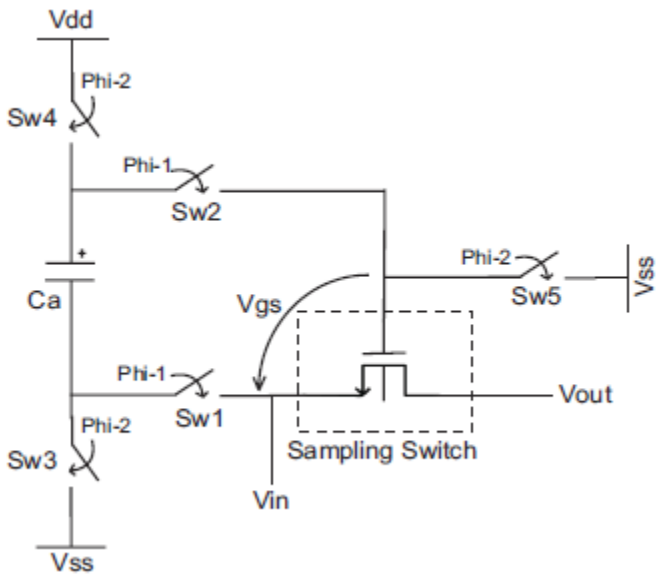


Fig. 4.4. Basic Bootstrap Circuit

Figure 4.4 shows the conceptual circuit of gate-source bootstrapping technique which has been implemented in this report. It shows the main sampling switch along with five additional switches (Sw1-Sw5) and a bootstrap capacitor. Although in the actual circuit there are more than five switches for specific reasons which will be described later on. This bootstrapping topology works with two phases of non-overlapping clocks Phi-1 and Phi-2. The term non-overlapping clocks mean that the two clock phases have the same frequency but they are not high simultaneously at any instant of time. The generation of two non-overlapping clocks is an essential requirement because it greatly enhances

performance by guaranteeing that charge is not advertently lost. It actually prevents the capacitor Ca from being partially discharged during the transition phase of the clock. Switches Sw3 and Sw4 charge the capacitor Ca to Vdd during Phi-2 phase of the Clock. The charged capacitor is then connected in series with the input voltage Vin during phase Phi-1 of the clock through switches Sw1 and Sw2. This results in the gate-source voltage of the sampling transistor approximately equal to Vdd which is the voltage across capacitor Ca. Switch Sw5 makes sure that the sampling switch is not conducting by connecting it to Vss during phase Phi-2 of the clock when capacitor Ca is being charged to Vdd. The advantage of this scheme is that switch conductance becomes independent of the time varying input voltage which linearizes the sampling switch.

### 4.2.2 TRANSISTOR LEVEL IMPLEMENTATION

The actual bootstrap switch circuit used in this report is adopted from [3] as it's shown in figure 4.5. The charge pump circuitry is shown in the dotted box. The idea behind using charge pump circuit is to output voltage that is several times higher than VDD, which in this case is needed to drive transistor MN-3 to charge Ca to the supply voltage, VDD. Unlike the other traditional DC-DC converters, which employ inductors, charge pump circuits are only made of capacitors and switches, thereby allowing integration on silicon. In the following section we will study the basic operation of such circuits before analyzing the operation of the modified version of charge pump in our bootstrap switch circuit.

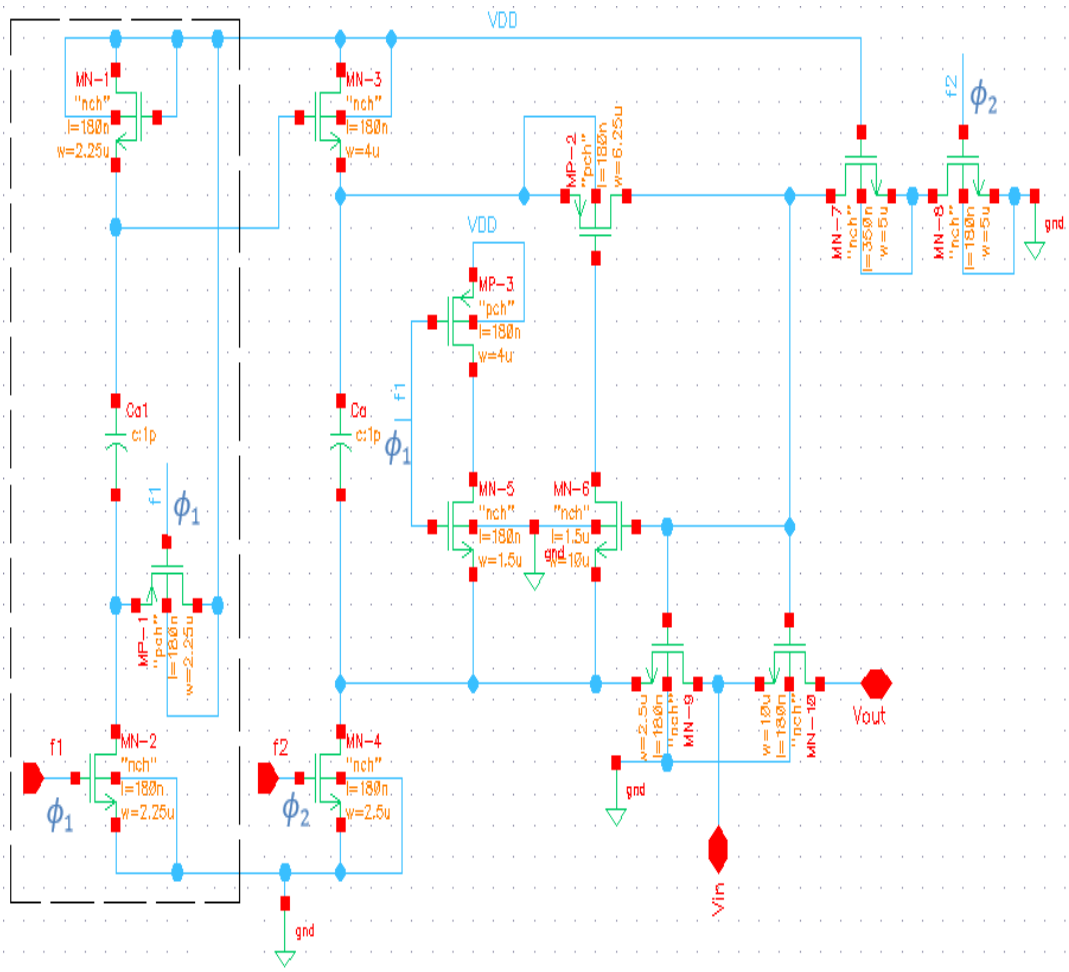


Fig. 4.5 Transistor Level Implementation of the bootstrap switch circuit

### 4.2.3 CLASSICAL CHARGE PUMP CIRCUIT OPERATION

Abo uses a classical charge pump as shown in figure 4.6. The cross-coupled NMOS transistors, M1 and M2, self-charge the capacitors C1 and C2 to $V_{dd}$ when $\phi_1$ is applied. Double clock signals between $V_{dd}$ and $2V_{dd}$ are obtained at the gates of MN-1 and MN-2 at alternative clock cycles $\phi_1$ and $\phi_2$ respectively. The disadvantage of this design is that NMOS transistors M1 and M2 in figure 4.6 are free from the reliability as their gate-source

voltage is approximately twice as much as the specified supply voltage (1.8 V for 0.18um process).

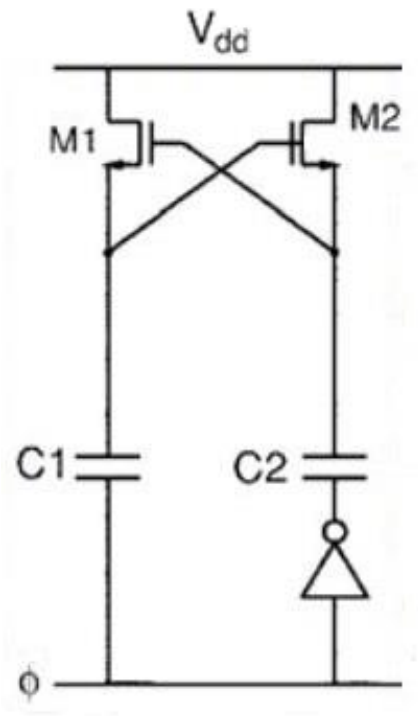


Fig. 4.6 Charge pump for clock doubling in Abo's architecture

### 4.2.4 CHARGE PUMP CIRCUIT IN THE IMPLEMENTED BOOTSTRAP SWITCH

As we previously noted, the charge pump circuit used in the actual design of the bootstrap switch is shown in the dotted box if figure 4.5. This solution provides clock doubling by ensuring the reliability constraint. None of the transistors in this scheme have terminal voltages $V_{gs}$, $V_{ds}$ and $V_{gd}$ exceeding the specified supply voltage of 1.8 V. During phase $\phi_1$ of the clock, capacitor Ca1 is pre-charged to a fixed value of $V_{dd} - V_t$, where $V_t$ is the voltage drop across diode connected transistor MN-1. During the next phase of Phi-1, MP-1 transistor connects the bottom plate of the capacitor Ca1 to $V_{dd}$. As a result the

top plate of the capacitor Ca1 rises to $2V_{dd} - V_t$. The gate voltage of transistor MN-3 toggles between $(V_{dd} - V_t)$ and $(2V_{dd} - V_t)$.

The transistors' sizes of the bootstrap switch in figure 4.5 are shown in table 4.2. I followed the strategy used in to size the different transistors especially MN-10 and MP-2 which are most critical for achieving high linearity.

| Transistor | Type | Width | Length |
|---|---|---|---|
| MN-1 | NMOS | 2.25 | 0.18 |
| MN-2 | NMOS | 2.25 | 0.18 |
| MN-3 | NMOS | 4 | 0.18 |
| MN-4 | NMOS | 2.25 | 0.18 |
| MN-5 | NMOS | 1.5 | 0.18 |
| MN-6 | NMOS | 10 | 1.5 |
| MN-7 | NMOS | 5 | 0.35 |
| MN-8 | NMOS | 5 | 0.18 |
| MN-9 | NMOS | 2.5 | 0.18 |
| MN-10 | NMOS | 10 | 0.18 |
| MP-1 | PMOS | 2.25 | 0.18 |
| MP-2 | PMOS | 4 | 0.18 |
| MP-3 | PMOS | 6.25 | 0.18 |

Table 4.2 Transistors' sizes of the bootstrap switch

The testing of the bootstrap switch in our 1.5-bit pipeline ADC stage design showed acceptable performance in terms of meeting and exceeding the linearity requirements as it showing in FFT plot of constructed input signal in figure 4.7 using transistor level flash ADC and bootstrap sampling switches.

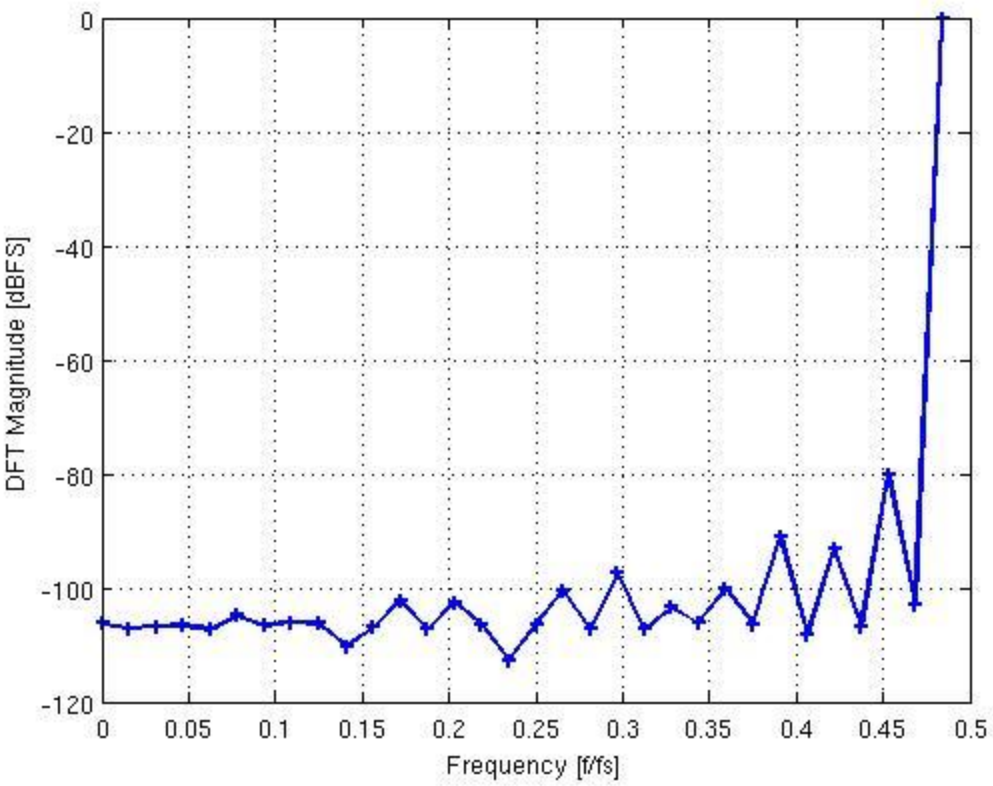


Fig. 4.7 The FFT plot of the constructed signal using
Transistor-level voltage comparators and bootstrap switches

THD = -79.19dB

## 4.3 OTA DESIGN

The selection of which op-amp to use in a particular application is often the key factor which determines the success or failure of the circuit. There is a wide range of the available op-amp designs that circuit designers can pick from for the applications of interest. So, with all of this variety, how can one go about selecting the right design? First, one must determine several characteristic like gain, unity gain bandwidth, slew rate, output swing, output resistance, and other parameters that this report is not intended to expose to.

