Jitter Reduction on High-Speed Clock Signals

Tina Harriet Smilkstein

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2007-96
http://www.eecs.berkeley.edu/Pubs/TechRpts/2007/EECS-2007-96.html

August 6, 2007

Jitter Reduction on High-Speed Clock Signals

by

Tina Harriet Smilkstein

B.A. (Nanzan University, Nagoya, Japan) 1989
M.S. (University of California at Berkeley) 2003

A dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Engineering - Electrical Engineering and Computer Science

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor Robert W. Brodersen, Chair
Professor Jan M. Rabaey
Professor Paul K. Wright

Fall 2007

The dissertation of Tina Harriet Smilkstein is approved.

Chair Date

Date

Date

University of California, Berkeley

Fall 2007

Jitter Reduction on High-Speed Clock Signals

by

Tina Harriet Smilkstein

Abstract

Jitter Reduction on High-Speed Clock Signals

by

Tina Harriet Smilkstein

Doctor of Philosophy in Engineering - Electrical Engineering and Computer Science

University of California, Berkeley

Professor Robert W. Brodersen, Chair

As clocking speeds increase, it becomes more and more important to be able to generate "clean", low-jitter clock signals. Traditionally, PLLs have been one of the most commonly used signal cleaning methods, but as higher frequencies are being used, the limits imposed by both the design complexity and performance of PLLs is being felt.

This work shows that a purely feedforward jitter removal circuit is possible for frequencies in the $800MHz$ to $5GHz$ range. The design is relatively simple and modular, which allows a designer to customize to the type of system where the circuit will be used. MOSFET devices were used and no special processing is required. In some cases it is recommended that certain analog blocks are placed near each other to minimize process variation effects but, other than that, no special layout considerations are required or recommended. The transistors used in the switching $NMOS$ in the integrator have a channel length of $110nm$. All other devices in the design are sized above $130nm$. No effort was made to minimize the number of transistors used. The final number of transistors used was 403 from the input single-to-differential converter through the output level detector. Exactly 300 of these were used in the input monostable block.

Simulations were done using ST Microelectronic $90nm$ technology. Simulations looked at the performance of the circuit in the presence of supply noise, GND noise, intrinsic noise,

and input jitter. All simulations were done at typical, fast, and slow corners. Attenuation of jitter was tested with input jitter from $0pspk-pk$ to the limit the system could process. For a system running at $1GHz$ and driven with a pulse of width $650ps$, the maximum jitter that can be processed is slightly less than $350pspk-pk$. The results varied from $-13.81dB$ (a 4.9x reduction in rms jitter) for $200mV$ of random noise on the supply, intrinsic noise, and worst case evaluations for 10% process variation, to $-14.68dB$ (a 5.4x reduction in rms jitter) for no supply or GND noise. Using an ideal source instead of the pre-processing input monostable block used in these results, gives a maximum jitter reduction of $-35.5dB$. The loss in performance can be attributed to the large amount of circuitry in the input block. These results show that this purely feedforward system is, in fact, effective in reducing jitter.

In implementing this system, a number of new blocks were developed including a differential Schmitt trigger, feedforward correction block to align signals, high-speed pulse-mode flip-flop, and a monostable that can produce a duty cycle close to 100%. The feedforward biasing circuitry was also unique as was the effort to create a completely feedforward design.

Professor Robert W. Brodersen
Dissertation Committee Chair

To Peeps. And Huey. And my dad.

Contents

List of Figures

List of Tables

Acknowledgements

aaaaah... I'm to the point where I'm writing acknowledgements for my dissertation. Where to begin? I can say with certainty that the person that had the most to do with me somehow getting through the actual writing of this document is Professor Bob Brodersen. Though I sometimes didn't get the lead he showed me in the beginning, I managed to catch on along the way and now can see how intellegent, thoughtful, insightful, and, yes, useful his direction was. Many things happened along the way but he stuck with me.

