Clock Edge Timing Adjustment Techniques for Correction of Timing Mismatches in Interleaved Analog-to-Digital Converters

by

Jason N. Shirtliff

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2010

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Time-interleaved analog-to-digital converters make use of parallelization to increase the rate at which an analog signal can be digitized. Using M channels at their maximum sampling frequency allows for an overall sampling frequency of M times the individual converters' sampling rate. However, the performance of interleaved systems suffers from mismatches between the sub-converters. Offset mismatches, gain mismatches, and timing mismatches all contribute to the degradation of the resolution of the ADC system.

Offset and gain mismatches can be corrected for in the digital domain with minimal extra processing. However, the effects of timing mismatches (specifically, the magnitude of the spurious tones that are introduced) are dependent on the frequency of the input, so digital correction is not a trivial task. This makes a circuit-based correction mechanism a much more desirable solution to the problem.

This work explores the effect of timing mismatches on interleaved analog-to-digital converter performance. A set of requirements is derived to specify the performance of a variable-delay circuit for the tuning of sample clocks. Since the mismatches can be composed of both fixed and random components, several candidate architectures are modeled for their delay and jitter performance. One candidate is selected for design, based on its jitter performance and on practical considerations.

A practical implementation of the clock-adjustment circuit is designed, featuring low-noise differential clock paths with high precision delay adjustment. A means of testing the circuit and verifying the precision of adjustment is presented. The design is implemented for fabrication, and post-layout simulations are shown to demonstrate the feasibility and functionality of the design.

Acknowledgements

Throughout the ups and downs of this project, numerous people have been kind enough to offer their support, advice, and encouragement. Without their help, this thesis would not have been possible.

First and foremost of these is my supervisor, Dr. David Nairn. His vast knowledge of circuits, from the broad concepts to the neat little tricks that make things just work, was invaluable. Many were the times where I descended upon his office, dejected about my inability to get the results I was looking for, and many were the times I left with a good objective viewpoint and something new to try. His ability to distill the important information from the noise saved me countless hours of work. Perhaps the most important thing he taught me was to use the simplest equations to come up with an educated guess, and then let the simulator do the hard work. We also shared a good few laughs about the insanity of the rest of the world from our viewpoint. Over the course of my degree, 273 banks have failed in the United States of America.

I would be remiss if I didn't give full credit to the countless pieces of advice I received over the course of this project from Dr. David Rennie. David is one of those fountains of knowledge in this department whose experience is invaluable in the mechanics of designing circuits and implementing them. There were times when I was at work well past midnight on my layout, and David was often around at that hour as well. He was surprisingly willing to answer questions at 3am. I wouldn't have finished the circuit without his assistance.

I'd like to thank my readers for their willingness to push their own work aside in order to pick my thesis apart and help me to put it back together in a much stronger fashion. Dr. William Bishop and Dr. Vincent Gaudet deserve all the thanks I can give them.

To the love of my life, Katherine Olsen, I'm finally finished and ready to start real life. Thanks for being patient and providing gentle encouragement to stick with this project.

To my parents and family, who I have sorely neglected for the past two years, thank you for your understanding and for your words of confidence and comfort throughout this journey. I promise I'm going to come home and visit more often now.

I would like to extend my congratulations to the friends of mine who managed to get out of here before I did (even if only by months or weeks): Bahman Hadji

and Adam Neale. Bahman, I'm coming out to join you in the real world. Adam, you're crazy for coming back yet again - finally our paths diverge.

I'd also like to wish the best to those friends and colleagues who are still toiling away: Zhao Li, Noman Hai, Jaspal Singh Shah, Pierce Chuang, and countless others. You all managed to help me in some small way during my time here.

One last recognition is due here: Kelly Martel, my high school electronics teacher. When I was in Grade 10, I told him, "It's nice that Ohm's Law works the way it does, but how can I use it to actually do something?" Kelly's response was, "You're going to be an engineer, aren't you?" While I can't give him all the credit for the path that led me here, Kelly gave me a starting point in circuits and a lot of other things in life for which I am truly indebted to him.

To the myriad who I haven't named here, please do not feel you were omitted because I didn't appreciate your contributions. I'm just trying to save a tree or two.

Dedication

In loving memory of my grandmother

Pauline Shirtliff

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Time-Interleaved Analog-to-Digital Converters

In increasingly many applications, the sample rates provided by state-of-the-art analog-to-digital converters (ADCs) are insufficient at the resolutions required. High-speed communication systems with optical or wireless signalling and RADAR systems are just two examples of these high-bandwidth applications. To keep up with these high-speed systems, one solution is to use time-interleaved ADCs, which take advantage of parallelization to increase the sample rate.

In these interleaved converters, two or more sub-ADCs are connected in parallel to the same input, and a clock generator sends sample signals to each converter in sequence, such that the next sub-ADC can begin conversion while the previous one is still finishing its conversion. In Figure 1.1, an interleaved system is shown where 4 sub-ADCs are used [1]. The sample clocks for each of the converters are shown in Figure 1.2. Theoretically, the overall sample rate is the maximum sample rate of one sub-ADC multiplied by the number of sub-ADCs in the system. However, there are many stringent requirements on the sub-ADCs that must be enforced, otherwise the performance of the interleaved system degrades significantly. Mismatches between the sub-ADCs and their separate signalling paths result in diminished resolution as well as limitations on the maximum speed of the system.

Therefore, time interleaved analog-to-digital converters require a high level of matching to achieve high resolution at high speed. Three main manifestations of mismatch exist: offset mismatch, gain mismatch, and timing mismatch. With an

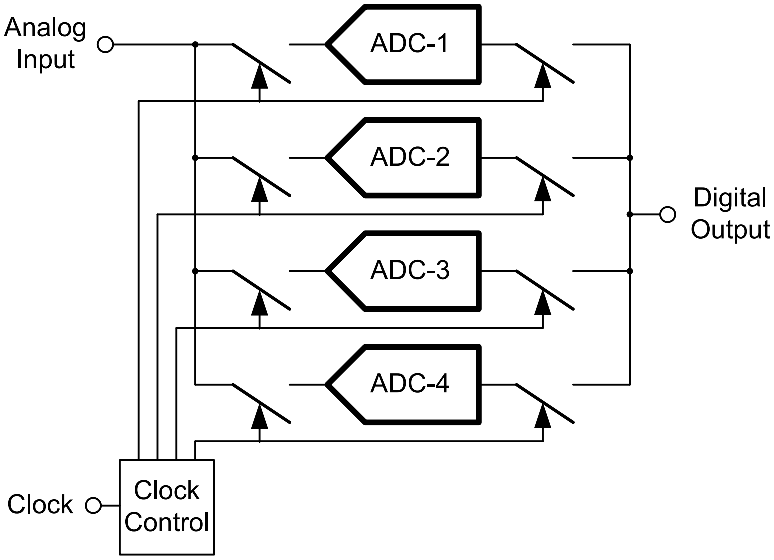


Figure 1.1: An interleaved ADC system with 4 channels

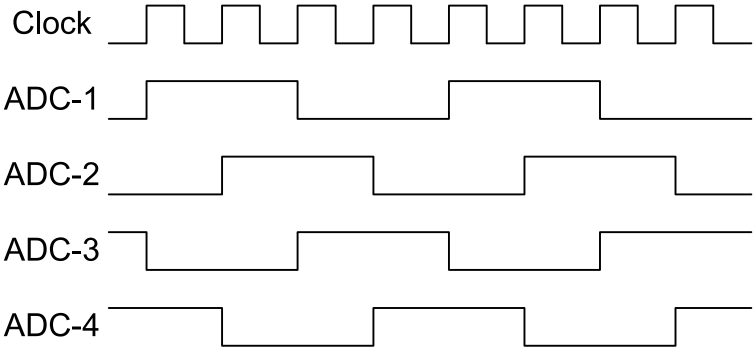


Figure 1.2: Sample timing of an interleaved ADC system with 4 channels

offset mismatch, the sub-ADCs have different zero levels, and therefore, different outputs for the same input, resulting in oscillation in the output for a constant input. With a gain mismatch, the sub-ADCs have different gains (slope of the analog-to-digital transfer function), resulting in spurious tones at certain frequencies related to the input frequency and the sampling frequency. Timing mismatches result in similar spurious tones, but their magnitudes are dependent on the input frequency.

Careful matching of the circuits and various DSP correcion algorithms have been able to minimize the impact of offset and gain mismatches. However, timing mismatches are more difficult to correct for, due to their frequency-dependent spurious tone magnitudes. This makes circuit-based correction much more desirable.

This work seeks to analyze the effects of timing mismatches in interleaved analog-to-digital converters and develop a circuit-based approach to correcting them, which involves adjusting the timing of the sample signals with a variable-delay circuit.

1.2 Organization

This thesis is divided up into four main sections. In Chapter 2, an overview of the effects of timing mismatches in interleaved analog-to-digital converters is presented. The symptoms of this mismatch are explained and the requirements on a circuit to alleviate these symptoms are derived. In Chapter 3, several different architectures for solving the timing mismatch problem are modeled and explored to determine suitability for implementation. The circuits are compared on their merits and one is selected for design, implementation, and fabrication for testing. In Chapter 4, the components of the selected architecture are designed at the block, and then the transistor level. Schematic simulations are provided to verify the design, and a testing methodology is presented. Finally, in Chapter 5, post-layout simulation results are presented to verify that the circuit can be tested and that it will provide results within some tolerance of the design requirements.

Chapter 2

Timing Skew Mismatches in Interleaved Analog-to-Digital Converters

2.1 Introduction

Sample time errors arise from both deterministic and random causes. In non-interleaved analog-to-digital converters (ADCs), the deterministic (or fixed) timing errors cause no problem for the system they are in because the error results in a uniform delay. However, in interleaved ADCs, when each sub-ADC has a different fixed delay, problems arise due to the timing mismatch between the channels. These sample time mismatches are often referred to as timing skews. They occur as a result of differing clock delays from the clock generator to each of the sub-ADCs, leading to sample intervals that are non-uniform. However, at the output, the samples are assumed to be ideal and accurate reconstruction of the wave cannot be done. The random sample time errors affect both interleaved and non-interleaved ADCs in much the same way and contribute to a reduction in the signal-to-noise ratio.

In this chapter, the effect of timing skews is examined visually and mathematically. The resulting error terms are evaluated to determine the effect of these mismatches on the resolution of the ADC. Finally, since clock jitter affects the system in a similar fashion, the effect of clock jitter is examined. Using the results of these analyses, a set of requirements is defined to guide the design of a variable clock skew buffer.

2.2 The Effect of Sample Time Mismatches

2.2.1 Analysis

To illustrate the effect of timing skew mismatches, consider a two-way interleaved ADC with identical gain and offset in its converters. It can be shown that a two-way interleaved ADC has the worst performance in terms of mismatches [2], because, with more converters, the mismatches inevitably average to less than the worst case.

If a full scale cosine input is applied to the ADC at a frequency f_i and the ADC operates at an overall sample rate of f_{clk}, the input can be expressed as

$$V_{in} = \frac{V_{FS}}{2} \cos\left(2\pi f_i t\right) \tag{2.1}$$

Ideally the sample period is $T_{clk} = 1/f_{clk}$ and the resulting samples appear as in Figure 2.1, where for illustration purposes $f_{clk} = 8f_i$. The samples taken by sub-ADC1 are marked with the symbol x and those taken by sub-ADC2 are marked with o. If a mismatch occurs in the sample timing between each sub-ADC, the average sample period remains equal to T_{clk}. However, the mismatch between the periods can be defined as the difference between the time of one sample plus the average sample period and the time of the subsequent sample on the other sub-ADC, which can be expressed as

$$\Delta T = |T_1 + T_{clk} - T_2| \tag{2.2}$$

Using this definition for the mismatch, then without loss of generality it can be divided equally between the two sub-ADCs. Hence, the sample intervals for sub-ADC1 occur at $0 + \frac{\Delta T}{2}$, $2T_{clk} + \frac{\Delta T}{2}$, $4T_{clk} + \frac{\Delta T}{2}$, ... and the sample intervals for sub-ADC2 occur at $T_{clk} - \frac{\Delta T}{2}$, $3T_{clk} - \frac{\Delta T}{2}$, $5T_{clk} - \frac{\Delta T}{2}$, ... The resulting samples appear as in Figure 2.2, where again the samples taken by ADC1 are marked with x and the samples taken by ADC2 are marked with o.

At the output of the ADC, any timing information regarding the samples is unknown, and the samples are assumed to be ideally spaced in time. Hence, on reconstruction of the wave at the output, distortion occurs.

To determine the effects of this distortion, first consider the output of each of the sub-ADCs individually. ADC1's output appears as a cosine wave with a full-

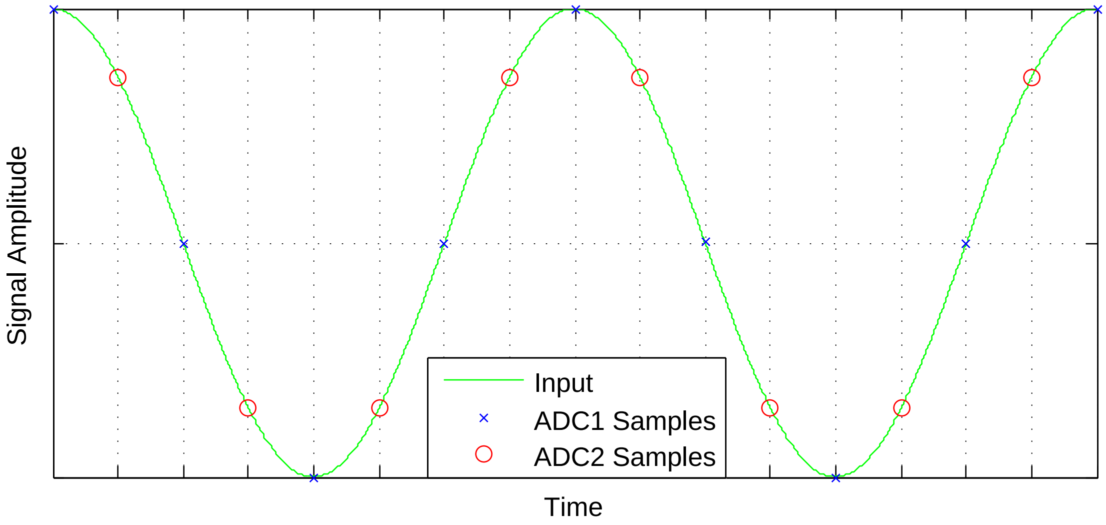


Figure 2.1: Input cosine wave with ideal interleaved sample intervals

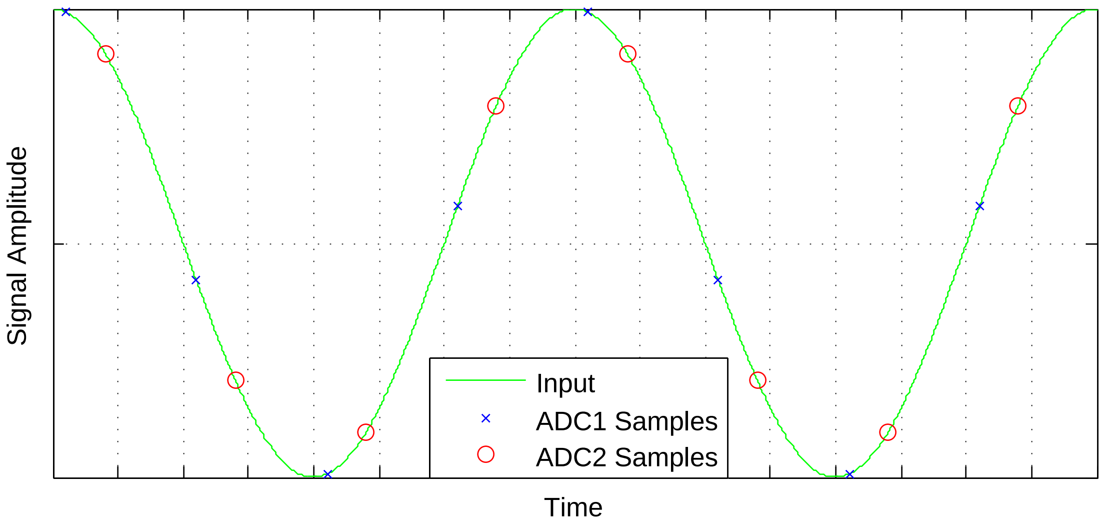


Figure 2.2: Input cosine wave with skewed interleaved sample intervals

scale magnitude and a positive phase shift $\frac{\Delta\phi}{2}$, while ADC2's output appears as a cosine wave with a full-scale magnitude and a negative phase shift $-\frac{\Delta\phi}{2}$.

$$V_{out1} = \frac{V_{FS}}{2} \cos\left(2\pi f_i t + \frac{\Delta\phi}{2}\right) \tag{2.3}$$

$$V_{out2} = \frac{V_{FS}}{2} \cos\left(2\pi f_i t - \frac{\Delta\phi}{2}\right) \tag{2.4}$$

These shifted waves are shown in Figure 2.3. The phase shift in each case is related to the original timing mismatch, ΔT, by the following relationship:

$$\Delta\phi = 2\pi f_i \Delta T \tag{2.5}$$

When the sub-ADC outputs are interleaved, the final output contains two components. The first is the dominant signal, which is the original input cosine wave with a slight reduction in magnitude. The second is an error term that manifests itself as a sine wave modulated by a frequency of $\frac{f_{clk}}{2}$ caused by the oscillation back and forth between each of the interleaved outputs.

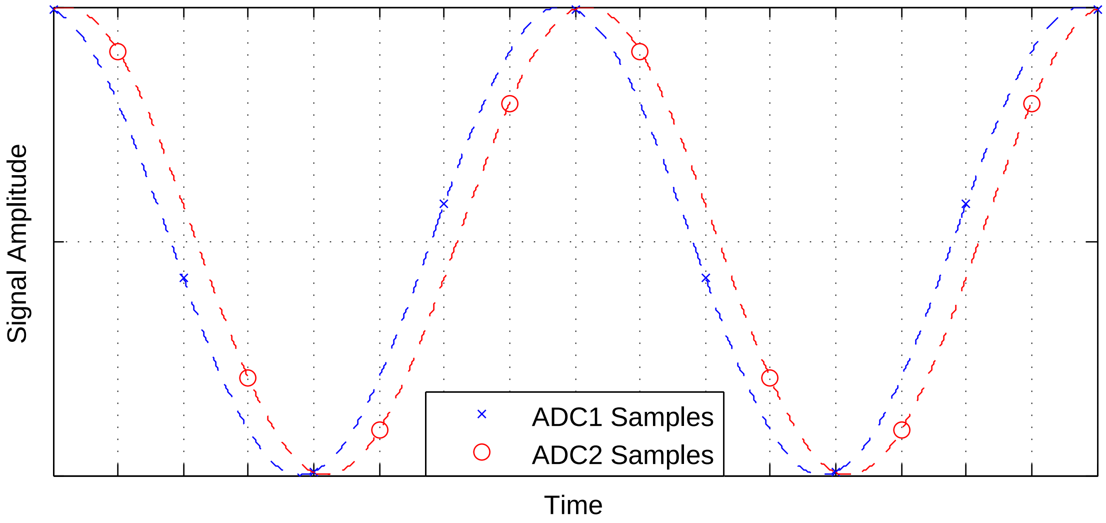


Figure 2.3: Outputs of each sub-ADC before recombination

Mathematically, the first output term is derived by taking the average of the two outputs.

$$V_{out\,av} = \frac{1}{2}\left(\frac{V_{FS}}{2}\cos\left(2\pi f_i t + \frac{\Delta\phi}{2}\right) + \frac{V_{FS}}{2}\cos\left(2\pi f_i t - \frac{\Delta\phi}{2}\right)\right)$$

$$V_{out\,av} = \frac{V_{FS}}{4}\left(\cos\left(2\pi f_i t + \frac{\Delta\phi}{2}\right) + \cos\left(2\pi f_i t - \frac{\Delta\phi}{2}\right)\right) \qquad (2.6)$$

Using the cosine identity $\cos(u \pm v) = \cos\, u\, \cos\, v \mp \sin\, u\, \sin\, v$, Equation (2.6) can be manipulated as follows:

$$V_{out\,av} = \frac{V_{FS}}{4}\begin{pmatrix} \cos(2\pi f_i t)\cos\left(\frac{\Delta\phi}{2}\right) \\ -\sin(2\pi f_i t)\sin\left(\frac{\Delta\phi}{2}\right) \\ +\cos(2\pi f_i t)\cos\left(\frac{\Delta\phi}{2}\right) \\ +\sin(2\pi f_i t)\sin\left(\frac{\Delta\phi}{2}\right) \end{pmatrix}$$

$$V_{out\,av} = \frac{V_{FS}}{2}\cos(2\pi f_i t)\cos\left(\frac{\Delta\phi}{2}\right) \qquad (2.7)$$

Since $\cos\left(\frac{\Delta\phi}{2}\right) = \cos(\pi f_i \Delta T)$ and $\pi f_i \Delta T$ is small, this reduces to the input signal.

$$V_{out\,av} \approx \frac{V_{FS}}{2}\cos(2\pi f_i t) \qquad (2.8)$$

The bound for the error term is derived by taking the difference between the two outputs.

$$V_{out\,err\_bound} = \frac{V_{FS}}{2}\cos\left(2\pi f_i t + \frac{\Delta\phi}{2}\right) - \frac{V_{FS}}{2}\cos\left(2\pi f_i t - \frac{\Delta\phi}{2}\right)$$

$$V_{out\,err\_bound} = \frac{V_{FS}}{2}\left(\cos\left(2\pi f_i t + \frac{\Delta\phi}{2}\right) - \cos\left(2\pi f_i t - \frac{\Delta\phi}{2}\right)\right) \qquad (2.9)$$

Using the same cosine identity as above,

$$V_{out\,err\_bound} = \frac{V_{FS}}{2}\begin{pmatrix} \cos(2\pi f_i t)\cos\left(\frac{\Delta\phi}{2}\right) \\ -\sin(2\pi f_i t)\sin\left(\frac{\Delta\phi}{2}\right) \\ -\cos(2\pi f_i t)\cos\left(\frac{\Delta\phi}{2}\right) \\ -\sin(2\pi f_i t)\sin\left(\frac{\Delta\phi}{2}\right) \end{pmatrix}$$

$$V_{out\,err\_bound} = -V_{FS}\sin(2\pi f_i t)\sin\left(\frac{\Delta\phi}{2}\right) \qquad (2.10)$$

The error itself appears as a sine wave modulated by half the clock frequency. This is because the output signal alternates between the two different converters in an ideal series of impulses of value 1, -1, 1, -1... spaced $\frac{1}{f_{clk}}$ apart. Therefore, one period of this impulse series would be equal to two periods of the clock frequency. The error is modeled as an amplitude modulated signal between this sinusoid impulse series and the bound.

$$V_{out\,err} = -V_{FS} \sin\left(\frac{\Delta\phi}{2}\right) \sin\left(2\pi f_i t\right) \sin\left(2\pi \frac{f_{clk}}{2}\right) \tag{2.11}$$

Using the sine identity $\sin u \sin v = \frac{1}{2}\left(\cos\left(u - v\right) - \cos\left(u + v\right)\right)$, the error term can be reduced to two tones whose magnitude is determined by the size of the phase skew.

$$V_{out\,err} = \frac{-V_{FS}}{2} \sin\left(\frac{\Delta\phi}{2}\right) \left(\begin{array}{c} \cos\left(2\pi\left(f_i - \frac{f_{clk}}{2}\right)t\right) \\ - \cos\left(2\pi\left(f_i + \frac{f_{clk}}{2}\right)t\right) \end{array}\right) \tag{2.12}$$

The total output is the sum of the average output plus the error term.

$$V_{out} = \frac{V_{FS}}{2} \left(\cos\left(\frac{\Delta\phi}{2}\right) \cos\left(2\pi f_i t\right) - \sin\left(\frac{\Delta\phi}{2}\right) \left(\begin{array}{c} \cos\left(2\pi\left(f_i - \frac{f_{clk}}{2}\right)t\right) \\ - \cos\left(2\pi\left(f_i + \frac{f_{clk}}{2}\right)t\right) \end{array}\right)\right) \tag{2.13}$$

This output is plotted in the frequency domain in Figure 2.4. The figure also contains the alias at $f_{clk} - f_i$ that is a result of the nature of a sampled-data system. Examining one Nyquist Zone, from $f = 0$ to $f = \frac{f_{clk}}{2}$, reveals that there is a strong tone at $f = f_i$ and a weaker, spurious tone at $f = \frac{f_{clk}}{2} - f_i$ that is introduced by the timing mismatch.

2.2.2 ADC Resolution

The spurious-free dynamic range (SFDR) of an ADC limits the resolution of the ADC. The spurious tones produced by the timing mismatch result in a reduction in the SFDR and hence the resolution of the ADC. The SFDR of a system is defined as the magnitude of the signal at the output divided by the magnitude of the worst case spurious tone.

$$SFDR = \frac{|V_{sig}|}{|V_{spur}|} \tag{2.14}$$

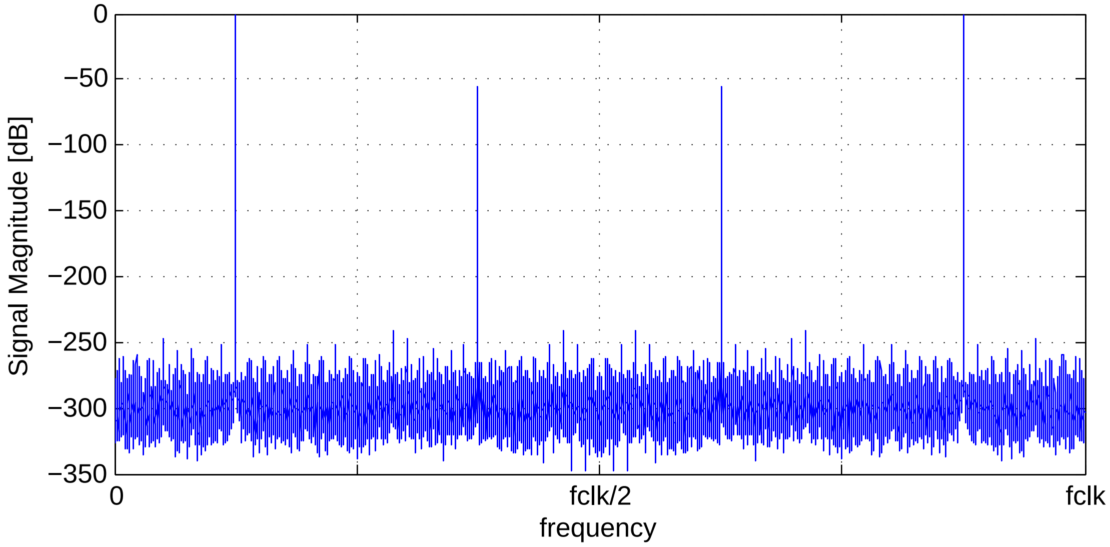


Figure 2.4: Frequency domain output of an ADC with time skews

In the case of an interleaved ADC, the signal and spurious tones conform to those derived above. From the previous derivation, the magnitude of the signal tone at the output is

$$|V_{sig}| = \frac{V_{FS}}{2}\cos\left(\frac{\Delta\phi}{2}\right) \tag{2.15}$$

and the magnitude of the spurious tone is

$$|V_{spur}| = \frac{V_{FS}}{2}\sin\left(\frac{\Delta\phi}{2}\right) \tag{2.16}$$

Hence, the SFDR for a two-way interleaved ADC is

$$SFDR = \frac{\frac{V_{FS}}{2}\cos\left(\frac{\Delta\phi}{2}\right)}{\frac{V_{FS}}{2}\sin\left(\frac{\Delta\phi}{2}\right)}$$

$$SFDR = \frac{\cos\left(\frac{\Delta\phi}{2}\right)}{\sin\left(\frac{\Delta\phi}{2}\right)} \tag{2.17}$$

For very small phase shifts and using the relationship between the phase shift and the timing mismatch, Equation (2.5), the SFDR can be written in terms of the absolute time skew, ΔT.

$$SFDR \approx \frac{1}{\frac{\Delta\phi}{2}}$$

$$SFDR \approx \frac{1}{\pi f_i \Delta T} \tag{2.18}$$

It is of interest to note that the SFDR is dependent on the input frequency. Thus, the resolution of the ADC is signal-dependent and the errors due to timing skews are worse at higher input frequencies. It is also of interest to note that the SFDR is not related to the sampling frequency, which may be counterintuitive.

2.2.3 The Effect of Clock Jitter in Interleaved ADCs

The errors in the clock edge associated with jitter are random and vary from cycle-to-cycle. The average of these errors is zero, but the noise in the output associated with clock jitter must be accounted for.

The change in sampled voltage associated with a change in sample time will be directly related by the slope of the input signal at that time. For the same input signal as in Equation (2.1), the slope of the input is

$$\frac{dV_{in}}{dt} = -V_{FS}\pi f_i \sin\left(2\pi f_i t\right) \tag{2.19}$$

The maximum possible error in voltage is associated with the maximum slope of the input signal.

$$\frac{dV_{in}}{dt}|_{max} = V_{FS}\pi f_i \tag{2.20}$$

This can be rewritten in the form

$$\Delta V_{in} = V_{FS}\pi f_i t_j \tag{2.21}$$

where t_j is the root-mean-square (RMS) jitter of the clock and ΔV_{in} is the corresponding RMS error that translates to the output of the ADC as a result of the sample jitter. This change in voltage at the output corresponds to an output noise voltage that reduces the signal-to-noise ratio (SNR) of the ADC, which in turn reduces the effective resolution of the ADC.

The SNR of an ADC, when accounting for only noise contributed by jitter, can be expressed as the ratio of the magnitude of the input signal to the magnitude of the jitter noise:

$$\begin{aligned} SNR &= \frac{\frac{V_{FS}}{2\sqrt{2}}}{V_{FS}\pi f_i t_j} \\ SNR &= \frac{1}{2\pi f_i t_j \sqrt{2}} \end{aligned} \tag{2.22}$$

It is clear that any increase in the jitter decreases the SNR of the ADC.

2.3 Requirements on Timing Matching

The requirement on the precision of the clock edges in an interleaved ADC comes from the SFDR and the desired resolution for the ADC. It is customary to reduce the magnitude of the spurious tones below the quantization noise, $\frac{V_{LSB}}{\sqrt{12}}$, where $V_{LSB} = \frac{V_{FS}}{2^N}$ and N is the resolution of the ADC.

$$\frac{V_{FS}}{2} \sin\left(\pi f_i \Delta T\right) \leq \frac{V_{FS}}{2^N \sqrt{12}} \tag{2.23}$$

Solving for the timing mismatch yields

$$\frac{\Delta T}{2} \leq \frac{1}{\pi f_i 2^N \sqrt{12}} \tag{2.24}$$

Assuming that the ADC is sampling the maximum frequency allowed (Nyquist sampling), the input frequency is $f_i = \frac{f_{clk}}{2} = \frac{1}{2T}$ and Equation (2.24) can be rewritten as

$$\frac{\Delta T}{T} \leq \frac{4}{\pi 2^N \sqrt{12}} \tag{2.25}$$

If the time skew in the ADC is less than or equal to this value, then the dynamic range will be greater than N bits. Figure 2.5 shows the maximum input frequency possible for an ADC with a given resolution for various sample timing mismatches. Table 2.1 provides some examples of the values of ΔT for given Nyquist input

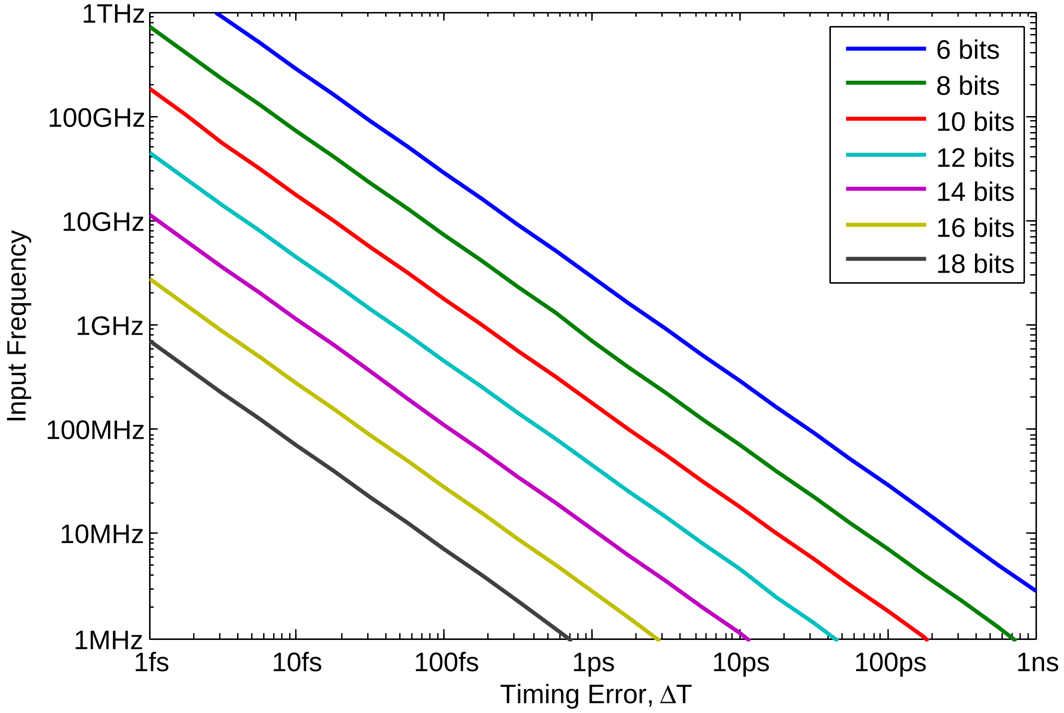


Figure 2.5: The maximum input frequency that is measurable with an interleaved ADC of a fixed resolution suffering a time error

frequencies and resolutions that can be considered state-of-the-art. For state-of-the-art interleaved systems, timing mismatch is required to be on the order of $\Delta T \approx 25\,\text{fs}$.

Using the common convention of ensuring that the magnitude of the output voltage noise due to clock jitter is below the quantization noise, $\frac{V_{LSB}}{\sqrt{12}} = \frac{V_{FS}}{2^N\sqrt{12}}$, the requirement on the jitter can be expressed in terms of the input frequency and the number of bits, starting from Equation (2.21). This gives

$$\frac{V_{FS}}{2^N\sqrt{12}} \geq V_{FS}\pi f_i t_j \tag{2.26}$$

Solving for the jitter, t_j gives the result

$$t_j \leq \frac{1}{\pi f_i 2^N \sqrt{12}} \tag{2.27}$$

which gives an upper bound for the RMS clock jitter at the sampling circuit. Noting the similarity between Equation (2.24) and Equation (2.27), Figure 2.5 also shows the maximum input frequency possible for an ADC with a given resolution for various sample timing errors. Table 2.1 provides some examples of the values of t_j

for given Nyquist input frequencies and resolutions that can be considered to be state of the art. It is notable that the effect of timing mismatch and the jitter are different by a factor of two. Consider that the timing mismatch is a value that can be positive or negative while the jitter is an RMS value which can only be a positive value.

Table 2.1: Practical requirements for maximum timing mismatch and jitter to achieve state of the art resolutions and sampling rates

N	f_{clk}	f_i	ΔT	t_j
16	100 MS/s	50 MHz	56.1 fs	28.0 fs
12	1.8 GS/s	900 Mhz	49.8 fs	24.9 fs
8	28 GS/s	14 GHz	51.3 fs	25.6 fs

2.4 Summary

Timing errors degrade the performance of interleaved ADCs. Timing skews between channels lead to spurious tones, which decreases the SFDR. Timing jitter increases the noise, which reduces the SNR. To obtain acceptable ADC performance in an interleaved ADC, both of these timing errors must be tightly controlled.