Since the OTA gain is one of the important factors that directly affecting the linearity of the pipeline ADC architecture as we discussed in the previous chapter, then selecting an OTA with enough gain would be the first thing to look for in the pursued OTA. The telescopic OTA architecture is of one of those OTA designs famous for high gain, but on the other side they are notorious for their low output voltage swing range and high output resistance. The last two attributes of the telescopic OTA would let us avoid using it in our design because as we allow more output voltage swing range as we maximize the Signal-to-Noise (SNR) ratio and vice versa. Secondly the high output resistance would affect the speed of the OTA as it would need more time to charge the load capacitor connected at the output, and hence would put a constrains on maximum frequency at which the whole design can operate.

Another OTA architecture that would solve the issues associate with the telescopic architecture with still achieving high gain is the two-stage folded cascode OTA architecture. Figure 4.8 shows the schematic of the implemented OTA, the design is adopted from [Ishii 2005] and modified by implementing the transistors of the input stage using NMOS transistors.

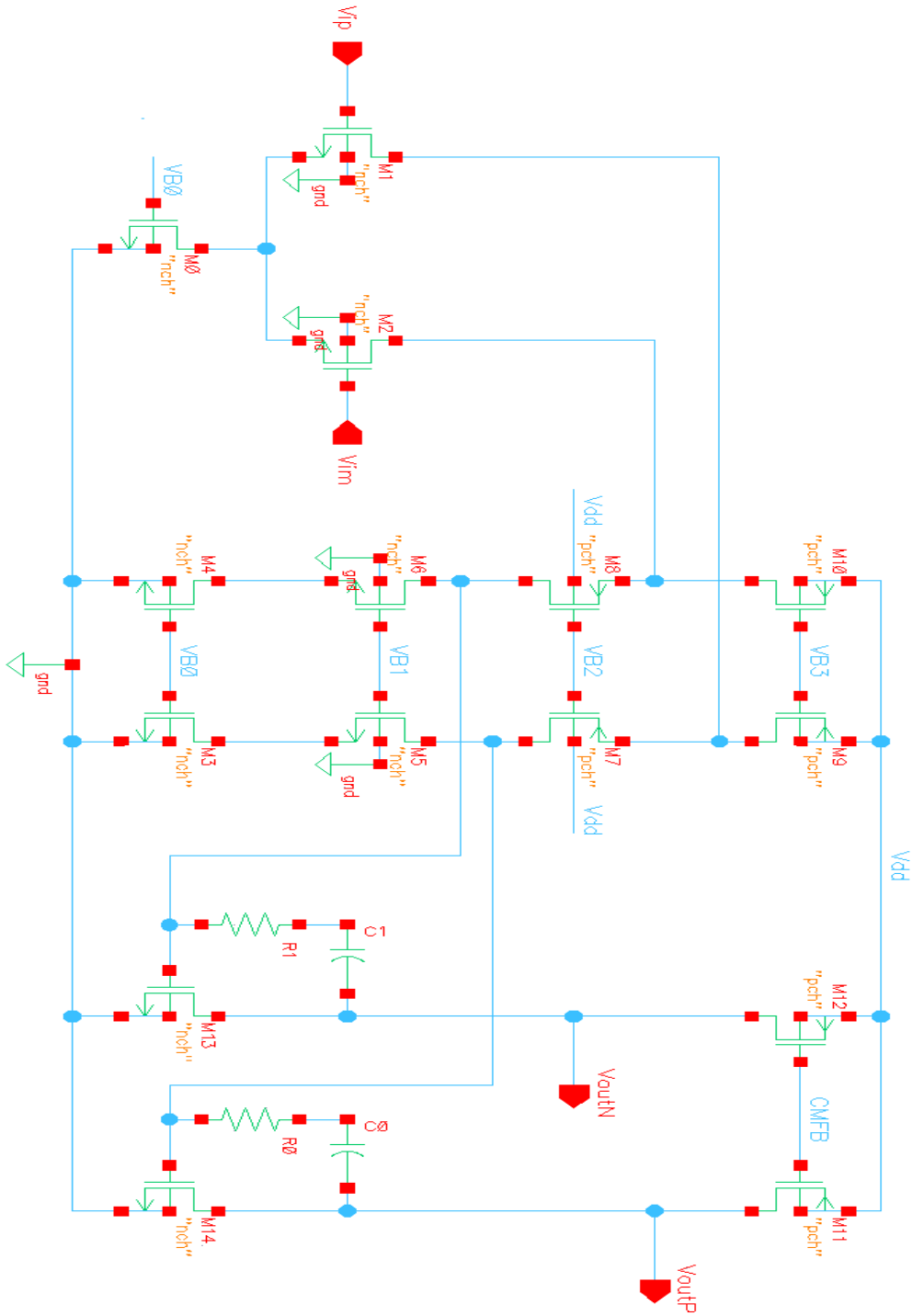


Fig 4.8 Two-Stage Folded Cascode OTA

### 4.3.1 OTA GAIN ANALYSIS

As we can see from the circuit diagram of the two-stage folded cascode OTA, instead of achieving high gain from a single stage design, we use two-stage each with gain much less than the final desired gain. By relaxing the gain constrains on each stage, this will also reduce the output resistance as there is no cascaded devices in the second stage, and also result in much more headroom available for the output signal swing. The accumulative gain of the whole design is the product of the gains of the two stages.

The gain of the first stage of the two-stage folded cascode OTA can be calculated as:

$$A_1 = G_{m1} R_{out1} \tag{4.3}$$

Where $G_{m1}$ is transconductance of the input transistor $M_1$ (or transistor $M_{2)}$ , and $R_{out1}$ is the output resistance of the first stage and can be calculated as the parallel combinations of resistors seen by looking up and down at the common drain node of transistors $M_7$ and $M_5$ (or the common drain node of transistors $M_8$ and $M_6$ ).

$$R_{out1} = \{g_{m7}\, r_{ds7}\, (r_{ds9} \,||\, r_{ds1})\} \,||\, \{g_{m5}\, r_{ds5}\, r_{ds3}\} \tag{4.4}$$

Similarly, the gain of the second stage can be written as the product of the transconductance of input transistors of second stage and the output resistance at the output nodes, which can be calculated as in equation 4.5.

$$A_2 = g_{m14}(r_{ds11} \,||\, r_{ds14}) \tag{4.5}$$

The total gain of the OTA is the product of $A_1$ and $A_2$ in equations 4.4 and 4.5 respectively and can be expressed as in equation 4.6.

$$A = g_{m14}(r_{ds11} \,||\, r_{ds14})\; g_{m1}\; [\{g_{m7}\, r_{ds7}\, (r_{ds9} \,||\, r_{ds1})\} \,||\, \{g_{m5}\, r_{ds5}\, r_{ds3}\}] \quad (4.6)$$

### 4.3.2 INPUT AND OUTPUT SIGNALS RANGES:

One of the interesting characteristics about the folded cascode OTA is that the maxim input signal range can exceed the voltage supply limits ($V_{dd}$). For the implemented OTA, the maximum and the minimum ranges for the input signal, and the maximum and minimum output swings is summarized in the table 4.3.

| Signal Range | Expression |
|---|---|
| Minimum Input Range | $V_{GS}\,(M1) + V_{DS,sat}(M0)$ |
| Maximum Input Range | $V_{th}(M1) + V_{D,sat}(M9)$ |
| Minimum Output Swing | $V_{D,sat}(M14)$ |
| Minimum Output Swing | $V_{dd} - V_{D,sat}(M11)$ |

Table 4.3 Minimum and Maximum Signals Ranges

As we can see in the above table that the output swing is enlarged in this OTA topology to expand between the limits that won't let transistors $M_{11}$ and $M_{14}$ (or transistors $M_{12}$ and $M_{13}$) leave the saturation operation region. The minimum input range is limited by the gate to source voltage of input transistor M1 and the drain to source voltage of the tail current source M0.

### 4.3.3 PHASE MARGIN ANALYSIS

The value of capacitors $C_0$ and $C_1$ are very critical as they act with their associated resistors as knobs to control the phase margin of the OTA. For an OTA to be used in data converter block, we worry not only about the stability of the OTA, but also we worry about how the transient response of the OTA for a certain phase margin looks like. If we design the OTA such that the phase margin is very small, the output of the OTA output might overshot before it settles. On the other hand if the phase margin is very large, the OTA output response will be over damped. Neither phase margin are acceptable in data converter designs world. The plot in figure 4.9 shows the different transient responses versus their phase margins values.

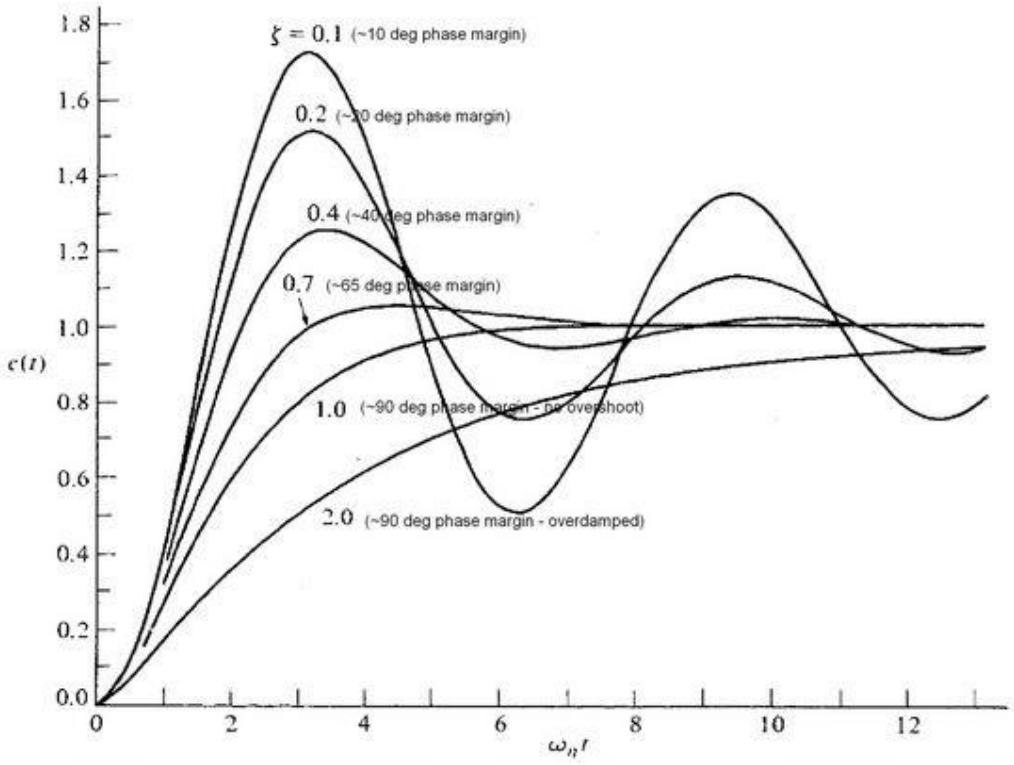


Fig. 4.9 2nd Order Transient Curves

Form Dorf, Richard C. Modern Control System. 3rd Edition

Integrating the OTA in the 1.5-bit stage design to obtain the required linearity for the whole stage was a challenging task as choosing the right values for the resistor and the capacitor in the feedback loop of the second stage is sensitive to achieve such high linearity. The AC analysis of the implemented OTA is shown in figure 4.10 with a DC gain more than 80dB and phase margin of 57˚ degree.

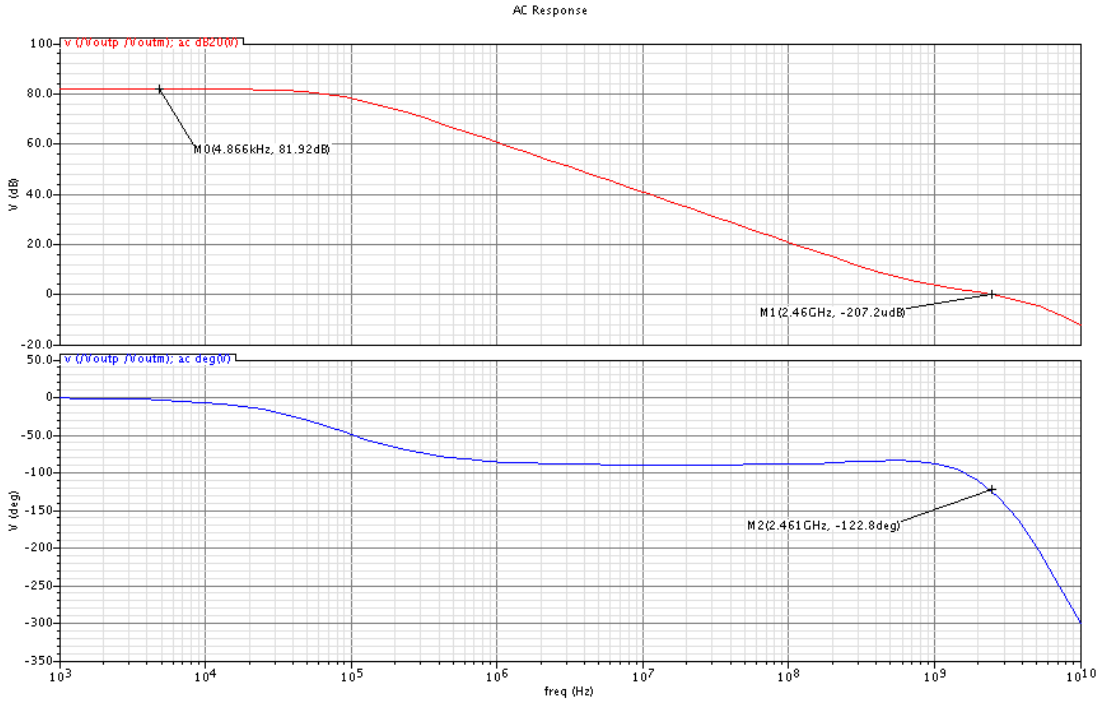


Fig. 4.10 AC Analysis Results

### 4.3.4 COMMON-MODE FEEDBACK DESIGN

For differential designs, the operating output voltage is very sensitive to small changes in device characteristics. Small changes in the bias voltage can cause very large changes in the operating output voltage. In order to control these variations, common-mode feedback is employed. Common-mode feedback applies a variable amount of bias current

in order to correct for excursions from the ideal output common-mode voltage. In order to achieve this, circuits are required to sense the common-mode voltage, to compare this sensed voltage to the desired common-mode voltage, and to apply a variable amount of current based on the difference between actual output common-mode and desired common-mode. For the purposes of design simulations, an ideal CMFB network was used to ensure that the base OTA design worked by itself. Figure 4.11 illustrates the schematic for the ideal CMFB network.