If I'm listing people that were great support above and beyond their job titles, I must thank Ruth Gjerde, and Tom Boot. How many times can one person answer the same question about procedure and still not get angry? Well, ask those two and use me as an example. Brenda Vanoni was also in the same group as Tom and Ruth - always there to answer questions, fill requests, or just supply some good gossip. She was instrumental in making me feel like I could make it through this experience. She was a fighter and had been showing us how much of a fighter she was over many years while fighting diabetes. She, sadly lost her fight April 2, 2007. It is an understatement to say we'll miss her.

Technical questions went to Sue Mellers, Kevin Zimmerman, and Brian Richards, three people that were always great, patient, and generous with their time. Another faculty member who I don't think knows how much he helped me is Borivoje Nikolić. He supported me in my personal life as well as my academic life when I really needed it.

Before I came to Berkeley, I lived almost 20 years in Japan. During that time I grew to be the person I am, made some great friends, and had some great mentors. Foremost is my seminar professor at Nanzan University in Nagoya, Muramoto Sensei, of the Information Technology department. He was the one that started me on this path and always showed support for my research and questions. Since sports was what originally brought me to Japan, I must thank my practice partners at Nagoya Denki high school and Aichi Kogyou University. I learned more from them than I can tell you. My university team at Nanzan University gave me some of the great friends of my life, and I miss them greatly, as I do my Tokyo friends.

The people that were great supports in my personal life were my mom, my big brother, Danny, Mieko Yamada and my partner Marcelle. My mom supported me in my dream of going back to school and putting off growing up a little longer. Danny was always the most interested in what I was doing and a personal support though separated by distance. Mieko was a person you only meet once in a lifetime. She was a samurai and taught me the importance of being true to a way of treating others and of carrying oneself. I never came near her grace and truth, but at least I have seen such a thing in my life. I'm guessing not everyone gets to meet such a person, or, if they do, they don't realize what a treasure they have in front of them. I am a better person having her in my circle of friends. And Marcelle, well Marcelle let me trash the house while I was writing, and didn't complain nearly as much as she had the right to. Having been through the whole PhD thing before, I used her unabashedly for her knowledge of the process.

And you can't do this thing without money. I'd like to thank my sources of income along the way: Cal, C2S2, and GAANN.

So I thank you all... And here's my thesis:

Chapter 1

Introduction

1.1 Introduction

Jitter is the random arrival time variation of a signal around what would be the ideally timed version of that signal. It is the enemy of any system whose accuracy depends on a signal arriving regularly at a specific time. For example, most sequential digital systems expect the clock signal, or, more correctly, the rising or falling edge of the signal used for clocking, to arrive at relatively even spaced intervals. If there is jitter on the clock signal which, as jitter does, randomly makes the expected signals arrive early at times and arrive late at others, the designer must take into consideration the maximum possible variation from the ideal and design their circuit accordingly, usually producing a circuit that is more complicated and has less than optimal performance. Analog to digital conversion (ADC) needs accurate clocks to sample data. Without accurately timed sampling, the digital values generated by the ADC will be inaccurate and, depending on the application and severity of the inaccuracies, it will be more difficult or impossible for the circuitry using the values to produce accurate results. More complex signal processing hardware can be built to deal with some jittery sampling, but this adds to design time, testing time, and other issues that come with added complexity. Using a phase-locked loop (PLL) has been the most common

jitter removal method in high performance circuits, but PLLs are complex circuits that require specialized knowledge to design for high-speed signals.

This thesis discusses an alternative jitter removal circuit dubbed here as the Jitter Attenuation Circuit (JAC). The goal of this circuit is to provide jitter reduction performance on par with commercially available PLLs, while being relatively simple to design and use as an on-chip solution. The main difference between the JAC and PLLs is that the JAC does not guarantee any phase alignment with its input. Its sole purpose is to remove jitter. In the following sections the effects of jitter, present methods to reduce jitter, and application of the JAC will be discussed.