Chapter 3

Methods of Selectively Skewing Clock Edges

3.1 Introduction

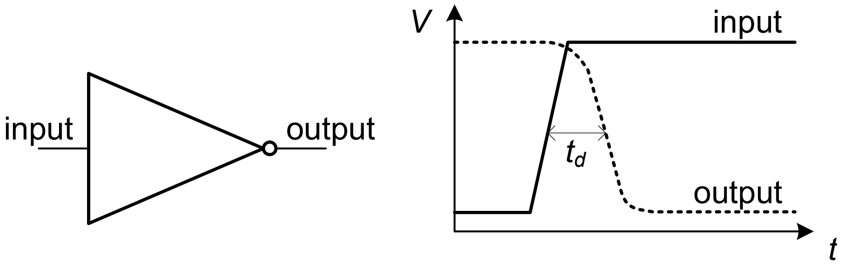
To solve the problem of timing mismatches in interleaved ADCs using circuit techniques, it is necessary to develop a circuit that allows for the skewing of a clock edge. Evidence shows that the maximum range over which clock edges must be skewed is on the order of hundreds of picoseconds. Poulton et. al. [3] cite a typical timing deviation of 50 ps and Conroy et. al. [4] report results showing 25 ps of timing mismatch, while Yuan and Svensson [5] claim a 500 ps sampling time error. In advanced CMOS processes, these deviations correspond to multiple gate delays. The previous chapter suggests that precision on the order of tens of femtoseconds, corresponding to fractions of a gate delay, are required. Furthermore, very low jitter is required to avoid degradation of the ADC's SNR. These requirements dictate the specifications of the clock adjustment circuit.

Several architectures exist that achieve the skewing of a clock edge by large amounts with fine precision. Therefore, it is useful to compare these circuit structures in terms of their jitter performance to select an optimal solution.

To enable a comparison of the different structures, this chapter introduces a basic model for the delay and jitter of a delay element. The model is then elaborated to include the effects of varying the delay, forming a complete model for the jitter of a variable delay buffer. Then, three potential solutions that achieve the required timing adjustment specifications are presented. These architectures are analyzed for their delay and jitter characteristics, first using the basic model and then using the complete model. Conclusions are made about their comparative performance, and using these conclusions, one of the architectures is selected for implementation.

3.2 Analysis of Delay Architectures

To create a foundation for the models on which the analysis of this chapter is based, consider the simple inverter, shown in Figure 3.1(a). As a rising edge is applied to the input, a falling edge is produced at its output, as illustrated in Figure 3.1(b).



(a)

(b)

Figure 3.1: (a) An inverter (b) For an input rising edge, and inverter's output falls after some intrinsic delay time, t_d

There is some propagation delay intrinsic to the gate, the majority of which is a result of the time required to charge or discharge the capacitance at the output of the gate. Therefore, the delay of the gate (t_d) is approximately proportional to the inverse of the slew rate (the rate of change of the output voltage with respect to time) at its output:

$$t_d \propto \frac{1}{SR} \tag{3.1}$$

The RMS jitter (t_j) at the output of the gate is approximately equal to the noise voltage at the output (v_n) divided by the slope of the output signal at the transition point. The slope of the output signal at the transition voltage is the slew rate. Therefore, the jitter is:

$$t_j \approx \frac{v_n}{SR} \tag{3.2}$$

Using the above equations, the jitter is written as a function of the delay as follows:

$$t_j \propto v_n t_d \tag{3.3}$$

The definition of the relationship between the jitter and the delay in Equation (3.3) is valid for any gate structure and forms a basis for the comparison between three different architectures for adjusting the edge of a clock signal.

3.2.1 Methods of Achieving Fractional Gate Delay Adjustments

The output of a buffer or an inverter can be modeled as a resistive-capacitive (RC) circuit driven by either a voltage or current source. To create a variable delay, any one of the three different elements in the RC output circuit can be adjusted to change the rise and fall time (and therefore, the delay) of the buffer's output. In Figure 3.2(a), the resistor is adjusted; in Figure 3.2(b), the capacitor is adjusted; in Figure 3.2(c), the output current is adjusted. To examine the effect these variable components have on the delay and the jitter, it is useful to derive equations for the delay and jitter in terms of R and C.

Assuming that the input of the subsequent stage of the circuit has a switching threshold of 50%, for a rising edge, the delay of the stage is equivalent to the 50% rise time of the output. Using the RC time constant, the delay of the gate is expressed as follows:

$$\begin{aligned} t_d &= -RC\ln(0.5) \\ t_d &= 0.69RC \end{aligned} \tag{3.4}$$

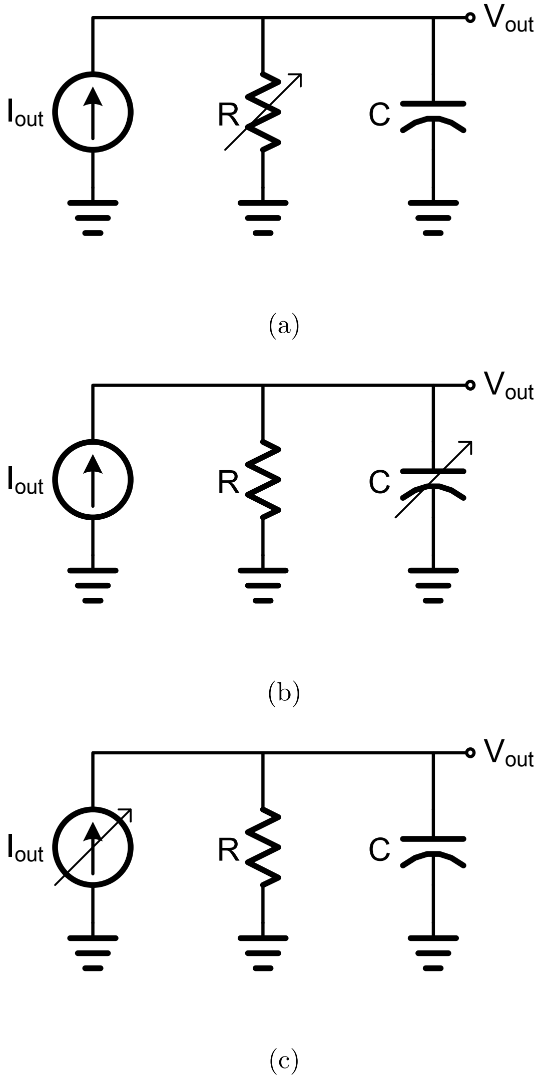
It is of note that the delay is independent of the voltage or current source. [1]

If a voltage step function is applied to the input of the buffer, then the output voltage of the buffer has the following form:

$$V_{out} = V_{in} - \exp\left(\frac{-t}{RC}\right) \tag{3.5}$$

The slope of this output is

[1] In CMOS technology, changing the voltage leads to a change in resistance, which leads to a change in the delay.



(a)

(b)

(c)

Figure 3.2: Variable delay buffer output models. (a) The resistor is adjusted (b) The capacitor is adjusted (c) The output current is adjusted

$$\frac{dV_{out}}{dt}(t) = \frac{1}{RC}\exp\left(\frac{-t}{RC}\right) \tag{3.6}$$

and the slew rate, which is the slope of the output signal as it crosses the threshold, becomes

$$\begin{aligned} SR = \frac{dV_{out}}{dt}(t_d) &= \frac{1}{RC}\exp\left(\frac{-0.69RC}{RC}\right) \\ SR &= \frac{1}{2RC} \end{aligned} \tag{3.7}$$

It has been shown [6] that the noise voltage at the output of an RC circuit is equal to the following:

$$v_n = \sqrt{\frac{kT}{C}} \tag{3.8}$$

and has no dependence on the value of the resistor.

Using Equation (3.8) for the noise voltage and the slew rate definition in Equation (3.2) on page 16, the following relationship between the jitter and the values of R and C is derived:

$$\begin{aligned} t_j &= \frac{\sqrt{\frac{kT}{C}}}{\left(\frac{1}{2RC}\right)} \\ t_j &= 2R\sqrt{kTC} \end{aligned} \tag{3.9}$$

This relationship shows that the jitter has a linear dependence on the output resistance of the buffer, and a square root dependence on the load capacitance. To illustrate the difference between these options, consider the following; if the circuit shown in Figure 3.2(a) is used, a doubling of the delay would require a doubling of the resistance. This doubling of the resistance would cause a doubling of the jitter as well. However, using the circuit in Figure 3.2(b) implies that a doubling of the delay requires a doubling of the capacitance, resulting in an increase of a factor of only $\sqrt{2}$ in the jitter. Thus, varying the capacitance is a better way of varying the delay in terms of jitter performance then varying the resistance.

The third method of changing the delay of the buffer is that the current, I_d, in the output stage of the buffer can be adjusted, as shown in Figure 3.2(c). This method works provided that the switching threshold of the buffer or inverter does not scale with the current. The result of an increase in current is that the slew rate of the output increases, and the delay decreases in approximately linear proportion. However, the increase in current also leads to an increase in the noise voltage.

To determine a relationship between the current, I_d, and the jitter, t_j, the delay is estimated using the saturation equations, as in Rabaey [7]. Therefore, it is acceptable to use the MOSFET noise characteristics in saturation. The noise current for a MOSFET in saturation is

$$i_n = \sqrt{\frac{8}{3}kTg_m} \tag{3.10}$$

Since $g_m \propto \sqrt{I_d}$, the noise voltage and the current in the buffer are related as follows:

$$v_n \propto \sqrt[4]{I_d} \tag{3.11}$$

This expression, combined with the linear relationship between slew rate and the current, means that varying the current in the buffer results in the following relationship, derived from Equation (3.2):

$$t_j \propto \frac{1}{{I_d}^{3/4}} \tag{3.12}$$

The slew rate is proportional to the current, so by Equation (3.1), the delay is inversely proportional to the current. This means that for a doubling in delay, a halving of current is necessary. When the current is halved, by Equation (3.12), the jitter increases by a factor of $2^{3/4}$.

The required adjustment factor and resulting effect on jitter for each of the three methods of adjusting the delay are shown in Table 3.1 and Figure 3.3. From the table and Equations (3.9) and (3.12), it is clear that the increase in jitter caused by varying the output current is greater than that caused by adjusting the output capacitance but less than the increase when varying the resistor.

As can be seen in Table 3.1 and Figure 3.3, the variable capacitance structure is preferred for its noise performance. Increasing the delay results in a smaller increase in jitter at the output than the other adjustment methods. However, there is a finite

Table 3.1: The adjustment necessary to double the delay for each of the parameters and their effect on the jitter

Parameter	Delay Adjustment	Effect on Jitter	
C	$\times 2$	$\times\sqrt{2}$	1.41
I_d	$\times 2$	$\times 2^{3/4}$	1.68
R	$\times\frac{1}{2}$	$\times 2$	2

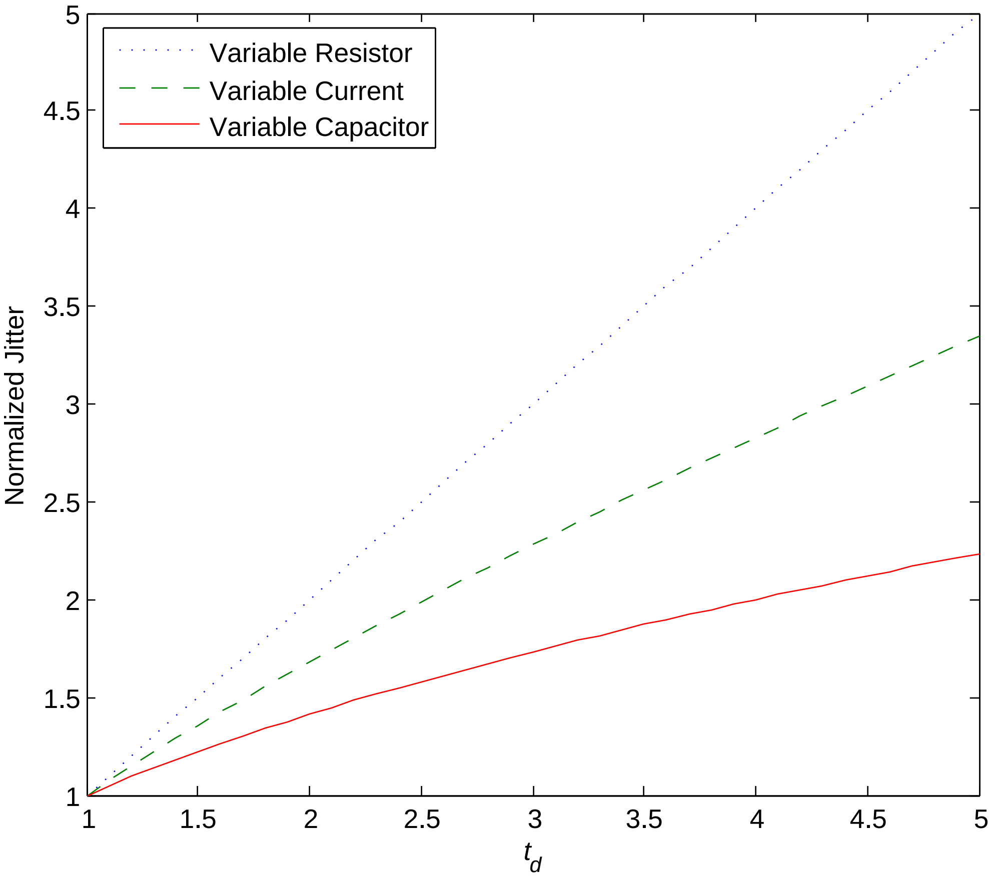


Figure 3.3: Normalized output jitter ($v_n = T_d = 1$) versus delay for each of the adjustment methods

limit on the minimum size of well-characterized capacitors in CMOS processes. Therefore, if the timing steps achievable using these capacitors are not precise enough to meet the requirement, it may be necessary to implement a second method of delay adjustment to complement the variable capacitance. The second preference is to adjust the current, since it provides the second best jitter performance.

3.2.2 Comparing Three Delay Architectures - A Simple Model

With the assumption that a gate may either have a fixed delay or a variable delay, there are three possible approaches to arrange a set of buffers to create adjustable delays with fine precision and a broad range. These are illustrated in Figure 3.4. The first, shown in Figure 3.4(a), uses a single variable-delay buffer. The second, in Figure 3.4(b), uses N variable-delay buffers in series. The third uses N fixed-delay buffers in series as coarse unit steps, with a multiplexer to select between taps at each stage in the series, and a variable-delay buffer to provide fine adjustment, as shown in Figure 3.4(c). It is assumed that the variable-delay buffers can be adjusted from a finite minimum delay (which, for simplicity, is treated as equal to the delay of the fixed-delay buffers) to some sufficiently large maximum delay.

Each of these architectures for delay adjustment can be analyzed for their delay characteristics and their jitter performance using the relationship in Equation (3.3). Based on a comparison of these analyses, the optimal architecture can be selected for implementation.

Single Variable Delay Buffer

Examining the first architecture, in Figure 3.4(a), the total delay is a variable delay that can be adjusted from some finite minimum delay to as large a delay as necessary to achieve the required maximum edge adjustment. The delay is expressed as

$$\begin{aligned} t_d &= T_d + \Delta D T_d \\ t_d &= T_d \left(1 + \Delta D\right) \end{aligned} \tag{3.13}$$

where T_d is the unit delay of a delay buffer and ΔD is the factor by which the delay is adjusted, expressed as a fraction of a gate delay. ΔD is a real number precise

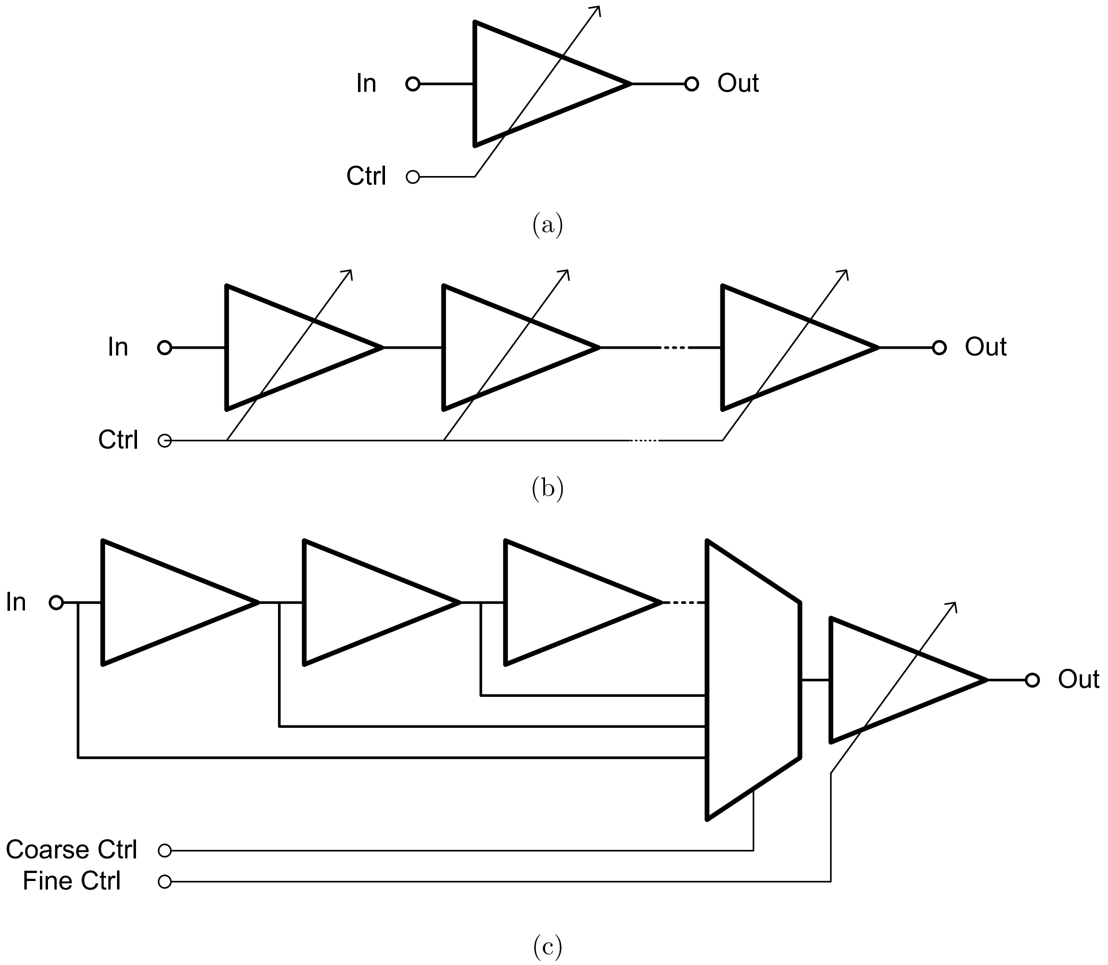


Figure 3.4: The three configurations of fixed- and variable-delay buffers to allow delay adjustment in fine increments with a broad range. (a) A single variable-delay buffer (b) N variable-delay buffers (c) N fixed-delay buffers and one variable-delay buffer

enough to achieve the required timing accuracy, and it may range high enough to achieve the maximum range of adjustment. In all cases, $\Delta D \geq 0$.

Using the relationship between delay and jitter in Equation (3.3), the jitter at the output of the circuit is written as

$$t_j \propto v_n T_d \left(1 + \Delta D\right) \tag{3.14}$$

Initially, it is useful to assume that changing the delay of the gate does not affect the noise voltage of the circuit and that v_n is a constant. This assumption will be revisited. Based on Equation (3.14), when v_n is independent of ΔD, t_j varies linearly with ΔD, as expected from Equation (3.3).

Many Variable Delay Buffers

Examining the second architecture, in Figure 3.4(b), the total delay is comprised of N delay stages in series, each of which is adjustable from some finite minimum gate delay to a sufficiently large maximum delay. Here, the delay is

$$\begin{aligned} t_d &= NT_d + \Delta D T_d \\ t_d &= NT_d \left(1 + \frac{\Delta D}{N}\right) \end{aligned} \tag{3.15}$$

Comparing this to Equation (3.13), it is clear that the fixed minimum delay in this case is much greater than in the first option, but the adjustability is equivalent. It is also clear that the precision of adjustment by each stage is required to be N times higher in order to achieve the same overall adjustment characteristics.

To determine the jitter in this architecture, begin by examining the jitter in one single element of the chain of buffers. Since the delay per stage of the circuit is $t_{d_0} = T_d \left(1 + \frac{\Delta D}{N}\right)$, the jitter per stage is

$$t_{j0} \propto v_n T_d \left(1 + \frac{\Delta D}{N}\right) \tag{3.16}$$

The jitter of each stage is a random variable that is uncorrelated between stages, so the total RMS jitter added to the clock from input to output is determined by adding the magnitudes of the jitter for each stage, as follows:

$$t_j = \sqrt{t_{j_1}^2 + t_{j_2}^2 + ...t_{j_N}^2} \tag{3.17}$$

which reduces to

$$t_j = \sqrt{N t_{j_0}^2} = t_{j_0}\sqrt{N} \tag{3.18}$$

when the RMS jitter for each stage is the same.

Substituting the value obtained for the jitter per stage in Equation (3.16) into Equation (3.18), the total jitter of the circuit in the second architecture is expressed as

$$t_{j_0} \propto v_n T_d \sqrt{N}\left(1 + \frac{\Delta D}{N}\right) \tag{3.19}$$

If $N = 1$, Equation (3.15) reduces to Equation (3.13). Similarly, if $N = 1$, Equation (3.19) reduces to Equation (3.14). This means that the first architecture is simply a special case of the second architecture where the number of stages is equal to one. The jitter performance of both cases is summarized in Table 3.2 and Figure 3.5.

When choosing a delay architecture, the preferred solution is the one that gives the lowest jitter for the desired range of delays. As illustrated by Figure 3.5, the optimal number of delay stages is different, depending on the amount of delay adjustment that is necessary. For greater delays, the use of more stages leads to lower jitter. The lower bound on the jitter, as illustrated by the dotted line in Figure 3.5 and the bold entries in Table 3.2, can found for all $\Delta D \geq 1$ by setting the derivative of Equation (3.19) equal to zero and solving for N. It is found that $N = \Delta D$ yields the minimum jitter, resulting in

$$t_{j_{min}} \propto v_n T_d 2\sqrt{\Delta D} \tag{3.20}$$

For $\Delta D < \sqrt{2}$, the minimum is clearly the case where $N = 1$. To achieve the minimum jitter over a wide range of delays, it will be necessary to select different values of N for different ΔDs.

Many Selectable Fixed Delay Paths and One Variable Delay Buffer

The third architecture consists of N fixed-delay buffers in series with taps between the buffers, allowing the selective bypassing of any number of the buffers, as shown

Table 3.2: Output jitter of the circuit for the first two architectures for varying delays and buffer chain lengths. The lowest jitter for a given delay is in bold.

		ΔD					
		0	0.5	1	2	3	4
N	1	**v_nT_d**	**$1.50v_nT_d$**	**$2.00v_nT_d$**	$3.00v_nT_d$	$4.00v_nT_d$	$4.50v_nT_d$
	2	$1.41v_nT_d$	$1.77v_nT_d$	$2.12v_nT_d$	**$2.83v_nT_d$**	$3.54v_nT_d$	$4.24v_nT_d$
	3	$1.73v_nT_d$	$2.02v_nT_d$	$2.30v_nT_d$	$2.89v_nT_d$	**$3.46v_nT_d$**	$4.04v_nT_d$
	4	$2.00v_nT_d$	$2.25v_nT_d$	$2.50v_nT_d$	$3.00v_nT_d$	$3.50v_nT_d$	**$4.00v_nT_d$**

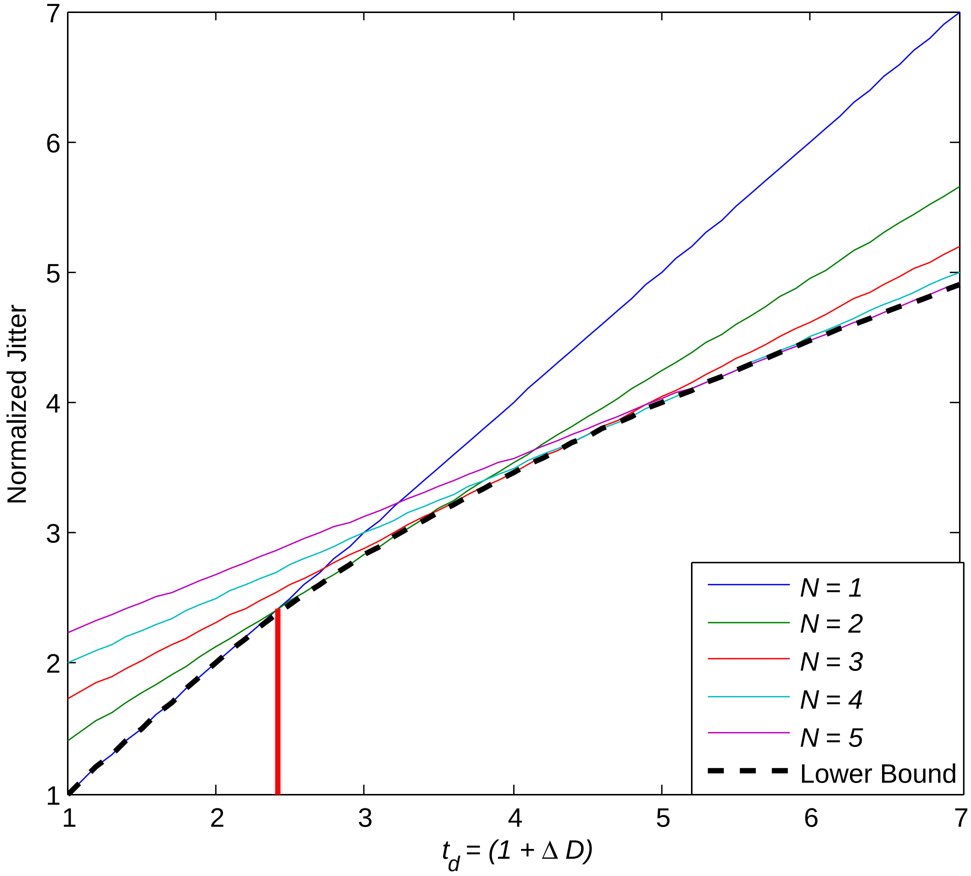


Figure 3.5: Normalized output jitter ($v_n = T_d = 1$) versus delay for various lengths of buffer chains, annotated with lower bound on jitter for a given delay with any length of chain. The marker at $\Delta D = \sqrt{2}$ indicates where the minimum delay transitions from $N = 1$ to $N = 2$.

in Figure 3.4(c). This is followed by one variable-delay buffer. The selectable fixed-delay paths allow a coarse delay adjustment by an integer number of gate delays and the variable-delay buffer allows a variable delay from a minimum of one unit gate delay to greater than two gate delays. For simplicity, the delay of the fixed buffers is assumed to be equal to the minimum delay of the variable delay buffer, although this is not required, so long as the difference between the variable-delay buffer's maximum and minimum delays is greater than or equal to the gate delay of the fixed-delay buffer. The resulting delay from this structure is

$$\begin{aligned} t_d &= \Delta N T_d + T_d + \Delta D T_d \\ t_d &= \Delta N T_d + \left(1 + \Delta D\right) T_d \end{aligned} \tag{3.21}$$

where $\Delta N = 0, 1, 2 ... k$ is the integer number of fixed gate delays currently selected in the coarse adjustment section and ΔD is the fractional adjustment provided by the variable-delay buffer, as before, which allows fine adjustment of the delay.

Determining the jitter for this architecture requires using a method similar to Equation (3.17), since the individual gates have random and uncorrelated jitter. Each fixed-delay buffer has jitter of

$$t_j \propto v_n T_d \tag{3.22}$$

When put together in series, these buffers contribute selectable jitter that is expressed by the following:

$$t_j \propto v_n T_d \sqrt{\Delta N} \tag{3.23}$$

The variable delay buffer adds the following:

$$t_j \propto v_n T_d \left(1 + \Delta D\right) \tag{3.24}$$

Combining these gives the total jitter for the circuit:

$$t_j = v_n T_d \sqrt{\Delta N + \left(1 + \Delta D\right)^2} \tag{3.25}$$

The delay provided by this implementation is the same as that provided by the first option. The minimum delay is T_d, with $\Delta N = 0$ and $\Delta D = 0$. The delay is increased by increasing ΔD until $t_d = 2T_d$. At this point, the coarse delay selection and fine delay adjustment change so that $\Delta N = 1$ and $\Delta D = 0$ again, such that the delay remains $t_d = 2T_d$. To increase the delay, this process is repeated, incrementing ΔN and resetting ΔD, as necessary, as shown in Table 3.3.

If this mechanism is used to increase the delay, then the same manipulations can be applied to the jitter equation to determine the jitter of the circuit. The results of these calculations are found in Table 3.3 and are plotted in normalized form in Figure 3.6, with the lower bound from Figure 3.5 for comparison.

Comparing the jitter resulting from the third architecture with the lower bound on the jitter resulting from the first and second architectures, it is clear that the third implementation for delay adjustment provides the best jitter performance for a given delay adjustment. However, it is necessary to revisit the assumption that the noise voltage does not change with ΔD.

3.2.3 Revisiting the Three Delay Architectures - A Complete Model

The complete jitter model of the delay architectures allows the generalization of the previous models to account for the change in the noise voltage as the delay of the variable buffers is adjusted. Each of the three architectures considered previously is revisited for this purpose, using the variable capacitance model described in Section 3.2.1.

Single Variable Delay Buffer

For the first architecture, shown in Figure 3.4(a), the delay adjustment using a single variable-delay buffer can be expressed in terms of R and C by including the adjustment factor into Equation (3.4):

$$t_d = 0.69RC\left(1 + \frac{\Delta C}{C}\right) \tag{3.26}$$

Using this equation and Equation (3.13), definitions of T_d and ΔD are written in terms of R and C:

Table 3.3: Output jitter of the circuit for the third architecture for varying ΔN and ΔD

ΔN	ΔD	t_d	t_j
0	0	T_d	$v_n T_d$
0	1	$2T_d$	$2.00 v_n T_d$
1	0	$2T_d$	$1.41 v_n T_d$
1	1	$3T_d$	$2.24 v_n T_d$
2	0	$3T_d$	$1.73 v_n T_d$
2	1	$4T_d$	$2.45 v_n T_d$
3	0	$4T_d$	$2.00 v_n T_d$

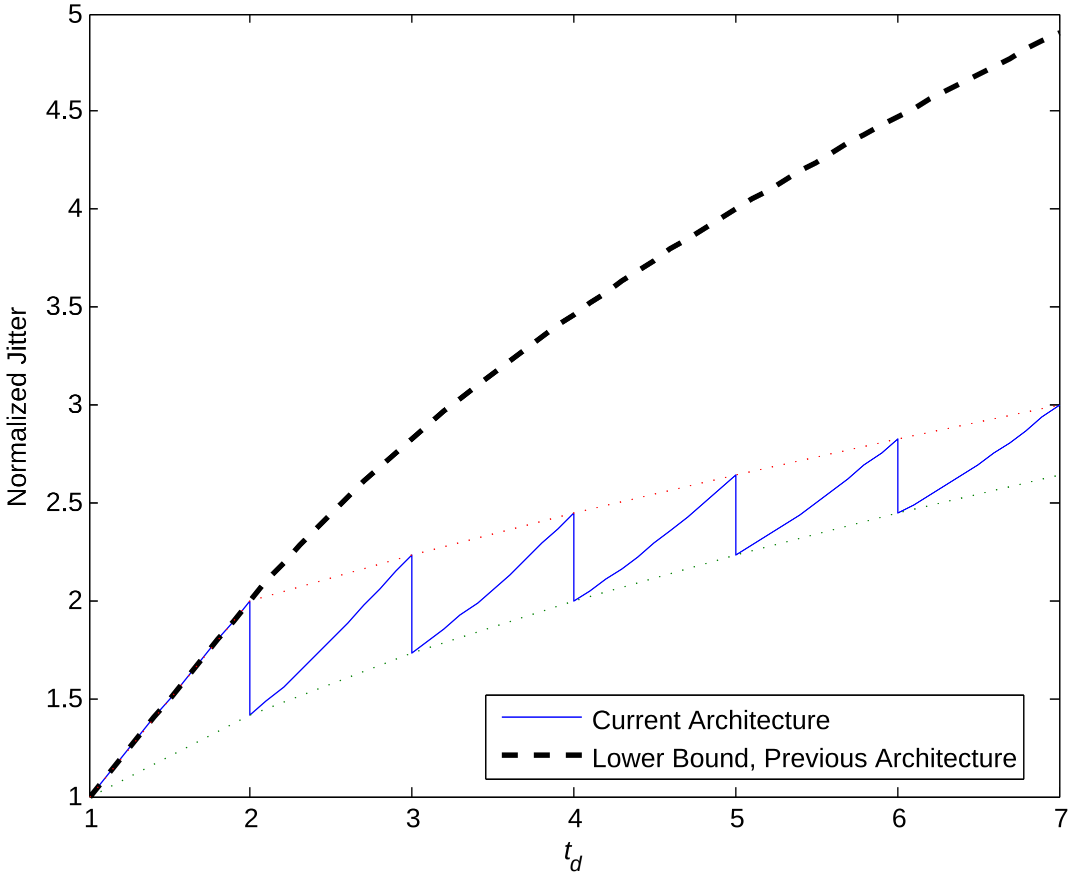


Figure 3.6: Normalized output jitter ($v_n = T_d = 1$) versus delay as ΔD and ΔN are adjusted to increase the delay, annotated with the lower bound on jitter from Figure 3.5

$$\Delta D = \frac{\Delta C}{C} \tag{3.27}$$

$$T_d = 0.69RC \tag{3.28}$$

Incorporating the capacitor adjustment into the equation for the jitter, Equation (3.2), and using the noise voltage from Equation (3.8) and the slew rate from Equation (3.7) gives

$$t_j \approx \frac{\sqrt{\frac{kT}{C}}\sqrt{\frac{1}{1+\frac{\Delta C}{C}}}}{\frac{1}{2RC}\left(\frac{1}{1+\frac{\Delta C}{C}}\right)}$$

$$t_j \approx 2RC\sqrt{\frac{kT}{C}}\sqrt{1+\frac{\Delta C}{C}} \tag{3.29}$$

To compare with the previous models, consider that $v_n = \sqrt{\frac{kT}{C}}$ is once again a constant, representing the default noise voltage at the minimum delay, because its variation with ΔD has been taken into account in Equation (3.29). Additionally, recalling the definitions of T_d and SR in Equations (3.27) and (3.7), respectively, consider that the $T_d \propto \frac{1}{SR}$ still holds. The jitter becomes

$$t_j \propto T_d v_n \sqrt{1+\Delta D} \tag{3.30}$$

This expression allows for the direct comparison with the previous model, Equation (3.14) on page 24. In Equation (3.29) the jitter has a square root dependence on the adjustment factor (and the delay) when the delay is controlled by an adjustable capacitance, while in Equation (3.14) there was a linear relationship.

Many Variable Delay Buffers

For the second architecture, shown in Figure 3.4(b), the delay per buffer is found by dividing the total delay in Equation (3.15) by the number of stages in the buffer chain:

$$t_{d_0} = T_d\left(1+\frac{\Delta D}{N}\right) \tag{3.31}$$

This variable delay per buffer is equal to the variable delay of one single buffer in Equation (3.26), which allows $\frac{\Delta D}{N}$ and T_d to be written in terms of R and C for this architecture:

$$\frac{\Delta D}{N} = \frac{\Delta C}{C} \tag{3.32}$$

$$T_d = 0.69RC \tag{3.33}$$

The total delay, from Equation (3.15), is rewritten in terms of R and C:

$$t_d = 0.69RCN\left(1 + \frac{\Delta C}{C}\right) \tag{3.34}$$

When the variable capacitor is included in the jitter calculation, the jitter per buffer is identical to the total jitter in the single buffer architecture in Equations (3.29) and (3.30). Adding the jitter from each stage gives the total jitter:

$$\begin{aligned} t_j &\approx 2RC\sqrt{\frac{kT}{C}}\sqrt{N}\sqrt{1 + \frac{\Delta D}{N}} \\ t_j &\propto T_d v_n \sqrt{N + \Delta D} \end{aligned} \tag{3.35}$$

Comparing this with the previous model, Equation (3.19) on page 25, the jitter again has a square root dependence on the adjustment factor ΔD, where previously there was a linear relationship.

Many Selectable Fixed Delay Paths and One Variable Delay Buffer

For the third architecture, shown in Figure 3.4(c), the delay is comprised of ΔN fixed delay buffers and one variable delay buffer. When written in terms of R and C, the delay is expressed as

$$t_d = 0.69RC\Delta N + 0.69RC\left(1 + \frac{\Delta C}{C}\right) \tag{3.36}$$

Using this expression and Equation (3.21), it is clear that the definitions of T_d and ΔD are the same as in Equation (3.27) for the single variable buffer case.