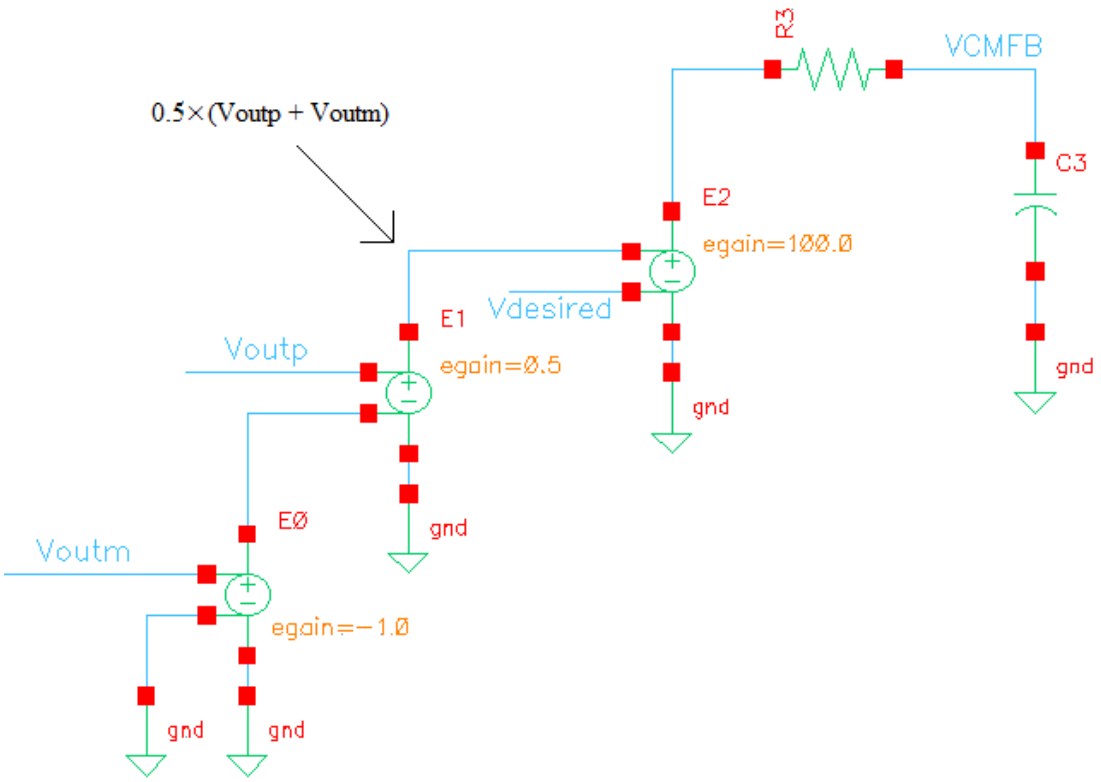


Fig. 4.11 Ideal CMFB Circuit

The voltage nodes, $V_{outp}$ and $V_{outp}$, are the positive and negative output voltages, respectively. An ideal Voltage-Controlled Voltage Sources are used to sense the deviation of the actual common mode voltage from its desired voltage value. The voltage difference between the two voltages is amplified by a gain factor of a hundred then passed to the RC low pass filter network that outputs the common mode feedback voltage. The output common-mode voltage will be very close to the desired common-mode voltage, which is the desired operation of the CMFB network.

### 4.3.5 OTA TRANSISTORS SIZING

After getting the initial design of the OTA to meet the gain requirement and integrating it into the 1.5-bit stage ADC, many iterations of simulating the whole design had to be performed before the OTA achieved the desired linearity. The simulation of the initial design of the OTA showed that it requires more time for the complete settling of the output to occur. By increasing the devices' width of the second stage, the whole response of the pipeline ADC stage was improving. In Cadence, we can increase or decrease the NMOS and PMOS devices' widths by certain ratio by changing the value in the "Multiplier" field under the property of the device. Table 4.4 summarizes the sizes of the devices used in building the OTA circuit along with the ideal CMFB network.

| Transistor | Type | Width (μm) | Length (μm) |
|---|---|---|---|
| M0 | NMOS | 26.6 | 0.26 |
| M1 | NMOS | 15.2 | 0.26 |
| M2 | NMOS | 15.2 | 0.26 |
| M3 | NMOS | 13.3 | 0.26 |
| M4 | NMOS | 13.3 | 0.26 |
| M5 | NMOS | 13.3 | 0.26 |
| M6 | NMOS | 13.3 | 0.26 |
| M7 | PMOS | 32.36 | 0.34 |
| M8 | PMOS | 32.36 | 0.34 |
| M9 | PMOS | 67.26 | 0.34 |
| M10 | PMOS | 67.26 | 0.34 |
| M11 | PMOS | 32.67 | 0.34 |
| M12 | PMOS | 32.67 | 0.34 |
| M13 | NMOS | 7.54 | 0.26 |
| M14 | NMOS | 7.54 | 0.26 |

Table 4.4 OTA Transistors Sizes

### 4.4 1.5-BIT STAGE SIMULATION RESULTS WITH TRANSISTOR-LEVEL COMPONENTS

The simulation of the whole stage using the implemented blocks including voltage comparators, bootstrap switches and MDAC showed a good performance in terms of linearity as the THD is still below -70dB (-71.224dB as measured). The FFT plot for the constructed signal from the whole stage with transistor-level component is shown in figure 4.12.

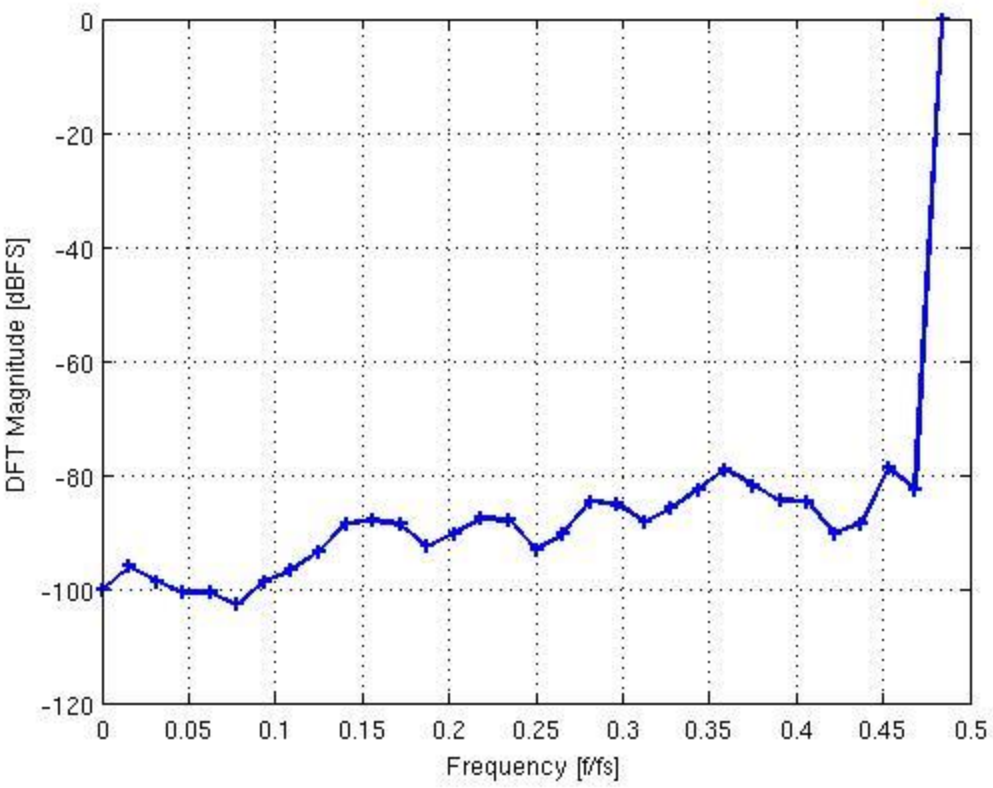


Fig 4.12 The FFT plot of the constructed signal using

Transistor-level components

# Chapter 5: Implementation of 10-bit pipeline ADC

## 5.1 Conventional Pipeline ADC Architecture

The proposed 10-bit pipeline ADC applies *Digital Error Correction* (DEC) and consists of eight pipeline stages, and a 2-bit flash ADC at the end as shown in figure 5.1. Each pipeline stage resolves 1.5 bits, which is represented by a 2-bit output. In the end, a total of 18-bits are generated from a single input sample.

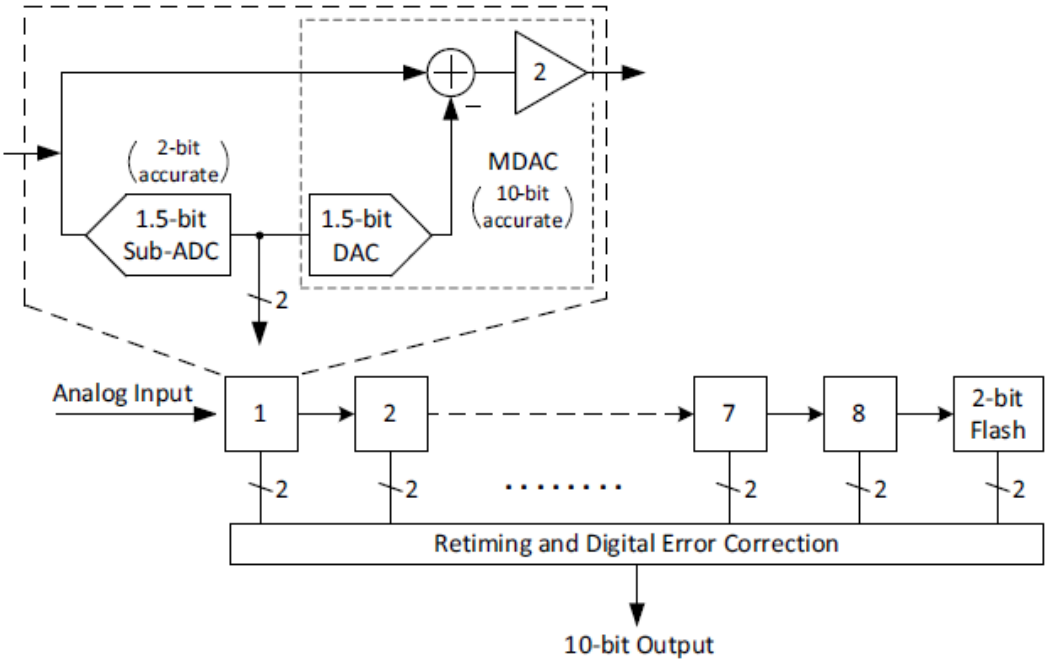


Fig. 5.1 A 10-bit 1.5-bit/stage pipeline ADC with digital error correction

## 5.2 Digital Error Correction

To apply the *digital error correction* technique, the bits from each adjacent pipeline stage overlap by half a bit and form the expected 10-bit output as shown in figure 5.2. The digital error correction will not correct for errors made in the final 2-bit flash conversion. Any error made at that conversion is suppressed by the large cumulative gain preceding the 2-bit flash.

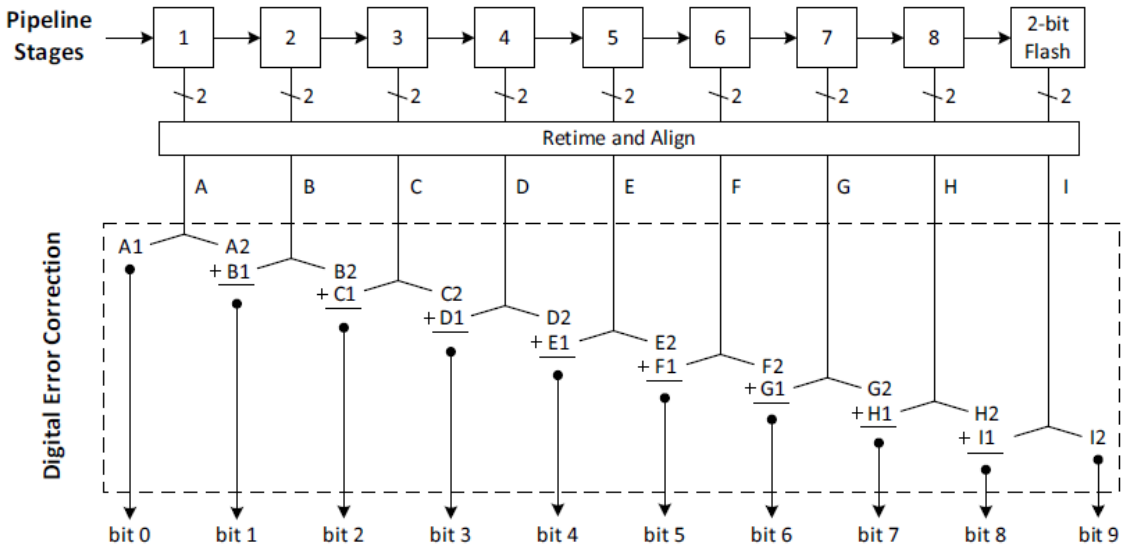


Fig 5.2 Digital error correction in a 1.5-bit/stage 10-bit pipeline ADC

## 5.3 Component Accuracy

Digital error correction does not correct gain or linearity errors in the individual DAC and gain amplifiers. The components of the first stage actually need about 10-bit accuracy, whereas the components in subsequent stages require less accuracy (for example, 8-bit accuracy for Stage 2, 6-bit for Stage 3, and so forth). This need for reduced accuracy is because the later stages' error terms are divided down by the preceding interstage gain(s).