1.2 Definition of Jitter

Figure 1.1 shows a square wave and what jitter on that square wave might look like. The center rising edge in Figure 1.1 is shown as a solid line. The solid line is the ideal signal, but in the presence of jitter, that edge may come earlier or later than expected as shown by the dotted lines on either side of the ideal edge. The aspect of jitter looked at is highly dependent on the system under consideration. For clock distribution systems, a small peak-to-peak time jitter on the clock edge is often most important whereas, in systems that require an accurate duty cycle, the duty cycle or pulse width jitter may be all that is cared about. Communication systems may be interested in jitter that is data dependent, or long-term jitter, a jitter measured as a variation in the accumulated width of many clock periods. The interest of the research presented in this thesis is centered on clocking and clock signals, and as such, the values looked at will be the peak-to-peak jitter (*pk-pk*), the root mean square jitter (*rms*), and jitter with long-term cyclic behavoir (*ltj*). The *pk-pk* jitter and *rms* jitter is shown in Figure 1.1. Long term periodic or pattern jitter will be a value added to *pk-pk* jitter.

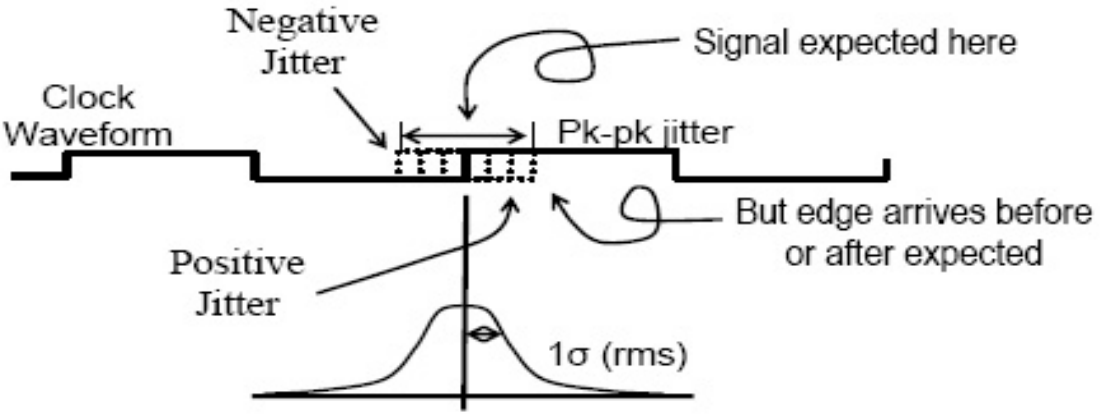


Figure 1.1: Jitter with Gaussian distribution.

1.3 Effects of Jitter

ADCs and digital systems are two places where clock signals are used and where clock jitter can have system-wide implications.

1.3.1 Jitter in ADC Systems

ADCs sample analog (continuous) signals and provide the data for digital processing. Transformation of a continuous time signal, $x_c(t)$ to a discrete time signal, $x_d[n]$, is done according to a clock, and, as such, is greatly at the mercy of how regular the clock might be. Jitter on the clock used to time when to sample the analog signal will give inaccurate data on the analog signal [*Oppenheim and Willsky*, 1997]. Figure 1.2 shows two pictures. Both pictures show a sinusoidal analog wave and a squarewave clock signal. The analog wave is sampled when the clock signal's rising edge reaches half its amplitude shown in Figure 1.2 by a horizontal line. In the picture on the left, there is no jitter on the clock signal, and the analog wave is accurately sampled and $x_d[n] = x_c(t)$. In the picture on the right, the center pulse of the clock signal has positive jitter and arrives slightly late. When this delayed clock pulse reaches its 50% value, the analog wave is sampled late and therefore inaccurately. The discrete sampled value $x_d[n]$ equals $x_c(t+\Delta t)$ instead of the correct $x_c(t)$.