Including the variable capacitor in the jitter calculation gives

$$t_j \approx 2RC\sqrt{\frac{kT}{C}}\sqrt{\Delta N + \left(1 + \frac{\Delta C}{C}\right)}$$

$$t_j \propto T_d v_n \sqrt{\Delta N + 1 + \Delta D} \tag{3.37}$$

Compared to the original model in Equation (3.25), the variable adjustment factor no longer has the square under the root, resulting in a square root relationship between the jitter and the delay adjustment.

The jitter for this case is now very similar to that of the second architecture in Equation (3.35). Consider that in this case, the mechanism for increasing the delay is such that $\Delta N + \Delta D$ is a linearly increasing value that increments by steps corresponding to the precision of adjustment, and $(\Delta N + \Delta D) \geq 0$. Similarly, in the previous architecture, ΔD increases linearly in identical step sizes, and $\Delta D \geq 0$. The result is that for the case where $N = 1$, Equation (3.35) is equal to Equation (3.37). The plot in Figure 3.7 confirms this finding.

Based on these models for the jitter performance, it is clear that the first architecture and the third architecture are the same in terms of jitter performance and delay adjustability. The second architecture (for $N > 1$) has higher jitter and higher minimum delay, so it will not be considered further for this application.

The first architecture is less practical than the third because of the variable capacitor it requires to achieve a wide range of delay with precise increments. The capacitor must be large, requiring a significantly large silicon die area to implement. Furthermore, the parasitics associated with switching the capacitance on or off to adjust the delay add a large parasitic load to the output of the buffer, even with the load capacitance turned off. This increases the required variable load capacitance, in order to achieve the appropriate $\Delta D = \frac{\Delta C}{C}$. The third architecture requires only that the variable capacitance be large enough to double the delay of the variable buffer. The fixed delay buffers take up significantly less die area than the capacitance required to achieve equivalent delay in the variable buffer of the first architecture, and the solution in the third architecture is expandable to achieve a very large delay if necessary. Therefore, the third architecture is the best-performing solution to the problem.

It is worthwhile to note that the second architecture is very similar to typical delay locked loop (DLL) implementations, and its suitability for that application should not be questioned based on this analysis. For DLLs, the requirement is typically for many uniform delays. The problem addressed by this thesis requires

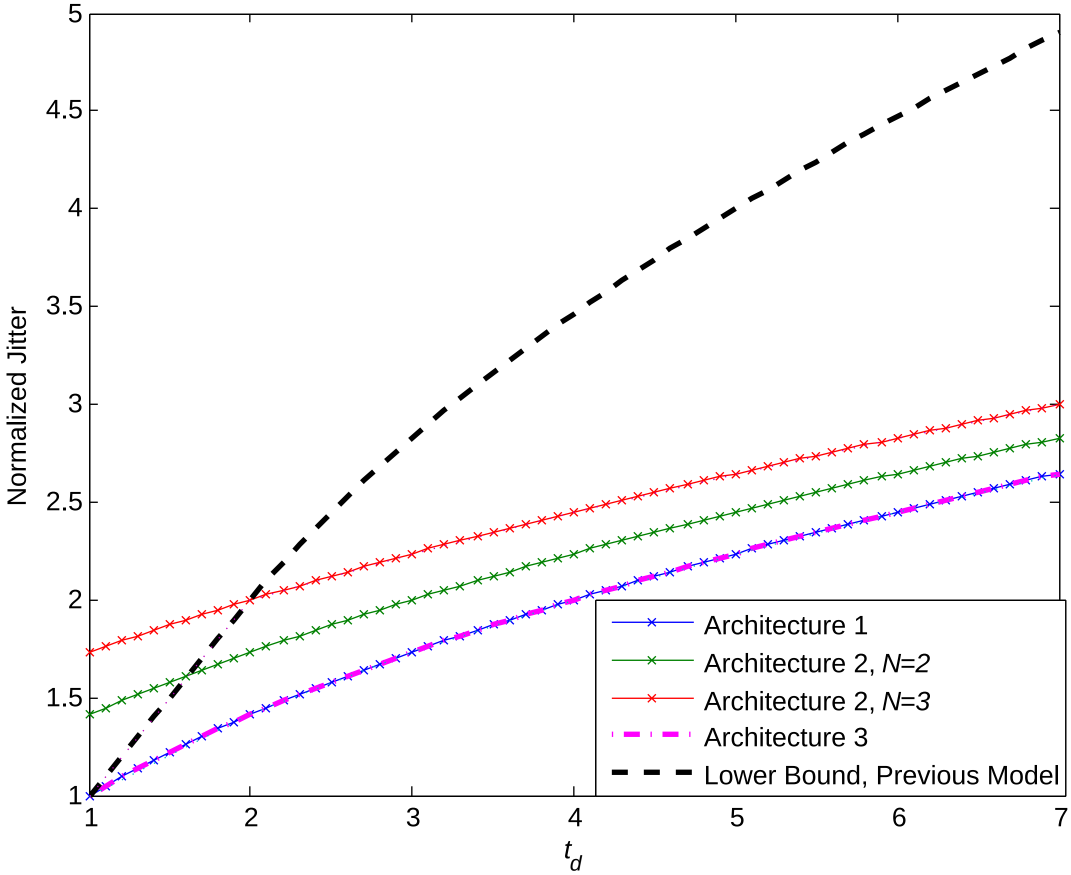


Figure 3.7: Normalized output jitter ($v_n = T_d = 1$) versus delay for each architecture using the revised jitter model, annotated with lower bound on jitter for the previous model

only one delay. Hence, these are different problems for which there are different optimal solutions.

3.3 Conclusion

A basic model and a complete model for a variable-delay buffer were developed. Based on this analysis, it was concluded that adjusting the load capacitance of a buffer offers better jitter performance than adjusting the current or the resistance, although practical limitations may require the use of current adjustment, since this is still better than the variable resistor solution.

Three architectures exist that allow the adjustment of a delay over a wide range with high precision. The first uses one variable-delay buffer, the second uses many variable-delay buffers in series, and the third uses fixed unit-delay steps for coarse adjustment plus one variable-delay buffer for precision. Comparing these buffers based on their delay adjustability and jitter performance using the aforementioned models suggests that the first and third options give equal optimality. Based on practical considerations, the third architecture is selected for implementation.

Chapter 4

Design of a Digitally-Controlled Clock Skew Circuit

4.1 Introduction

As shown in Chapters 2 and 3, it is necessary to design a circuit with a delay adjustment precision of 50 fs and tuning range of at least 50 ps, with jitter of less than 25 fs. The chosen architecture consists of a number of selectable fixed-delay paths and a variable-delay buffer. These are referred to as the coarse delay-adjustment circuit and the fine delay-adjustment circuit, respectively, as shown in Figure 4.1. It is necessary to design the components of this: the fixed-delay buffers, path selection circuit, and the variable delay buffer.

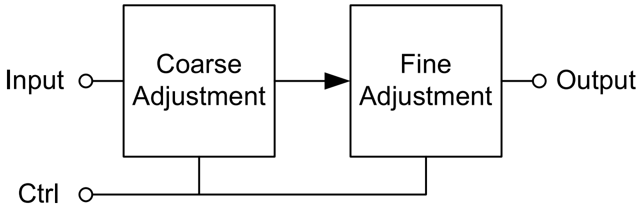


Figure 4.1: A system providing delay adjustment using a coarse delay adjustment and a fine delay adjustment

The design will be implemented for testing in 0.18 μm CMOS technology. Each circuit component has been designed and simulated in Cadence Design Systems' electronic design automation software.

This chapter begins by discussing some general considerations for the design. It proceeds to discuss the design of the fixed-delay circuit structure and the variable-delay buffer, first at the gate level and then at the transistor level. Then these components are brought together to complete the clock adjustment circuit. Finally, a means for testing the circuit is considered and an effective testing methodology is designed.

4.2 Single-Ended and Differential Circuits

A general consideration when commencing a design such as this one is with what type of structure the clock path will be implemented. The two alternatives are single-ended and differential circuit structures. A single-ended implementation takes one input and produces one output. All voltages are referenced to ground, or occasionally the supply voltage. A differential implementation has two complementary inputs and two complementary outputs, and the signal values are taken as the difference between the two:

$$V_{out} = V_{out}^{+} - V_{out}^{-} \tag{4.1}$$

A single-ended implementation has the drawback that the output is coupled directly to the supply voltage. Therefore, any noise in the supply will be coupled to the output. The differential option eliminates this problem because the supply noise will be coupled equally to the positive and negative outputs. When taken differentially, the noise cancels:

$$V_{out} = \left(V_{out}^{+} + V_{sup\_noise}\right) - \left(V_{out}^{-} + V_{sup\_noise}\right) \tag{4.2}$$

This means that the differential circuits will be significantly less susceptible to supply noise, implying that the total output noise will be lower than the single-ended case. Since this circuit is noise-sensitive, the design will use a differential clock path.

4.3 Fixed-Delay-Step Circuit Design

The coarse delay structure provides delays in steps of some fixed value by selecting from several different paths with varying numbers of delay elements. The number of paths required depends on the number of fixed-delay buffers required to span the entire tuning range. The minimum tuning range is 50 ps, as determined in the previous chapter. However, it is prudent to design for a much wider range for proof of concept and maximum flexibility. Therefore, the fixed-delay paths are designed to cover a total range of 200 ps.

One structure that provides the required functionality is the one illustrated in Figure 4.2 [8]. In this structure, the design of the simple delay cells is trivial because they are simply differential buffers. However, the structure requires the design of a high-quality, low-jitter differential multiplexer with several inputs.

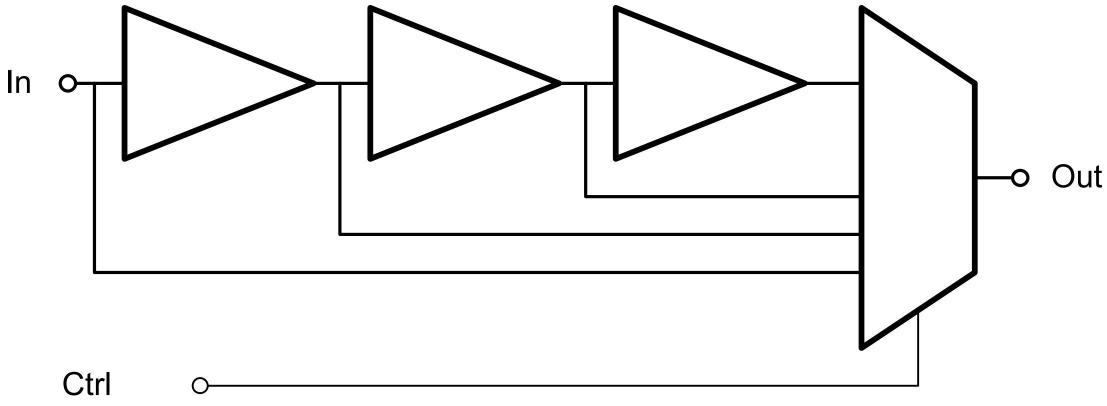


Figure 4.2: A system for providing fixed delay steps featuring many fixed-delay buffers and a multiplexer

As an estimate of the complexity of the circuit, the typical gate delay in 0.18 μm CMOS can be approximated to be in the range of 25 ps to 50 ps. Therefore, to cover the desired tuning range, it is necessary to use at least 4 gates, and as many as 8. Because of signal-swing requirements, it is difficult to design a differential multiplexer with more than two inputs. Therefore, to multiplex these signals in differential form, it is necessary to cascade several layers of 2-input multiplexers, as in Figure 4.3 where there are four selectable paths with delays of $0 + 2t_{mux}$ to $6t_{buffer} + 2t_{mux}$.

If the total required delay adjustment range is greater than the one provided above, another layer of multiplexers is necessary, exponentially increasing the complexity of the circuit. Therefore, alternative solutions are examined.

For a more flexible solution, consider the circuit shown in Figure 4.4. The input signal (*In*) is the clock, and the control signal (*Ctrl*) is a digital value used to control where the clock propagates in the circuit.

For a NAND gate, the controlling input is a 0 (that is, when an input is a 0,

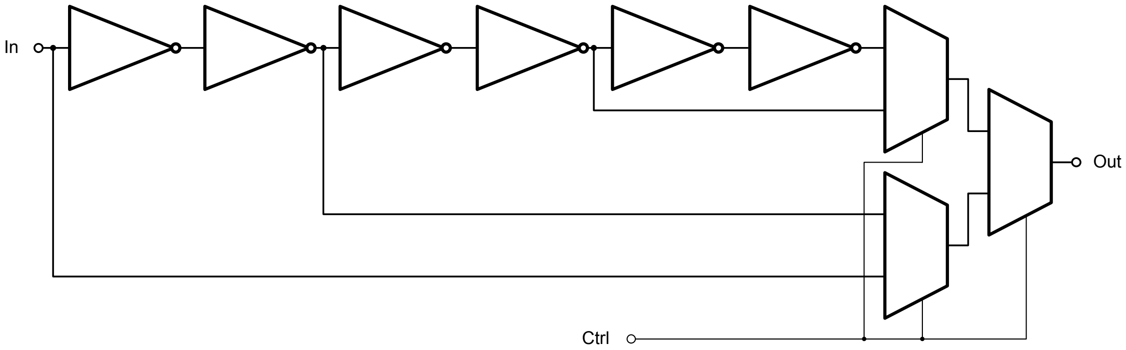


Figure 4.3: A system for providing four different delay paths using multiplexers to select the active path

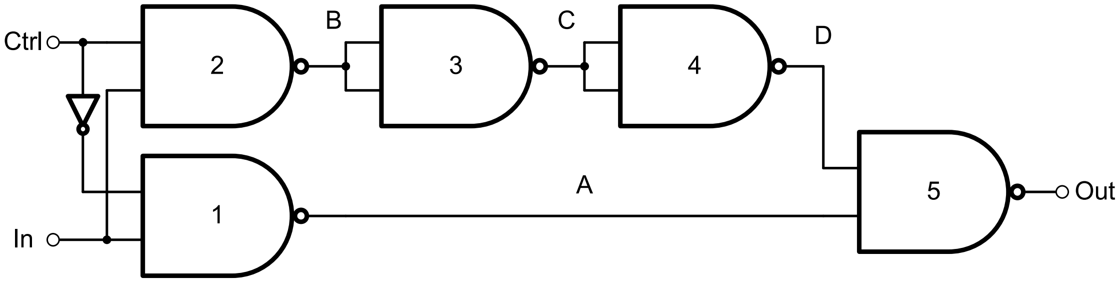


Figure 4.4: A system for providing two different delay paths using a NAND gate switching structure

the output is forced to be 1, irrespective of the value of the other input), while a passive input is a 1 (whereby the value of the other input determines the value of the output; the other input being a 0 results in a 1 at the output, while a 1 results in a 0 at the output, and a toggling of the second input results in a toggling of the output).

Referring to Figure 4.4, if the control signal is a 0, then the inputs to Gate 1 are 1 and a clock, and the clock propagates, inverted, to the net marked A. The inputs to Gate 2 are 0 and a clock, and the 0 forces B to be 1. Gates 3 and 4, with their inputs tied together, act as inverters, so C is 0 and D is 1. Thus the inputs to Gate 5 are 1 and the inverted clock, and the output of the circuit is the clock. In this example, the clock has passed through 2 NAND gates and has therefore been delayed by $2t_{NAND}$, where t_{NAND} is the gate delay from the input to the output of the NAND gate. The truth table for this circuit, summarizing this scenario, is shown in Table 4.1.

Table 4.1: Truth table for a coarse delay structure using NAND gates to select between two different paths

In	Ctrl	A	B	C	D	Out	Delay
clock	0	$\overline{clock}$	1	0	1	*clock*	$2t_{NAND}$
clock	1	1	$\overline{clock}$	*clock*	$\overline{clock}$	*clock*	$4t_{NAND}$

Alternatively, if the control signal is a 1, then the inputs to Gate 1 are 0 and a clock. Therefore, A becomes a 1. The inputs to Gate 2 are 1 and a clock, meaning B becomes the clock, inverted. Once again, Gates 3 and 4 act as inverters, so C becomes the clock and D becomes the clock, inverted again. The inputs to Gate 5 are the inverted clock and 1, resulting in an output of the clock. Now, the clock has passed through 4 NAND gates and has experienced a delay of $4t_{NAND}$. This example is summarized in the truth table in Table 4.1.

This structure is expandable to create as many delay paths as desired by recursively replacing the two NAND gates with shorted inputs by a copy of the whole circuit as many times as desired. To illustrate this, Figure 4.5 shows the case where there are four different paths through the circuit, providing similar delay to the buffers-and-multiplexer solution in Figure 4.3. A condensed truth table for this circuit is shown in Table 4.2, showing that the delay of the paths increments by steps of $2t_{NAND}$ from one to the next.

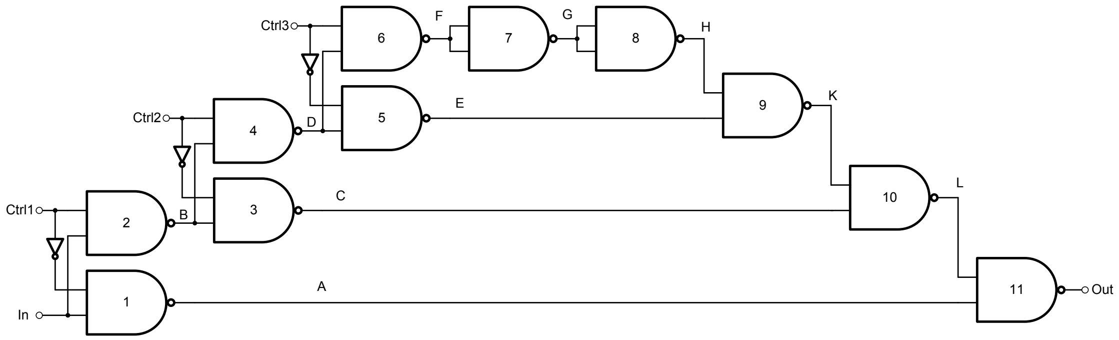


Figure 4.5: A system for providing four different delay paths using a NAND gate switching structure

Table 4.2: Truth table for a coarse delay structure using NAND gates to select between four different paths

In	Ctrl1	Ctrl2	Ctrl3	Gates Passed Through	Delay
clock	0	0	0	1, 11	$2t_{NAND}$
clock	1	0	0	2, 3, 10, 11	$4t_{NAND}$
clock	1	1	0	2, 4, 5, 9, 10, 11	$6t_{NAND}$
clock	1	1	1	2, 4, 6, 7, 8, 9, 10, 11	$8t_{NAND}$

Comparing the original buffer-and-multiplexer architecture with this NAND architecture, several important distinctions can be made. First, the NAND architecture contains just one type of logic gate, repeated many times. The buffer-and-multiplexer system requires the design and optimization of a buffer and a multiplexer. Therefore, there is less design effort required for the NAND implementation. Furthermore, the NAND implementation is easily expandable to as many steps as desired with no extra overhead, while the expansion of the buffer/multiplexer solution requires increasing the complexity of the multiplexer exponentially. Thus, the NAND switching architecture is selected for implementation.

4.3.1 Design of MOS Current Mode Logic NAND Gate

To implement the NAND switching architecture for coarse delay increments, it is necessary to design and optimize a differential NAND gate for low jitter. The class of digital logic that provides differential signaling is called current mode logic (CML). When implemented in CMOS, it is also called MOS current mode logic (MCML) [9].

Rabaey and Musicer [10] designed MCML gates for low noise. A NAND gate similar to theirs is shown in Figure 4.6, the difference being the extra transistors (M2c,d) added to the inverting output pulldown path to balance the rise times. The logic is performed by transistors M1a,b and M2a-d. M0 sets the tail current in the gate. M3a,b are load transistors whose equivalent resistance, along with the tail current, determines the voltage swing seen at the output. When the output is high, all of the current is switched to one side of the gate, and the output voltage on that side of the gate is the supply minus the drop across the PMOS device. On the other side, the PMOS device pulls the output up to the supply voltage.

The design of the load PMOS devices for this type of logic affects the logic levels, the speed of the gate, and the noise at the output of the gate. Based on Maneatis' study of low-jitter delay generation [11], for low noise and linearity, the load devices should be partitioned into two symmetric devices in parallel, one biased externally and the other diode-connected, as shown in Figure 4.7(a).

For the purpose of this circuit, the linearity is not critical, because it is the threshold crossing of the clock signal that determines the sample instance. As long as that instance is well-defined and tightly controlled, the linearity of the circuit may be poor without significantly affecting the performance of the ADC. Therefore, if it is possible to trade linearity for better noise performance, the optimality of the

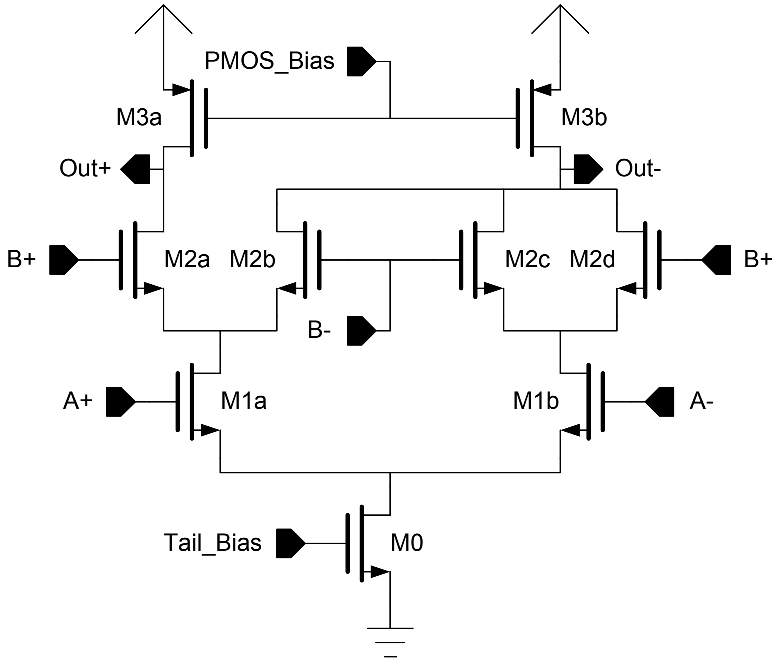


Figure 4.6: MCML NAND gate

design improves. Through a series of simulations, discussed later in Section 4.4.1, it has been determined that the noise can be reduced at the cost of linearity by changing the symmetric load devices into diode-connected PMOS devices, as shown in Figure 4.7(b). Doing so also reduces the complexity of the circuit because there is no longer a need to generate a bias voltage for the PMOS load devices.

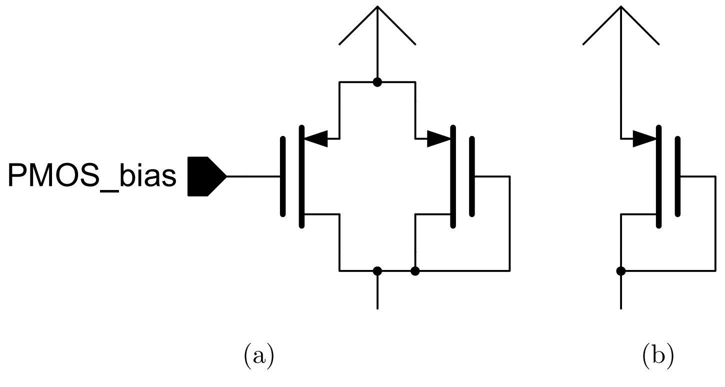


Figure 4.7: (a) Symmetric PMOS load devices for low jitter and high linearity (b) A diode-connected PMOS load device for lower jitter but poor linearity

The implementation of the MCML NAND uses diode-connected PMOS load transistors for simplicity and noise performance. In addition, for low noise, the current must be high, as shown by Weigandt [12]. Therefore, as a starting point, a tail current of 1 mA is selected. With a fixed current, the device sizes can be adjusted to achieve the optimal jitter for that current. Then the current and device width can be increased, keeping the current density constant, to achieve the required results.

Since it is desirable to operate at high frequencies, the capacitive loading should be minimized by using devices that are as small as possible. As such, the minimum lengths are used for the input NMOS devices and the load PMOS devices.

The load devices are sized to provide an appropriate swing at the output. Since the NAND will be loaded by other NAND gates and by the variable delay buffer, the output swing must be sufficient to switch the current in subsequent gates. In order to switch the current in a subsequent gate, the output swing must be greater than $\sqrt{2}V_{ov}$ of the subsequent gate [13]. Additionally, the swing should be high enough to dominate over any noise in the system, but must ensure that all of the devices remain in saturation. Therefore, a voltage swing in the range of 300 mV to 400 mV is acceptable.

As a starting point, 350 mV is selected as the swing. The output swing is determined by the voltage drop across the PMOS devices. The maximum voltage occurs when the PMOS device turns off (i.e., $V_{max} = V_{DD} - |V_{t_p}|$). The minimum voltage occurs when the entire tail current is passing through the drain of the PMOS device (i.e., $I_{D_p} = I_{tail}$) and the output voltage is equal to $V_{min} = V_{DD} - |V_{t_p}| - |V_{ov_p}|$. Hence, the swing at the output of the gate is

$$V_{SW} = V_{max} - V_{min} = |V_{ov_p}| = \sqrt{\frac{2I_{D_p}}{\mu_p C_{ox} \frac{W}{L}}} \tag{4.3}$$

Using this formula and the minimum length (i.e., $L_P = 0.18\,\mu\text{m}$), the PMOS device width can be determined to be $W_P \approx 50\,\mu\text{m}$.

The overdrive voltage of the NMOS transistors should be no higher than $\frac{V_{swing}}{\sqrt{2}} = \frac{350\,\text{mV}}{\sqrt{2}} \approx 250\,\text{mV}$. The devices are sized so that the rise and fall times are balanced. A quick series of simulations determines that an appropriate starting size is $W_N \approx 20\,\mu\text{m}$.

The circuit is optimized for low jitter by increasing the current density in the devices. Keeping the current density constant ensures that the overdrive voltages remain as designed. A series of simulations are done to determine the final device sizes and tail current for low jitter. The final device sizes are shown in Table 4.3. The designed tail current is 8 mA.

Table 4.3: MCML NAND transistor sizes

	M0	M1a,b	M2a-d	M3a,b
W	200 μm	80 μm	80 μm	210 μm
L	0.18 μm	0.18 μm	0.18 μm	0.18 μm

4.3.2 Simulation Results

A plot of the delay performance of the NAND structure is shown in Figure 4.8. As the control signals are changed, shown in green and blue at the top of the figure, the total delay of the circuit steps up from 110.5 ps to 352.8 ps in steps of about 121.1 ps, as shown in red.

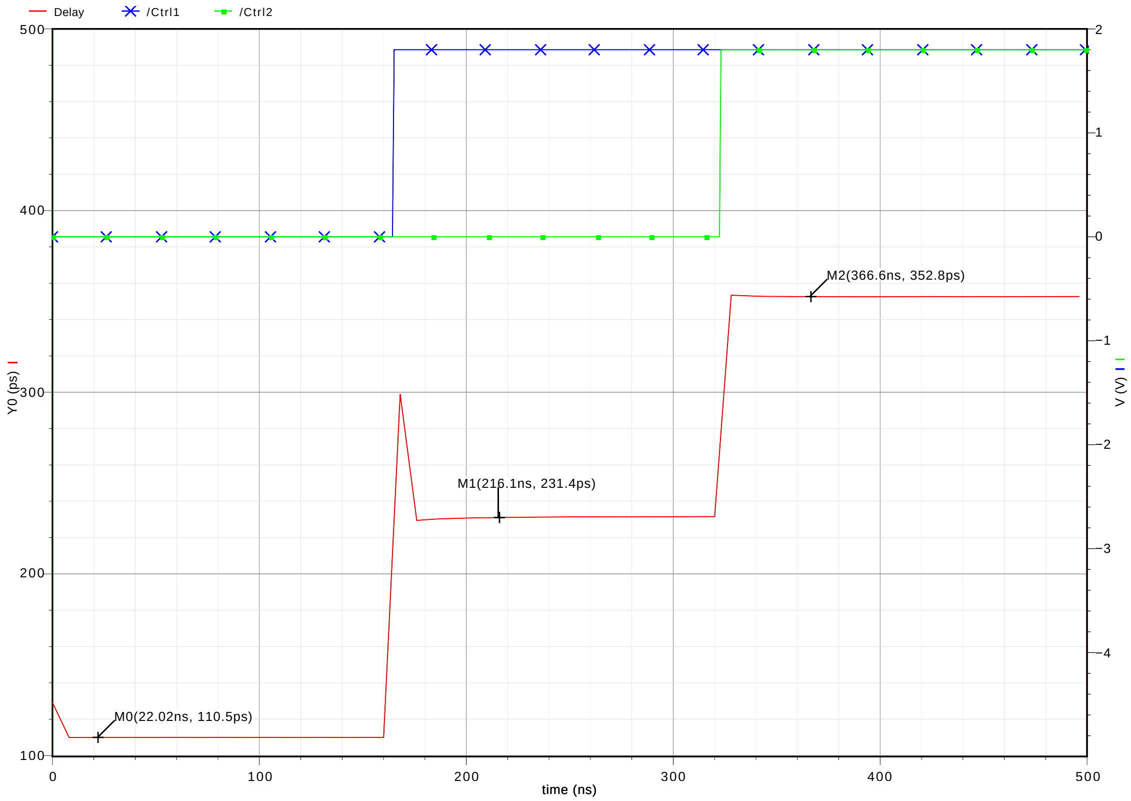


Figure 4.8: Delay results from the NAND gate switching circuit

A plot of the output and noise of the NAND circuit, with the longest path activated (noisiest case), is shown in Figure 4.9. The simulations were run following the procedure outlined in Appendix B. Using the values for the slope and the noise voltage at each of the rising and falling edges, the jitter can be determined as follows:

$$t_{j_{rise}} = \frac{v_n}{\frac{dV}{dt}} = \frac{2.901 \times 10^{-4}}{3.733 \times 10^{9}} = 77.7\,\mathrm{fs} \tag{4.4}$$

$$t_{j_{fall}} = \frac{v_n}{\frac{dV}{dt}} = \frac{2.519 \times 10^{-4}}{3.307 \times 10^{9}} = 76.2\,\mathrm{fs} \tag{4.5}$$

While this jitter value does not meet the jitter requirement specified in the previous chapter, due to an impending tape-out deadline, the results were deemed sufficient to provide meaningful results. If necessary during testing, data can be collected over a long period of time and averaged to minimize the effects of the jitter. A table summarizing the delay and noise characteristics of the NAND circuit is shown in Table 4.4.

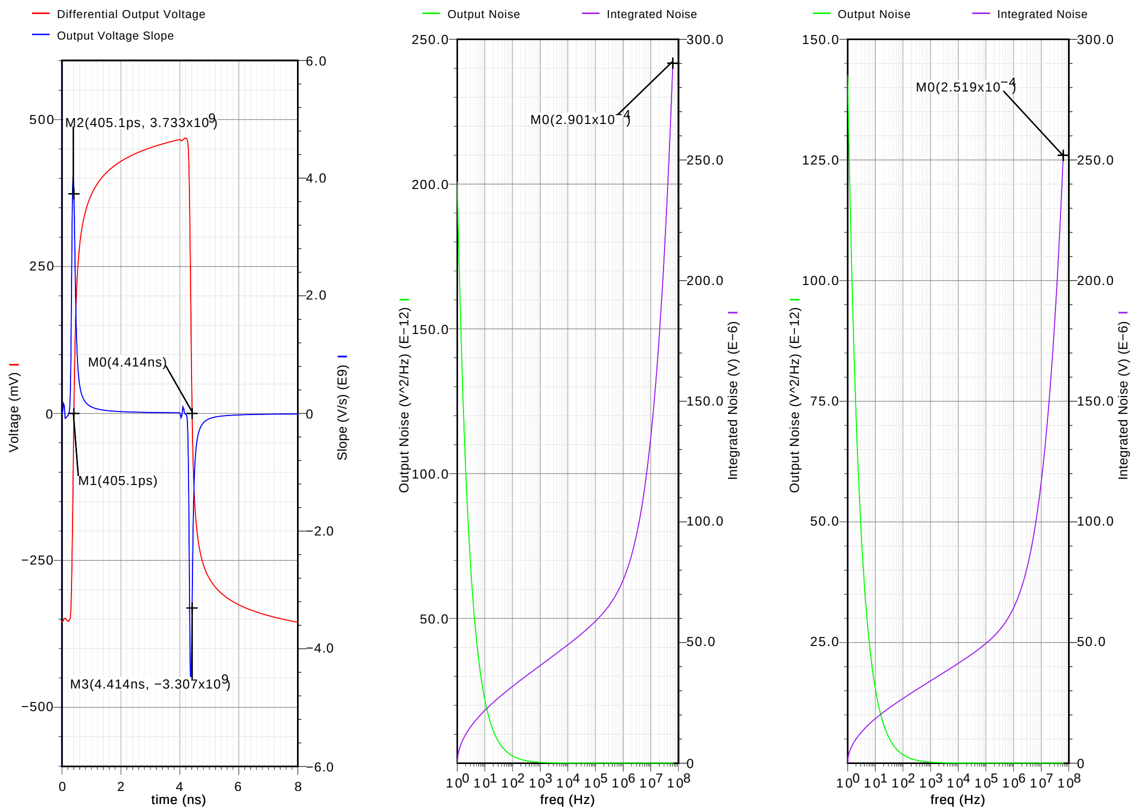


Figure 4.9: Noise simulation results from the NAND gate switching circuit. The middle plot shows the noise for the rising edge and the right plot shows the noise for the falling edge

Table 4.4: Summary of NAND gate delay switching circuit performance

Min. Delay	Max. Delay	Delay Step	$t_{j_{rise}}$	$t_{j_{fall}}$
110.5 ps	352.8 ps	121.1 ps	77.7 fs	76.2 fs

4.4 Variable-Delay Buffer Circuit Design

The variable-delay buffer is designed to provide adjustable delays in steps of 50 fs that span the steps provided by the NAND gate coarse delay switching circuit. Making the assumption that the minimum delay of the buffer will be approximately equal to one step of the NAND switching circuit, the maximum delay provided by the variable-delay buffer must be greater than double the minimum delay.

Ideally, the circuit must vary the delay by adjusting the load capacitance, as shown in Chapter 3. If the variable capacitance precision is not high enough, the current in the buffer can be varied to obtain the desired delay steps, at greater cost in terms of noise.

The buffer is designed for low noise using Weigandt's delay stage as a basis [12], which is shown in Figure 4.10. The device is biased with a replica circuit that allows the bias voltage of the PMOS load and the NMOS current source to automatically compensate for variations.

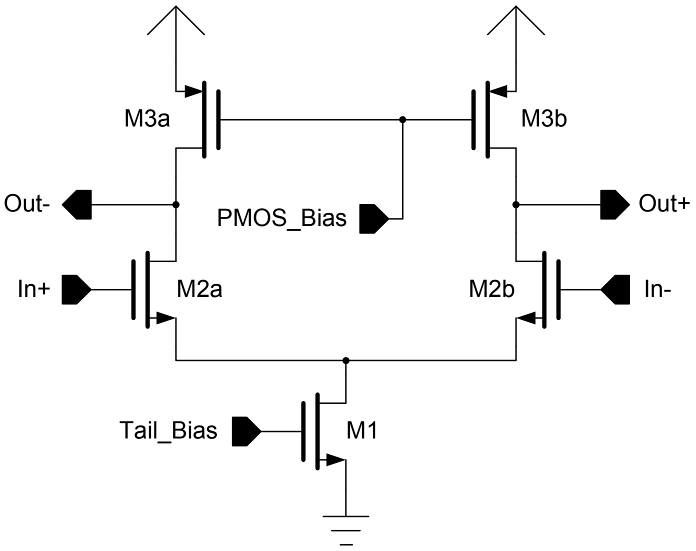


Figure 4.10: Basic differential delay stage

Maneatis [14] improved this delay stage by increasing the linearity of the output using symmetric load devices where two PMOS devices are placed in parallel to one another for each load, as shown in Figure 4.11. One PMOS transistor is biased with the replica bias circuit and the other is diode-connected, and the non-linearities of each device ideally cancel out. In Maneatis' work, the PMOS devices are equally-sized.