This fact is often exploited to save additional power by making the pipelined stages progressively smaller.

### 5.4 Timing and Clocking Mechanism of Multiple Stages

Stages work on complementary clocks, i.e. for the first stage, we use $\phi 1$ clock for sampling and $\phi 2$ clock for amplification, while for the second stage we use $\phi 2$ clock for sampling and $\phi 1$ clock for amplification and so on. Figure 5.3 shows the phases of the clocks to the T/H amplifiers are alternated from stage to stage such that when a particular T/H in the ADC enters the hold mode it holds the sample from the preceding T/H, and the preceding T/H returns to the track mode. The held analog signal is passed along from stage to stage until it reaches the final stage in the pipelined ADC—in this case, a flash converter.

The delay that each sample is encountering from the moment it is acquired till the digital output representing this sample is produced is depicted in figure 5.3. Once all the stages is filled with data to be processed, the system produce results for every clock cycle. The overall throughput of the pipeline ADC is one digitized sample per clock cycle no matter how many stages exist between the input and the output of the system.

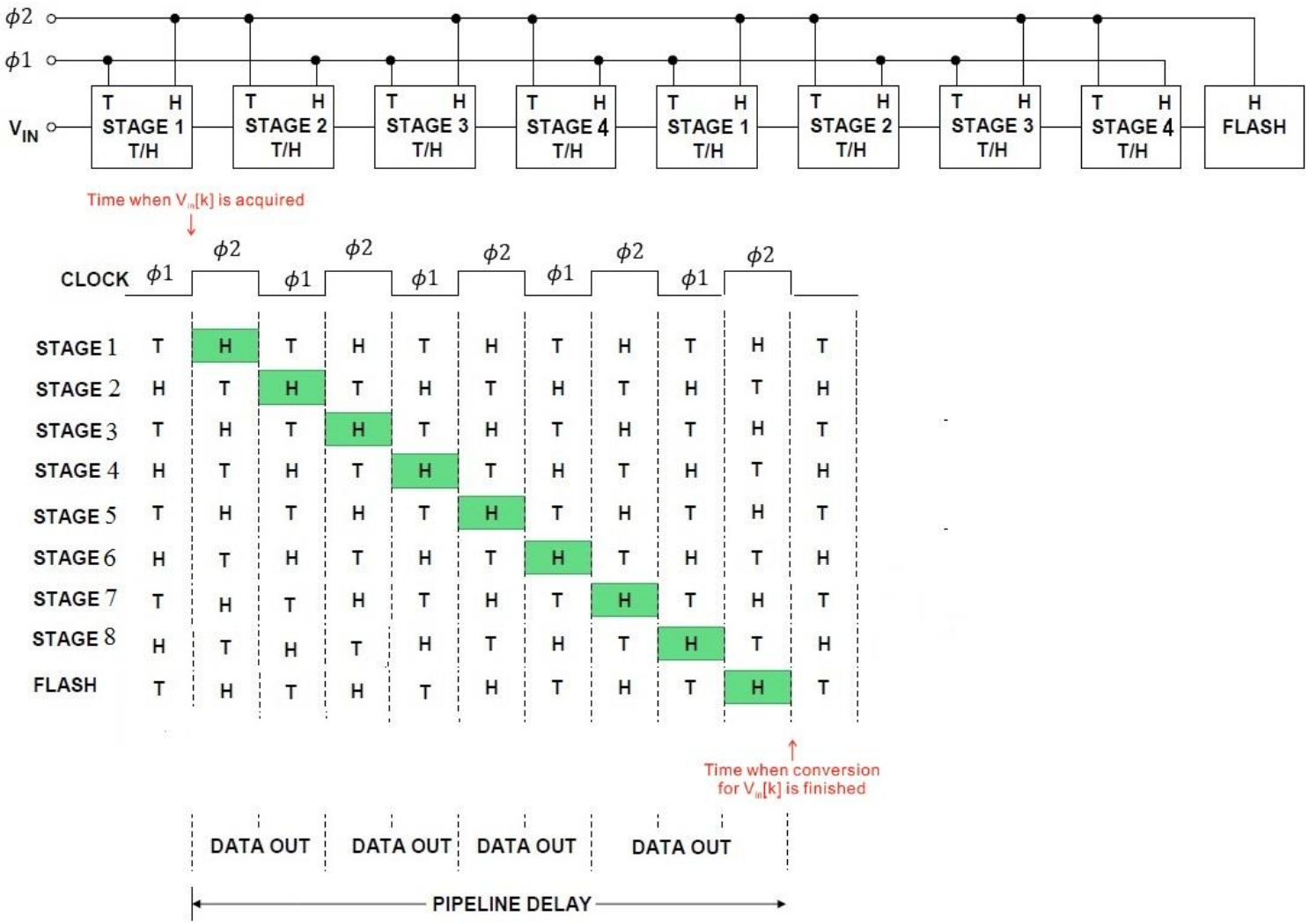


Fig. 5.3 Clocking Mechanism of The Pipeline ADC

## 5.5 STAGE SCALING

Since the pipeline ADC architecture consists of more than one stage, unlike other data converters that might be having only one stage-for example the flash ADC, there was a need to know whether or not the scaling of the pipeline ADC stages should be different or not? Understanding the noise-power relationship helps us as circuit designer choose the optimal scaling to achieve our design targets. The example of the pipeline ADC that is showing in figure 5.4 assumes interstage gain of 2 same as the proposed design in this report, and as we know that the thermal noise introduced at each stage is proportional to the capacitor size as described in equation 3.7

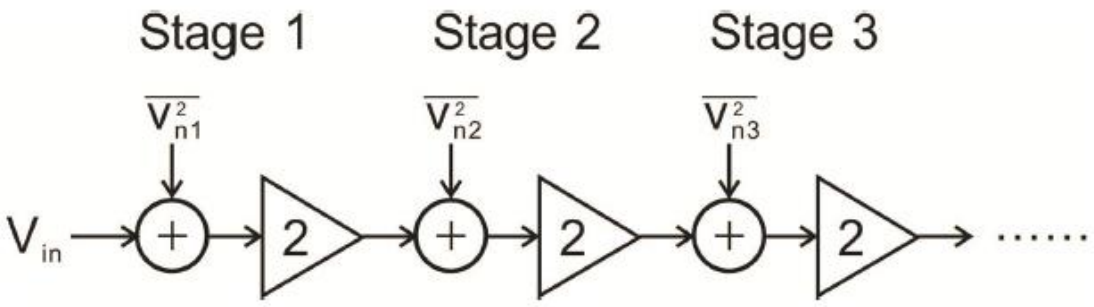


Fig. 5.4 Noise modeling in pipeline ADC stages

The presence a gain block between each stage causes the noise at the input of any stage to be suppressed by a factor equal to the square of the product of all gain terms in previous stages. If the first stage is built with capacitors of the size $C_1$, and the second stage is built with capacitors of the size $C_2$ and so on. Then we can write the total noise $N_{total}$ as the summation of all stages' noise as in equation 5.1.

$$N_{total} \propto KT \left[\frac{1}{C_1} + \frac{1}{4C_2} + \frac{1}{16C_3} + \cdots\right] \tag{5.1}$$

As we can see from equation 5.1 that the first stage contributes the most to the total noise, and later stages in the pipeline contributes the less. If we made a crude assumption and choose not to scale the pipeline stages at all; i.e.$C_1 = C_2 = C_3 = \cdots$, that means that we burn the same amount of power for all the stages while the noise contribution of these stages varies by a lot as we discussed. This crude assumption would be wasting of the power budget and of course is not an optimum solution.

Another suggestion that would come to one's mind is to scale the pipeline stages such that the noise contributions of all the stages are equal, which means the first few stages with larger thermal noise than the rest of the stages would require more power to suppress that noise [power consumption is proportional to capacitor size]. The later stages in the pipeline sequence would burn much less power since their thermal noises are not significant. Figure 5.4 shows the noise contribution versus power consumption of the pipeline stages for the two previous proposed scaling solutions.

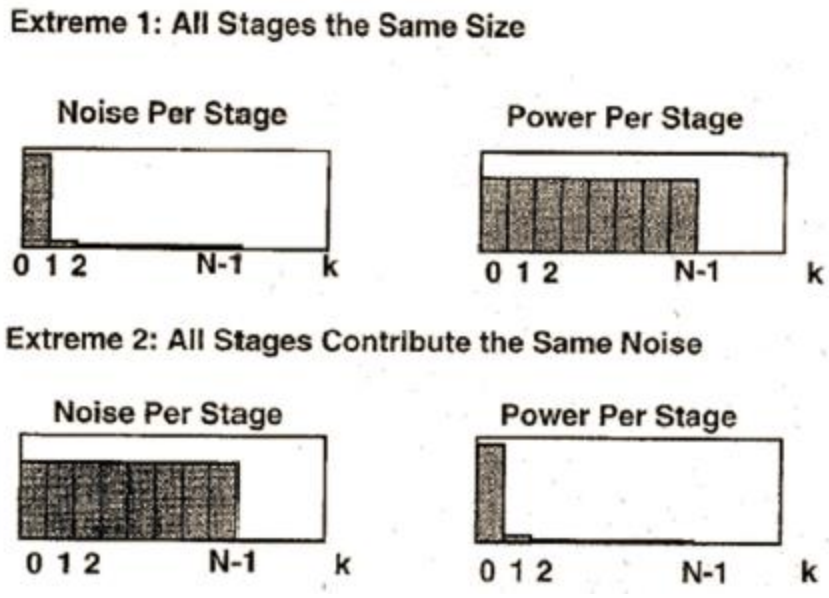


Fig. 5.5 Noise Contribution vs. Power Consumption

As we can see from the above sketch in figure 5.4 that in the first example we burn unnecessarily a lot of power in the later stages of the pipeline ADC as there is no noise that needs all that power to be suppressed, while on the other hand the second solution let most of the power budget to be consumed in the first and the second stages.

Between those two extreme solutions the optimal solution lies. Actually if we find the capacitors' sizes at which the power-noise product of the pipeline ADC is minimized, then that would be the optimal scaling solution for the pipeline ADC. The total power-noise product is proportional to the capacitors of the pipeline stages as it is showing in equation 5.2.

$$N_{total} X\, P_{total} \propto \left[\frac{1}{C_1} + \frac{1}{4C_2} + \frac{1}{16C_3} + \cdots\right] \times [C_1 + C_2 + C_3 \ldots] \tag{5.2}$$

According to Cauchy–Schwarz theorem, the noise-power product term is governed by the inequality given in equation 5.3.

$$\left[\frac{1}{C_1} + \frac{1}{4C_2} + \frac{1}{16C_3} + \cdots\right] \times [C_1 + C_2 + C_3 \ldots] \geq \left[1 + \frac{1}{2} + \frac{1}{4} + \cdots\right] \tag{5.3}$$

The noise-power product term is minimized when:

$$C_1 = 2C_2 = 4C_3 = \cdots = 2^i\, C_{i+1} \tag{5.4}$$

Interestingly enough, the optimal scaling factor as showing in equation (5.4) is identical to the inter-stage gain.

## 5.6 Combining the Bits of the Pipeline ADC Stages

The final digital output bits of the proposed 10-bit pipeline ADC is the combination of the nine stages digital outputs as showing in figure 5.5. For those pipeline ADC which built without redundancy, combining the output bits of the different stages doesn't need extra hardware as all we need to do is bits alignments such that each stage's output is presented in the right order in the final output. In our case, because we have some redundancy built in the proposed design, combing all the stages' output bits is not as simple as bits alignments case. We can think of redundancy as "overlapping" of the output bits of the different pipeline ADC stages. To produce the final output we would need to build an output stage with half-adders, full-adders, and storage units.

Since we have nine stages in our design, and each stage produces two bits once the conversion of its analog voltage to digital output is complete, the overlapping between each stage and the subsequent stage is one bit. If the output if stage i is $D_i$, we can get obtain the finial outputs by doing the addition of the different bits as following.