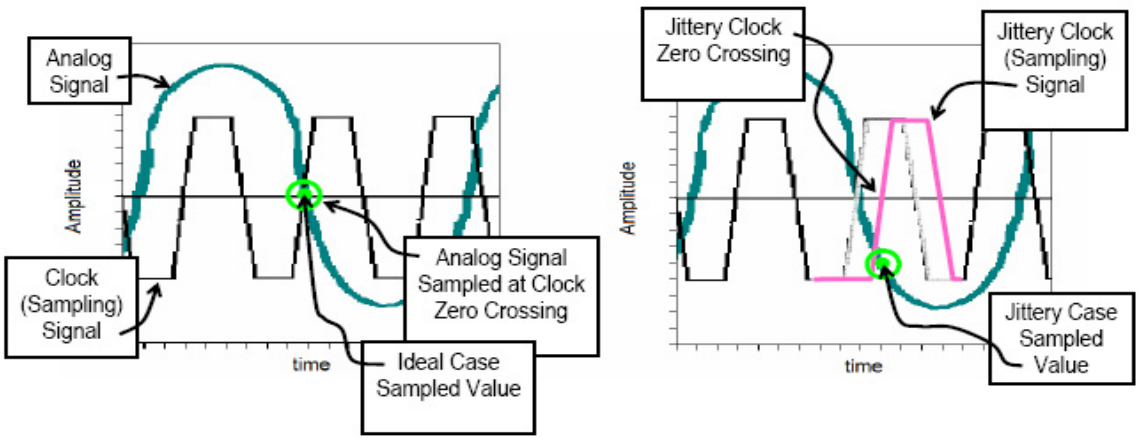


Figure 1.2: Affect of clock jitter on sampling of analog waves.

This inaccurate data is then delivered to the circuitry which knows nothing about the jitter or that the data is inaccurate.

The effect on accuracy for a sinusoidal wave can be quantified in terms of bits of accuracy. Given the analog signal:

$$S\left(t\right) = 2^{N-1} sin\left(2\pi F_S t\right) \tag{1.1}$$

where N is the number of bits of the ADC, and F_S is the analog signal frequency, the analog signal will have the its steepest slope as calculated in Equation 1.1.

$$\left.\frac{\partial S(t)}{\partial t}\right|_{t=0} = 2\pi F_S 2^{N-1} cos\left(2\pi F_S t\right)\Big|_{t=0} = 2\pi F_S 2^{N-1} \tag{1.2}$$

The error in the value sampled in the worst case can be found by multiplying this slope by the time jitter:

$$T_{jitter} \times slope = V_{Error} \tag{1.3}$$

Rearranging and solving for time jitter:

$$T_{jitter} = \frac{V_{Error}}{2\pi F_S 2^{N-1}} \tag{1.4}$$

The equation:

$$T_{jitter} = \frac{\Phi_{jitter}}{2\pi F_C} \tag{1.5}$$

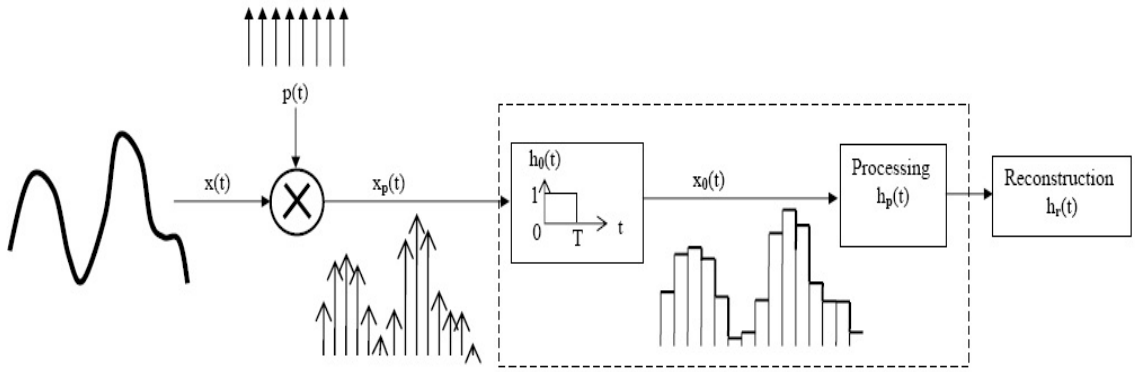


Figure 1.3: Zero order hold system.

where F_C is the clock frequency relates time jitter to phase jitter. Phase jitter has units of radians and is equivalent to the time jitter in terms of the clock period. Using Equation 1.4 and Equation 1.5, an equation relating phase jitter to V_{Error} can be derived:

$$\phi_{jitter} = \frac{F_C}{F_S} \times \frac{V_{Error}}{2^{N-1}} \tag{1.6}$$

A few things can be seen from this equation. First, the higher the number of times the analog signal is sampled, in other words, the higher the clock frequency, F_S, the smaller the phase jitter or phase error. Also, if the Nyquist rate is assumed ($2 \times F_S \leq F_C$), and the V_{Error} allowed is known, the worst case jitter can be calculated. The LSB of the ADC would be twice the amplitude divided by the number of bits. In this example the LSB would have a value of 1 ($LSB = 2 \times \frac{2^{N-1}}{2^N} = 1$). If a maximum error of $\frac{1}{4}V$ was required and the minimum clock frequency was used, an equation for phase jitter can be derived:

$$\phi_{jitter} = \frac{2 \times F_S}{F_S} \times \frac{\frac{1}{4}}{2^{N-1}} = 2^{-N} \tag{1.7}$$

The error described by Equations 1.1 through 1.7 is the maximum possible error in the data collected for digital processing. The error will, for example, effect the digital values in the zero hold system shown in Figure 1.3.

If discrete values gathered are used in signal processing hardware, using, for example, algorithms based on Fourier series, the effect of incorrect digital values will be additive in the frequency domain and will guarantee incorrect reconstruction of the original waveform.

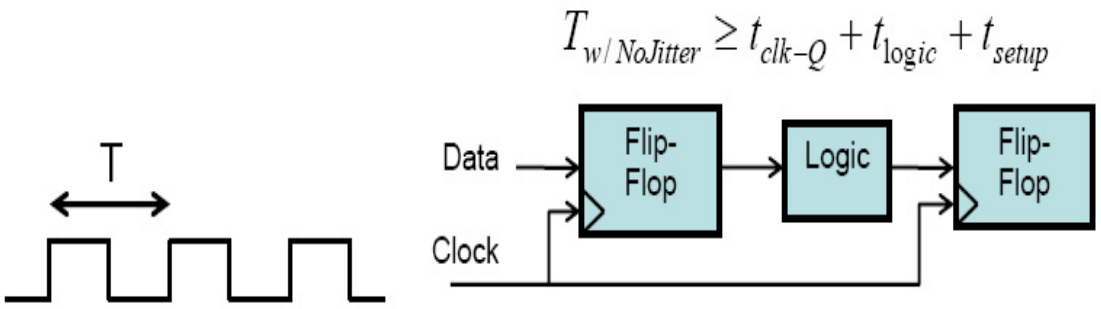


Figure 1.4: A digital system with no jitter.

The effect of jitter on the ADC clock is, therefore, detrimental or fatal to accuracy of circuits which use sampled analog signal data.