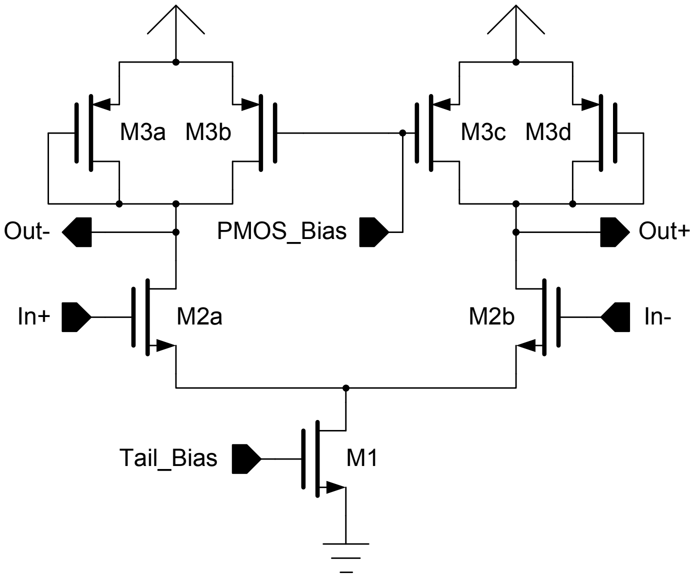


Figure 4.11: Basic differential delay stage using symmetric load PMOS devices

4.4.1 PMOS Load Devices

These circuits by Weigandt and Maneatis are designed for use in phase-locked loops (PLL), delay-locked loops (DLL), and voltage controlled oscillators (VCO) as delay stages. In these applications, their linearity is important. For the adjustment of a clock edge, the most significant concerns are the precise timing of the threshold crossing and the jitter. Therefore, it is not necessary to follow Maneatis' design exactly.

As an experiment, a simple delay buffer was designed using Maneatis' circuit diagram. The total combined widths of the PMOS load devices was kept constant, such that $W_{diode} + W_{bias} =$ const, where W_{diode} is the width of the diode-connected PMOS and W_{bias} is the width of the biased PMOS device. Then the ratio of $\frac{W_{bias}}{W_{diode}}$ was varied, and the noise was simulated in Cadence to determine how the trade-off between the load devices affected the jitter. All other circuit parameters were kept constant.

The results of this simulation are shown in Figure 4.12. The data point that is not shown on the plot because of the logarithmic scale is the one where the ratio is zero, such that there is no biased PMOS device, and the load is purely the diode-connected device. This data point shows that the downward trend below a ratio of 0.05 continues, and at $\frac{W_{bias}}{W_{diode}} = 0$ the jitter is $t_j = 70.5\,\text{fs}$.

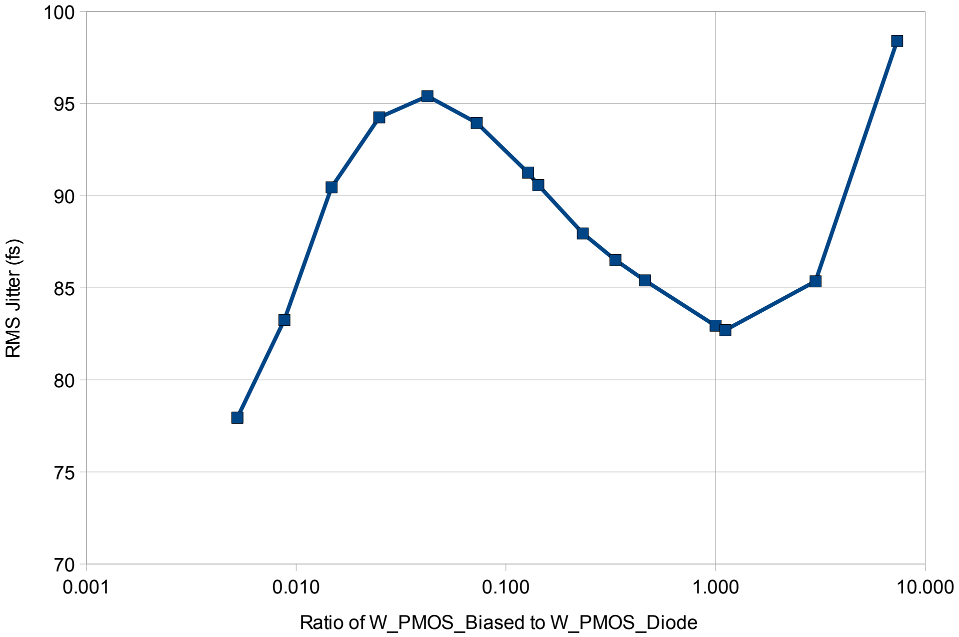


Figure 4.12: The jitter with respect to the ratio between the widths of the diode-connected PMOS device and the biased PMOS device

This plot is interesting in that it has one local minimum near the symmetric point, where $\frac{W_{bias}}{W_{diode}} = 1$. This point is the one that also maximizes linearity. For PLLs, DLLs, and VCOs, this is excellent because the jitter is optimal at the same point that the linearity is maximal, per Maneatis [14]. However, it is possible to achieve even better performance, considering the needs of this circuit do not include optimal linearity as long as the timing of the clock edges is precise and accurate.

One reason for the improvement in jitter as the size of the diode-connected PMOS is increased is that the gate of the diode-connected PMOS device is connected to the output of the gate. Hence, as its size is increased, the capacitance at the output also increases and begins to dominate. As shown in Chapter 3, increasing the capacitance at the output of the buffer reduces the jitter when all other parameters are constant. This makes sense because a larger capacitor is able to absorb more thermal noise current (and other noise sources) with less voltage variation.

Another reason for the improvement in noise is that the biased PMOS device is also acting like a common source amplifier and amplifying the noise from the biasing circuit into the buffer. While the gain of this amplifier is low, this adds to the total noise at the output of the buffer. Eliminating this bias simplifies the required circuitry and improves the noise performance.

An added bonus of the larger capacitance at the output is that the total capacitance in the $\frac{\Delta C}{C}$ term is larger, meaning that for a fixed capacitor step size, a smaller delay step is possible. This leads to greater precision, which is desirable in this application. With these benefits in mind, the delay buffer is designed as shown in Figure 4.13.

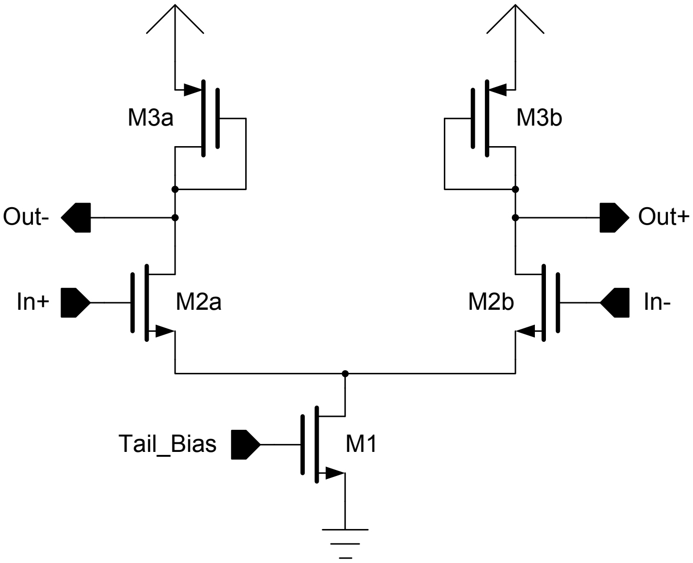


Figure 4.13: Differential delay stage using diode-connected PMOS load devices

4.4.2 Delay Buffer Design

The delay stage shown in Figure 4.13 has several degrees of freedom that can be used to achieve optimal noise performance of the system. The current, voltage swing, supply voltage, and transistor sizes can be adjusted to achieve the best possible results. However, this many degrees of freedom can make the design difficult to optimize, and the models for deep sub-micron CMOS technologies are complicated. A design procedure using the simplest models for MOS transistors has been followed to obtain a starting point with which simulations can be run. Since the simulator's models are more accurate, final optimization has been done using the simulator.

Weigandt concludes that the timing jitter in a differential delay stage is inversely proportional to the supply current [12]. He also shows that timing jitter is inversely proportional to the square root of the capacitance at the output. Consider that the load capacitance usually consists of the diffusion capacitance of the drains of the output devices as well as the gate capacitance of the input of the subsequent stage. Hence, the jitter is inversely proportional to the square root of the width of the transistors:

$$t_j \propto \frac{1}{\sqrt{W}} \tag{4.6}$$

Using these two conclusions, there are two ways to provide low jitter. The first is to increase the current in the devices and the second is to increase the size of the devices to increase the load capacitance.

To obtain an initial design for the simulator, a relatively high tail current is selected. For this design, $I_{tail} = 1\,\mathrm{mA}$ was chosen. To switch the current in the differential circuit completely to one side or the other, a differential input swing of $\pm\sqrt{2}V_{ov_n}$ is needed [13], where V_{ov_n} is the overdrive voltage of the input transistors. To be switchable by the output swing of the NAND gates that will be driving the buffer, an overdrive voltage of $V_{ov_n} = \frac{V_{swing}}{\sqrt{2}} \approx 250\,\mathrm{mV}$ is required. Using the saturation region current model for the transistors and typical device parameters for the $0.18\,\mu$m CMOS process from TSMC, the required $\frac{W}{L}$ ratio for the NMOS input transistors can be determined as follows:

$$\frac{W}{L} = \frac{2I_{d_n}}{\mu_n C_{ox} V_{ov_n}^2} \approx 114 \tag{4.7}$$

Using minimum-length devices for the input, this results in transistors of the size $W_N = 20.5\,\mu\mathrm{m}, L_N = 0.18\,\mu\mathrm{m}$.

Since it may be desirable to connect these delay buffers in series with one another, the output swing must be at least equal to the input swing requirement. The output voltage swing is determined by the resistive drop across the PMOS load devices. At each output node, the maximum voltage occurs when the devices are turned off when the voltage rises above $V_{DD} - |V_{t_p}|$. Similarly, the minimum voltage at each node occurs when the entire current in the gate is switched to that side, and the voltage drops to $V_{DD} - |V_{GS_p}|$ where V_{GS_p} is determined using the current equation. The voltage swing at the output of the gate is then

$$V_{swing} = V_{DD} - |V_{t_p}| - V_{DD} + |V_{GS_p}| = |V_{ov_p}| \tag{4.8}$$

Since the required swing is 350 mV, the required device size can be found using the saturation current equation:

$$\frac{W}{L} = \frac{2I_{d_p}}{\mu_p C_{ox}|V_{ov_p}|^2} \approx 267 \tag{4.9}$$

For minimum-length devices, this yields a size of $W_P = 48\,\mu\text{m}, L_P = 0.18\,\mu\text{m}$.

When connecting several of these devices in series, it is ideal to have regenerative gates for sub-optimal input voltages to be corrected in the circuit. To achieve this, it is necessary for the circuit's small signal gain to be greater than unity. However, large gains also amplify the noise from earlier stages at the output, so the gain must not be so high that it impedes the noise performance of the circuit. The gain of the differential circuit is given by the following:

$$\begin{aligned} A_v = g_{m_n} R_L = \frac{2I_{d_n}}{V_{ov_n}} \frac{V_{swing}}{I_{tail}} &= \frac{2\frac{I_{tail}}{2}}{V_{ov_n}} \frac{\sqrt{2}V_{ov_n}}{I_{tail}} \\ A_v &= \sqrt{2} \end{aligned} \tag{4.10}$$

This gain of about 1.4 provides a margin of safety over unity, but is not so large that the noise in the circuit will be amplified significantly.

The final part of the circuit that must be designed is the tail current source. The ideal current source has high resistance so that its current is not dependent on the voltage across it. However, the device must be wide enough to carry the required current safely.

For the current source transistor, the important considerations are the small signal gain and the output resistance. The output resistance should be high to prevent variations in the tail current. The small signal gain should be low to minimize the amount of noise amplified from the biasing circuit into the tail of the differential pair [12]. To keep the small signal gain low, the overdrive voltage is designed to be relatively high so that the g_m of the transistor is low. This is limited by the need to keep the other transistors in saturation and the signal swing requirements at the output. The size $W = 50\,\mu\text{m}$ and $L = 0.18\,\mu\text{m}$ is selected as a starting point.

With the devices sized as above, the design was entered into Cadence and simulated for jitter and delay. First, the width of the PMOS load devices was swept to determine the optimal size for low jitter. Several different constant current values were used. The results of the sweep are shown in Figure 4.14. From these results, it is clear that as the current is increased, the jitter decreases. Additionally, the optimal jitter for constant current appears to occur when the ratio between the NMOS width and the PMOS width is in the range of 2.5 to 2.75.

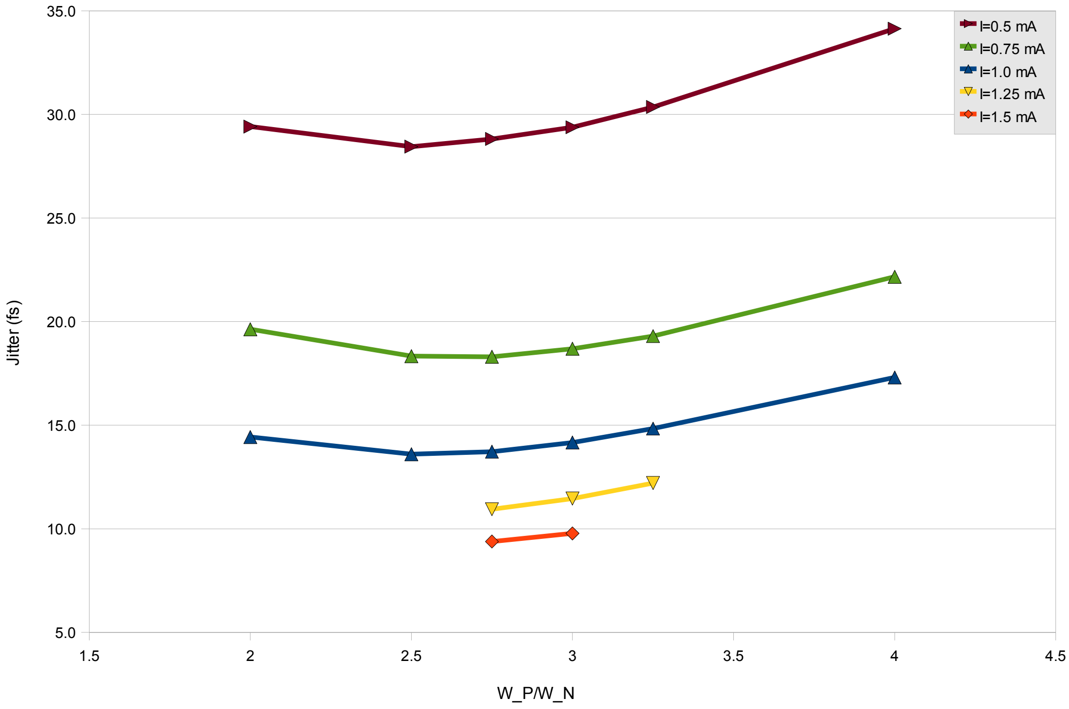


Figure 4.14: The jitter with respect to the ratio between the widths of the load PMOS (W_P) and the input NMOS (W_N) for different tail currents

Next, the tradeoff between current and jitter was examined by keeping the device sizes constant and sweeping the tail current. The results of these simulations are shown in Figure 4.15. As the current is increased, the jitter decreases in an inverse relationship, as expected. It is notable that using the ratio $\frac{W_N}{W_P} = 2.5$, the circuit breaks down with current greater than 1 mA, while the circuit with ratio $\frac{W_N}{W_P} = 2.75$ remains functional up to 1.5 mA.

To find the optimal point where further increases in current result in diminishing returns in terms of jitter, the figure of merit $FOM = I_{tail} \times t_j$ is used. As the current is increased, the jitter decreases, so the minimum of this figure of merit is the point at which the best tradeoff between current and jitter exists. This FOM is plotted in Figure 4.16, and the optimal point occurs at $I_{tail} = 1\,\text{mA}$.

The last step of the design optimization process is to increase the current and the device sizes, keeping the current density in the devices constant, until the jitter in

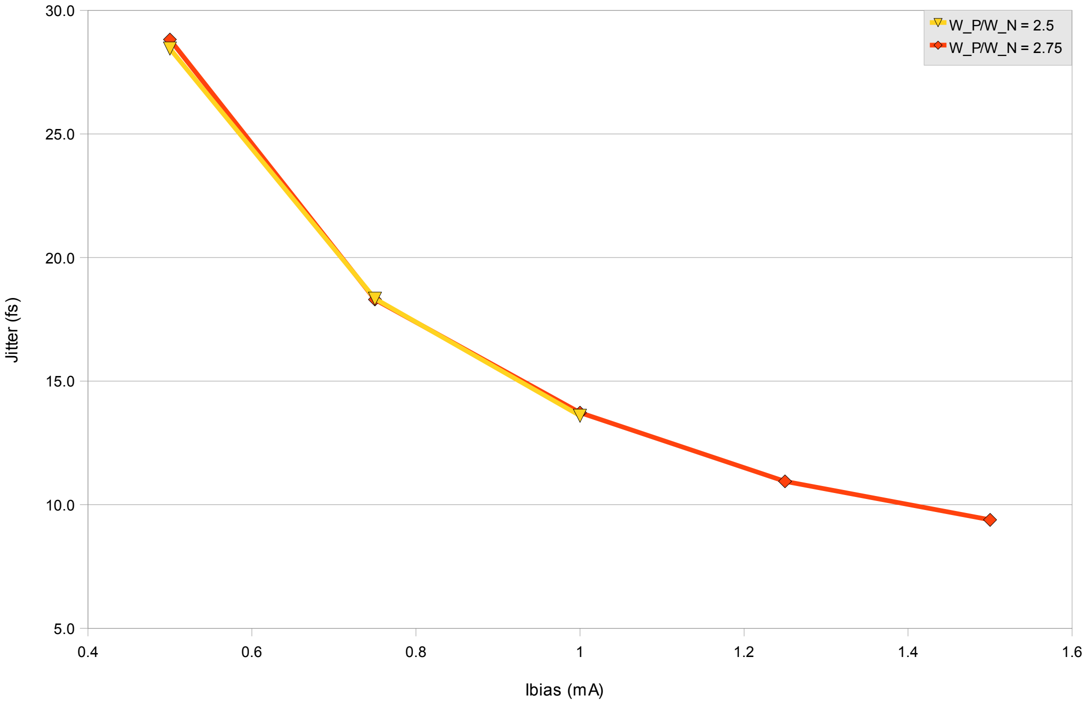


Figure 4.15: The jitter as a function of the tail current for fixed device sizes

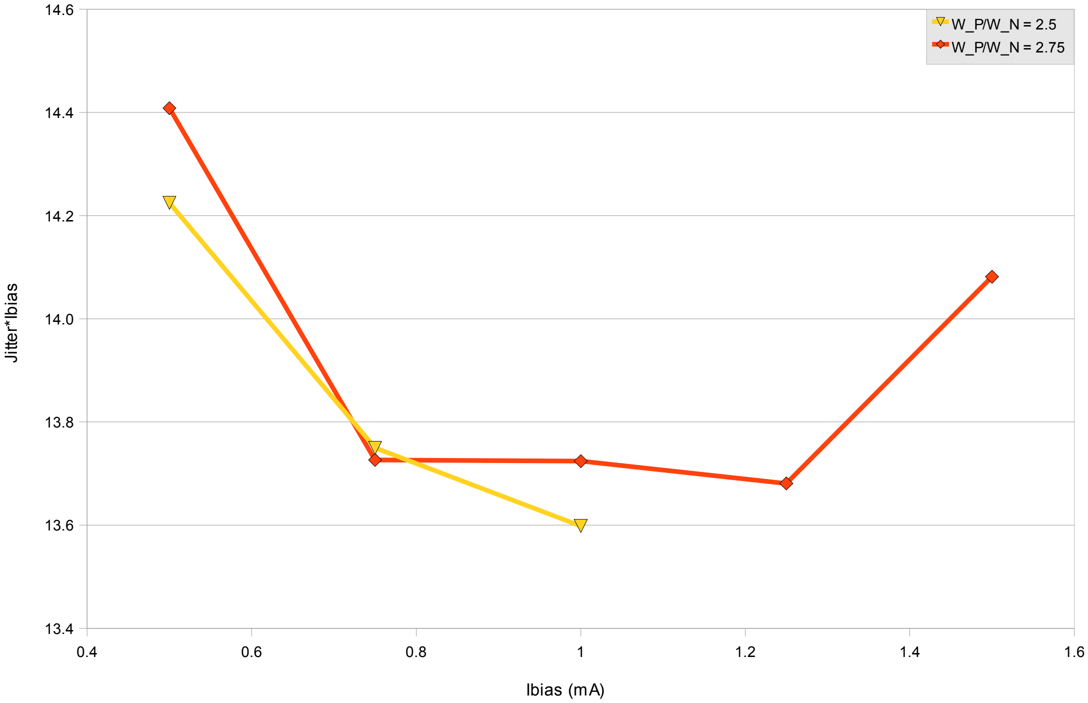


Figure 4.16: The figure of merit used to optimize the tradeoff between current and jitter

the buffer is reduced below the threshold. Following this optimization, the buffer's transistors have the sizes shown in Table 4.5. After optimization, the tail current is designed to be 8 mA. In the layout, the transistors are implemented as fingered devices, each with a width of 2.5 μm. This is done to share drain diffusions, and therefore reduce the capacitance seen at the output of the buffer.

Table 4.5: Delay buffer transistor sizes

	M1	M2a,b	M3a,b
W	200 μm	80 μm	210 μm
L	0.18 μm	0.18 μm	0.18 μm

The buffer's performance is shown in Table 4.6. The buffer has an intrinsic delay of 13.2 ps, which means that to achieve the adjustment range required to span the steps provided by the NAND circuit, the buffer must be able to increase its delay by a factor of 5. However, when the adjustable load capacitors are added, this adjustment factor will be reduced, as discussed in the next section. Additionally, the delay buffer has output jitter of 26.4 fs. Combined with the worst-case output noise of the NAND gate, the total output jitter is $t_j = \sqrt{(13.2 \times 10^{-15})^2 + (76.2 \times 10^{-15})^2} = 77.3\,\text{fs}$.

Table 4.6: Summary of delay buffer performance

Delay	t_j
24.588 ps	13.2 fs

4.4.3 Delay Adjustment with Variable Capacitor

In terms of jitter performance, the best way to adjust the delay of a buffer is to vary its load capacitance, as shown in the previous chapter. Therefore, to provide a digitally-controlled variable capacitor, circuit structures providing switchable capacitive loads are explored.

There are several structures that implement a switchable capacitive load. The ideal solution is fully differential, such that any mismatches would affect both sides of the differential buffer equally. Each of the simple circuit structures discussed here are repeated as necessary in order to provide the total load needed to be able to change the delay from one gate delay to the more than six gate delays required to span the NAND circuit delay steps.

One differential capacitive circuit proposed to accomplish switchable steps is shown in Figure 4.17. The capacitor is connected differentially between the inputs, which are connected to the output of the delay buffer. Transistors are used as switches to connect the capacitor to the external circuit when desired. Unfortunately, it is not an ideal candidate for implementation because there is no way to control the voltage at the source terminal, and therefore no way to set the V_{GS} of the transistor to guarantee that it will be on and fully conductive at all times that it is desired to be.

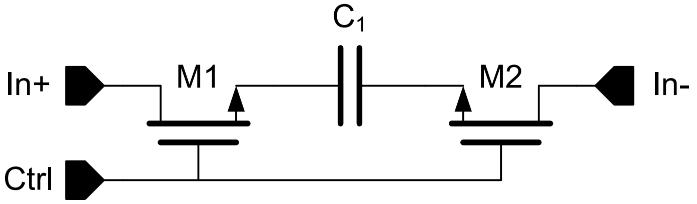


Figure 4.17: A differential switchable load capacitor configuration using transistors as switches to connect the inputs to the capacitor between them

A second differential solution is proposed in Figure 4.18. This circuit does not suffer from the problem of having floating transistor terminals. When $C_1 = C_2 = C$, the capacitance seen between the differential inputs can be switched from $\frac{C}{3}$ to $\frac{C}{2}$ by turning on the switches and effectively shorting the two nodes together. This solution adds a fixed capacitive load by default and increments the load by bypassing the middle capacitor. It is also more susceptible to mismatch than the single-capacitor design, though less so than a single-ended capacitor solution.

To analyze the effectiveness of this circuit, begin by considering the following. If the objective of this circuit structure is to enable switching over the range of one gate delay to at least two gate delays, then the capacitive load added when all the switches are closed must be an effective doubling of the load when all the switches are open. Expressed mathematically,

$$\begin{aligned} t_{d\_min} &= 0.69RC \\ t_{d\_max} = 0.69R\left(C + \Delta C\right) &\geq 2t_{d\_min} \\ C + \Delta C &\geq 2C \end{aligned} \tag{4.11}$$

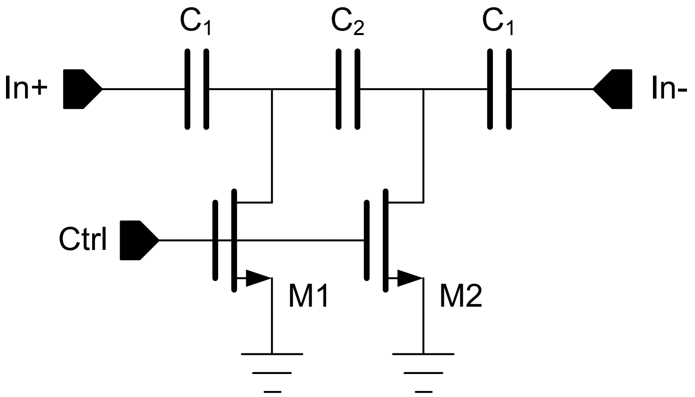


Figure 4.18: One unit of a differential switchable load capacitor configuration that uses transistors as switches to bypass the middle capacitor, varying the load seen from the inputs

Examining the switching unit circuit structure in Figure 4.18, when the switches are open, the total capacitance seen between the inputs is expressed as follows:

$$C_{open} = C_1//C_2//C_1 = \frac{C_1 C_2}{C_1 + 2C_2} \tag{4.12}$$

When the switches are closed, the capacitance changes to the following:

$$C_{closed} = C_1//C_1 = \frac{C_1}{2} \tag{4.13}$$

In practice, these individual capacitors C_1 and C_2 are made to be as small as possible to provide small unit steps, and the units are repeated many times to provide delay control. However, in this analysis, C_1 represents all of the C_1 unit capacitors lumped together (same for C_2), since the two cases that are needed for consideration are the minimum and maximum delay cases where the switches are either all open or all closed. Proportionally, $\frac{C_1}{C_2}$ is the same, regardless of whether the unit circuit or the lumped circuit is considered.

The total capacitance seen by the buffer's output in the minimum delay situation is given by the following:

$$C_{min} = C_{int} + C_{open} \tag{4.14}$$

where C_{int} is the intrinsic parasitic capacitance of the buffer's output and the input of the subsequent gate. The maximum delay situation connects the following total load to the buffer:

$$C_{max} = C_{int} + C_{closed} \tag{4.15}$$

Subtracting Equation (4.14) from Equation (4.15) gives the change in capacitance as

$$\Delta C = C_{int} + C_{closed} - C_{int} - C_{open} = C_{closed} - C_{open} \tag{4.16}$$

To at least double the load capacitance, ΔC must be greater than or equal to the minimum capacitance, C_{min}, in Equation (4.14):

$$C_{closed} - C_{open} \geq C_{int} + C_{open} \tag{4.17}$$

Substituting for C_{open} and C_{closed} from Equations (4.12) and (4.13), respectively, the following result is obtained:

$$C_1 \geq 2C_2 + 2C_{int}\left(1 + 2\frac{C_2}{C_1}\right) \tag{4.18}$$

This leads to the conclusion that $C_1 > 2C_2$ by some safety margin that is dependent on the output capacitance of the delay buffer. Since the devices in the delay buffer are large, both the drain capacitance and the gate capacitance of the following stage contribute to this intrinsic capacitance, so it is not insignificant.

The effect of selecting different ratios between C_1 and C_2 can be seen in Table 4.7, where everything is normalized to $C_2 = 1$. As $\frac{C_1}{C_2}$ is increased from 2, the size of the capacitor step increases as C_2 becomes more and more dominant over C_{open}. This translates into a loss of precision for a variable-delay buffer using this differential capacitor structure to change the delay. As seen in the table, the step size passes one unit capacitor when $\frac{C_1}{C_2}$ is slightly greater than 3.

Table 4.7: The effect of increasing the ratio of C_1 to C_2 on the capacitor step size

C_1	C_2	C_{open}	C_{closed}	ΔC
$2C$	C	$0.5C$	$1C$	$0.5C$
$3C$	C	$0.6C$	$1.5C$	$0.9C$
$4C$	C	$0.67C$	$2C$	$1.33C$

To determine if the step sizes provided by this circuit configuration are small enough to meet the needs of the variable delay buffer, simulations [1] were run where the variable load capacitance is adjusted one switching unit at a time. The delay through the buffer is computed in each case. The results, shown in Table 4.8 and Figure 4.19, confirm the finding that as the capacitor ratio is increased, the steps become less precise. Furthermore, these results show that the step sizes provided by this circuit configuration are rather large, and if an alternative solution is possible that would give more precise steps, it should be pursued.

Table 4.8: Simulation results of increasing the ratio of C_1 to C_2 on the delay step size

C_1	C_2	$\frac{C_1}{C_2}$	Delay Step
40 fF	20 fF	2	2.52 ps
60 fF	20 fF	3	3.81 ps
80 fF	20 fF	4	5.36 ps

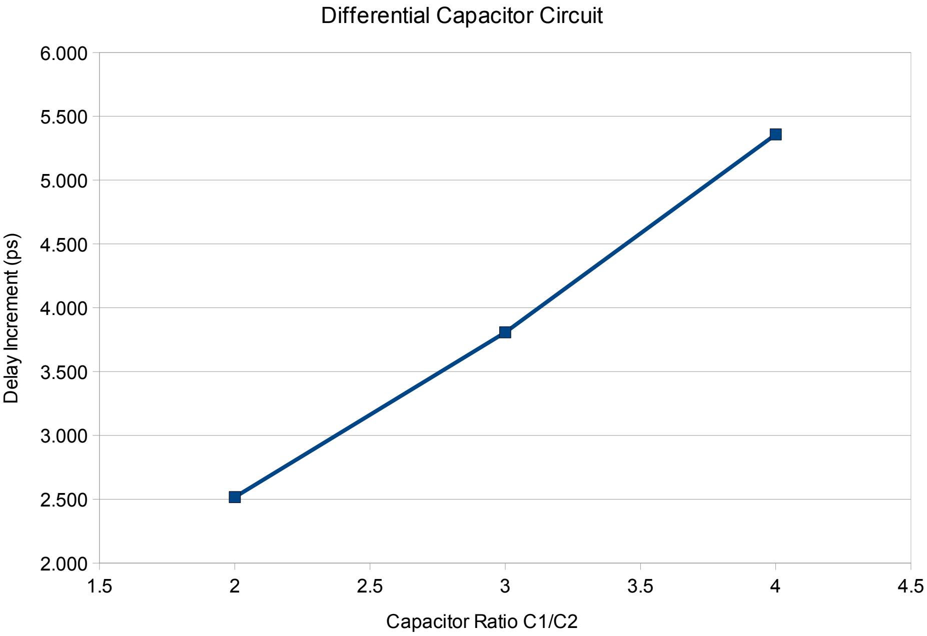


Figure 4.19: Simulation results of increasing the ratio of C_1 to C_2 on the delay step size [Note: these simulations were not run with the final delay buffer design]

[1] These simulations are not run using the final design for the delay buffer, but used the same buffer circuit for comparison purposes.

Noteworthy about this capacitor circuit is that there are three capacitors in the structure, and two must be more than twice the size of the third. Since capacitance is proportional to area, this means that for each repeated capacitance switch, there are more than 5 minimum-sized units of capacitance required. A single-ended solution can provide the same adjustment with just 2 units of capacitance in a structure that is somewhat more susceptible to mismatch. While increasing area is not generally of great concern in modern CMOS processes, this design is limited by the amount of area allocated to it on a shared fabrication run.

In the interest of reducing the area required by the design, a single-ended capacitor structure is explored. The single-ended circuit is shown in Figure 4.20. A circuit such as this must be attached to both the inverting and non-inverting outputs of the delay buffer. This introduces a susceptibility to mismatch, as the loads are independent from one another, but through careful layout these can be made to match reasonably well.

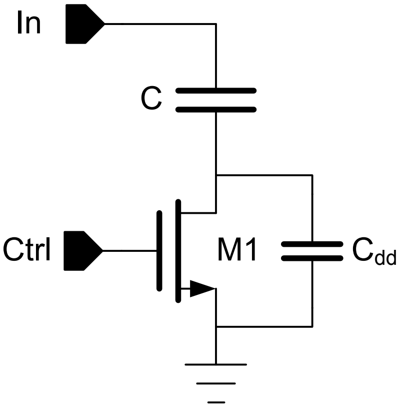


Figure 4.20: One unit of a single-ended switchable load capacitor configuration that uses transistors as switches to ground the bottom plate of the capacitor, varying the load seen from the inputs (In)

When the transistor switch is open, the bottom plate of the capacitor is connected in series with the total capacitance of the transistor's drain. Since the drain capacitance is at least an order of magnitude smaller in comparison to the MIM capacitor, when they are connected in series, the equivalent capacitive load is

$$C_{open} = C//C_{dd} \approx C_{dd} \tag{4.19}$$

The circuit contributes very little to the buffer's load, since C_{dd} is very small. When the transistor is turned on and the bottom plate of the capacitor is connected directly to ground, the load capacitance is

$$C_{closed} = C \tag{4.20}$$

and the change in load capacitance from open to closed is

$$\Delta C = C - C_{dd} \approx C \tag{4.21}$$

Using this circuit structure, it is clear that it is trivial to increase the delay of the gate twofold by doubling the load capacitance, so long as sufficient single-ended switchable capacitors are connected.

In the single-ended circuit, each additional capacitor of value C adds approximately C to the total load capacitance when its switch is closed. From Table 4.7 above, the differential circuit must be configured with a ratio $\frac{C_1}{C_2}$ of slightly greater than 3 to accomplish the same increment of loading. However, at this ratio, there is a much greater intrinsic capacitance loading the buffer, and so the resulting step size (proportional to $\frac{\Delta C}{C}$) is smaller for the differential circuit. This is confirmed by the simulation results [2] shown in Table 4.9. The delay adjustment step size for the single-ended circuit is matched by that of the differential circuit with ratio of approximately 4. Since it is possible that the ratio of C_1 and C_2 in the differential circuit would have to be this high anyway, due to the intrinsic capacitance, the precision of each of these implementations can be seen as roughly the same.

Table 4.9: Simulation results of the delay step size provided by the single-ended capacitor circuit

Delay Step
5.194 ps

The single-ended capacitor circuit is preferred for its ease to doubling the delay of the gate as well as its smaller area. Therefore, it is selected for the final implementation.

When simulated with the final design for the delay buffer, the capacitive loading increases the minimum delay from 24.6 ps to 39.7 ps. Additionally, the delay ad-

[2] These simulations were not run using the final design for the delay buffer, but used the same circuit for comparison.

justment steps provided by the circuit are much more precise than in the previous simulation, at about 500 fs. The results are shown in Table 4.10 and Figure 4.21.