D1 $d_{1,1}$ $d_{1,0}$

D2 $d_{2,1}$ $d_{2,0}$

D3 $d_{3,1}$ $d_{3,0}$

D4 $d_{4,1}$ $d_{4,0}$

D5 $d_{5,1}$ $d_{5,0}$

D6 $d_{6,1}$ $d_{6,0}$

D7 $d_{7,1}$ $d_{7,0}$

D8 $d_{8,1}$ $d_{8,0}$

D9 $d_{9,1}$ $d_{9,0}$

---

$$D_{out}[9:0] = d_{out,9}\ d_{out,8}\ d_{out,7}\ d_{out,6}\ d_{out,5}\ d_{out,4}\ d_{out,3}\ d_{out,2}\ d_{out,1}\ d_{out,0}$$

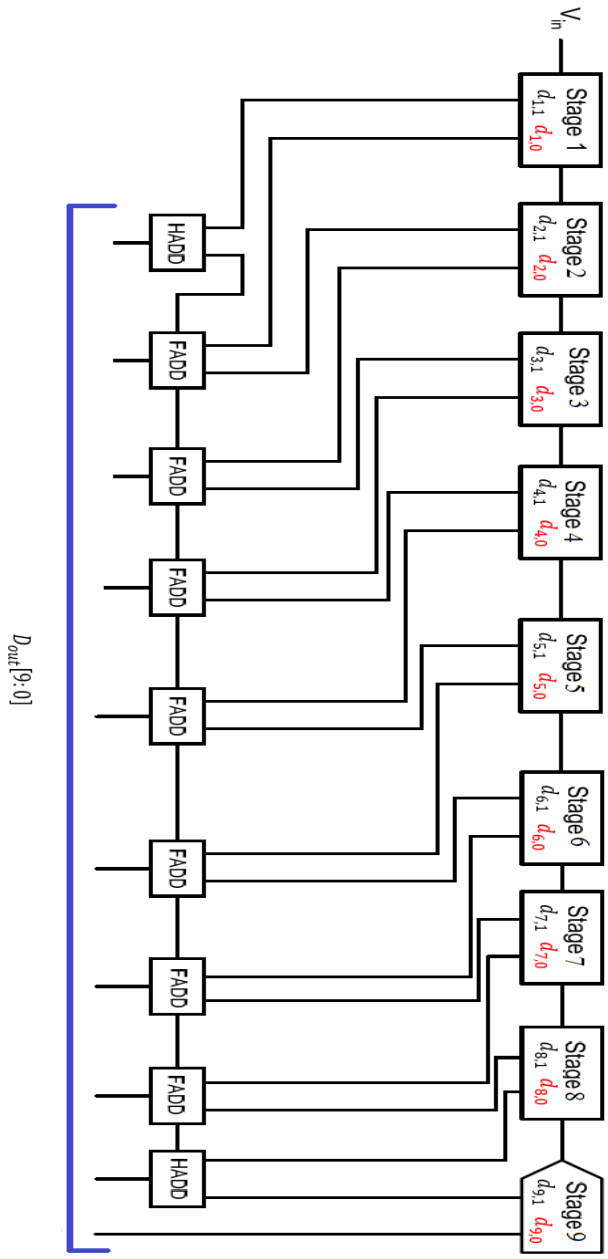


Fig. 5.6 combining bits of the pipeline ADC stages

$$D_{out}[9:0] = D_1 + \frac{1}{2}D_2 + \frac{1}{4}D_3 + \frac{1}{8}D_4 + \frac{1}{16}D_5 + \frac{1}{32}D_6 + \frac{1}{64}D_7 + \frac{1}{128}D_8 + \frac{1}{256}D_9$$

### 5.6.1 STAGES DIGITAL OUTPUTS

A high level schematic abstraction for the 10-bit pipeline ADC in showing in figure 5.7 with a close view of two adjacent intermediate stage to show the complementary clocking mechanism that we described earlier.

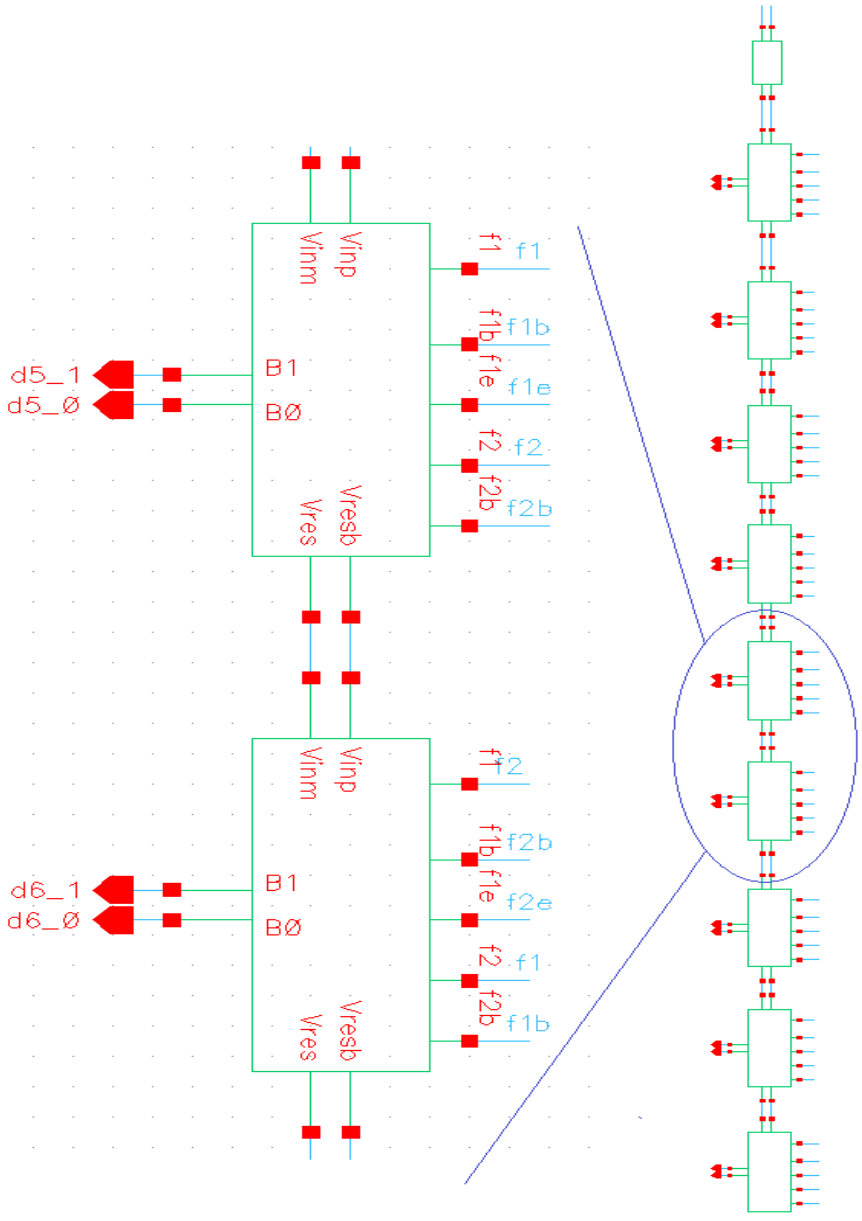


Fig. 5.7 10-bit Pipeline ADC

The output of the any stage is available every 10ns. We start harvesting the output of the first stage at time = 19ns (the first clock cycle is not used for the purpose of this simulation) then at time = 29ns and time = 39ns and so on, i.e. just 1ns before the end of the "hold" phase which is happing during $\phi_1$ low. Figure 5.7 shows the first stage output over the entire simulation time which is 1µs. Figure 5.8 show a close view for the first stage output and clock $\phi_1$.

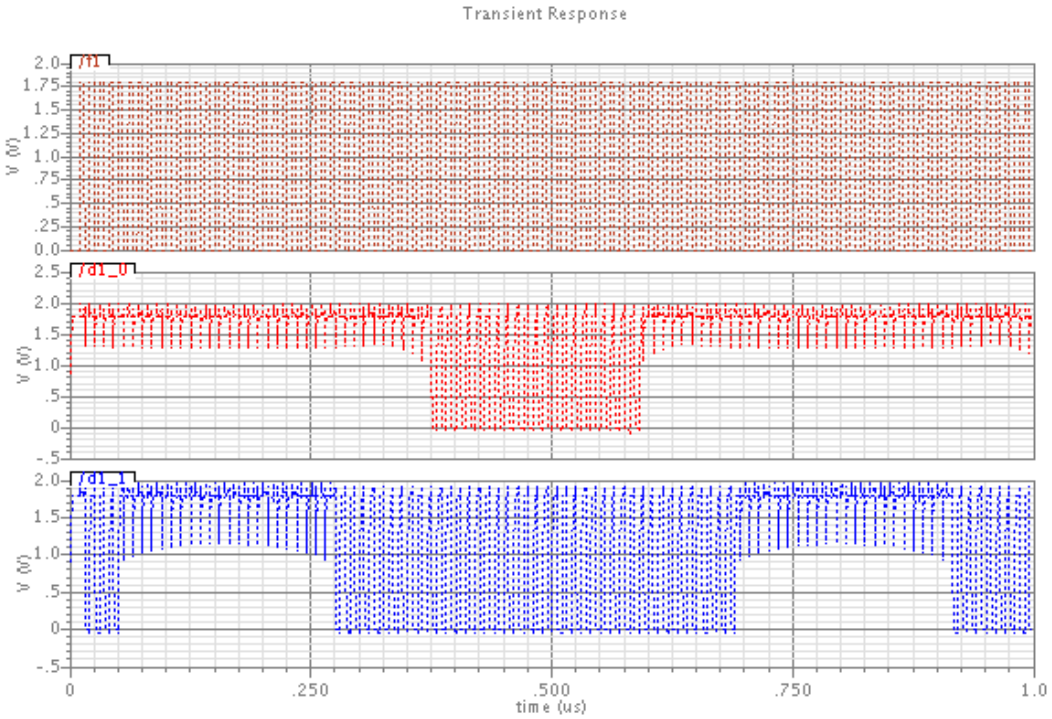


Fig. 5.8 First stage output

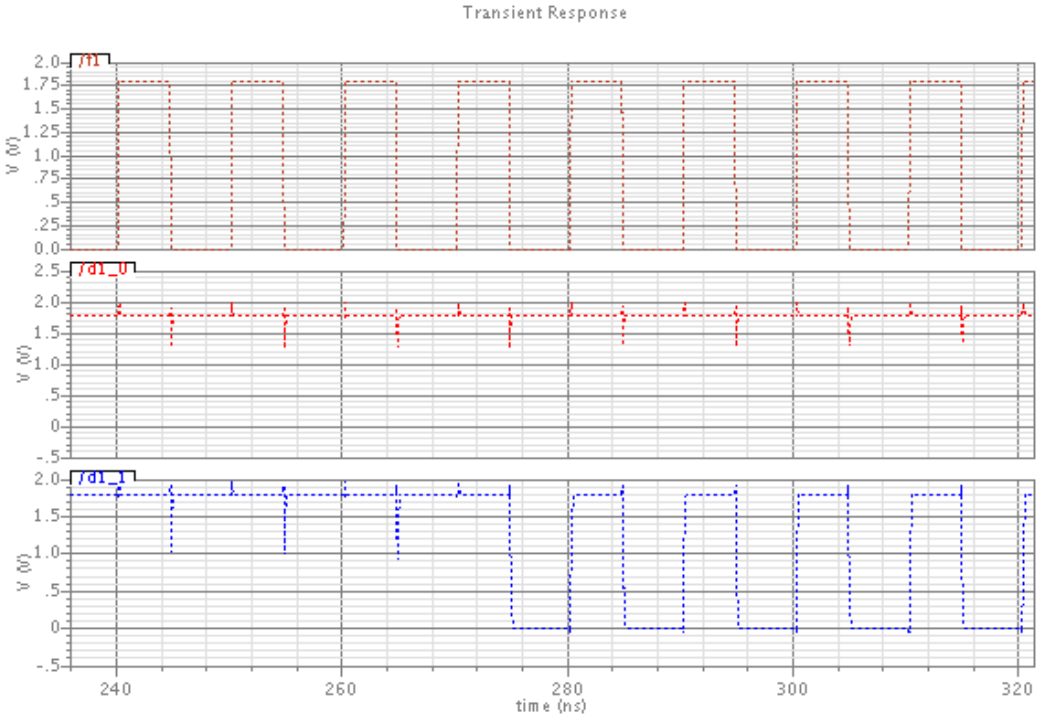


Fig 5.9 Close view for the first stage output

The output of the second stage is ready half clock cycle (i.e. 5ns) after the first stage's output being harvested. That means that we can collect the output of the second stage at time = 24ns, time = 34ns, and so on. Figure 5.9 shows the second stage's simulated output and its clock $\phi_2$. A close view in figure 5.10 shows the output readiness during the "hold" phase ($\phi_2$ low).