1.3.2 Jitter in Digital Systems

Clock signals in digital systems can have deterministic non-idealities (skew), and random non-idealities (jitter). Skew is most often dealt with at design time by matching clock signal path lengths, making sure impedances are matched, and by buffering among other methods to protect the integrity of the signal. Jitter is random and cannot be designed out. Jitter on clock signals used in digital system constrain the maximum speed at which the digital circuit can be run. Figure 1.4 shows a digital system with no jitter on its clock signal. In order for the system to be safe from setup time violations, the period must be greater than the sum of the $t_{clock-to-Q}$ time, the logic delay, and the setup time. This equation is shown in Figure 1.4.

In a digital system that has to accommodate a jittery clock, the equation is altered to include the peak-to-peak time jitter. The equation for the period in a digital system with a jittery clock is shown in Figure 1.5. The value t_{pk-pk} was used instead of t_{0-pk+} to cover the case where the clock into the first flip-flop has $-t_{0-pk-}$ on it and the second clock to the second flip-flop has t_{0-pk+} on it. Note that the longer period required to accommodate a jittery clock means a reduced frequency.

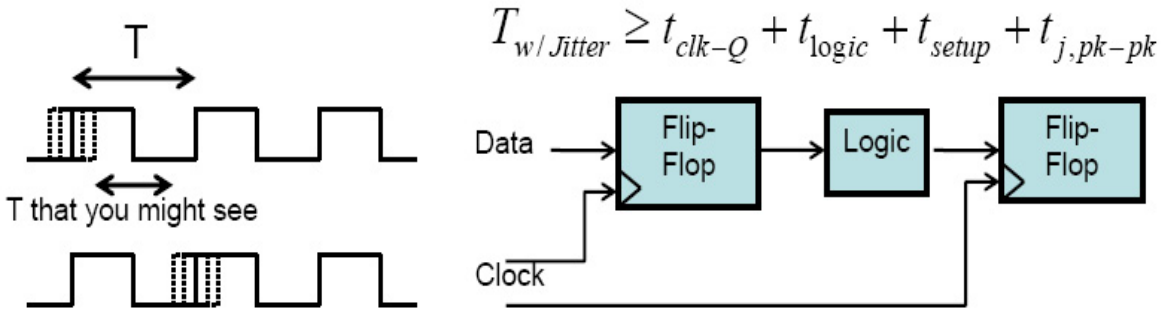


Figure 1.5: A digital system with jitter.

Jitter can also introduce hold time violations. In an ideal system, $t_{hold} < t_{clk-Q} + t_{logic}$ must be true to avoid hold time violations. In a system with jitter this constraint becomes $t_{hold} < t_{clk-Q} + t_{logic} - t_{pk-pk}$. Again, the value t_{pk-pk} is used to cover the case where the clock to the first flip-flop has t_{0-pk+} jitter on it and the cock to the second flip-flop has $-t_{0-pk-}$ jitter on it. Hold time violations can cause failure of a digital system and cannot be ignored.

The effect of jitter on a digital system can mean the system must be run at a lower clock rate at its best and system failure at its worst.

1.4 Methods Used to Reduce Jitter

This section discusses methods used to remove jitter from clock signals.

1.4.1 Hardware Solutions

If the clock signal has an acceptably low jitter for a system, jitter removal is, of course, not needed. A representative selection of low-jitter off-chip crystals are shown in Table 1.1. Note that the maximum frequencies available are a little over $1GHz$ and the best $pk-pk$ rms jitter was around $1ps$. Delivering this signal onto the chip where it will be used may cause degradation of the clock signal and add noise. Coupling, impedance mismatches in

the signal path, induction on chip pins and other environmental and physical influences may delay or degrade the signal such that jitter is added to the clock edge, the edge is moved or becomes badly defined, or the signal loses amplitude. Because of this, the numbers in Table 1.1 that have been measured at the crystal output, will be better than what is seen where the clock signal is actually used.