Table 4.10: Simulation results of the delay step size provided by the single-ended capacitor circuit using the finalized buffer design

Min Delay	Delay Step
39.7 ps	0.497 ps

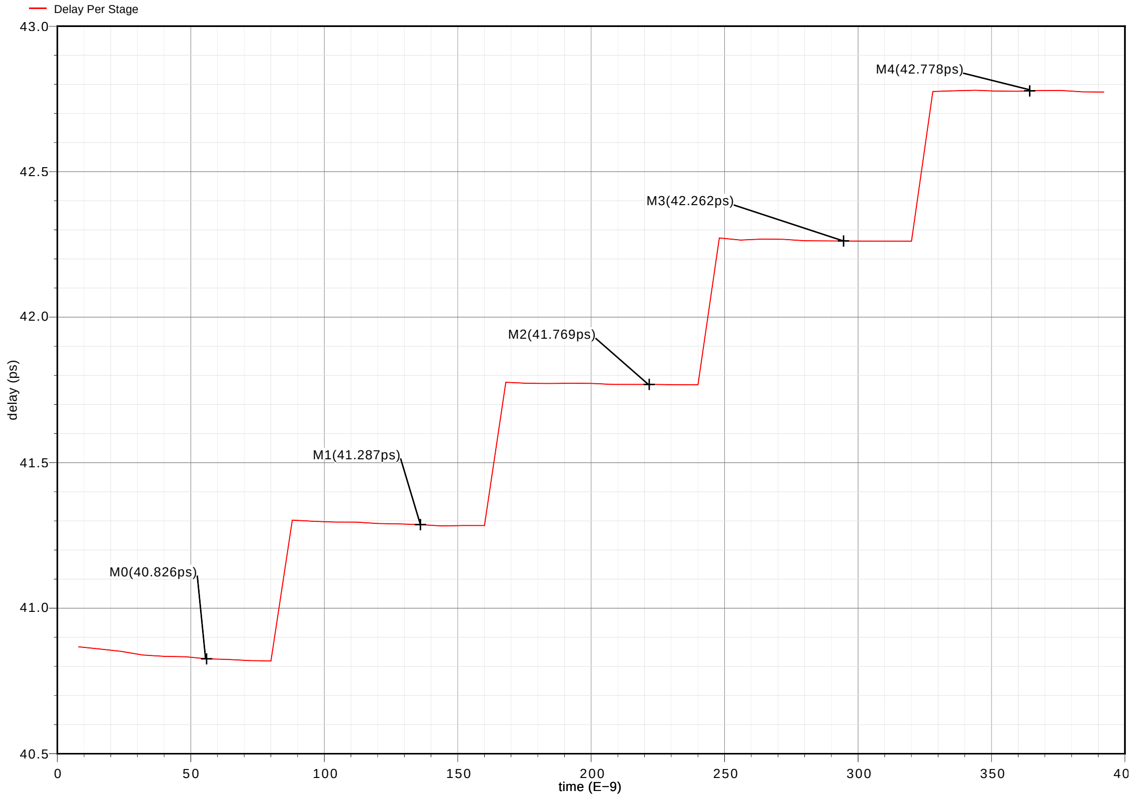


Figure 4.21: Delay results from the single-ended capacitor circuit connected to the delay buffer

Capacitor Array Design and Layout

The most reliable and well-characterized capacitor available in the 0.18 μm CMOS process is the metal-insulator-metal (MIM) capacitor. This capacitor uses the fifth metal layer and the capacitor top metal (CTM) layer, which are very close in proximity to one another and provide reasonably high capacitance per unit area.

The smallest capacitor that can be built is 4 μm by 4 μm [15], which results in a capacitance of approximately 20 fF, as shown in Figure 4.22. This is the value of capacitance used in the simulations in the analysis above. In the figure, the yellow rectangles represent metal layer 6 (M6), the light blue polygons represent M5, and the dark gray rectangle is the CTM.

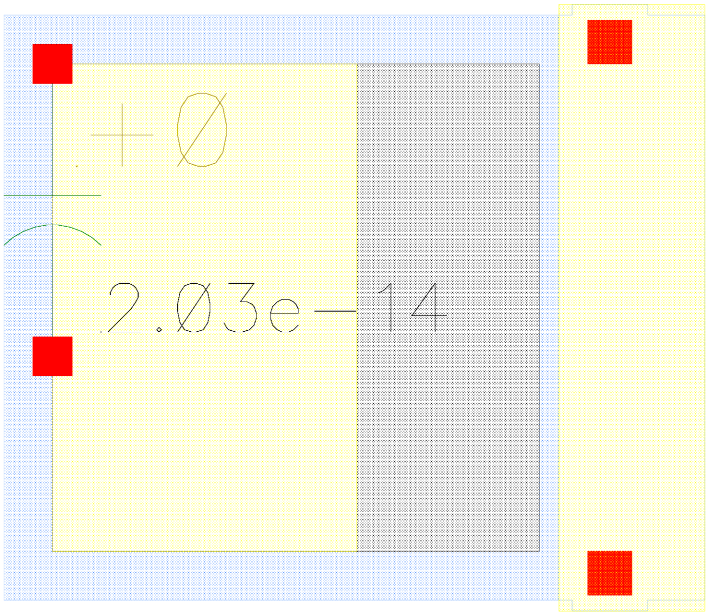


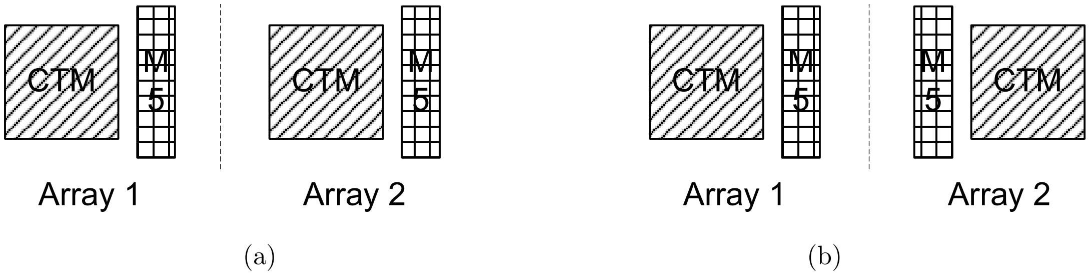
Figure 4.22: Extracted view of the smallest possible MIM capacitor in 0.18 μm CMOS

In order to increase the delay of the buffer from its minimum delay of 39.7 ps to 160.8 ps (to span the NAND gate steps of 121.1 ps), with steps of 0.497 ps, 243 steps are required. Since the circuit will be digitally controlled and it is desirable to have some overlap for safety margin, the circuit is designed with 256 capacitors, in binary groups, providing 8 bits of delay control. To solve any monotonicity errors, several extra unit capacitors are added so that they may be switched on or off under circumstances where the binary groups of capacitors provide non-monotonic delay steps.

The capacitors are carefully laid out in an array to ensure that the parasitics within the array are the same for each of the capacitors. To this end, the array of capacitors is surrounded by disconnected dummy capacitors on all sides so that the outer connected capacitors are surrounded by metal at the same relative locations and spacing as the ones in the middle of the array. This helps to ensure that each unit capacitor has the same value including all parasitics.

One further consideration when laying out the capacitor arrays is the matching from one side of the differential circuit to the other. In order to alleviate the potential for mismatch, the arrays are not mirrored, but laid out in the same direction on each output. The reason for this is that any misalignment occurring during the fabrication process will be applied in the same direction to each array, rather than in the relative opposite direction that would result from having them mirrored.

To help explain this, consider the diagrams in Figures 4.23(a) and 4.23(b). If the misalignment in the fabrication process has the CTM layer slightly to the right of where it is supposed to be (as illustrated in these diagrams), then in Figure 4.23(a), the CTM moves closer to the M5 interconnect beside it in both arrays. However, in Figure 4.23(b), the same misalignment results in the CTM in Array 1 moving closer to the M5 interconnect, while the CTM in Array 2 moves away from the M5 block. Since the proximity of two metals affects the capacitance between them, the parasitics seen by each of these CTM squares are not the same and a mismatch occurs between the differential outputs. Since this is very undesirable, the capacitors are laid out in the same orientation in both arrays.



(a)

(b)

Figure 4.23: (a) Capacitor array components laid out in the same orientation on both sides of the differential buffer output (b) Capacitor array components laid out symmetrically on each side of the differential buffer output

It is clear from the results in Figure 4.21 that there is a need for more precise delay adjustment than that provided by the variable load capacitance. Therefore, it is necessary to investigate varying the delay by adjusting the current in the buffer.

4.4.4 Delay Adjustment with Variable Current

The delay in the gate can be increased by decreasing the current in the delay buffer. By decreasing the tail current, the capacitive load at the output does not charge as quickly, resulting in a slower-slewing edge and therefore a greater delay. For precise control of the buffer's delay, very small decrements of the current are necessary.

This method of delay adjustment is not linear, and therefore cannot be relied on to span large ranges. Since the capacitor switching provides delay steps of 500 fs, and steps of 50 fs are required, then 10 steps must be provided by the current-adjustment circuit. However, to ensure that there are no gaps in the range and to use binary weightings in the digitally-controlled adjustment circuit, 4 bits of current adjustment are designed, giving 16 delay steps.

One method of varying the current in the delay buffer is to utilize switchable current-starving transistors in the tail. The current-starving transistors are placed in series with and below the current source transistor, as shown in Figure 4.24. The current-starving transistors have the effect of reducing the V_{GS} of the current source transistor, leading to a reduction in tail current. One large transistor that is always on is placed in parallel with many small unit transistors in binary groupings. The tail current is digitally controlled by switching the small transistors on to increase the current or off to reduce it.

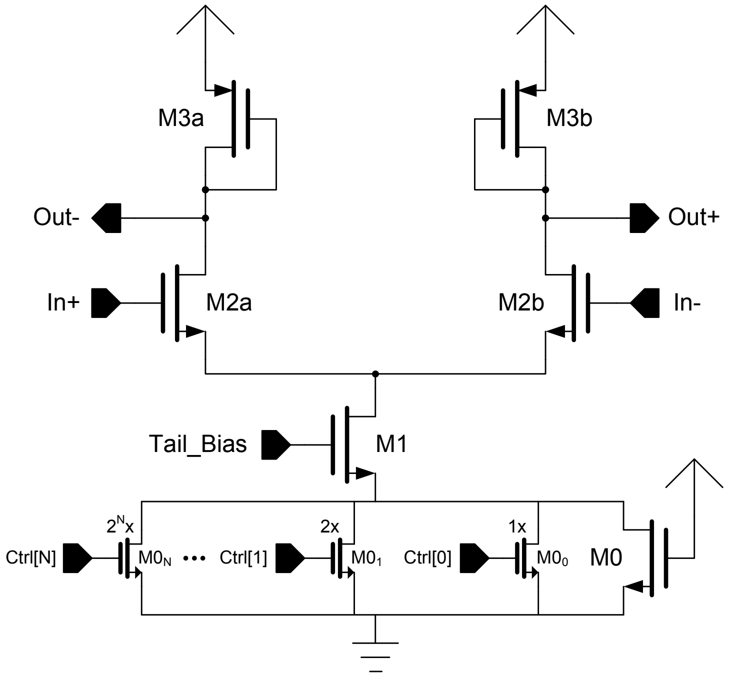


Figure 4.24: Complete differential delay stage with variable current-starving transistors in the tail

The required granularity of the delay is approximately

$$\frac{\Delta t_d}{t_{d_{min}}} = \frac{50\,\text{fs}}{40\,\text{ps}} = 0.00125 \tag{4.22}$$

Hence, if the assumption is made that the current varies linearly with the width of the current-starving device (which is not generally accurate, but to a first-order approximation over very small ranges of the device width, it is reasonable), then the ratio of the width of the small unit devices to the width of the large device should be the same as the required delay granularity:

$$\frac{W_{M0_a}}{W_{M0}} = 0.00125 \tag{4.23}$$

Taking the small device's size to be 420 nm for easy implementation (this is the minimum size transistor that can be laid out with the standard parameterized cell), this means that the large device must be 336 μm wide.

In simulation, it turns out that this gives far too great of precision and delay steps that are almost indistinguishable. Through a series of simulations, the widths of the current-starving transistors are optimized. The final values are shown in Table 4.11. Additionally, there are 5 bits of control designed into this circuit, to allow for characterization of the linearity of varying the current.

Table 4.11: Current-starving transistor sizes

	M0	M0_N
W	76 μm	420 nm
L	180 nm	180 nm

To avoid monotonicity errors in the delay steps resulting from the binary switching of the current-starving transistors, several extra unit transistors are added in parallel so that a switching scheme can prevent these non-idealities in practice. While these devices are provided, the implementation of detection and correction of the monotonicity errors is beyond the scope of this project.

Simulation results for the adjustment of the delay using variable current starving are shown in Figure 4.25 and Figure 4.26. In Figure 4.25, the delay steps provided by the current switching circuit are demonstrated, showing that the delay steps are about 32 fs in size. In Figure 4.26, the output and noise voltage for the worst-case control code for the circuit with variable current is shown. The jitter can be calculated as follows:

$$t_j = \frac{v_n}{\frac{dV}{dt}} = \frac{1.047 \times 10^{-4}}{1.981 \times 10^{9}} = 52.8\,\text{fs} \tag{4.24}$$

This is the overall jitter for a circuit containing four copies of the buffer in series (as explained in Appendix B); the jitter per stage is then $\sqrt{\frac{(52.8\,\text{fs})^2}{4}} = 26.4\,\text{fs}$. The delay performance and worst-case jitter results are summarized in Table 4.12.

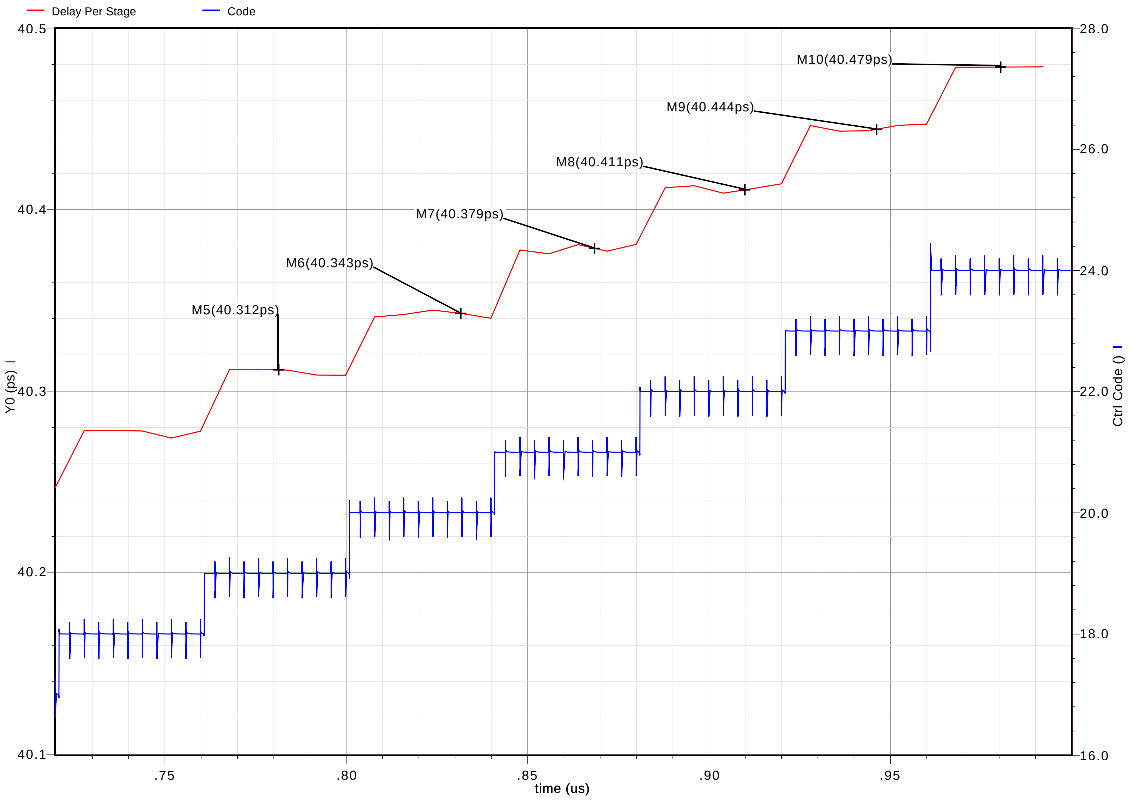


Figure 4.25: Delay steps provided by decreasing the buffer's tail current

Table 4.12: Summary of performance obtained by adjusting only the current in the delay buffer

Min. Delay	Max. Delay	Average Delay Step	Worst-case t_j
39.7 ps	40.8 ps	34.4 fs	26.4 fs

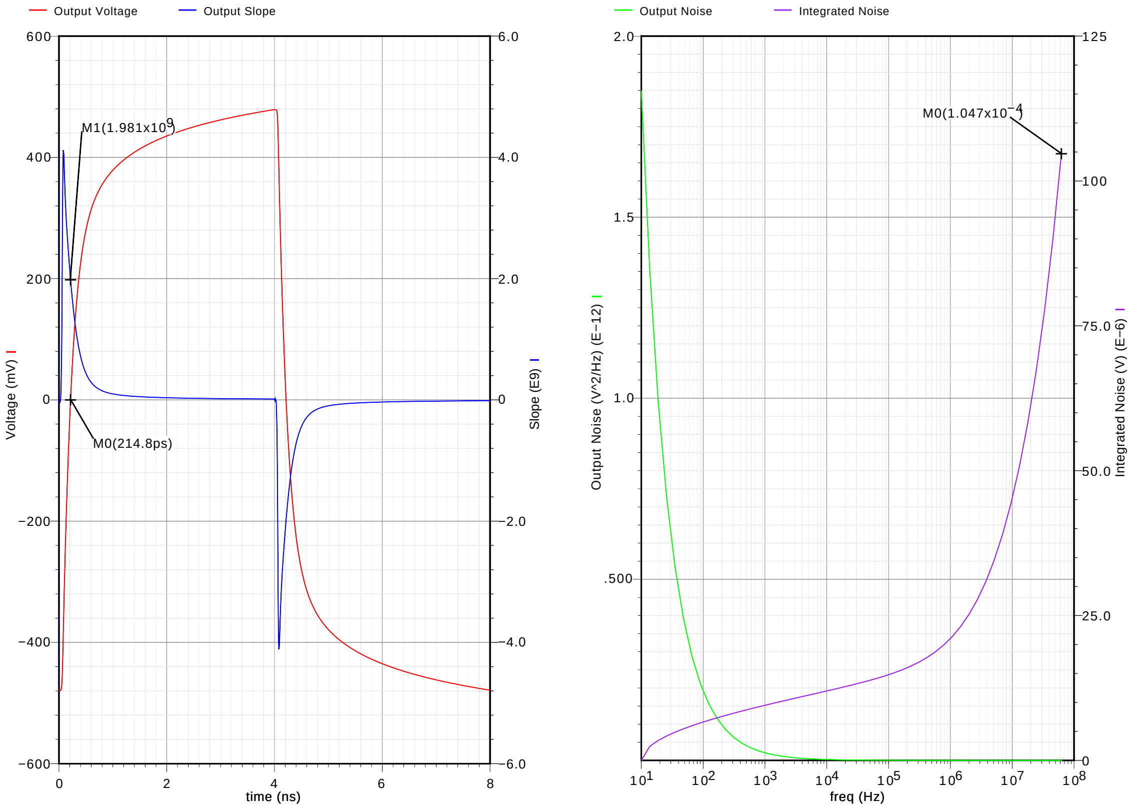


Figure 4.26: Output voltage slope and noise voltage for the worst-case control code using current switching

4.4.5 Variable-Delay Buffer Simulation Results

A plot of the delay performance of the variable-delay buffer using both current and capacitor switching is shown in Figure 4.27. In these simulations, the current-adjusting code and the load capacitance-adjusting code are concatenated, with the current-starting code being the least significant bits.

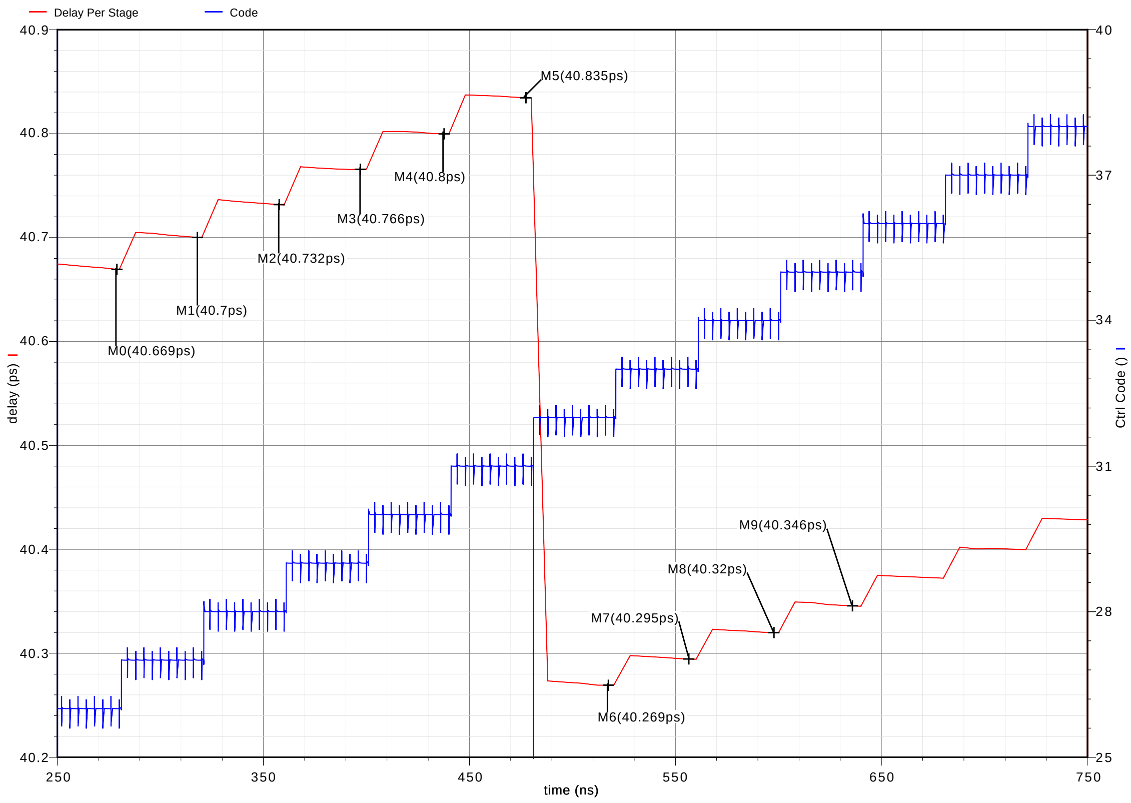


Figure 4.27: Delay steps provided by decreasing the buffer's tail current and adjusting the load capacitance

A longer plot showing two capacitor-adjusted delay steps spanned by current steps is shown in Figure 4.28. It is apparent in this plot that the steps provided by the current-starving circuit are non-linear.

A plot of the output and noise of the variable-delay buffer for the worst-case control code is shown in Figure 4.29. Using the values for the slope and the noise voltage at each of the rising and falling edges, the jitter can be determined as follows:

$$t_j = \frac{v_n}{\frac{dV}{dt}} = \frac{0.839 \times 10^{-4}}{1.481 \times 10^{9}} = 56.6\,\mathrm{fs} \tag{4.25}$$

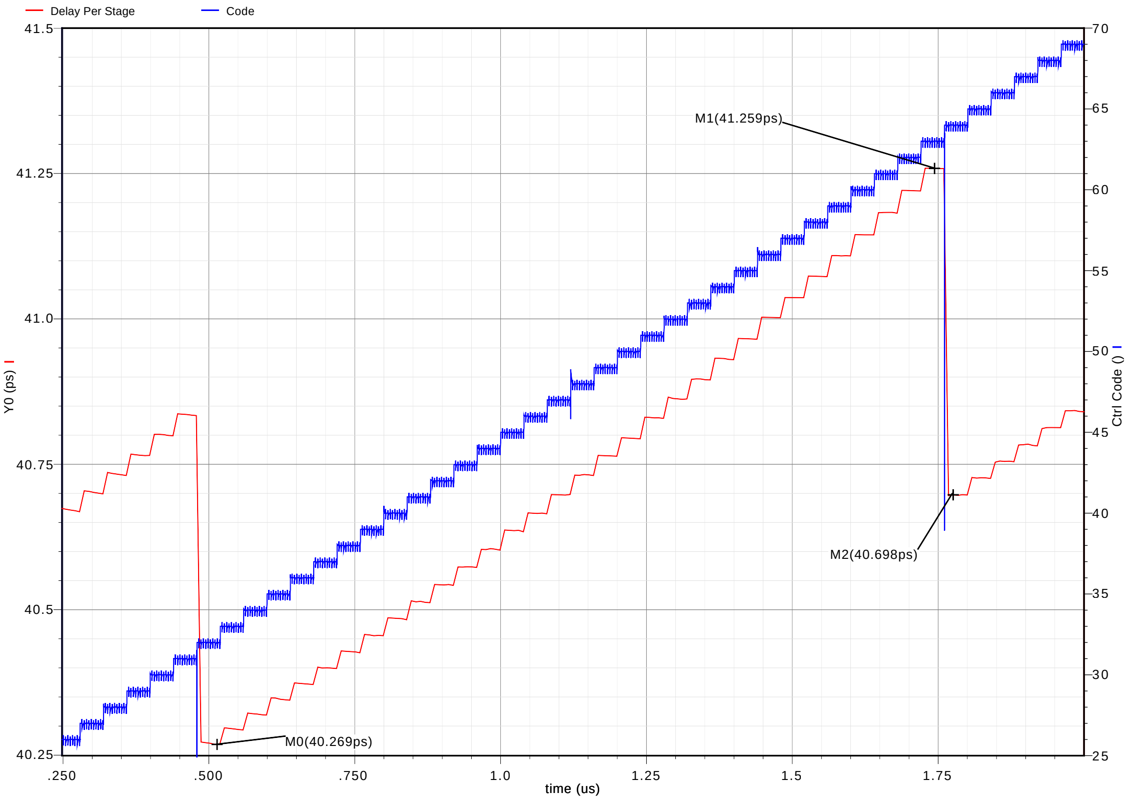


Figure 4.28: Delay steps provided by decreasing the buffer's tail current, spanning one step provided by adjusting the load capacitance

Once again, this simulation was run with four of the buffers in series (see Appendix B). Therefore, the jitter per stage is $\sqrt{\frac{(56.6\,\text{fs})^2}{4}} = 28.3\,\text{fs}$. Combined with the worst-case output noise of the NAND gate, the total output jitter is estimated to be $t_j = \sqrt{(28.3\times10^{-15})^2 + (76.2\times10^{-15})^2} = 81.3\,\text{fs}$. A summary of the delay and jitter characteristics of the variable-delay buffer is shown in Table 4.13.

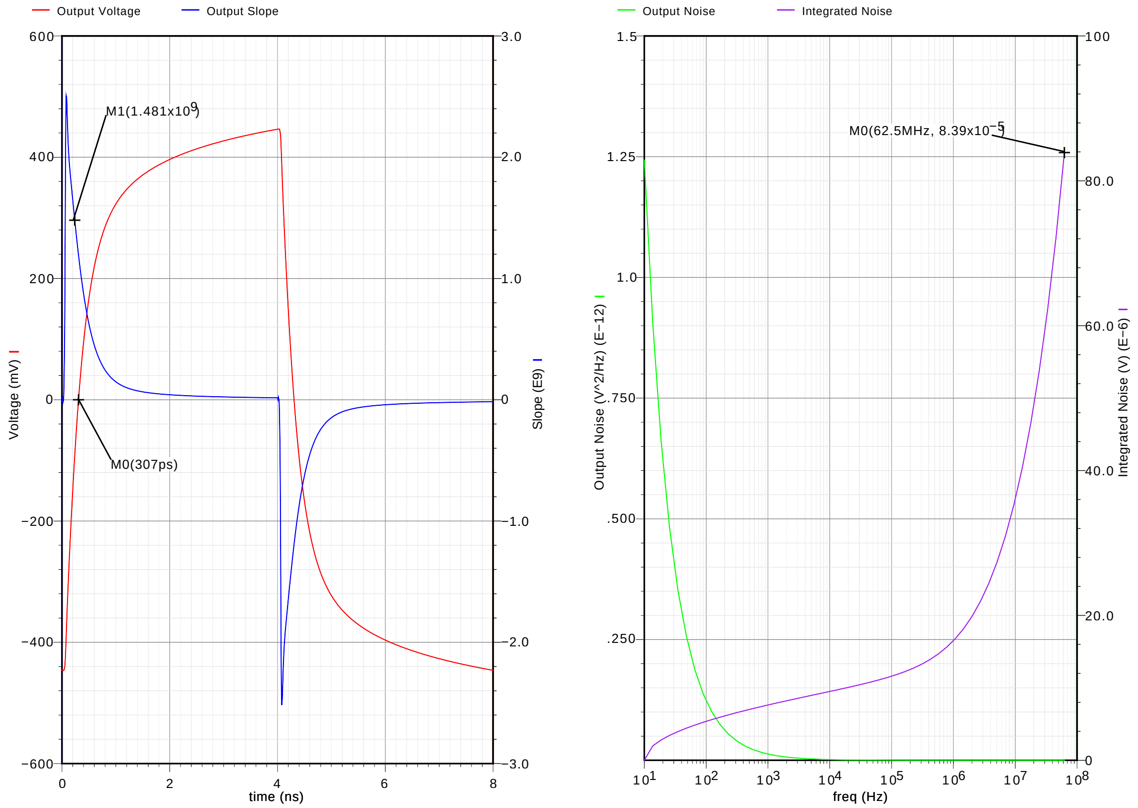


Figure 4.29: The output voltage and slope (left) and the integrated noise on the rising edge of the output (right) for the worst-case control code in the variable-delay buffer

Table 4.13: Summary of performance obtained by adjusting the load capacitance and the current in the variable-delay buffer

Min. Delay	Max. Delay	Average Delay Step	Worst-case t_j
39.7 ps	229.4 ps	34 fs	28.3 fs

4.5 Complete Clock Adjustment Circuit Design

The complete system is implemented as illustrated in Figure 4.30. There are two levels of NAND gate switching, providing three different delay paths, followed by a variable delay buffer with 8 bits of capacitor switching and 5 bits of current switching to provide fine delay adjustment precision. The control code shown in the subsequent results can be varied from 0 to 8191.

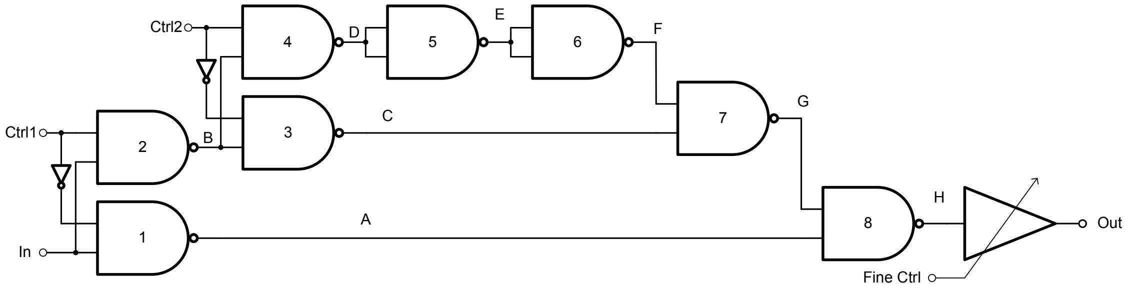


Figure 4.30: A system for providing four different delay paths using a NAND gate switching structure

The power consumption of this circuit can be reduced by further decreasing the sizes of the devices, with a cost of increased jitter. Due to lack of time, the power consumption is not optimized in this design.

4.5.1 Simulation Results

Simulations are done with an input frequency of 125 MHz. The test ADC available in the lab operates at this maximum sample rate, so this is the fastest that the circuit will be required to operate. In practice, interleaved ADCs often operate at much higher frequencies, but this is merely a test circuit for proof of concept, so this frequency is adequate.

A plot of the delay performance of the complete circuit, as provided by varying the tail current and the load capacitance, is shown in Figure 4.31. In this simulation, the shortest path through the NAND gates was activated, and the control code into the variable-delay buffer is varied as shown. It is noteworthy that in some of the early codes in these simulations, the delay is slowly decreasing; this is due to some transients that have not settled out in the simulator. The exact source of these transients has not been determined.

A second plot for a different control value is shown in Figure 4.32. Here, the longest path in the NAND circuit is activated, and the control code is applied to

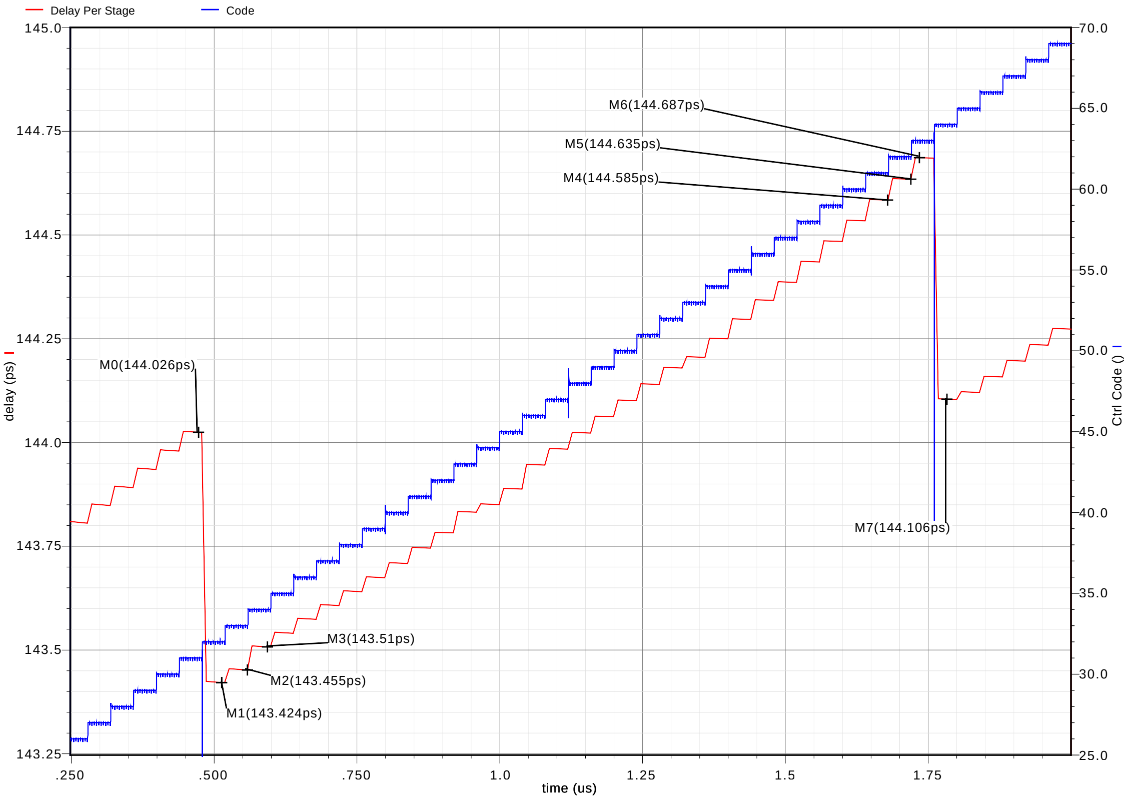


Figure 4.31: Delay steps provided by increasing the buffer's tail current, spanning one step provided by adjusting the load capacitance, with the complete delay adjustment circuit connected. The shortest NAND path is activated and the control code is incremented from 26 to 69.

the variable-delay buffer as shown. As can be seen in this plot, the simulations were done without monotonicity error detection and correction. The detection and correction of these errors is beyond the scope of this work; the means is provided in the design but the algorithm is not implemented.

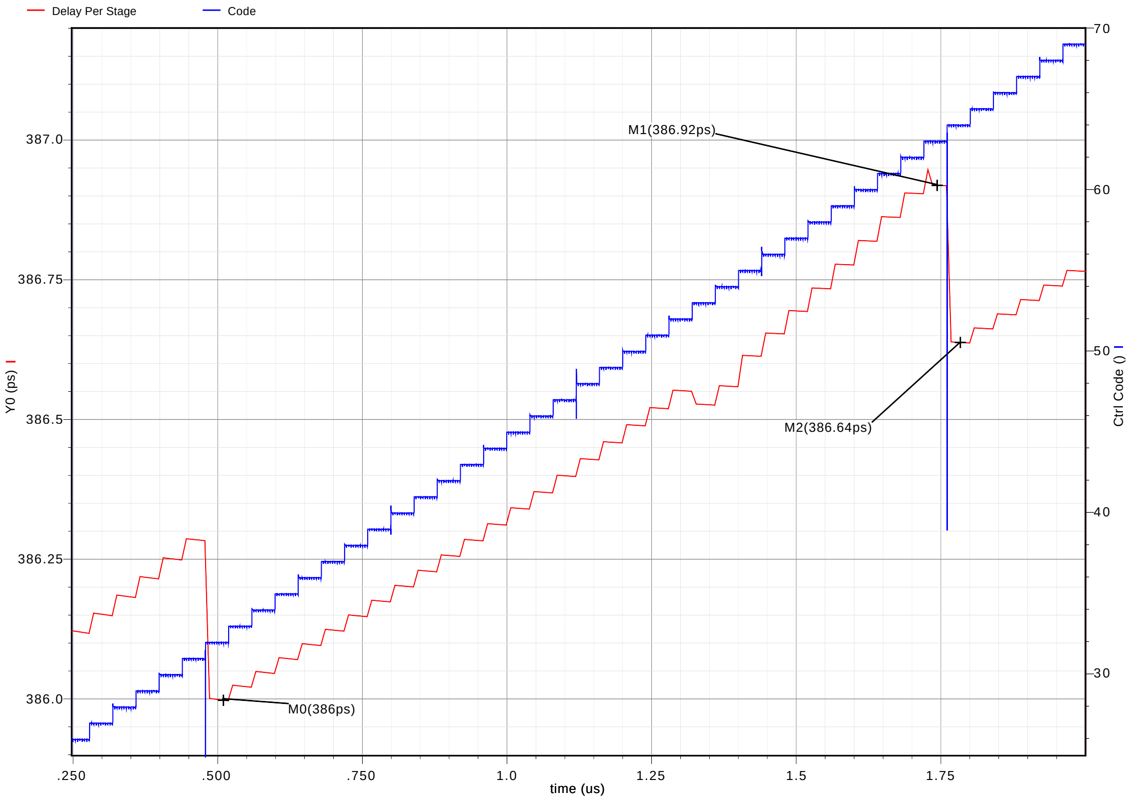


Figure 4.32: Delay steps provided by increasing the buffer's tail current and by adjusting the load capacitance, for the longest NAND path and control codes 26 to 69

A third plot of the delay steps is shown in Figure 4.33. In this simulation, the code was incremented much higher than the previous simulations. From this result, it is evident that as the capacitive load is increased, the current-starving steps become less precise.