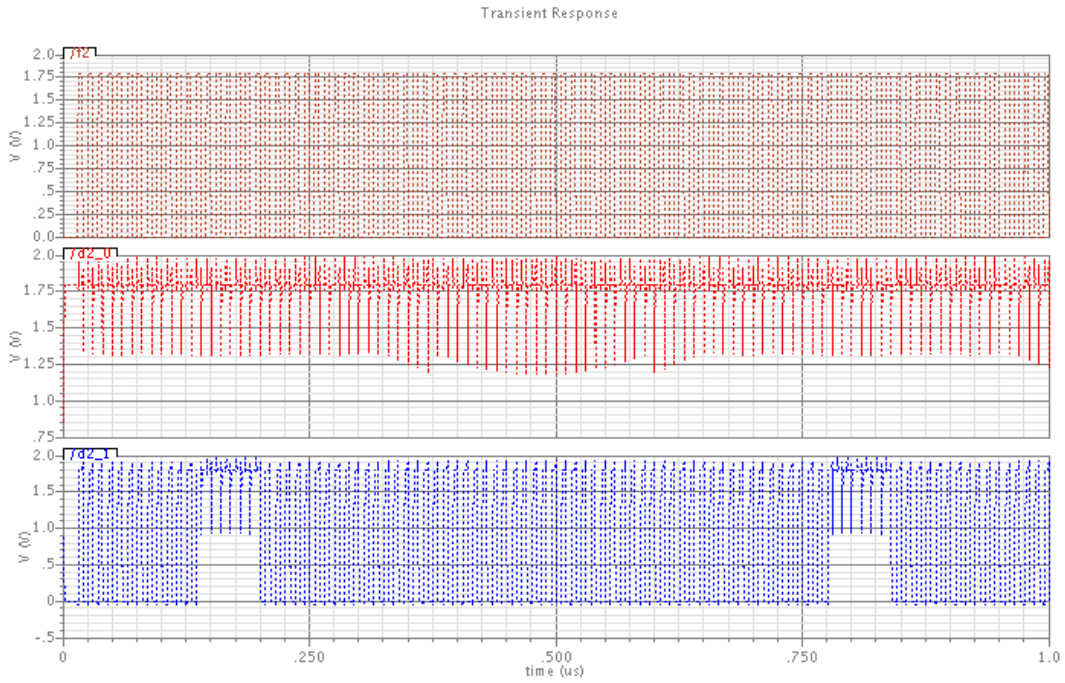


Fig. 5.10 Second stage output

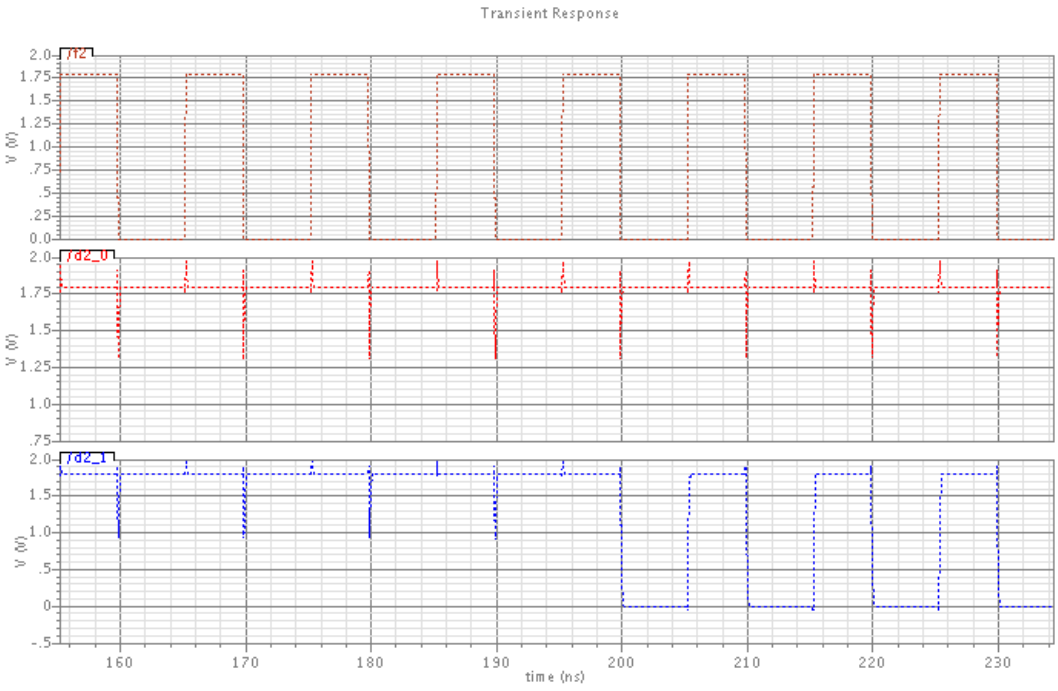


Fig 5.11 Close view for the second stage output

Similarly, the output of the third stage is available half clock cycle (i.e. 5ns) after the second stage's output being harvested. The third stage output is getting collected at time = 29ns, time = 39ns, and so on. The third stage output is showing in figure 5.11 and figure 5.12.

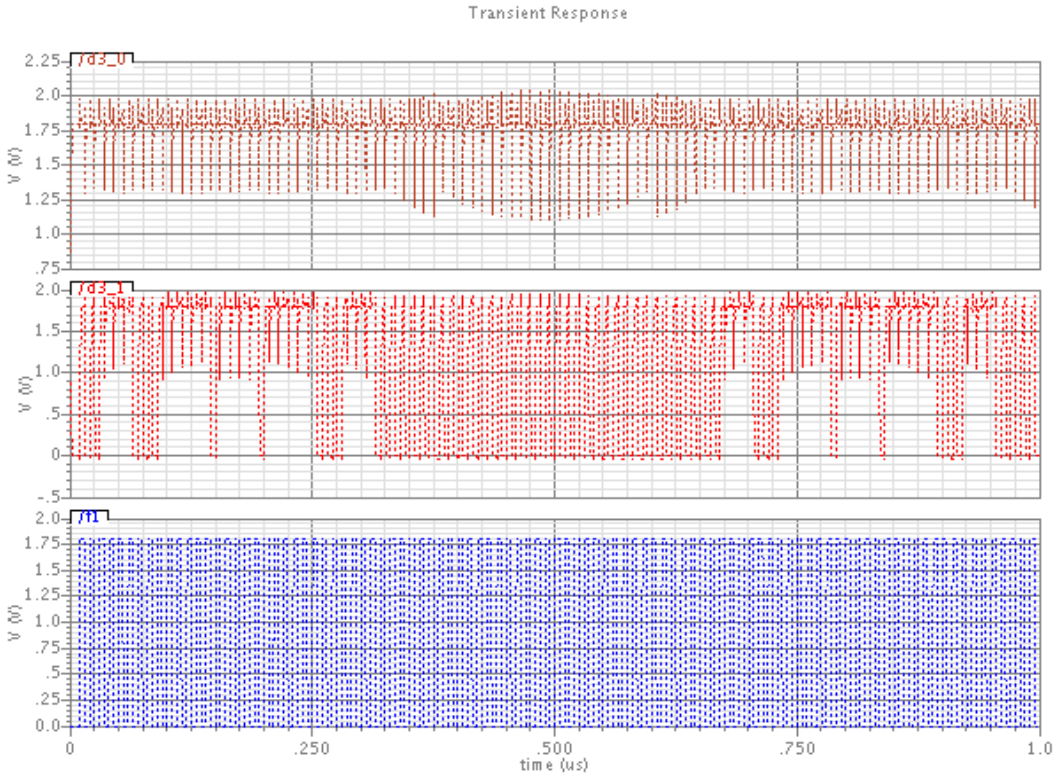


Fig. 5.12 Third stage output

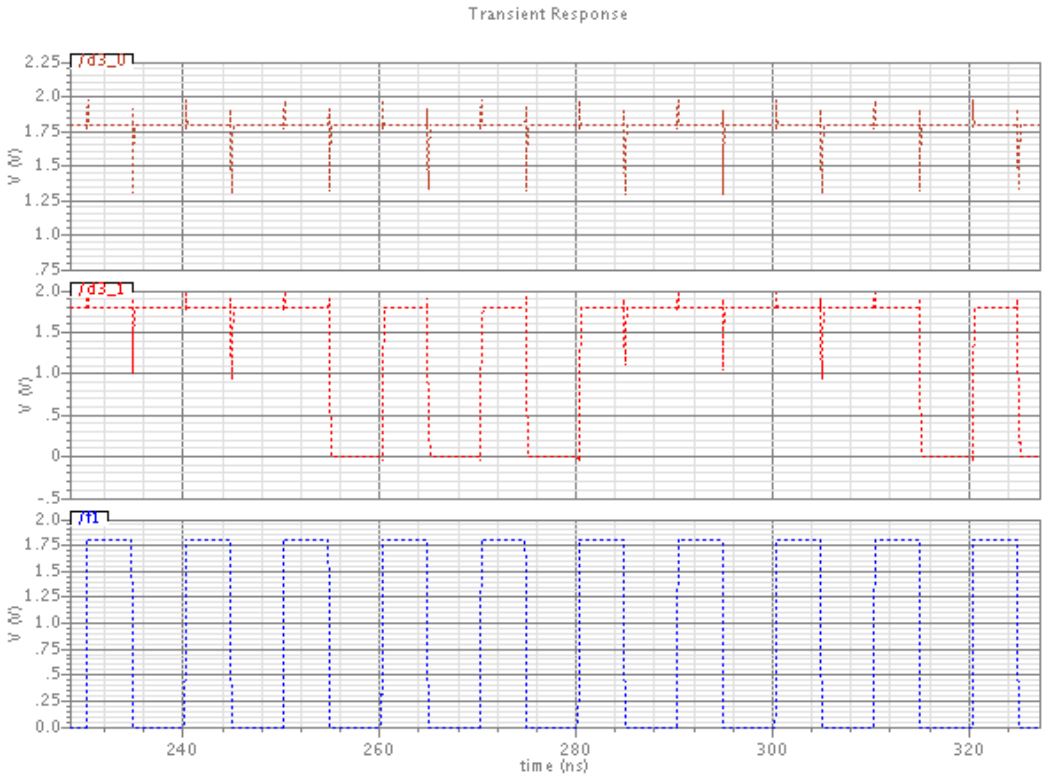


Fig 5.13 Close view for the third stage output

The rest of the stages' outputs along with their relevant clocks are showing in the figures below.

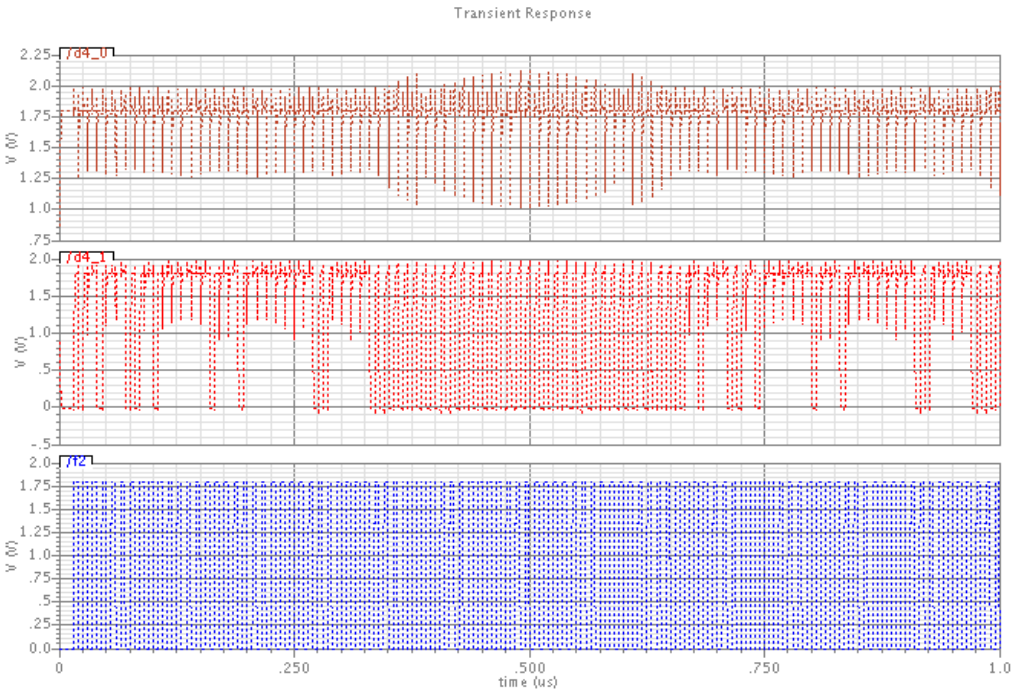


Fig. 5.14 Fourth stage output

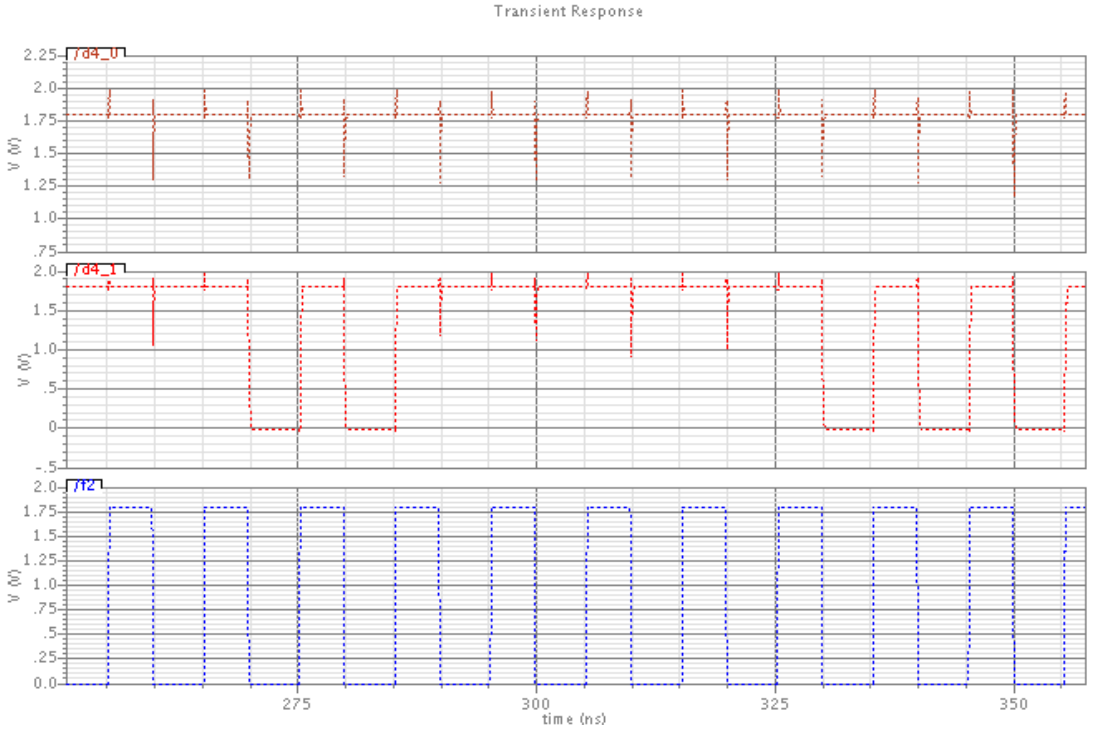


Fig. 5.15 Close view for fourth stage output

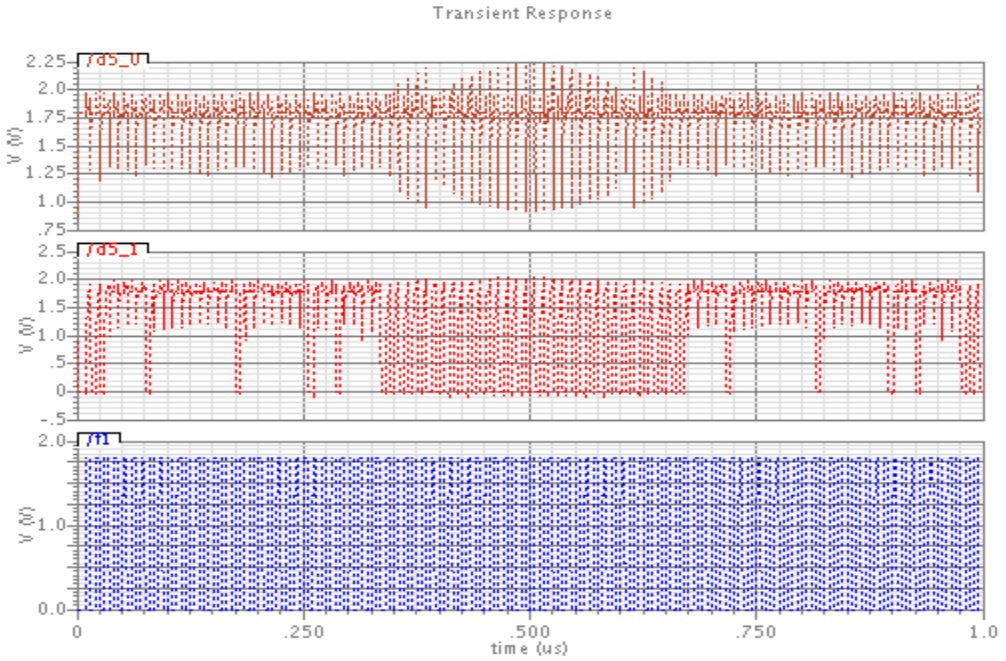


Fig. 5.16 Fifth stage output

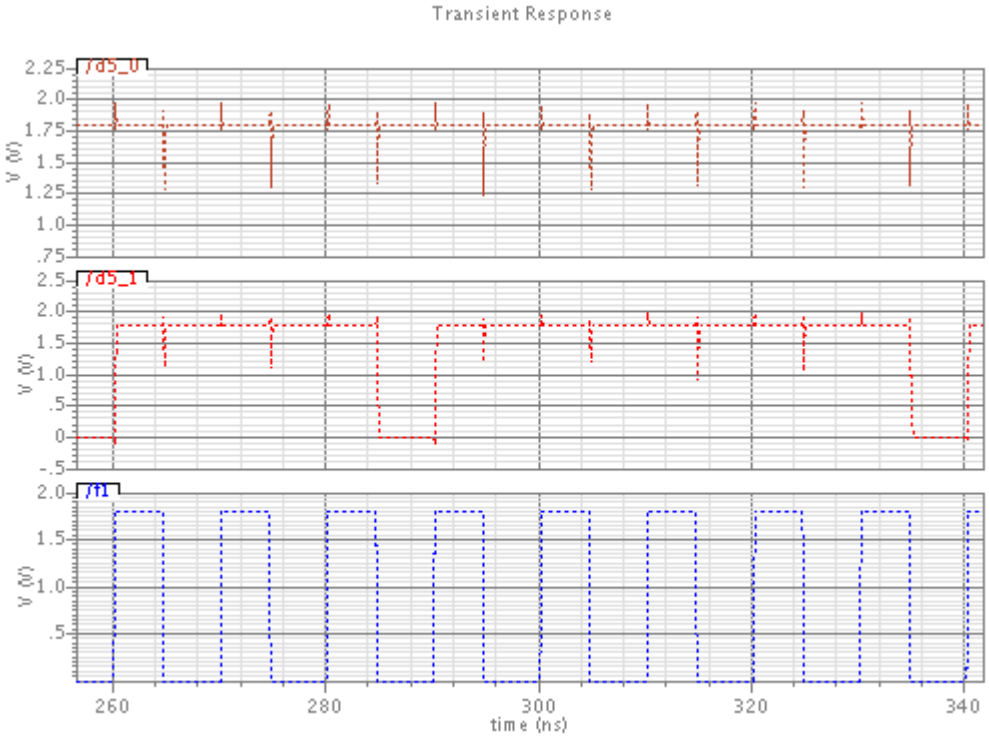


Fig. 5.17 Close view for fifth stage output

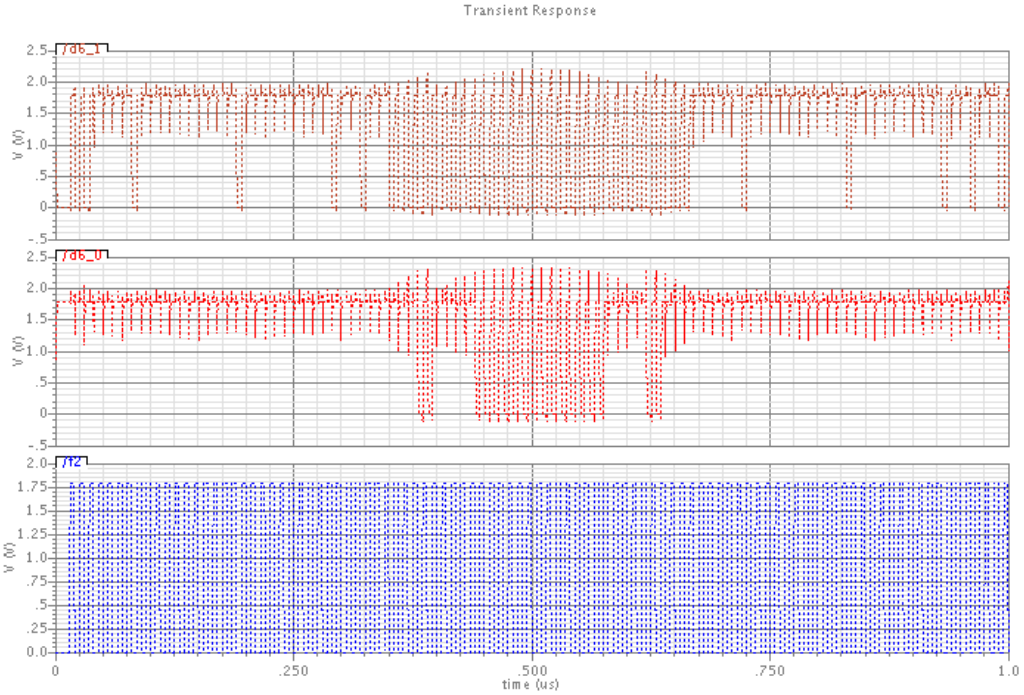


Fig. 5.18 Sixth stage output

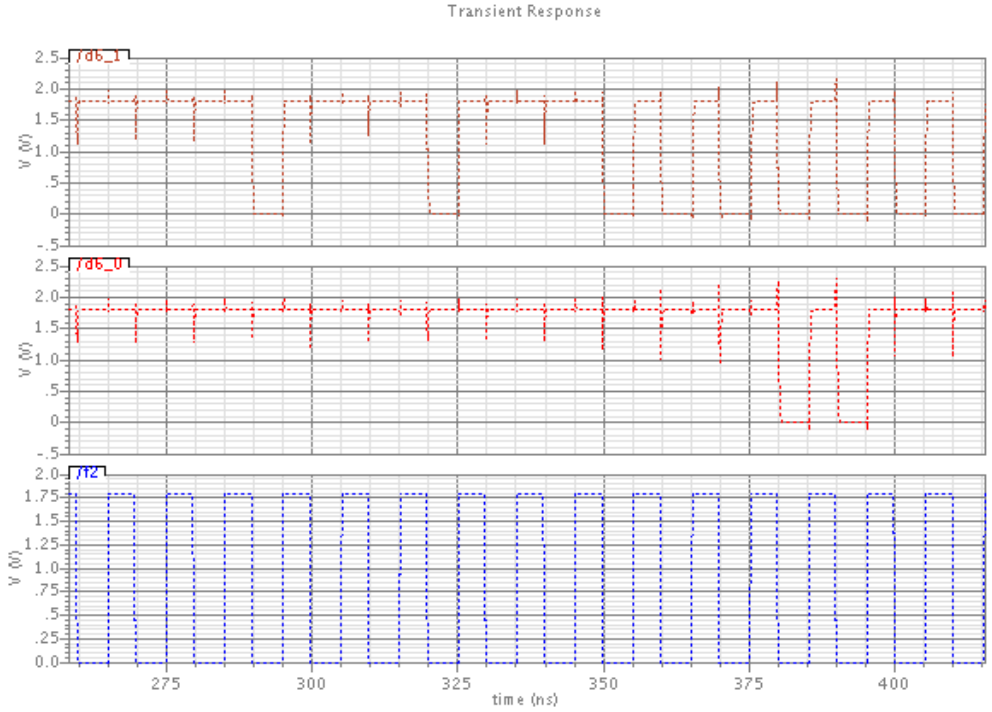


Fig. 5.19 Close view for sixth stage output

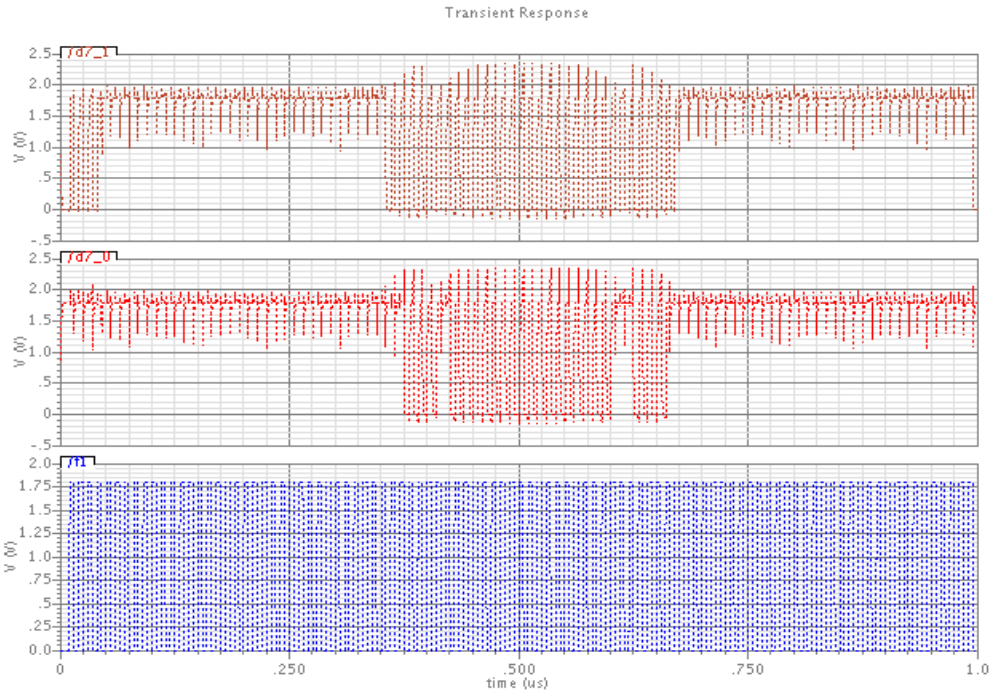


Fig. 5.20 Seventh stage output

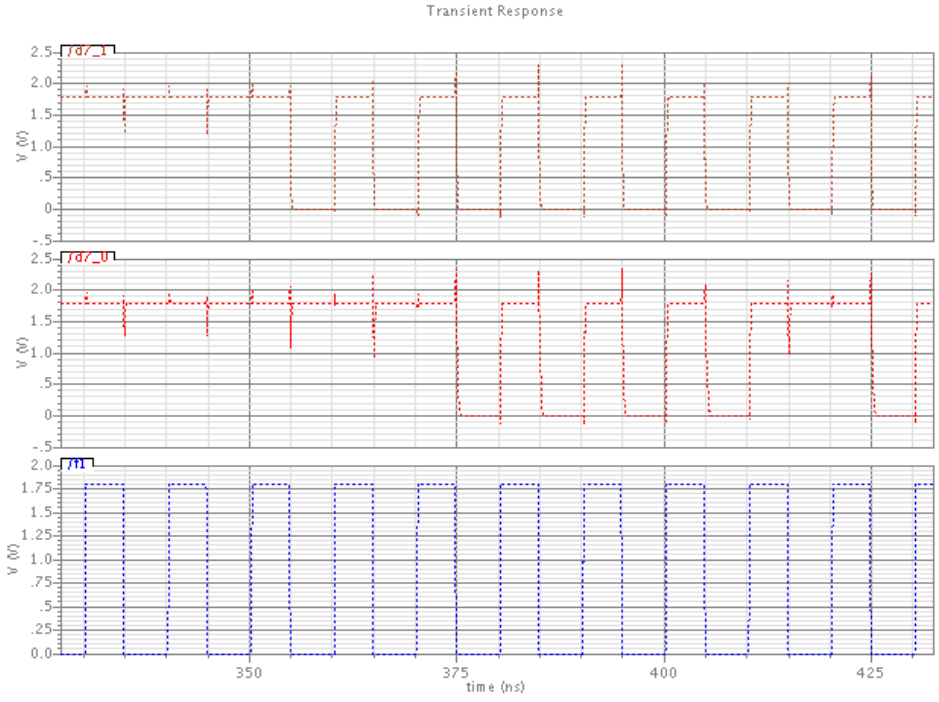


Fig. 5.21 Close view for seventh stage output

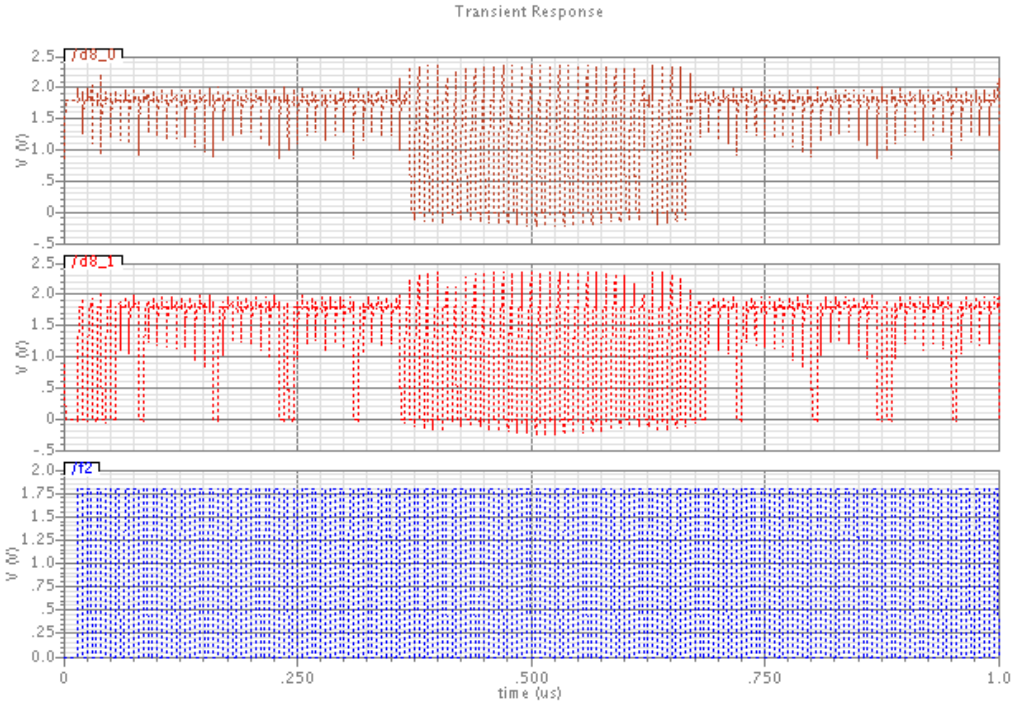


Fig. 5.22 Eighth stage output

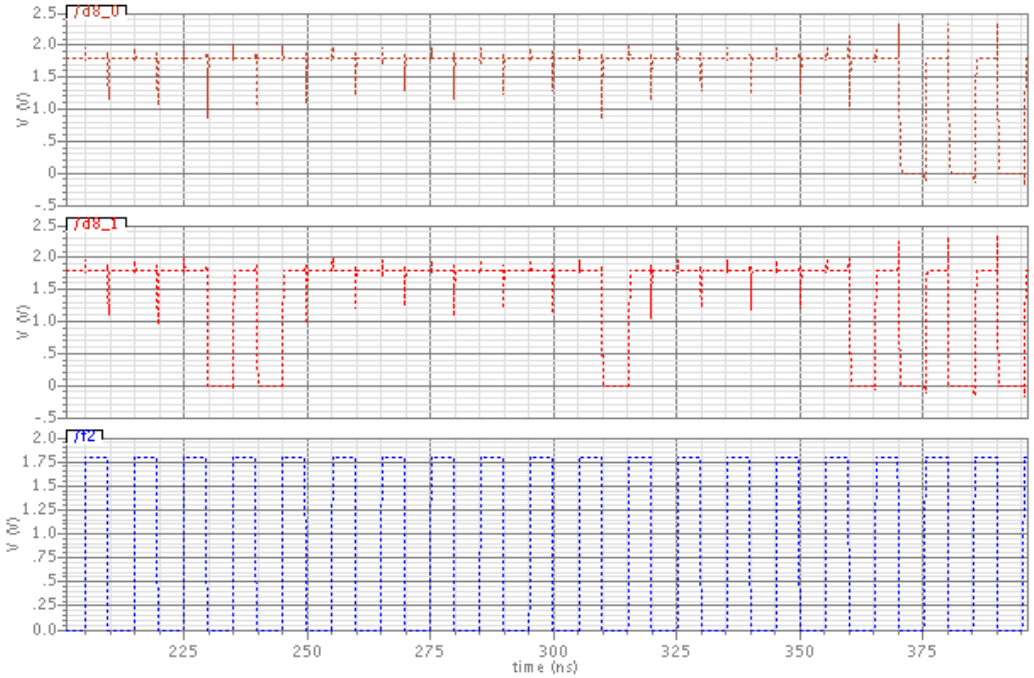


Fig. 5.23 Close view for eighth stage output

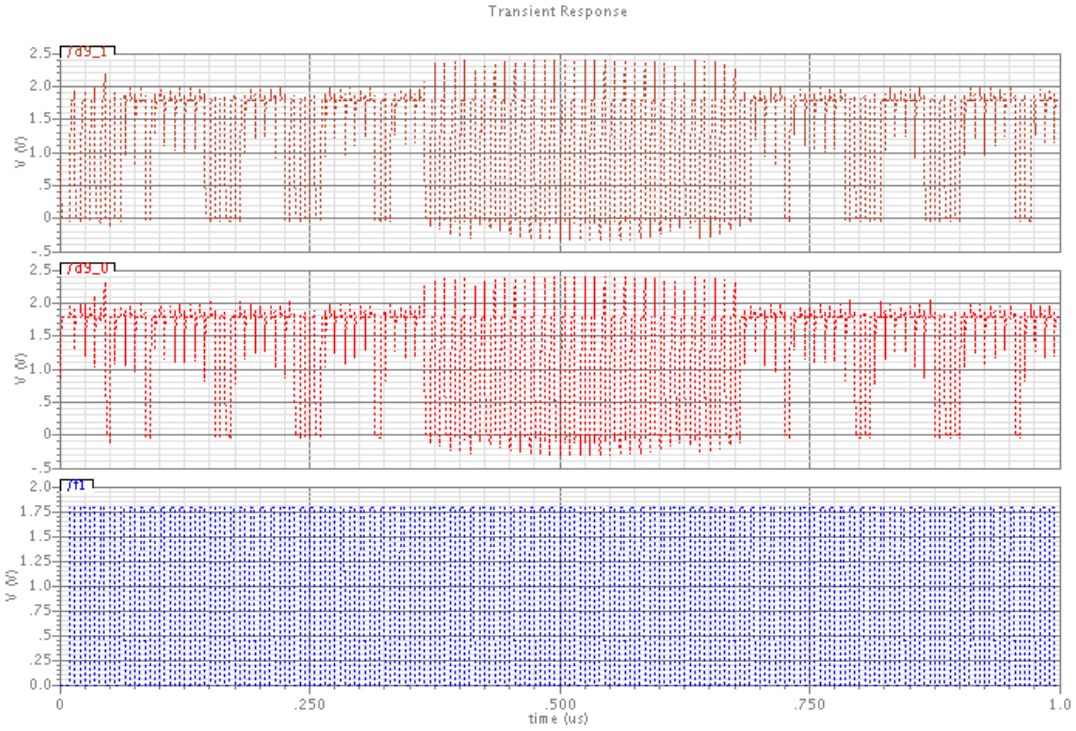


Fig. 5.24 Ninth stage output

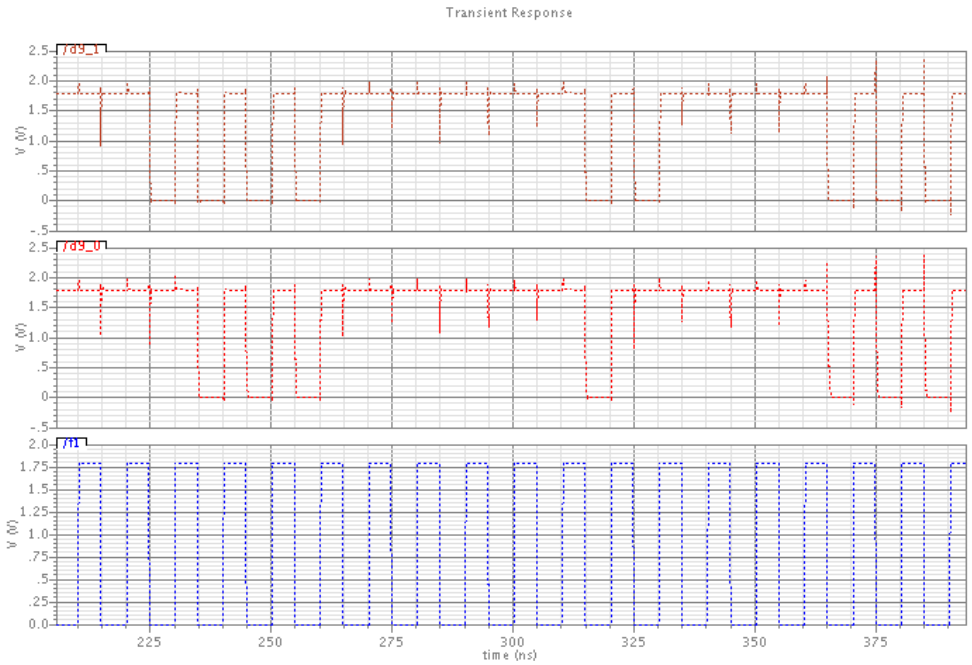


Fig. 5.25 Close view for ninth stage output

### 5.6.2 STAGE OUTPUT CONVERSION

Before doing the addition of the stages' outputs as we discussed earlier, we should first convert each stage two-comparator's output into a two bit binary number as follow:

| Comparators' Output | Binary Output |
|---|---|
| 00 | 00 |
| 01 | 01 |
| 11 | 10 |

Table 5.1 Stage Output Conversion

Then we perform the addition (as we showed in section 5.4) of the all "binary outputs" of the nine stages to get the final binary output. As an example, if we have 3-bit pipeline ADC with two stages, the combined 3-bit binary output possible values would be as follow.

| 1st Stage Binary Output | 2nd Stage Binary Output | Combined 3-bit Output |
|---|---|---|
| 00 | 00 | 000 |
| 00 | 01 | 001 |
| 00 | 10 | 010 |
| 01 | 00 | 010 |
| 01 | 01 | 011 |
| 01 | 10 | 100 |
| 10 | 00 | 100 |
| 10 | 01 | 101 |
| 10 | 10 | 110 |

Table 5.2 Combined output bits for 2-stage pipeline ADC

As we can see from the previous example, the final combined out of this 3-bit ADC never exceeds “110”, i.e. the outputs bits can’t be all “1’s”,  which means that we never run into an “over flow” situation as a results of summing up the different stages’ outputs. This also could be another way of explaining the “redundancy” that is implemented in the pipeline ADC.

## 5.7 CONCLUSION

The pipelined ADC is the architecture of choice for sampling rates from a few Msps up to 100Msps+. Design complexity increases only linearly (not exponentially) with the number of bits, thus providing converters with high speed, high resolution, and low power at the same time. Pipelined ADCs are very useful for a wide range of applications, most notably in digital communication where a converter's dynamic performance is often more important than traditional DC specifications like differential nonlinearity (DNL) and integral nonlinearity (INL). The data latency of pipelined ADCs is of little concern in most applications.

Table 5.4 summarizes the performances of the ADC measured at sampling rate of $f_s$ = 100MS/s.

| | |
|---|---|
| Technology | 0.18 μm CMOS process |
| Resolution | 10 bit |
| Sampling rate | 100 MS/s |