It is worthwhile to note that in practice, this circuit will not be used as tested in these simulations. Specifically, the control code will not be changed every few clock cycles, since it is unlikely that there will be a significant amount of high fequency drift in the timing mismatches in interleaved ADCs. These simulations are done to demonstrate the ability to vary the delay by changing the control code.

The output slope and noise voltage of the circuit for the worst-case control code are shown in Figure 4.34. Using these results, the jitter can be calculated as follows:

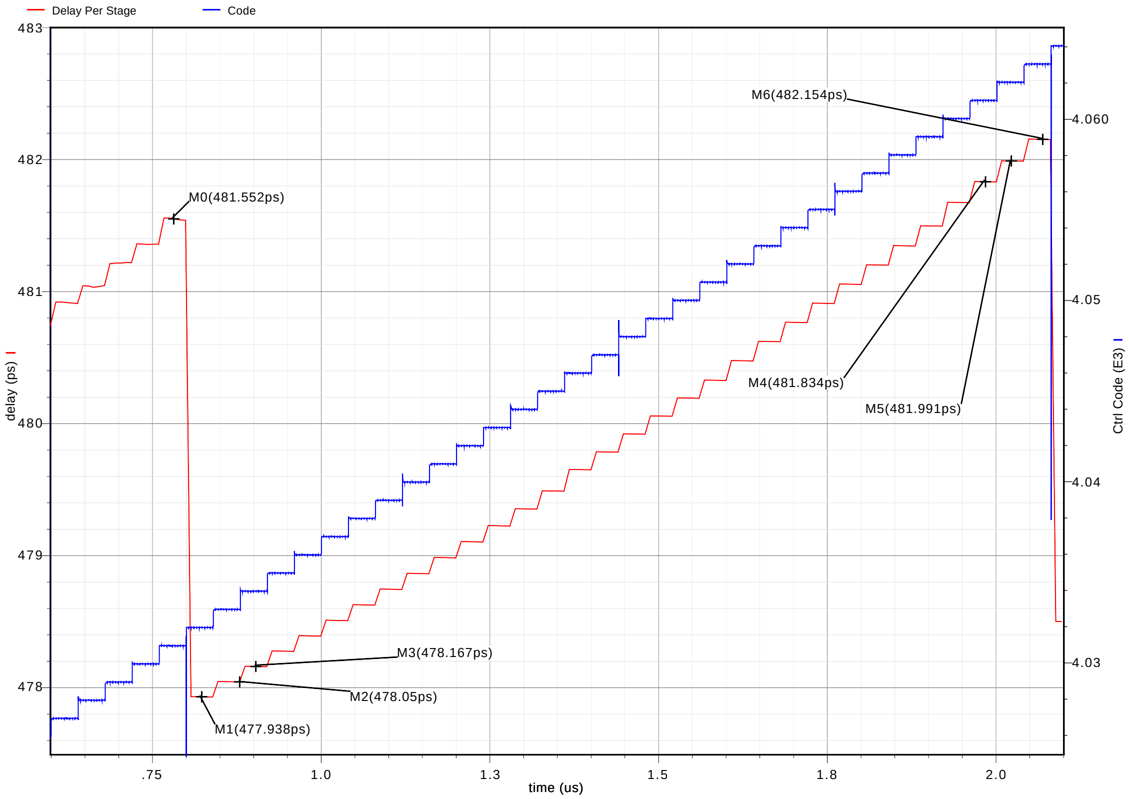


Figure 4.33: Delay steps provided by increasing the buffer's tail current and by adjusting the load capacitance, for the longest NAND path and control codes 4027 to 4064

$$t_j = \frac{v_n}{\frac{dV}{dt}} = \frac{1.736 \times 10^{-4}}{1.751 \times 10^{9}} = 99.1\,\text{fs} \tag{4.26}$$

This is slightly higher than the estimated value of 81.3 fs calculated previously. A summary of the delay and jitter characteristics of the complete circuit is found in Table 4.14. The average step size is calculated from a sample of the timing steps at different control codes.

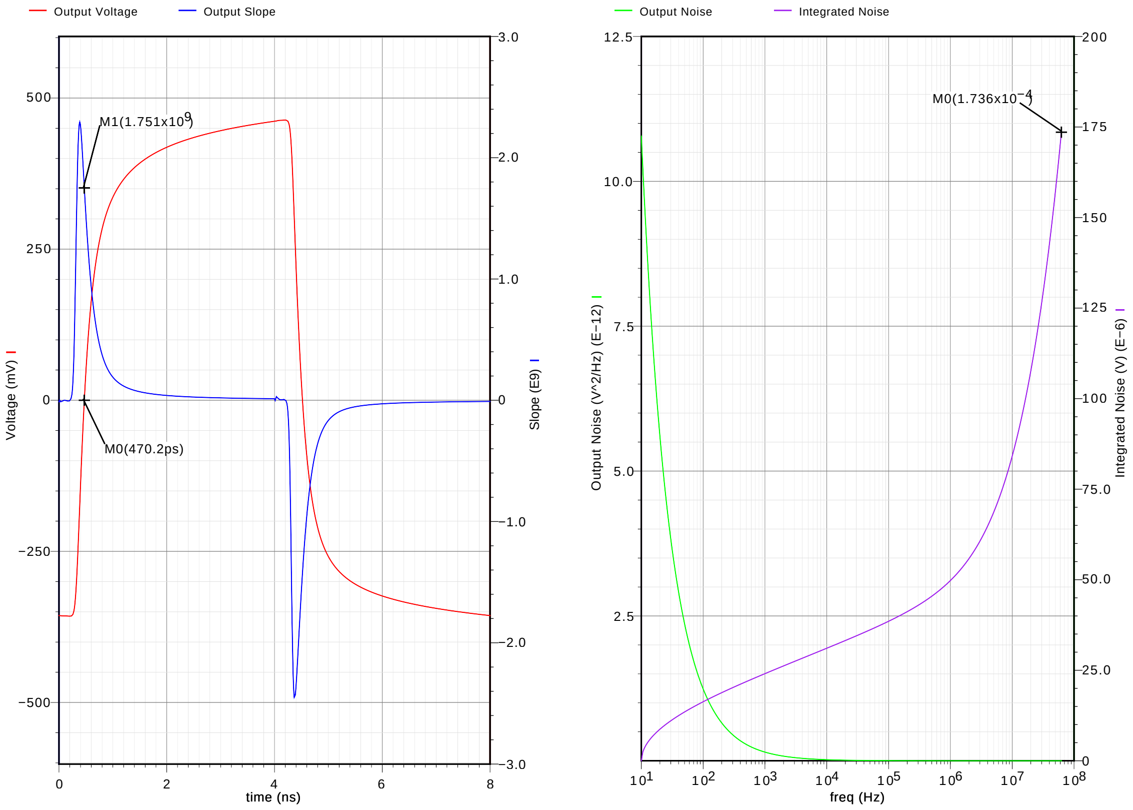


Figure 4.34: The output voltage and slope (left) and the integrated noise on the rising edge of the output (right) for the worst-case control code in the complete timing adjustment circuit

4.6 Design for Testability

In designing this circuit for proof of concept, the ability to demonstrate its functionality is a requirement. Therefore, it is necessary to provide a method by which the circuit can be tested so that experimental results can be shown.

Table 4.14: Summary of performance obtained by selecting different paths in the NAND circuit and adjusting the load capacitance and the current in the variable-delay buffer

Min. Delay	Max. Delay	Average Delay Step	Worst-case t_j
142.6 ps	483.0 ps	92 fs	99.1 fs

Using current lab equipment, it is not possible to directly measure small increments of time on the order of tens of femtoseconds. Oscilloscopes with sampling rates of tens of gigahertz are available. The lab facility is equipped with an 8 GHz oscilloscope. At 8 GHz, the sampling interval is 125 ps, which is 4 orders of magnitude larger than the timing precision required. Therefore, an indirect way to measure the precision is required.

The target application for the variable delay circuit is interleaved ADCs. Therefore it makes sense to test the circuit in an interleaved ADC, and the effects of the timing skew can be used to measure what the timing mismatch is, as described in Chapter 2. Adjustment can be done manually to reduce the spurious tones introduced by the mismatch and verify that the circuit can adjust the edge of the clock precisely enough. However, it is impossible to design an ADC core in the time available, and the silicon area is not allocated, even if an ADC core is available. Furthermore, adding that level of complexity to the system greatly increases the risk of failure.

One possible solution is to implement the variable delay circuit as a discrete component and use it to interleave two discrete ADCs with each other. This would allow for the measurement of the timing mismatch characteristics between the two ADCs and the ability of the variable-delay circuit to mitigate those effects. However, between any two ADCs there are also other mismatches such as gain and offset mismatches that have similar effects on the performance of an interleaved system. In particular, gain mismatches have similar symptoms to timing mismatches [1]. These mismatches would have to be dealt with in some other way in order to accurately prove the functionality of the timing adjustment circuit.

In order to eliminate any possible source of mismatch from an interleaved system other than timing, the same ADC must be used for all channels of an interleaved ADC system. That is, one discrete ADC must be interleaved with itself using a mechanism where the sample clock is split into two paths, one of which has an adjustable delay (using the variable delay circuit) and the other of which has a fixed delay equal to approximately the middle of the range of the variable delay. The fixed delay path will be accomplished by using a second copy of the delay buffer with its inputs fixed such that the delay is in the middle of the range.

The circuit recombines the outputs of the two delay paths such that each path is selected on alternating clock periods. Thus, the sample timing is determined alternately by one clock path, and then the other, and the timing mismatch between the two clock paths is a model for the timing mismatch between two clock paths to different ADCs in a true interleaved system.

The generation of interleaved timing signals is provided by a simple circuit composed of NAND gates shown in Figure 4.35. A clock is connected to one input and a clock with half the frequency is connected to the other input. The slower clock is used as the controlling input of the NAND gates, and because it is inverted in one path, each input NAND gate is forced high for half of the slow clock period (which is equal to one period of the fast clock), alternately. While one NAND gate's output is forced high, the other NAND gate passes through the clock. Thus, one period of the clock is sent to the variable delay path and one period is sent to the fixed delay path.

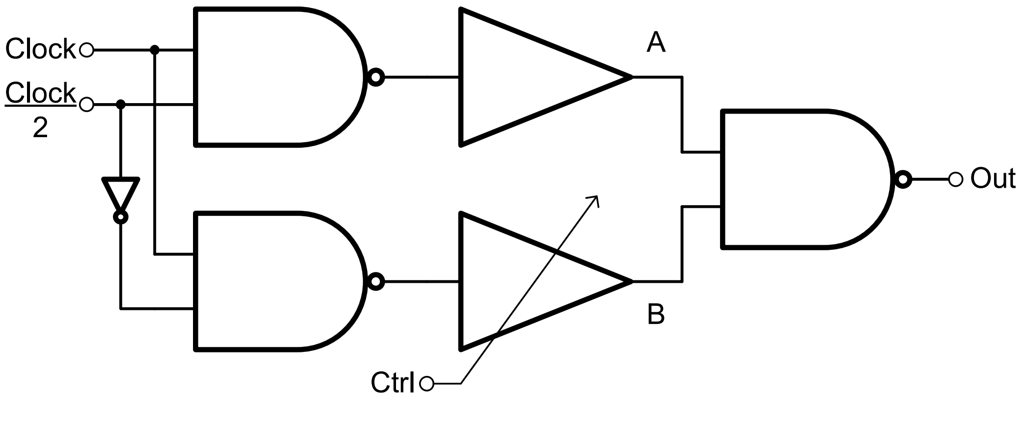


Figure 4.35: A method of splitting the clock path and recombining it in order to interleave two clock paths to the same ADC

The outputs of each of these clock paths look like a pulse train with a duty cycle of approximately 75%. This is useful for the recombination of the two paths into an interleaved clock signal, because this function can be performed by a simple NAND gate, as shown in Figure 4.35. When one output is switching, the other output is high, so the NAND gate is always in the pass-through mode. The output of the NAND gate appears to be a normal clock, as illustrated in Figure 4.36, although the edges may not be perfectly spaced due to timing mismatch in the two clock paths.

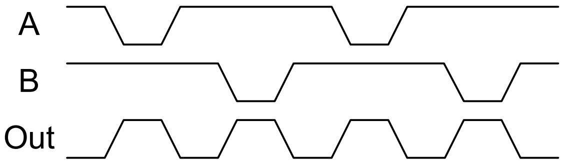


Figure 4.36: Pulse train recombination through a NAND gate

Since a design for an MCML NAND gate already exists, the only extra design effort that is required is the division of the clock. A clock can be easily divided using a D flip-flop with its inverting output connected to its input, and its clock connected to the fast clock in the system. An important consideration when designing this component is that the half speed clock must not toggle at the same time as the fast clock, so that glitches do not propagate through the system. The delay through the D flip-flop must be sufficient to avoid this problem.

Using this design, the effectiveness of the circuit can be tested and verified. A complex system of mismatches is simplified so that the source of mismatch under examination can be isolated from other nonidealities, providing accurate and verifiable results. However, the circuit requires the design of a D flip flop.

4.6.1 Design of an MCML D Flip-Flop

A flip-flop is composed of two stages of D latches, with clocks inverted from one another. An MCML D latch is shown in Figure 4.37. The device sizes were designed using the same procedure as with the NAND gate and the delay buffer. Since the D latch is a very small part of the overall circuit and time is of the essence, only a modest amount of time is spent optimizing the D latch. The tail current in the D latch is 4mA, and the final device sizes are found in Table 4.15.

Table 4.15: MCML D latch transistor sizes

	M0	M1a,b	M2a-d	M3a,b
W	200 μm	60 μm	60 μm	120 μm
L	0.18 μm	0.18 μm	0.18 μm	0.18 μm

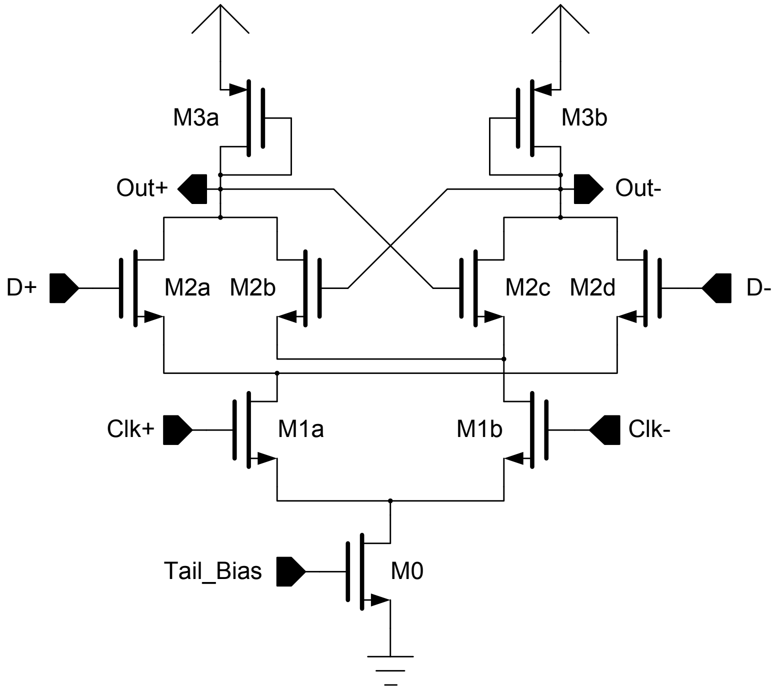


Figure 4.37: MCML D latch

4.7 Implementation

With the results at an acceptable level, the circuit was laid out for final testing and fabrication. The final block diagram of the system is shown in Figure 4.38.

One important consideration during the layout process is the distribution of power to the gates. Since this design was not optimized for power consumption (for lack of time), the current flowing through each gate is very high. With high currents come increased I^2R losses. In order to avoid supply voltage droop due to these losses, it is important to ensure that there is a very low-resistance path from the power supply and ground pads to the gates. Since the resistivity of the metal layers in the $0.18\,\mu$m process is around $0.078\,\Omega/\Box$, and ideally there should be no more than $0.1\,\Omega$ to $0.2\,\Omega$ between the supply pads and the circuits, there should be no more than about $1\,\Box$ to $2\,\Box$ of metal between the pads and the furthest gates. With this in mind, the top two metal layers are used as high-density grids in conjunction with a multiple-layer ring around the outside of the core to distribute power and ground.

The circuit is laid out for implementation within a $1\,\text{mm}\times 1.2\,\text{mm}$ rectangle allocated by Canadian Microelectronics Corporation (CMC) on a shared fabrication run, with the identifier code ICFWTJ4S. An image of the final layout is shown in

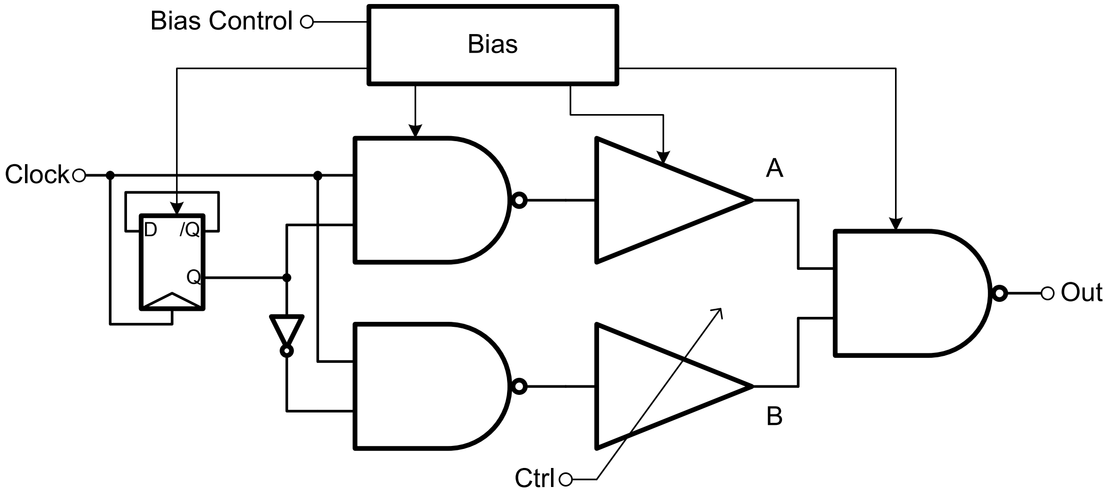


Figure 4.38: The complete circuit for implementation

Figure 4.39. This image does not include the top two metal layers, as their coverage would obscure the majority of the image. The design is submitted for fabrication on the scheduled September 15, 2010 fabrication run. Further details on the final implementation can be found in Appendix A.

4.8 Conclusion

The clock timing adjustment circuit is implemented using differential circuit structures to mitigate the effects of supply noise. The fixed-delay circuit is implemented as a series of NAND gates, and the NAND gate is implemented in MCML form, with an emphasis on low noise. The variable-delay circuit is implemented using variable load capacitors for the most significant bits of the adjustment control signal and tail current adjustment for the least significant bits. The circuit is designed for low noise and precise delay adjustment. Simulation results show that the circuit is capable of adjusting the delay to a precision of approximately 92 fs over a range of 143 ps to 483 ps, with an RMS jitter at the output of 99.1 fs for the worst-case control code.

A method for testing the circuit is presented that involves interleaving an ADC with itself using two separate clock paths to allow measurement of timing increments on the order of tens of femtoseconds. The clock edge adjustment circuit is incorporated into both clock paths of this testing circuit. The final implementation of the chip contains this testing circuit with one path having a digitally-controlled

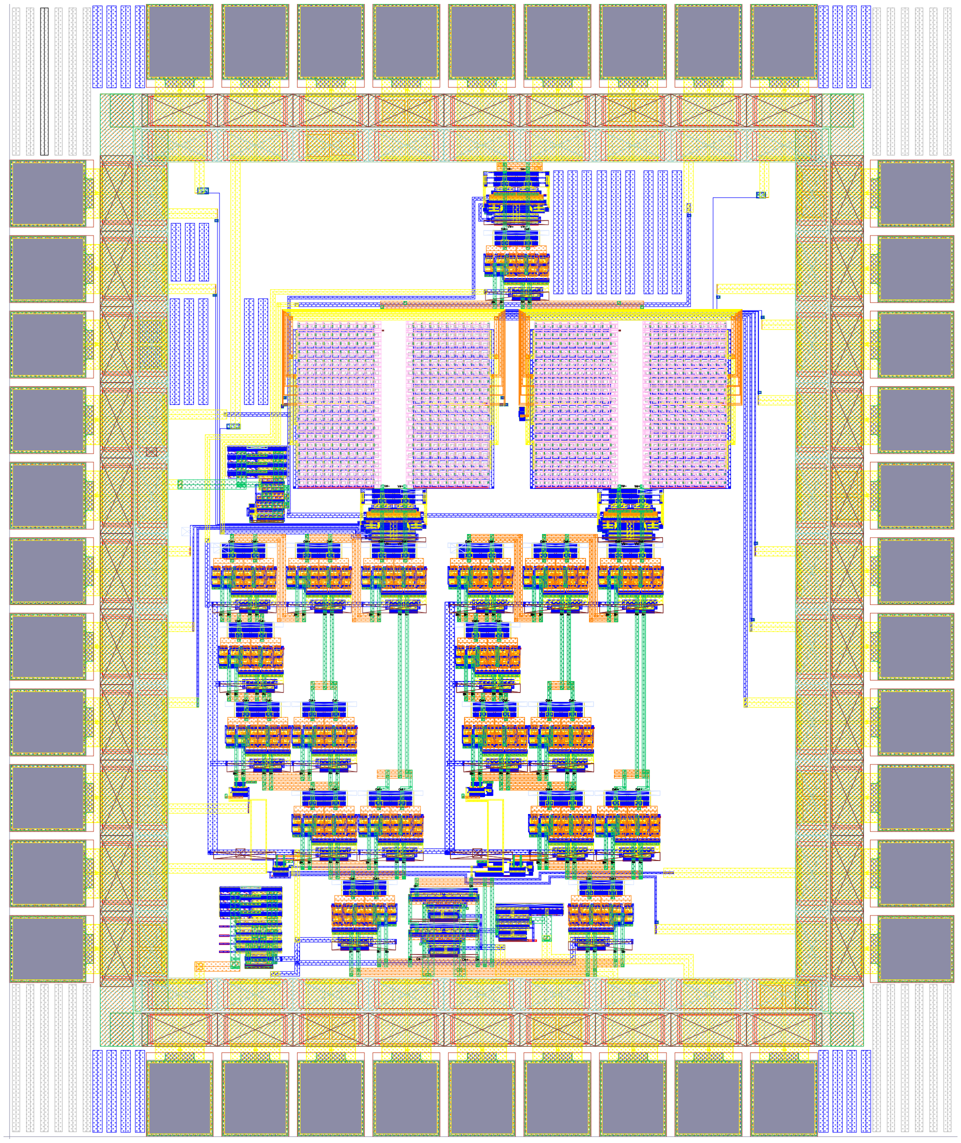


Figure 4.39: Layout of the dual clock path circuit

delay and the other path having a fixed delay. The circuit is laid out and submitted for fabrication.

Chapter 5

Post-Layout Simulation Results

The design has been submitted for fabrication on a shared run through the Canadian Microelectronics Corporation (CMC) Microsystems. Since testing results are not available, post-layout simulations are done to verify that the system works as designed.

This chapter shows several different simulation results for the post-layout simulations of the delays provided by the circuit. Then it shows simulation results for the noise in the circuit, from which the output jitter can be calculated.

5.1 Delay Steps

The simulations were designed to allow for measuring the delays provided by a number of different control codes. In order to show the delay steps without running one simulation per control code, the input control code was set up to increment slowly over time, allowing a consecutive sequence of control codes to be tested. In order to provide an accurate and repeated measurement of the delay for each control code, the period of the control code incrementation was set to allow for ten periods of the input clock in each control code.

Three plots of the delay performance for the extracted delay unit show that the system provides precise delay steps across the entire range of delay tuning. Each plot shows a different path activated in the NAND switching circuit as well as a different set of codes swept in the variable-delay buffer. In Figure 5.1, the shortest NAND path is activated and the variable-delay buffer is swept through relatively low delay codes from 26 to 69[1]. In Figure 5.2, the medium-length NAND path is activated and the variable-delay buffer is swept through a set of midrange control codes. In Figure 5.3, the longest NAND path is activated and the variable-delay buffer is higher in its controllable range. The results are summarized in Table 5.1. The average step size is calculated from a sample of the timing steps at different control codes.

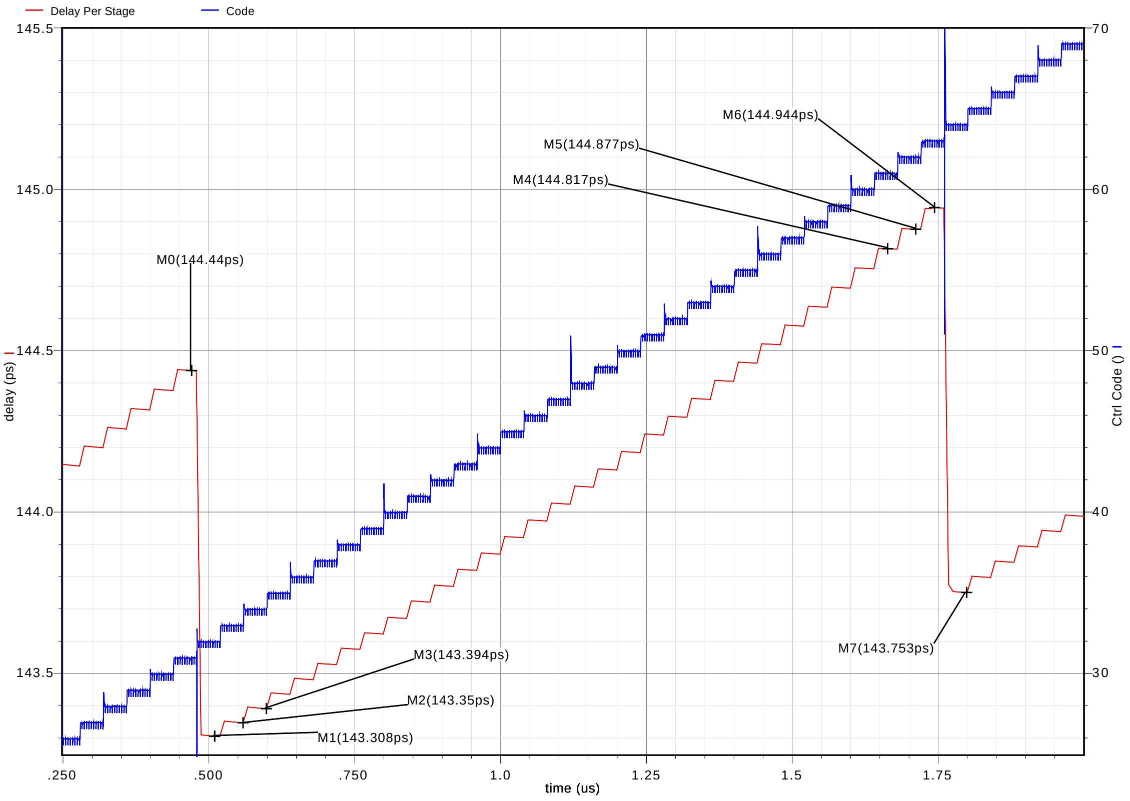


Figure 5.1: Delay steps (red) for control codes (blue) near the minimum using the extracted variable-delay circuit

It is worthwhile to note that although the range is much greater than the requirement of 50 ps, the average precision is less than the requirement of 50 fs. However, at the low end of the control range, the precision is on the order of 35 fs, while at the high end of the range, the precision is on the order of 120 fs. Due to the time constraints of the fabrication process, no further optimization was possible. Since this is a proof-of-concept design, these results are acceptable, and a future iteration of the design can improve this precision.

[1] To cover this range, the simulation takes approximately 3 hours to complete. Therefore, to cover the entire range of delays (i.e. control codes 0 to 8191 over each NAND gate step), it is estimated that the simulation would take almost 70 days to complete.

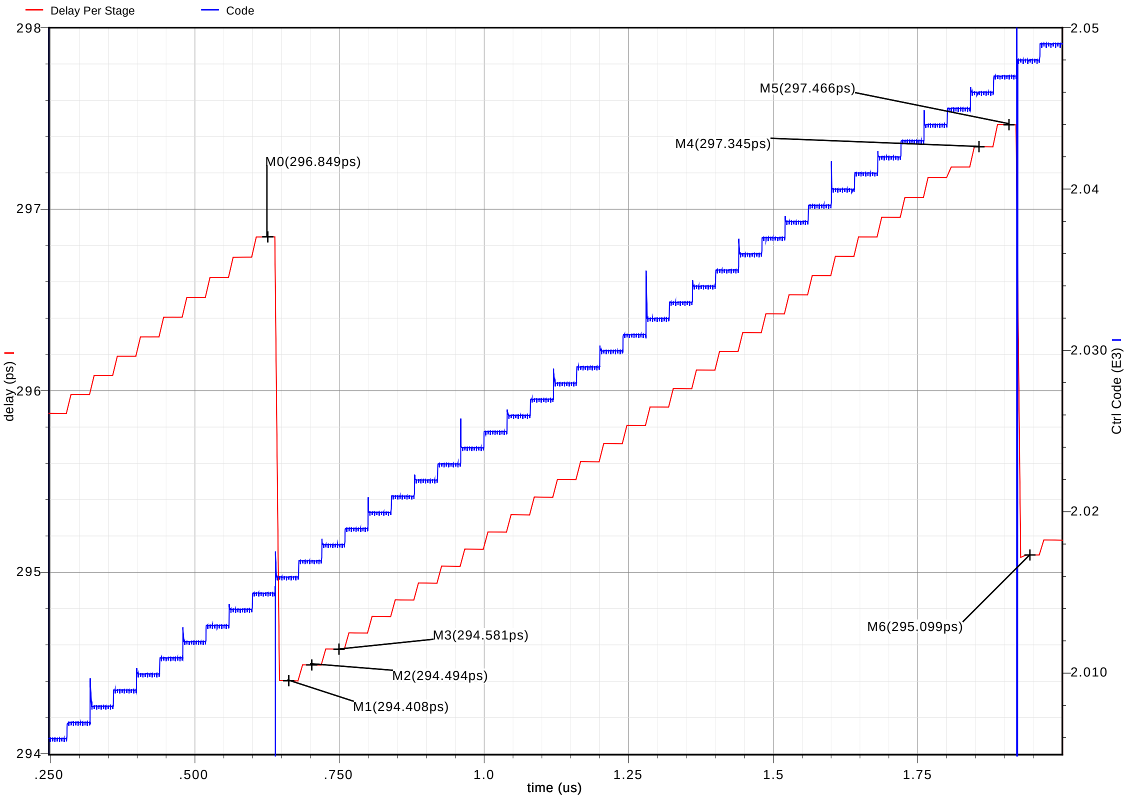


Figure 5.2: Delay steps (red) for midrange control codes (blue) using the extracted variable-delay circuit

Table 5.1: Summary of the delay performance obtained by selecting different paths in the NAND circuit and adjusting the load capacitance and the current in the variable-delay buffer for the extracted variable-delay circuit

Min. Delay	Max. Delay	Average Delay Step
142.6 ps	458.9 ps	92 fs

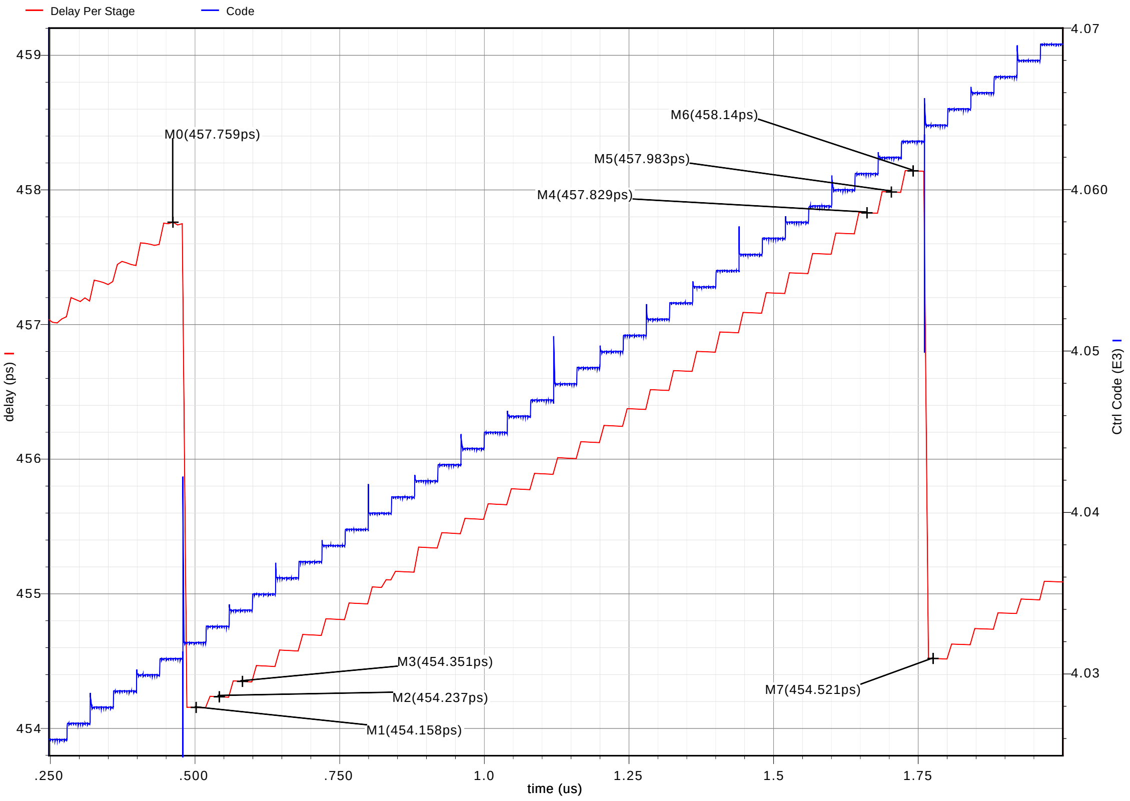


Figure 5.3: Delay steps (red) for control codes (blue) higher in the range using the extracted variable-delay circuit

5.2 Noise and Jitter Performance

Jitter performance is estimated by doing a periodic noise simulation, capturing the output slope and the noise at the rising edge transition. These results are then used in the equation

$$t_j = \frac{v_n}{\frac{dV}{dt}} \tag{5.1}$$

to provide an estimate of the jitter at the output of the device. More details about the noise simulation can be found in Appendix B.

The simulation is performed by activating the worst-case scenario for jitter, which is the one with the longest delay. The largest number of devices are contributing to the noise in this case (the most NAND gates are part of the clock path in this case) and the slope of the output is the lowest (due to the large capacitive load added to the buffer's output. The results of the noise simulation are shown in Figure 5.4, where the differential output voltage and its slope are shown on the left and the output noise spectrum and integrated output noise voltage are shown on the right.

From the results of this simulation, the noise can be calculated using the output slope and the integrated noise voltage in Equation (5.1):

$$t_j = \frac{1.776 \times 10^{-4}}{1.89 \times 10^{9}} = 94.0\,\mathrm{fs} \tag{5.2}$$

which is slightly better than the jitter predicted by the circuit simulations. It is higher than desired for the application, as explained in Chapter 2, but it is still very low compared to the low-noise signal generators used in the lab, which have RMS jitter on the order of picoseconds. Therefore, this jitter is deemed acceptable for this proof-of-concept design. A future iteration of the design can be used to improve the jitter performance.

Noise simulations for the extracted circuit are very time-consuming due to the large number of parasitic circuit components extracted from the layout by the extractor. The above simulation takes more than four days to complete using the computing resources available to the research group.

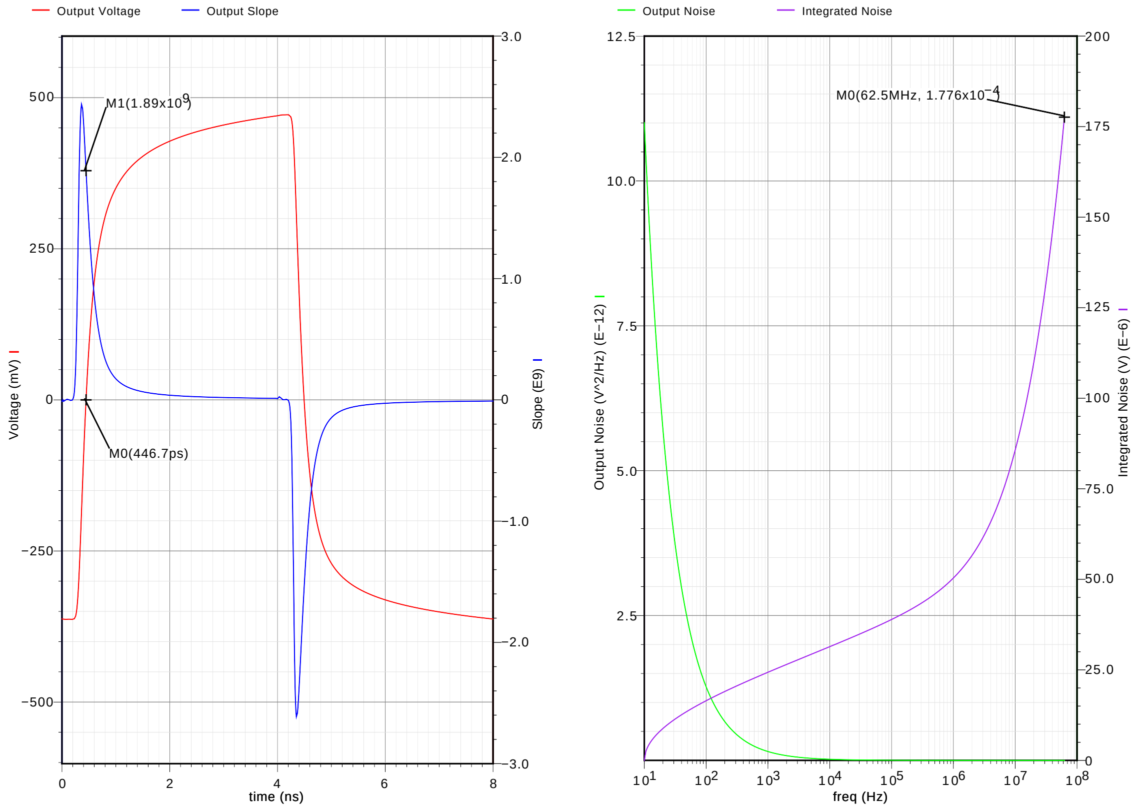


Figure 5.4: The output voltage and slope and the integrated noise on the rising edge of the output for the worst-case control code using the extracted variable-delay circuit

5.3 Conclusion

The extracted simulations verify that the circuit provides well-controlled delay steps with low jitter. The post-layout simulation results show an average delay step of 92 fs over a range of 142.6 ps to 458.9 ps, with an RMS jitter of 94.0 fs. While the precision and the jitter results do not meet the original goals of 50 fs and 25 fs, respectively, due to time constraints on the submission of the chip for fabrication, no further optimization was possible.

Chapter 6

Conclusions

This thesis has examined the effects of timing skew mismatches on interleaved analog-to-digital converters. Using this analysis, a set of specifications for a timing-adjustment circuit was developed. It was shown that the required precision of timing for typical ADCs is below 100 fs, while the jitter requirement is approximately half of the precision requirement.

Using these requirements as a basis, several general structures for adjusting the delay were discussed. These structures were modeled for their delay behavior and their jitter performance. Since the requirements on jitter are very stringent for this circuit, the most practical option with the optimal jitter performance was selected for implementation. The selected circuit uses a coarse delay path selection followed by a fine delay adjustment to span a large delay adjustment range.

The system was designed using a set of NAND gates that allows selection of one of several different delay paths for the coarse adjustment and a variable-delay buffer with variable capacitive load and adjustable tail current for delay tuning. Since the delay increments are far too small to measure directly, an indirect testing methodology was developed. The testing circuit was implemented in the TSMC 0.18 μm CMOS technology and submitted for fabrication in September 2010.

In lieu of measurement results, post-layout simulations were run on the extracted circuits including parasitics to verify that the circuit will provide the delays it was designed for. These simulations show that the circuit provides delay steps of approximately 92 fs over a range of 142.6 ps to 458.9 ps, with an output RMS jitter of 94.0 fs.

6.1 Other Applications

A device with the ability to adjust the timing of an edge to such precision as this one has many applications. The foremost application, that of interleaved analog-to-digital converters, was the focus of this work. Nevertheless, in many high-speed digital applications, clock skews limit performance and processor speed. Hence, the ability to precisely adjust the clock to tune the performance of these systems would be beneficial. The same techniques employed in this work can be applied to these systems as well.

One example of a digital system that requires precise timing is an SRAM. In an SRAM, various timed signals are required to generate the appropriate access sequence for the cells. The timing is important because data can be destroyed by read accesses that take too long, but reads that are too short can fail to provide an accurate sample of the data for the sense amplifier. The ability to adjust the timing provides an increase in yield as well as a performance boost for the fastest parts.

Another application where precise delays are valuable is in the metastability analysis of flip-flops. The ability to analyze flip-flop setup and hold times with high precision can lead to design insights and application-specific improvements. Additionally, well-characterized flip-flop metastability can be used in the design of better random-number generators.

6.2 Future Work

There are many things that could be done to improve the design and further test its ability. Firstly, the design needs to be tested in a practical application in order to verify that it fulfills its purpose. As soon as the fabricated parts are available, these tests can be performed.

Once the design is proved to be viable, additional work must be done to optimize the power consumption of the circuit. This was not done during the design phase due to tight time constraints. While decreasing the power consumption typically leads to higher noise, there are other design variables that can be manipulated to reduce the jitter while minimizing the power consumption, such as the device sizes.

Finally, the circuit should be incorporated into a fully integrated interleaved ADC system to further verify and prove its functionality. This is the last stage of the verification process and the ultimate goal of this research.

APPENDICES

Appendix A

Implementation Details

A.1 Schematics and Transistor Sizes

The circuit was implemented as shown in Figure 4.38, which is reproduced here in Figure A.1 for reference. The layout with block labels is shown in Figure A.2. The details for the sizing of the transistors in each circuit block are found in the following subsections.

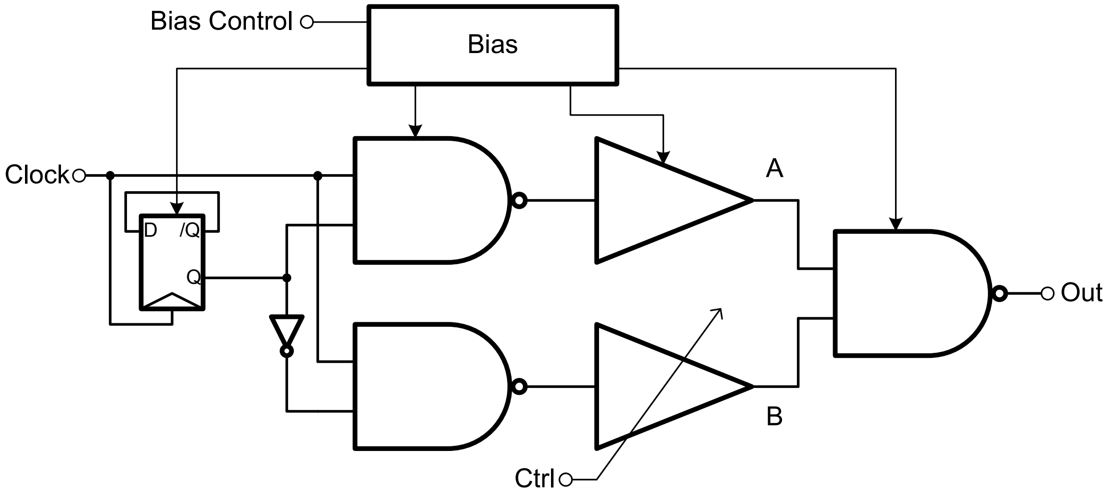


Figure A.1: The complete circuit for implementation

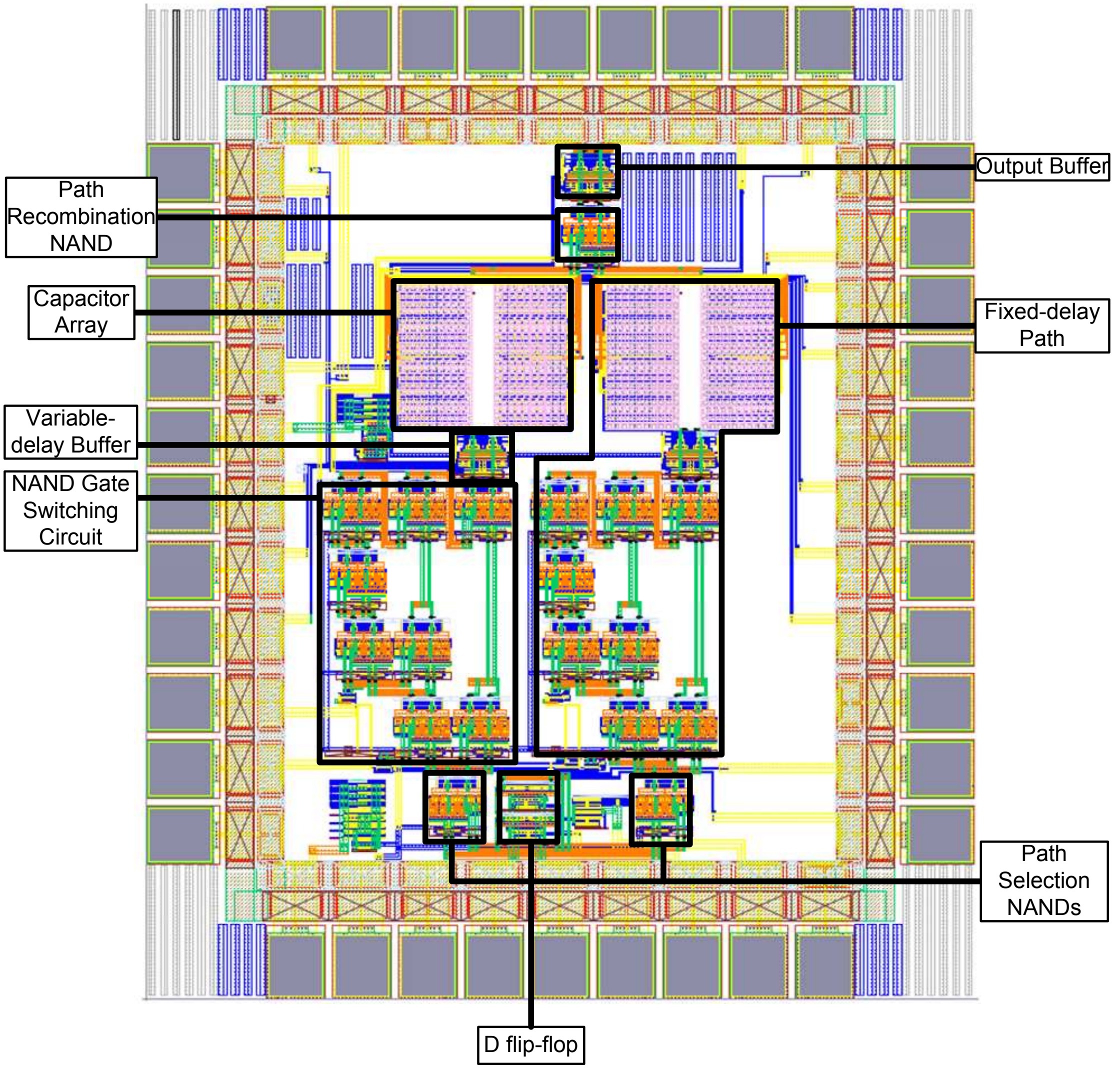


Figure A.2: The layout with blocks labelled

A.1.1 D Flip-Flop

The D flip-flop consists of two MCML D latches in series that latch on opposite phases of the clock. The schematic for the D latch is reproduced in Figure A.3. The device sizes for the D flip-flop are listed in Table A.1.

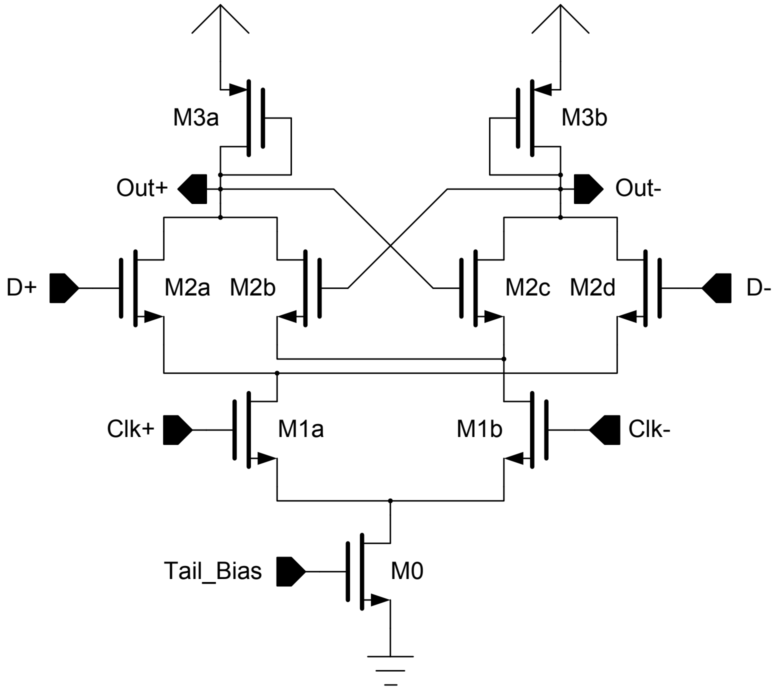


Figure A.3: MCML D latch

Table A.1: MCML D latch transistor sizes

	M0	M1a,b	M2a-d	M3a,b
W	$200\,\mu$m	$60\,\mu$m	$60\,\mu$m	$120\,\mu$m
L	$0.18\,\mu$m	$0.18\,\mu$m	$0.18\,\mu$m	$0.18\,\mu$m

Biasing

The bias circuit for the D latch is shown in Figure A.4. The bias current I_{ref} is supplied externally through a resistor of $260\,\Omega$. The device sizes for the D latch bias circuit are listed in Table A.2.

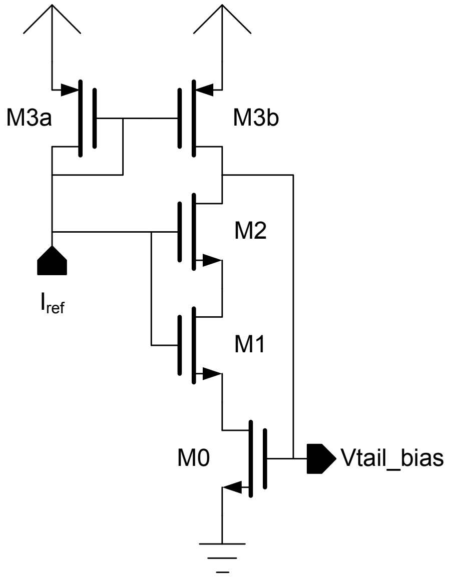


Figure A.4: MCML D latch bias circuit

Table A.2: MCML D latch bias circuit transistor sizes

	M0	M1	M2	M3a,b
W	$200\,\mu$m	$120\,\mu$m	$120\,\mu$m	$240\,\mu$m
L	$0.18\,\mu$m	$0.18\,\mu$m	$0.18\,\mu$m	$0.18\,\mu$m

A.1.2 MCML NAND

The schematic for the MCML NAND gate is shown in Figure A.5. The device sizes for the MCML NAND gate are listed in Table A.3.

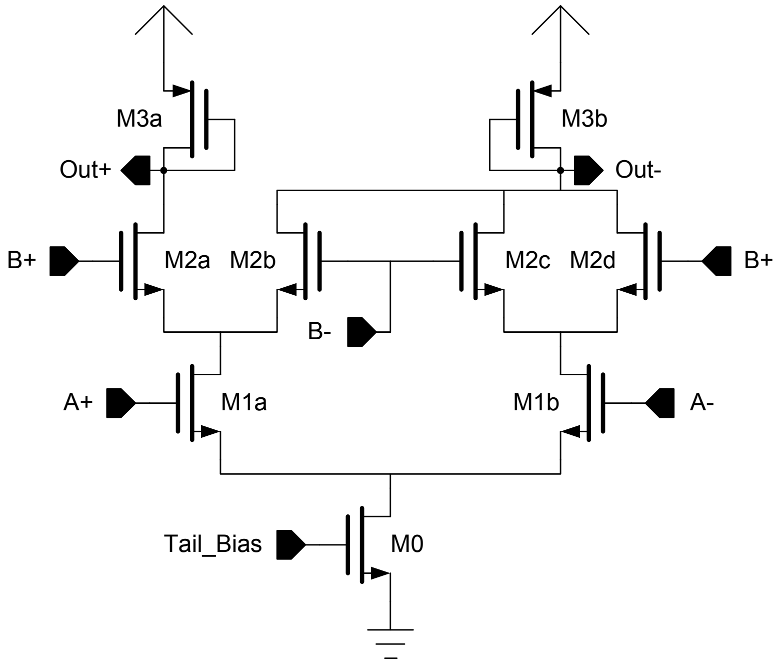


Figure A.5: MCML NAND gate schematic

Table A.3: MCML NAND transistor sizes

	M0	M1a,b	M2a-d	M3a,b
W	200 μm	80 μm	80 μm	210 μm
L	0.18 μm	0.18 μm	0.18 μm	0.18 μm

Biasing

The bias circuit for the MCML NAND gate is shown in Figure A.6. The device sizes for the MCML NAND gate bias circuit are listed in Table A.4.

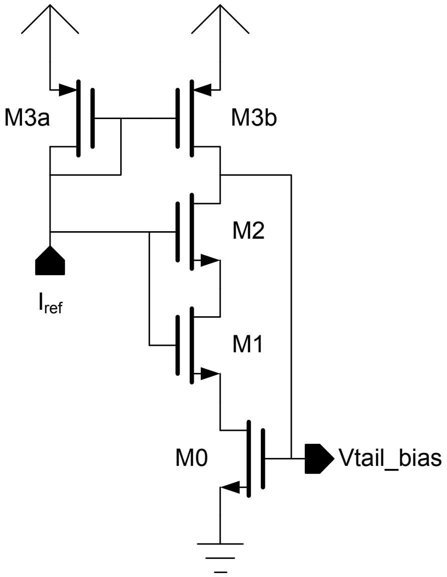


Figure A.6: MCML NAND bias circuit schematic

Table A.4: MCML NAND bias circuit transistor sizes

	M0	M1	M2	M3a,b
W	200 μm	320 μm	320 μm	440 μm
L	0.18 μm	0.18 μm	0.18 μm	0.18 μm

A.1.3 Variable Delay Buffer

The schematic for the variable-delay buffer is reproduced in Figure A.7. The device sizes for the variable-delay buffer are listed in Table A.5.

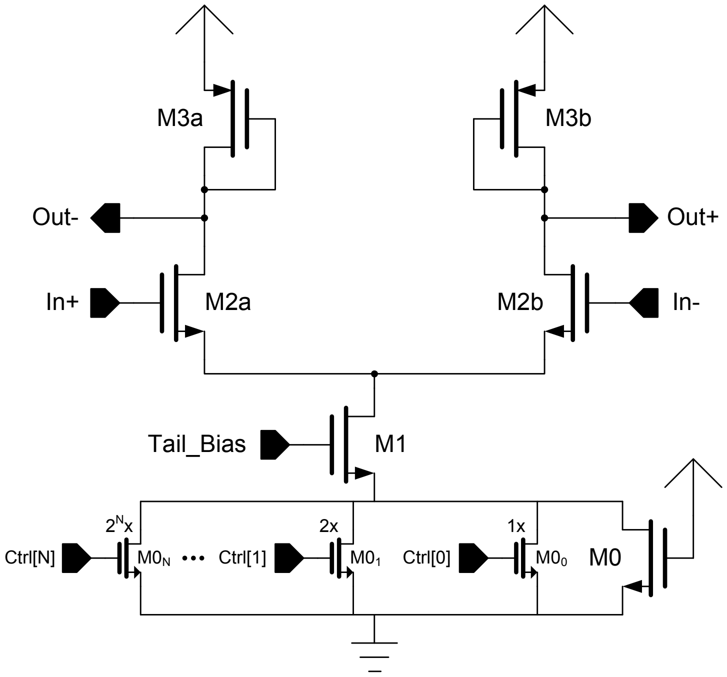


Figure A.7: Variable-delay buffer schematic

Table A.5: Variable delay buffer transistor sizes

	M0	$M0_N$	M1	M2a,b	M3a,b
W	76 μm	420 nm	200 μm	80 μm	210 μm
L	180 nm	180 nm	0.18 μm	0.18 μm	0.18 μm

Biasing

The bias circuit for the variable-delay buffer is shown in Figure A.8. The device sizes for the variable-delay buffer bias circuit are listed in Table A.6.

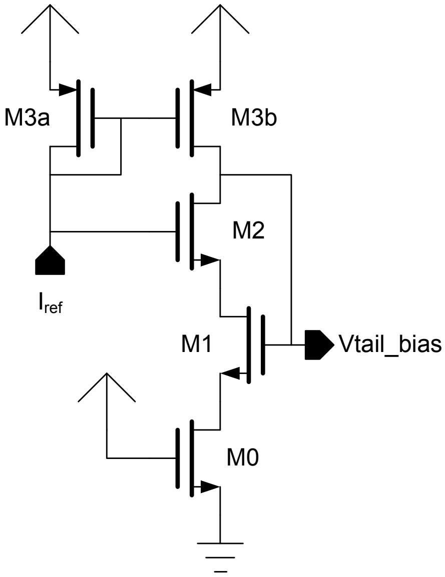


Figure A.8: Variable-delay buffer bias circuit schematic

Table A.6: Variable-delay buffer bias circuit transistor sizes

	M0	M1	M2a,b	M3a,b
W	76 μm	200 μm	160 μm	420 μm
L	180 nm	0.18 μm	0.18 μm	0.18 μm

Capacitor Array

The schematic for one unit of the variable-load capacitors is shown in Figure A.9. The total load circuit contains 256 of these in binary-weighted groups for each of the two outputs of the differential variable-delay buffer. The device sizes for the capacitor array are listed in Table A.7.

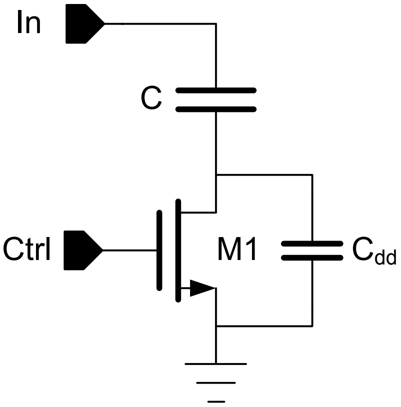


Figure A.9: One unit of the single-ended switchable load capacitor configuration

Table A.7: Capacitor and switch sizes

	Capacitor Top Metal	M1
W	4 μm	420 nm
L	4 μm	180 nm

A.1.4 Output Buffer

The output buffer consists of one copy of the variable-delay buffer with no current starving and no variable-capacitor load. This buffer has been verified for its ability to drive a 50 Ω load.

A.2 Pinout and Packaging

The layout of the chip including the pad names is shown in Figure A.10. The labels are explained in the following paragraphs.

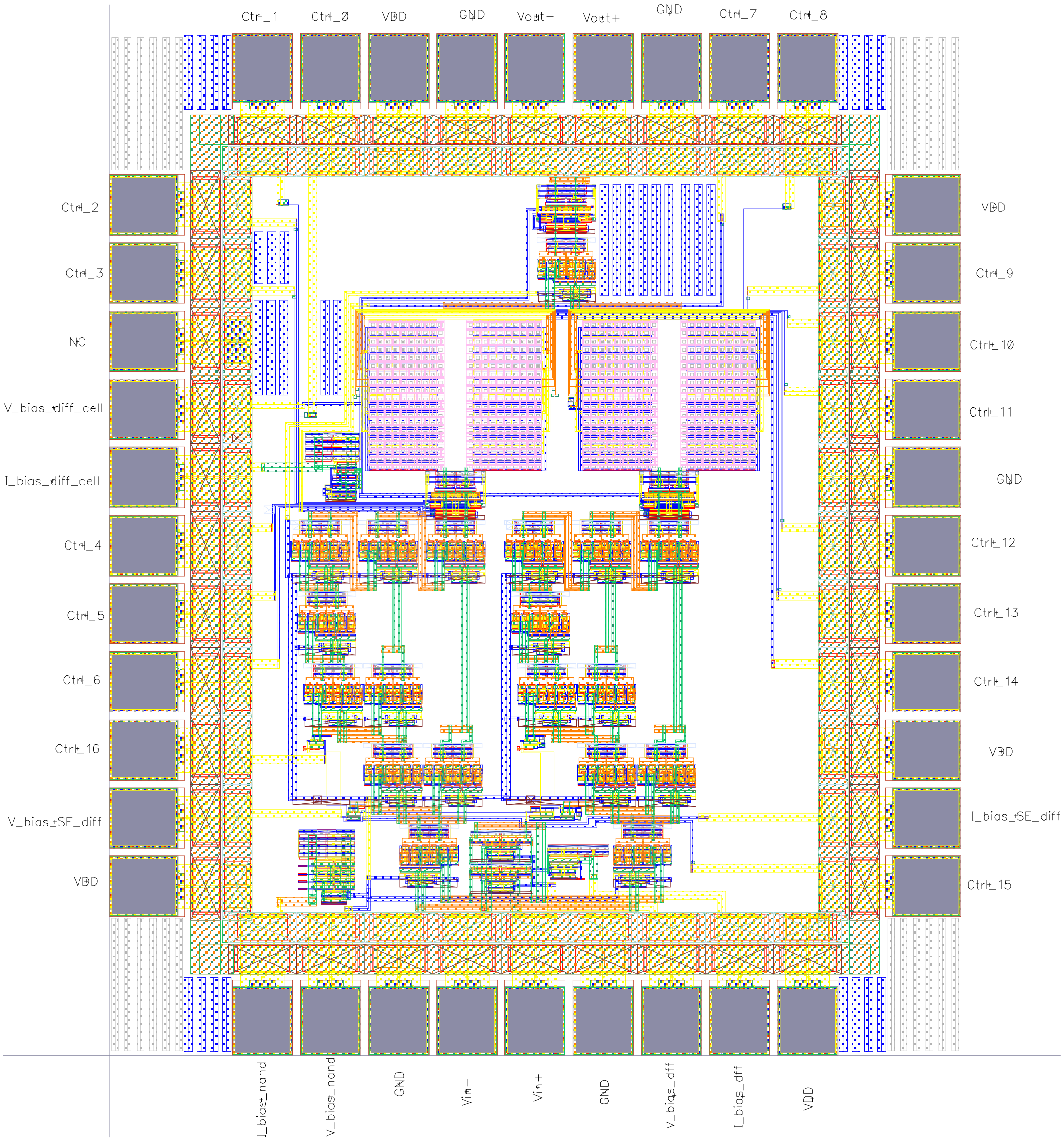


Figure A.10: Layout of the entire chip including pad names

The differential clock input pads (Vin+ and Vin-) are placed beside each other and surrounded on either side by ground pads for good matching and isolation. The differential clock outputs (Vout+ and Vout-) receive similar treatment.

The interface of the circuit includes a digital control vector of 17 bits. For lack of time, it was not possible to implement a digital shift register to hold the control code, so the digital control is connected directly to pads on the periphery of the chip in order to connected to externally-supplied signals. The control pads are labelled Ctrl_*N* where *N* is the bit number, which ranges from 0 to 16. The most-significant bit is Ctrl_16.

The digital control vector includes two bits of thermometer code for controlling the NAND gate coarse delay switching, Ctrl_15 and Ctrl_16. The NAND gate switching requires differential signals for control, so a simple differential pair was designed where one input is connected to a bias voltage (near $\frac{V_{DD}}{2}$) and the other is connected to the single-ended control bit. The differential voltage between the single-ended input and the bias voltage produces a differential swing on the outputs that is sufficient to switch the NAND gate.

Also included in the digital control vector are seven bits of control for the variable capacitor load in the variable delay buffer. Due to space constraints, it was not possible to fit the required eight bits' worth of capacitors in the layout. The capacitive load control uses Ctrl_8 through Ctrl_14 in order of least-significant to most-significant.

The remaining control bits are connected to the current-starving control transistors. These eight bits provide enough current adjustment to span over two steps of the capacitive delay adjustment. These control bits are inverted from the others; increasing the applied digital value turns on more transistors and increases the current in the tail. The current control bits are Ctrl_0 through Ctrl_7.

The chip was designed to be biased with several resistors supplying current mirrors with the appropriate current to bias the various circuit blocks. These current mirrors were connected to pads on the chip for bonding to the package. The current mirrors were used to provide the appropriate bias voltages to the tails of the differential gates. These bias nodes were also connected to the outside world in order to provide a means of verifying the current flowing in the gates and, if necessary, to force the bias nodes to a specific voltage and manually control the current. The bias currents flowing through external resistors are connected to pads labelled I_bias_*X* where *X* is the component being biased. Similarly, the bias voltages are connected to pads labelled V_bias_*X*.

The remaining pads are used for V_{DD} and GND. There are a total of 5 ground pads, including the four used to shield the differential input and output pads, and 6 supply voltage pads.

The circuit was packaged in the CQFP44 package offered by CMC. The bonding diagram is shown in Figure A.11. In the bonding diagram, the upward edge of the chip corresponds to the top edge of the layout diagram in Figure A.10. A pin table including names and descriptions of the pins is shown in Table A.8.

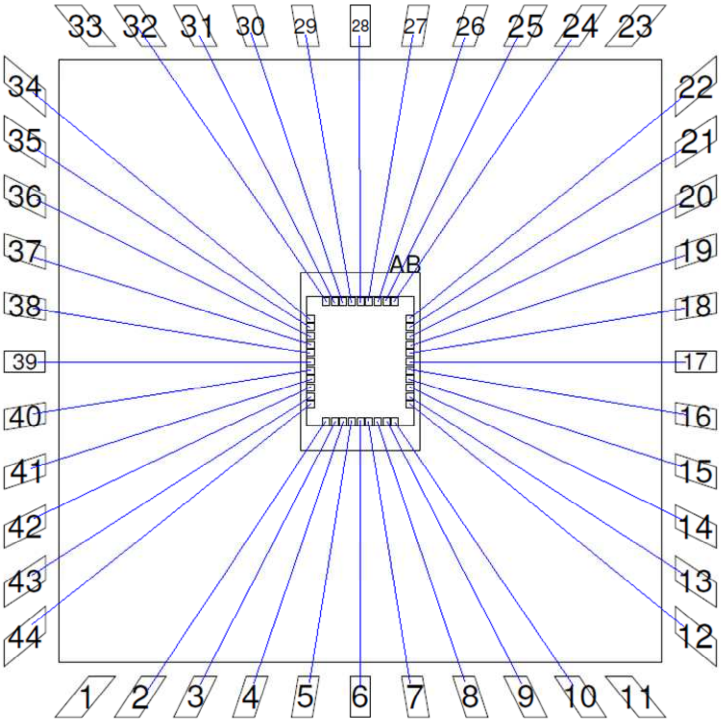


Figure A.11: Bonding diagram

Table A.8: Chip pin names and descriptions

Pin Number	Name	Description
1	NC	Not used
2	I_bias_NAND	NAND bias current
3	V_bias_NAND	NAND bias voltage
4	GND1	Ground
5	Vin+	Positive input clock
6	Vin-	Negative input clock

Continued on Next Page…

Table A.8 – Continued

Pin Number	Name	Description
7	GND2	Ground
8	V_bias_DFF	D Flip-Flop bias voltage
9	I_bias_DFF	D Flip-Flop bias current
10	VDD1	VDD
11	NC	Not used
12	Ctrl_15	Digital control bit 15
13	I_bias_SE_Diff	SE to Diff convertor bias current
14	VDD2	VDD
15	Ctrl_14	Digital control bit 14
16	Ctrl_13	Digital control bit 13
17	Ctrl_12	Digital control bit 12
18	GND3	Ground
19	Ctrl_11	Digital control bit 11
20	Ctrl_10	Digital control bit 10
21	Ctrl_9	Digital control bit 9
22	VDD3	VDD
23	NC	Not used
24	Ctrl_8	Digital control bit 8
25	Ctrl_7	Digital control bit 7
26	GND4	Ground
27	Vout+	Positive output clock
28	Vout-	Negative output clock
29	GND5	Ground
30	VDD4	VDD
31	Ctrl_0	Digital control bit 0
32	Ctrl_1	Digital control bit 1
33	NC	Not used
34	Ctrl_2	Digital control bit 2
35	Ctrl_3	Digital control bit 3
36	VDD5	VDD
37	V_bias_diff_cell	Delay buffer bias voltage
38	I_bias_diff_cell	Delay buffer bias current
39	Ctrl_4	Digital control bit 4

Continued on Next Page...

Table A.8 – Continued

Pin Number	Name	Description
40	Ctrl_5	Digital control bit 5
41	Ctrl_6	Digital control bit 6
42	Ctrl_16	Digital control bit 16
43	V_bias_SE_Diff	SE to Diff convertor bias voltage
44	VDD6	VDD

A.3 Test Circuit Board

A printed circuit board (PCB) was designed to be used to test the chip. The test PCB provides connections for the required power and grounding, as well as the digital control signals. The PCB was implemented as a 2-layer PCB and designed using EAGLE.

The schematic for the board is shown in Figure A.12. The power is provided by an external DC power supply through a banana plug connector, and is decoupled by several capacitors sized at $47\,\mu$F and $22\,\mu$F. Further decoupling capacitors of size $1\,\mu$F are placed near to the chip's supply pins.

Biasing is provided by four potentiometers to provide adjustable resistance to tune the current in the circuit. The bias current nodes are also decoupled to provide steady biasing. The bias voltage nodes are connected to test points to allow measurement of the voltages.

The digital control signals are provided using two banks of switches, shown to the right and left in the schematic diagram. Each of these signals is decoupled on the board to reduce the noise introduced by the control signals.

The input clock signals are connected to the board using two SMA connectors and are terminated with $50\,\Omega$ resistors. Additionally, the input signals have the option to connect a dual-Schottky diode between them in order to clip the sine waves of the signal generator into pseudo-square waves. The majority of testing was done with pseudo-square waves, so if a sinusoidal input does not provide the desired results, this dual-diode can be inserted to more accurately reproduce the test conditions used in simulation.

The output signals leave the board through two SMA connectors. Since the ADC evaluation board that will be used for testing has a single-ended sample clock input, two options are possible. The first is to use both of the differential outputs with a differential-to-single-ended converter. The alternative is to use one of the differential outputs as a single-ended output, and terminate the other output using a dummy 50 Ω load, which is also provided on the board.