| Supply Voltage | 1.8V |
| Total Harmonic Distortions (THD) | -71.2dB @ (31/64) ×100MHz input (single stage) |
| Input referred noise | < 200 uV rms |

Table 5.3 Performance Summary of ADC

Throughout this report, we have successfully been able to:

1- Characterize the System-level functionality of a Pipeline ADC by simulating its different sub-block systems using ideal circuit components.
2- Design the majority of the analog subsystem of the ADC in Cadence.

3- Compare the behaviors of the ideal and the non-ideal circuit blocks by checking the compliance of the whole design with the linearity requirement after replacing each ideal circuit block with its transistor-level equivalent block.

## 5.8 FUTURE WORK

Some future work on the pipeline ADC would include implementing the “combine and align bits” block in transistor level design.

# Appendix A: Total Harmonic Distortion Algorithm

```
clear all;
Vdac=0;
N=64;

C=200e-15;
Cx=16e-15;

vin_check=zeros(1,64);
vdac_check=zeros(1,64);
v_constructed=zeros(1,64);
THD=zeros(1,40);
count=0;
for A=100:100:4000
    count = count + 1;
    for x=0:N-1
        vin=0.9+0.5*sin(2*pi*(31/64)*x);
        vin_check(1,x+1) = vin;
       if vin <= 0.7625
           Vdac = 0.4;
           vdac_check(1,x+1)=Vdac;
       elseif 0.7625 < vin && vin < 1.0125
           Vdac = 0.9;
           vdac_check(1,x+1)=Vdac;
       elseif vin >= 1.0125
           Vdac = 1.4;
           vdac_check(1,x+1)=Vdac;
       end
       v_constructed(1,x+1)=C*vin/(Cx/A+C*(1+2/A))+Vdac*(Cx/A+2*C/A)/(Cx/A+C*(1+2/A));
    end
    s=abs(fft(v_constructed));
    distortion=0;
    for i=2:31
        distortion = distortion + s(i).^2;
    end
    THD(1,count) = 10*log10(distortion/s(32).^2);
end

A = 100:100:4000;
figure(3)
plot(A,THD)
```

## Bibliography


[1] Understanding pipeline ADCs: their architecture and comparison with other ADCs. Maxim Integrated Tutorials at: *http://www.maximintegrated.com/app-notes/index. mvp /id/634*

[2] Abo A.M., Gray P.R. *"A 1.5-V, 10-bit, 14.3 MS/s CMOS pipeline analog-to digital converter," IEEE JSSC, Vol 34, May 1999, Page(s):599 – 606*

[3] Tadeparthy P., Das M. *"Techniques to improve linearity of CMOS Sample and-Hold circuits for achieving 100 dB performance at 80 Msps", IEEE Circuits and Systems, 2002*

[4] R. Jacob Baker. "*CMOS circuit design, layout, and simulation. Third edition. IEEE Series on Microelectronic Systems."*

[5] Paul R. Gray, Paul J. Hurst, Stephen H. Lewis, and Robert G. Meyer. *Analysis and design of analog integrated circuits. Wiley, 4th edition, 2001.*

[6] Nan Sun and Boris Murmann. *Data Converter Course Material.*

[7] B. Razavi. *Principles of data conversion system design. IEEE Press. 1995.*

[8] Nan Sun and Boris Murmann. "*Pipeline ADCs. Advanced Analog IC Design Course Material"*

[9] Steensgaard J. *"Bootstrapped low-voltage analog switches", IEEE Circuits and Systems*, 1999

[10] B. Murmann and B. Boser, *"A 12-bit 75MS/s pipelined ADC using open-loop residue amplification," IEEE Journal of Solid-State Circuits, vol. 38, no. 12, pp. 2040{2050, December 2003.*