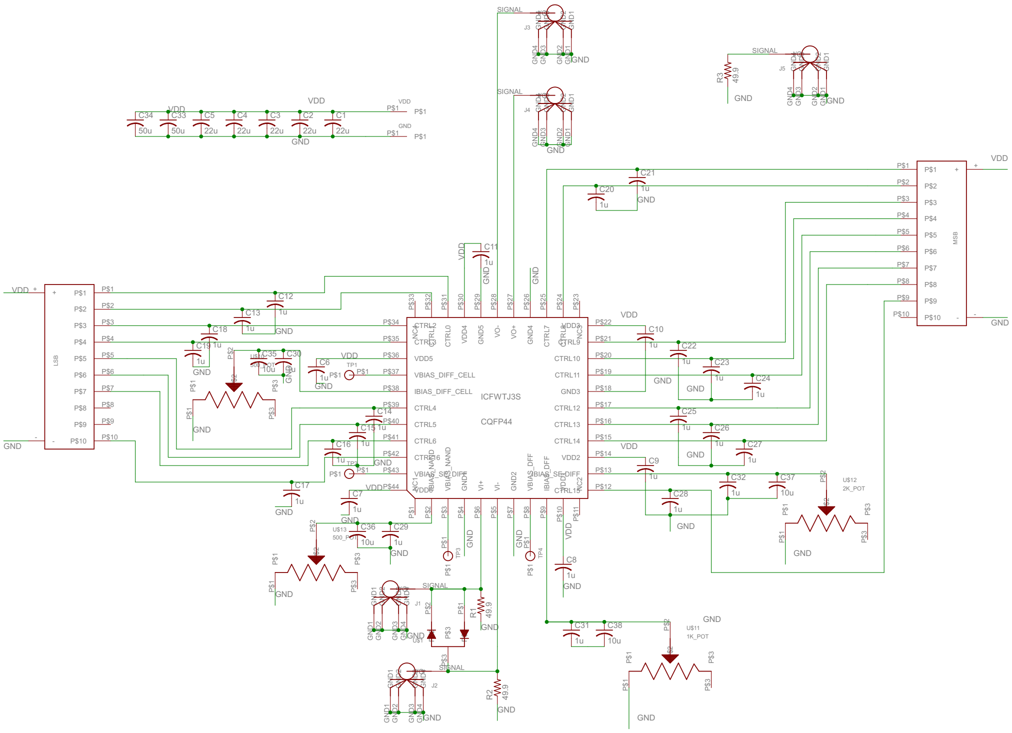
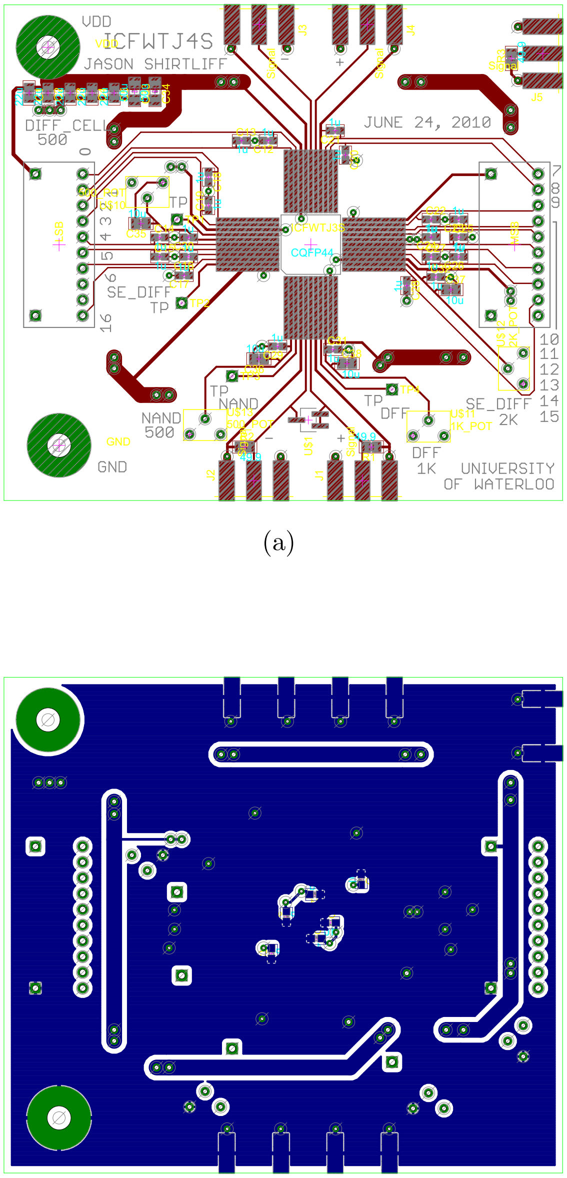


Figure A.12: PCB schematic diagram

The top layer of the board is shown in Figure A.13(a). The bottom layer, used largely as a ground plane, is shown in Figure A.13(b). The board was fabricated by Alberta Printed Circuits.



(a)

(b)

Figure A.13: (a) Top layer of the PCB (b) Bottom layer of the PCB

A.4 Test Procedure

The device will be tested using the AD9265 ADC evaluation board from Analog Devices (ADI). The board contains one AD9265 ADC, which is a 16-bit, 125 MSPS, 1.8 V ADC. The board interfaces with ADI's VisualAnalog software running on a PC via a FIFO device that connects between the ADC's output and the PC's USB port.

The VisualAnalog software allows for post-processing of the sampled data provided by the ADC. It can be configured to display an FFT (Fast Fourier Transform) of the data to view the spectral components of the output. Using this mode, the input tone and all spurious tones can be shown and their magnitudes recorded. Using these magnitudes, it is possible to view the effect of adjusting the timing of the clock edge in one path of the self-interleaved ADC. Since the timing is manually adjusted, the switches can be used to optimize the timing to minimize the magnitude of the spurious tones that result from the timing mismatch between paths.

A.4.1 Connection Details

The AD9265 Evaluation Board offers a single-ended sample clock input. Because the output provided by the clock adjustment circuit is differential, two options are available to connect the device under test to the ADC evaluation board. The first is to take one of the two differential outputs and treat it as single-ended, while terminating the other output with a dummy 50 Ω load. The alternative is to use a 2-way 180° power combiner to combine the differential outputs into a single-ended signal. The Mini-Circuits ZFSCJ-2-1 is a power combiner that will work in this application at the frequencies desired for this circuit.

Since the output of the signal generator is single-ended, conversion in the reverse direction is necessary at the input of the clock-adjustment circuit. The same part from Mini-Circuits works as a divider as well and is sufficient for both conversions. One additional consideration is the common mode bias at the input, since the circuit is not internally biased at the input. For this, it is necessary to use a Mini-Circuits ZFBT-6G bias tee that combines a high-frequency signal with a DC bias. The optimal DC bias point at the input is 0.9 V in simulation.

The required connections from the signal generator to the ADC evaluation board are shown are shown in Figure A.14. Not shown is the FIFO buffer used for data capture and the USB connection to the PC.

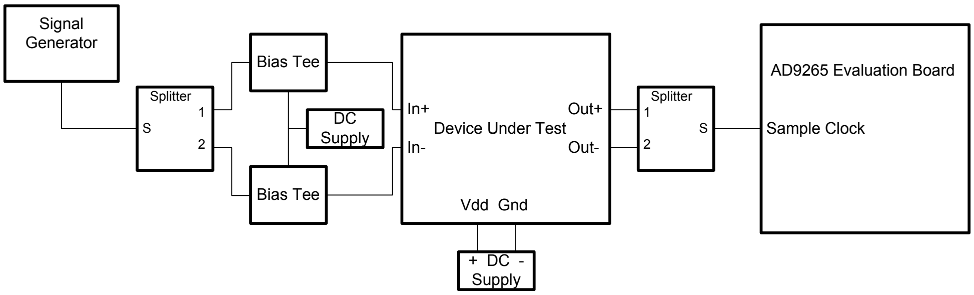


Figure A.14: Test bench connections

A.4.2 Input Voltages and Stimuli

The chip is designed to work with a supply voltage of 1.8 V. The input swing must be at least 600 mV, differentially, if the signal is a square wave, or 800 mV if the signal is a sine wave.

The fixed delay path is designed to have a delay that is equivalent to the following scenario: the middle path of the NAND switching circuit is activated, the capacitive load is at its minimum, and the current starving is at its minimum. This corresponds to an input vector of (from bit 16 to bit 0) 01000000011111111_b. The delays may not match exactly at this value, since the path recombination NAND gate has different delays from each input to the output. Incrementing and decrementing the control code should allow for adjustment in order to find the exact delay match, as described above.

The bias potentiometers should be set to their designed resistor values for testing. The nominal values are listed in Table A.9.

Table A.9: Bias resistor nominal values

Resistor	Value
I_bias_diff_cell	131 Ω
I_bias_NAND	131 Ω
I_bias_se_diff	950 Ω
I_bias_DFF	260 Ω

Appendix B

Performing Jitter Analysis Using Cadence

B.1 Definition of Jitter

Jitter is defined as the variation in the time at which a signal crosses a threshold. Jitter arises from noise in the circuit under consideration, and depends on the slope of the signal as it crosses the threshold.

Graphically, jitter can be depicted as the range over which an edge crosses a threshold, as shown in Figure B.1 and Figure B.2. As can be seen in Figure B.1, an edge with a small slope has a wide region where the noise may cause the signal to cross the threshold. In Figure B.2, an edge with a higher slope has a much narrower window where the signal crosses due to noise, and so its jitter is much less.

Mathematically, the jitter can be expressed as the magnitude of the noise divided by the slope of the signal with respect to time.

$$t_j = \frac{v_n}{dV/dt} \tag{B.1}$$

Therefore, to analyze the jitter in a circuit, it is necessary to determine the total noise in the circuit and the slope of the selected signal at the threshold.

B.2 Jitter Analysis in Cadence

Cadence provides a means to analyze the jitter in a circuit using the SpectreRF simulator [16]. To perform this analysis, a periodic input must be applied and a

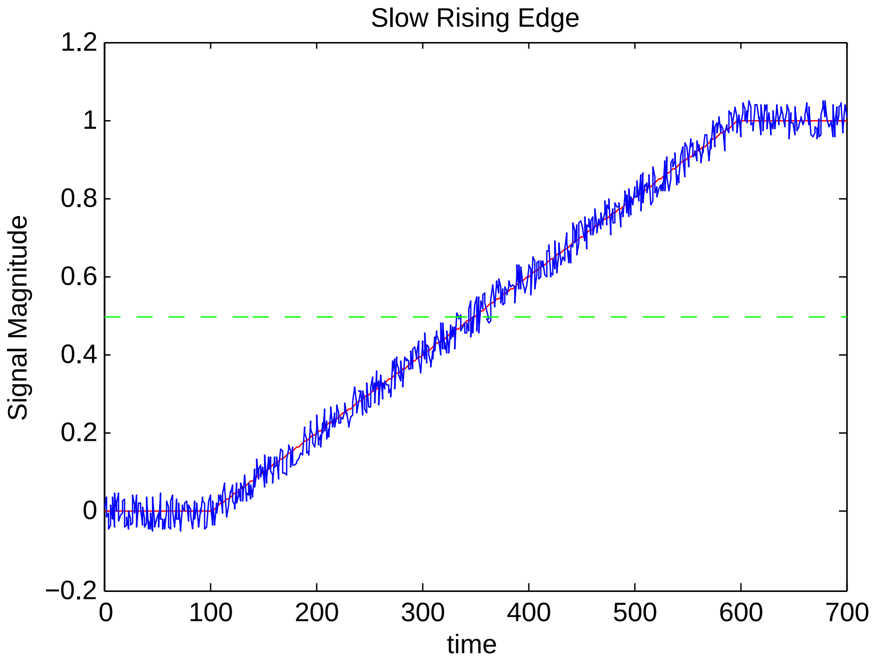


Figure B.1: A slow rising edge with noise crossing a threshold

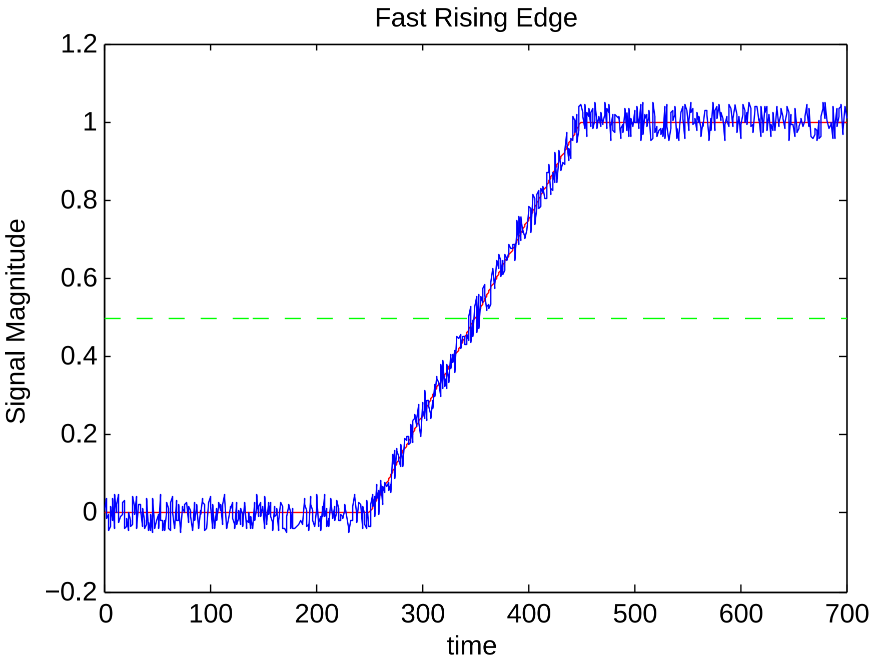


Figure B.2: A fast rising edge with noise crossing a threshold

periodic steady state (PSS) analysis done. Then, using the results from the PSS analysis, a periodic noise (PNoise) analysis can be performed, after which the jitter in the circuit can be calculated. The configuration options for the PSS and PNoise analyses are presented below.

The simulator cannot produce reliable results when one single gate such as an inverter is simulated for jitter. The integrated noise values are not consistent when the input source frequency changes. Logically, the root mean square (RMS) jitter for an edge should not change depending on how frequently the edges occur because the edge events should be independent of each other. However, it appears that there is some correlation between edges in the simulator when a single gate is simulated, because the total noise appears to be a function of the input frequency. Hence, to use this method to measure the jitter in a simple circuit, it must be possible to create a repeating structure using the circuit. The jitter for a single element can be mathematically determined using the jitter for the entire structure.

B.2.1 Periodic Steady State Analysis Configuration

To perform jitter analysis on a chain of four inverters, it is necessary to apply a square wave input to satisfy the periodic requirement. The frequency of the input signal should be within the bandwidth of the circuit under test, and an appropriate value for the rise time should be selected. Then, in the Virtuoso Analog Design Environment, the simulator must be configured with the correct libraries for Spectre and the appropriate design variables for the schematic.

To enable the PSS analysis, from the *Choosing Analyses* dialog, the radio button for *pss* analysis was selected.

At the top, the frequency of the input source appeared in the list of *Fundamental Tones*. With the *Beat Frequency* radio button selected, the same frequency was entered in the *Beat Frequency* field.

In the *Output Harmonics* drop-down menu, *Number of Harmonics* was selected, and a field appeared next to it where the number of harmonics the simulator should consider must be entered. It was been determined that the number of harmonics necessary to perform jitter analysis on a simple inverter circuit is 32. Fewer harmonics can be computed, but a threshold exists below which the PNoise analysis does not have enough data for the outer sidebands to provide meaningful results (as discussed below).

Under *Accuracy Defaults*, conservative was selected to obtain the most accurate (and slowest) simulation. The circuit was given extra time to arrive at a steady state; this was specified in the *Additional Time for Stabilization* field. It was decided to allow at least one period of the input source for stabilization, as the circuit may have had transients that must converge prior to the steady state analysis so that the results are reliable.

Finally, the initial transient results were saved, and the *oscillator* and *sweep* check boxes were deselected.

Final PSS Configuration

The simulator settings used are shown in Figure B.3.

B.2.2 Periodic Noise Analysis Configuration

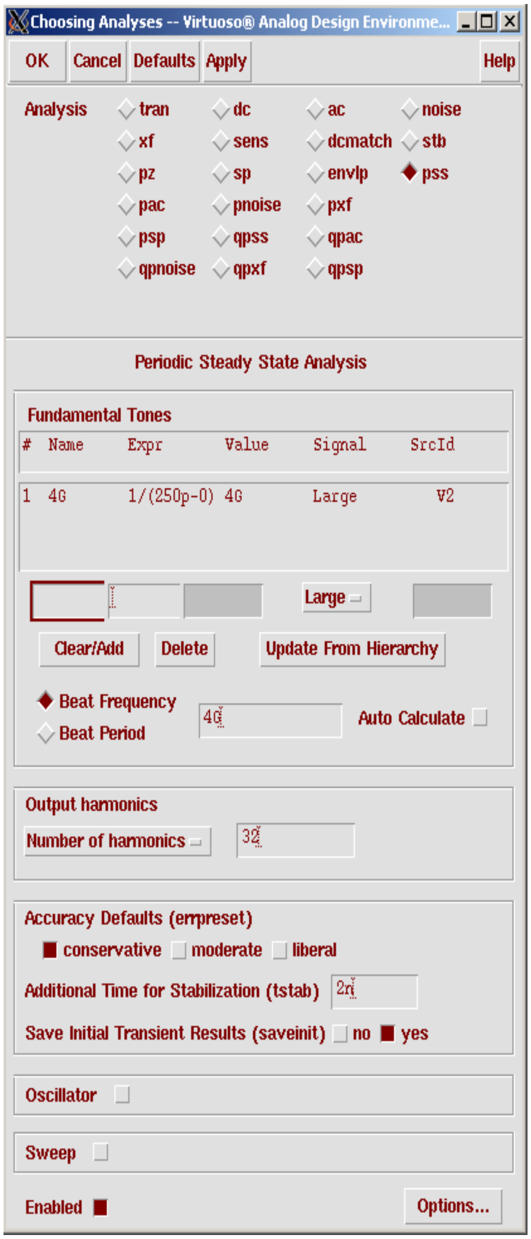
To enable the PNoise simulation, the radio button for *pnoise* analysis was selected.

For the sweep type, absolute was selected and the frequency range that matched the Nyquist band for the previously selected input frequency was input into the *Start/Stop* fields. The *Sweep Type* was selected to be logarithmic with 40 points per decade.

For the purpose of this simulation, the *Output* was selected to be a voltage whose positive node was the loaded output node of the inverter and whose negative node was the ground net. The *Input Source* was also a voltage, and the input square wave source was selected as the source.

The *Reference Sideband* selected was 0 because the output frequency was expected to be the same as the input frequency. This setting allows for the simulation of mixers and other frequency converters, and by setting it to zero, the simulator assumes that the output frequency for a given input frequency will be the same.

Jitter was selected in the *Noise Type* drop-down. The *Threshold Value* was set to the switching threshold of the inverter, since the noise at that point is what will be transferred into the next stage. Finally, only rising edges were considered so *rise* was selected in the *Crossing Direction* menu.



Analysis ○ tran ○ dc ○ ac ○ noise
○ xf ○ sens ○ dcmatch ○ stb
○ pz ○ sp ○ envlp ● pss
○ pac ○ pnoise ○ pxf
○ psp ○ qpss ○ qpac
○ qpnoise ○ qpxf ○ qpsp

Periodic Steady State Analysis

Fundamental Tones

#	Name	Expr	Value	Signal	SrcId
1	4G	1/(250p-0)	4G	Large	V2

Large

● Beat Frequency 4G
○ Beat Period
Auto Calculate □

Output harmonics

Number of harmonics 32

Accuracy Defaults (errpreset)

■ conservative □ moderate □ liberal

Additional Time for Stabilization (tstab) 2n

Save Initial Transient Results (saveinit) □ no ■ yes

Oscillator □

Sweep □

Enabled ■

Figure B.3: Settings used to run a periodic steady state analysis on a chain of four inverters with an input source operating at 4 GHz

Number of Sidebands

The number of sidebands required to obtain reliable results was found to vary with the frequency of the input signal. It was determined that increasing the sidebands parameter caused the total noise in the circuit to asymptotically approach a point of saturation, above which the number of sidebands had very little effect on the noise. A simple relationship between the AC bandwidth of one stage of the circuit and the number of sidebands required to capture enough of the noise information was determined.

Sidebands are found in the output at harmonic multiples of the input frequency. In noise analysis, they appear in the response of circuits at high frequencies. The noise at these frequencies is calculated and summed to approximate the total noise response of the circuit. When integrating to the Nyquist frequency for a given input frequency, many of the upper sidebands have a noise response which is minimal in the frequency band under consideration. The upper sidebands whose noise response is approximately zero within the Nyquist band do not need to be considered, and so the number of sidebands is selected to include the majority of the noice-contributing sidebands so an accurate evaluation of the noise in the circuit can be performed.

The number of multiples found to be required was approximately a factor of four larger than the expected number of sidebands that would fit within the bandwidth of a single stage of the circuit. For a simple inverter whose bandwidth is 30 GHz, and an input frequency of 1 GHz, exactly 30 sidebands fit within the bandwidth of the circuit. Therefore, following the aforementioned factor of four rule, it was expected that approximately 120 sidebands would be needed to characterize the noise in the circuit. Simulation results show that the noise calculated in the circuit level off almost completely when the sideband parameter is increased beyond 128 sidebands.

A summary of the number of sidebands required at various input frequencies for a basic inverter circuit with a bandwidth of approximately 30 GHz is found in Table B.1. These values were found to hold regardless of the number of inverters placed in a chain for simulation.

As discussed above, the number of harmonics calculated in the PSS analysis affects the validity of the PNoise simulation results. SpectreRF gives a warning when the number of sidebands selected in the PNoise simulation is so large that there is not enough harmonic data to give meaningful results. It was found that at an input frequency of 250 MHz, using 20 harmonics and 128 sidebands in the same

Table B.1: Number of sidebands required for reliable PNoise analysis for various input frequencies for a chain of inverters (with an individual bandwidth of 30 GHz)

Input Frequency	Sidebands Required
250 MHz	> 256
500 MHz	> 128
1 GHz	128
2 GHz	64
4 GHz	32

30 GHz inverter resulted in this warning. Increasing the number of harmonics to 32 allowed a simulation with 128 sidebands to complete successfully, but the warning occurred again with 256 sidebands.

Final PNoise Configuration

The simulator settings used for the PNoise simulation are shown in Figure B.4. With these settings applied, the simulation was run.

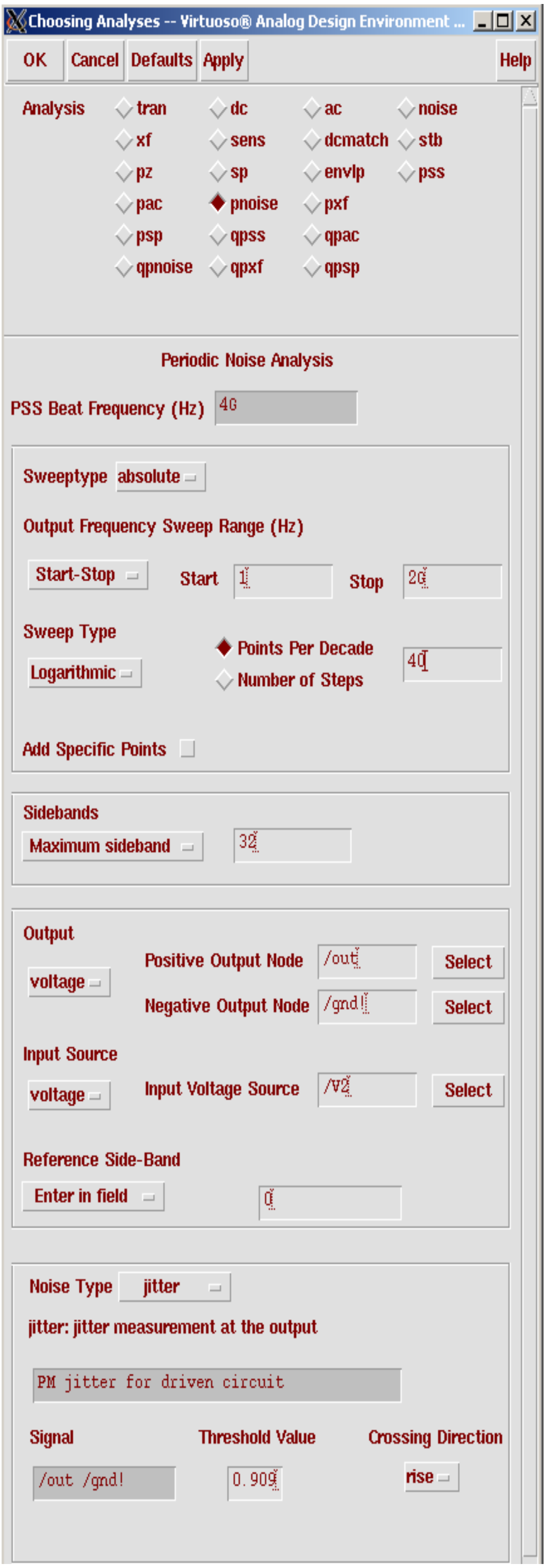
B.2.3 Plotting Results

To plot the results of the simulation, the *Direct Plot* form in the Virtuoso Analog Design Environment was used.

First, the *tstab* radio button was selected and the output node transient was plotted by selecting the node on the schematic. Then, using the calculator, the slope of the transient was plotted using the derivative function.

Finally, with the *tdnoise* radio button selected on the *Direct Plot* form, the total integrated noise was plotted. The *Function* button was switched to *Integ Output Noise*, the modifier was set to *Magnitude* and the start and stop frequency fields were entered as the Nyquist range as used in the PNoise settings.

Markers were placed at the time of the crossing event in the transient and slope plots to determine the value of the slope at that point, which is needed to calculate the jitter. These were placed on the plot at the jitter event time as shown on the total noise plot, after the stabilization time (as entered in the PSS settings). An example plot using the settings above is shown in Figure B.5.


Choosing Analyses -- Virtuoso® Analog Design Environment ...
OK
Cancel
Defaults
Apply
Help

Analysis ◇ tran ◇ dc ◇ ac ◇ noise
◇ xf ◇ sens ◇ dcmatch ◇ stb
◇ pz ◇ sp ◇ envlp ◇ pss
◇ pac ◆ pnoise ◇ pxf
◇ psp ◇ qpss ◇ qpac
◇ qpnoise ◇ qpxf ◇ qpsp

Periodic Noise Analysis

PSS Beat Frequency (Hz) 4G

Sweeptype absolute

Output Frequency Sweep Range (Hz)

Start-Stop Start 1 Stop 20G

Sweep Type

Logarithmic ◆ Points Per Decade 40
◇ Number of Steps

Add Specific Points □

Sidebands

Maximum sideband 32

Output

voltage Positive Output Node /out Select
Negative Output Node /gnd! Select

Input Source

voltage Input Voltage Source /V2 Select

Reference Side-Band

Enter in field 0

Noise Type jitter

jitter: jitter measurement at the output

PM jitter for driven circuit

Signal	Threshold Value	Crossing Direction
/out /gnd!	0.909	rise

Figure B.4: Settings used to run a periodic noise analysis on a chain of four inverters with an input source operating at 4 GHz

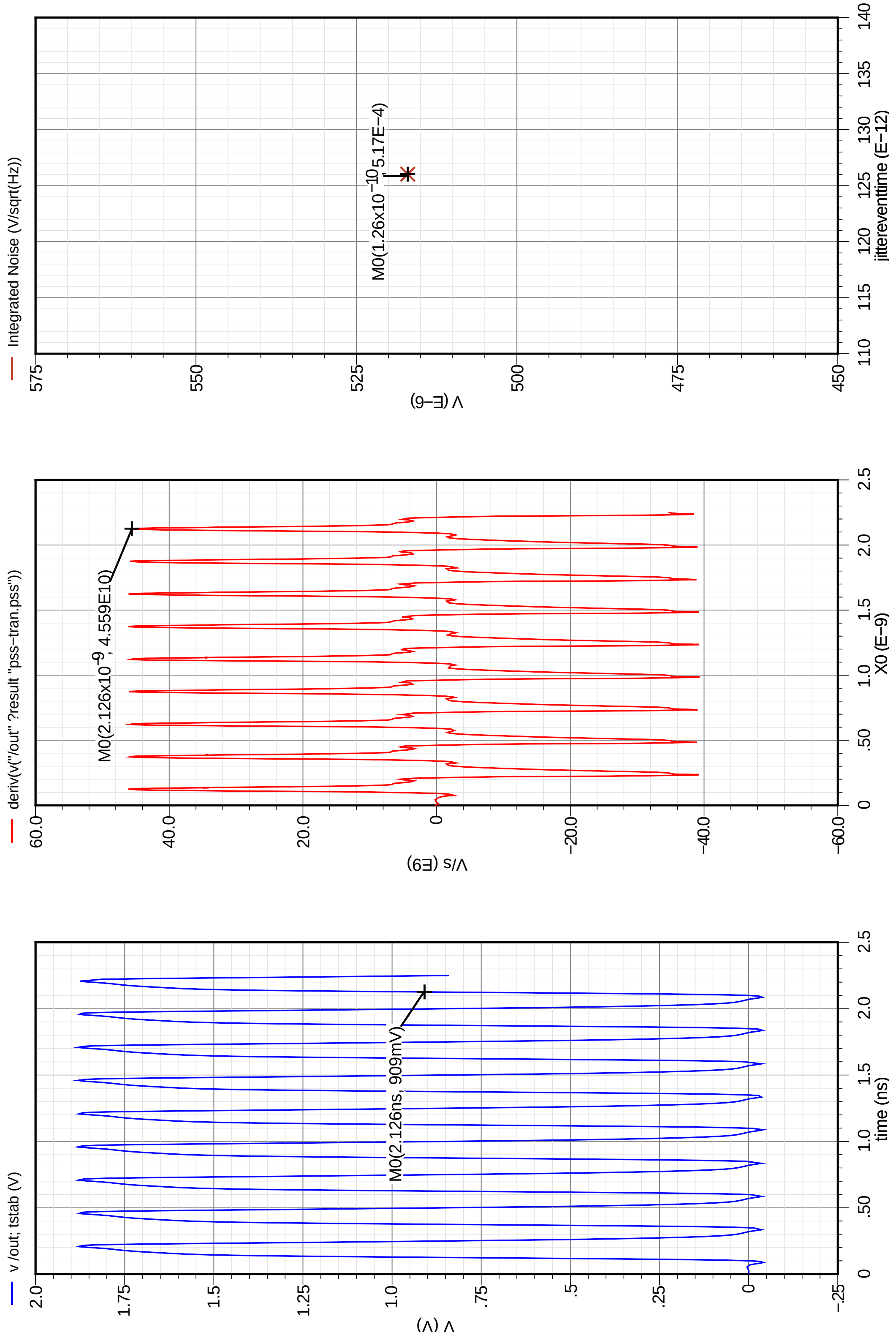


Figure B.5: Plotted results of a periodic noise analysis on a chain of four inverters with an input source operating at 4 GHz

B.2.4 Jitter Calculation

The plot in Figure B.5 gives us the total noise in the circuit and the slope of the edge at the output of the last inverter in the chain of four. With these values, the jitter in the whole circuit can be calculated, and then using the total jitter, the individual jitter for one inverter can be calculated. The circuit used for these results was a simple inverter with device sizes as shown in Table B.2.

Table B.2: Inverter transistor sizes for sample simulations

	NMOS	PMOS
W	$40\,\mu$m	$144\,\mu$m
L	$0.18\,\mu$m	$0.18\,\mu$m

Using the values above, the total jitter in the circuit is calculated as $t_j = \frac{v_n}{dV/dt} = \frac{5.17\times10^{-4}}{4.559\times10^{10}} = 11.3\,\text{fs}$. Because the noise in each inverter is uncorrelated with the rest of the inverters, the noise adds in the form $t_{j_{tot}} = \sqrt{t_{j_1}^2 + t_{j_2}^2 + \ldots}$. Hence, if the circuit structure is repeating, the noise in each stage is the same and $t_{j_{tot}} = \sqrt{N} t_{j_i}$, where N is the number of stages. This means that for a single inverter in the above example, the jitter value is $t_{j_i} = \frac{t_{j_{tot}}}{\sqrt{4}} = \frac{11.3\,\text{fs}}{2} = 5.65\,\text{fs}$.

This value for the jitter in the inverter has been verified by running simulations on chains of inverters 4, 8, and 16 units long at input frequencies of 1 GHz, 2 GHz, and 4 GHz. The results are presented in Table B.3. The values for 4 GHz begin to deviate as the number of circuit stages increases. This can be attributed to the fact that the circuit with more stages is approaching its bandwidth and is no longer fully swinging from rail to rail, resulting in a distorted waveform and altered slope.

Table B.3: Jitter in inverter chains of varying lengths at several input frequencies

	Inverter Chain Length					
	4		8		16	
Input Frequency	Total	Per Stage	Total	Per Stage	Total	Per Stage
1 GHz	11.2 fs	5.60 fs	16.0 fs	5.66 fs	22.9 fs	5.72 fs
2 GHz	11.2 fs	5.60 fs	16.1 fs	5.69 fs	23.0 fs	5.75 fs
4 GHz	11.3 fs	5.65 fs	16.7 fs	5.90 fs	24.9 fs	6.22 fs

References

[1] D. Nairn, “Time-interleaved analog-to-digital converters,” in *Proceedings of the IEEE 2008 Custom Integrated Circuits Conference*, 2008, pp. 289–296. 1, 76

[2] Y.-C. Jenq, “Digital spectra of nonuniform sampled signals: Fundamentals and high-speed waveform digitizers,” *IEEE Transactions on Instrumentation and Measurement*, vol. 37, pp. 245–251, Jun. 1988. 5

[3] K. Poulton, J. Corcoran, and T. Hornak, “A 1-GHz 6-bit ADC system,” *IEEE Journal of Solid-State Circuits*, vol. SC-22, pp. 962–970, Dec. 1987. 15

[4] C. Conroy, D. Cline, and P. Gray, “An 8-b 85-MS/s parallel pileline A/D converter in 1-μm CMOS,” *IEEE Journal of Solid-State Circuits*, vol. 28, pp. 447–454, Apr. 1993. 15

[5] J. Yuan and C. Svensson, “A 10-bit 5-MS/s successive approximation ADC cell used in a 70-MS/s ADC array in 1.2-μm CMOS,” *IEEE Journal of Solid-State Circuits*, vol. 29, pp. 866–872, Aug. 1994. 15

[6] D. Johns and K. Martin, *Analog Integrated Circuit Design*. New York: John Wiley & Sons, Inc., 1997, ch. 4. 19

[7] J. Rabaey, A. Chandrakasan, and B. Nikolić, *Digital Integrated Circuits: A Design Perspective*, 2nd ed., ser. Electronics and VLSI Series. Upper Saddle River, NJ, USA: Prentice Hall Press, 2003, ch. 5.4. 20

[8] D. Nairn, “Converter systems having reduced-jitter, selectively-skewed interleaved clocks,” U.S. Patent 7 429 944, 2008. 37

[9] M. Mizuno, M. Yamashina, K. Furuta, H. Igura, H. Abiko, K. Okabe, A. Ono, and H. Yamada, “A GHz MOS adaptive pipeline technique using MOS current-mode logic,” *IEEE Journal of Solid-State Circuits*, vol. 31, pp. 784–791, Jun. 1996. 41

[10] J. Musicer and J. Rabaey, "MOS current mode logic for low power, low noise CORDIC computation in mixed-signal environments," in *ISLPED 2000: Proceedings of the 2000 International Symposium on Low Power Electronics and Design*. New York, NY, USA: ACM, 2000, pp. 102–107. 41

[11] J. Maneatis and M. Horowitz, "Precise delay generation using coupled oscillators," *IEEE Journal of Solid-State Circuits*, vol. 28, pp. 1273–1282, Dec. 1993. 41

[12] T. Weigandt, "Low-phase-noise, low-timing-jitter design techniques for delay cell based VCOs and frequency synthesizers," Ph.D. dissertation, University of California at Berkeley, 1991. 42, 46, 50, 51

[13] A. Sedra and K. Smith, *Microelectronic Circuits*, 5th ed. New York, NY: Oxford University Press, 2004, ch. 7.1. 43, 50

[14] J. Maneatis, "Low-jitter process-independent DLL and PLL based on self-biased techniques," *IEEE Journal of Solid-State Circuits*, vol. 31, pp. 1723–1732, Nov. 1996. 47, 48

[15] *TSMC 0.18UM Mixed Signal/RF 1P6M Salicide 1.8V/3.3V Design Rule*, Taiwan Semiconductor Manufacturing Company, Ltd., 2006. 62

[16] *Virtuoso SpectreRF Simulation Option User Guide*, Cadence Design Systems, Inc., 2005. 111