Maker	Model	Frequency	*rms*	*pk-pk*
Oscilent Corporation	489-XX.XM-5DN-TTS	$19.44 - 180MHz$	$1ps$	Not Stated
Oscilent Corporation	492-XX.XM-5DN-TCO	$19.44 - 180MHz$	$3.5ps$	Not Stated
Vectron International	VCC6-Q/R	$10 - 270MHz$	$4.8ps$	$38ps$
Vectron International	VCC6-Q	$270.1 - 800MHz$	$4ps$	$30ps$
Vectron International	SO-720	$270.1 - 800MHz$	$2.5ps$	$16ps$
Fox Electronics	RFX300	$600 - 1250MHz$	$3ps$	$30ps$
Epson Electronics America	EG-2121/2102CA	$53.125 - 700MHz$	$3ps$	$25ps$
Epson Electronics America	EG-2021/2002CA	$62.5 - 170MHz$	$3ps$	$25ps$
Crtstek Crystals Corporation	CVHD-930	$10 - 49.125MHz$	$5ps$	Not Stated
Crtstek Crystals Corporation	CVHD-960	$14 - 49.125MHz$	$1ps$	Not Stated

Table 1.1: Examples of low-jitter crystals.

Using an on-chip clock generation or cleaning method is a way to remove some of the non-idealities caused by delivering a clock from off-chip. As noted, the big advantage of using the off-chip solution is that it allows designers to avoid the designing and testing of complicated on-chip clock circuitry. An option which allows an on-chip solution but still reduces the design and testing time of on-chip clocking circuitry is the use of IP macro

blocks. Table 1.2 shows the performance of some commercially available IP clock generation or clock cleaning blocks.

Maker	Model	Frequency	*rms*	*pk-pk*
Parthus Technologies	PLLXpert Online Online custom design tool	$-450MHz$	$< 5ps$	Not Stated
Analog Bits, Inc.	DLL Technologies	$50 - 200MHz$	Not Stated	$50ps$
Analog Bits, Inc.	Video Capture PLL	$25 - 160MHz$	Not Stated	$125ps$
Partha Ceva Ceva	Video and Flat Panel Display Applications	$13 - 230MHz$	Not Stated	$< 14ps(1\sigma)$
Silicon and Software Systems	S3PLLPROGC90	$1200MHz$	$8.5ps$	Not Stated

Table 1.2: Examples of IP crystal alternatives.

The performance of almost all of the IP clock generation and cleaning blocks in Table 1.2 is below that of the off-chip solutions. The one exception is the IP generated by the Parthus Technologies PLLXpert Online design tool. It is quite surprising that there is not more of an advantage to using an on-chip solution at this point in time, though many IP companies have the performance data for their designs hidden and better solutions may exist than found in Table 1.2.

1.4.2 Phase-locked loop (PLL)

Phase-locked loops (PLLs) are devices used to align the phase of a generated clock signal to an input reference clock signal. They often offer multiplication and division of the reference clock frequency, and, as a byproduct of their use of a low-pass filter, they also offer some level of jitter removal.

PLLs use feedback to adjust the phase of the output signal to match that of the input signal. The input clock is divided to match a divided version of the output signal and a phase detector compares these two signals. The phase detector outputs a value related to

the difference in phase between the two signals. Jitter also causes changes in the output of the phase detector because an edge moved by jitter can also be viewed as a disagreement in phase of the output clock and input reference clock. Changes, when caused by jitter (or noise), are often of a high frequency and the low-pass filter, which the output of the phase detector passes through, reduces the effects of these random high-frequency additions to the proper output of the phase detector. The low-pass filter, in effect, holds a proper value for the voltage controlled oscillator (VCO) [*van Roon*, 2006]. The result that the filter removes jitter is a bonus. The low pass filter output is then transformed into a proper bias value for the voltage controlled oscillator which generates the output clock signal. A basic PLL block diagram is shown in Figure 1.6.

A PLL has other sources of noise that can not be ignored. The VCO generates noise and that noise introduces variations in the output signal. These variations will accumulate as time offsets in the generated clock edges over time. Increasing the bandwidth of the low-pass filter reduces the variation generated by the VCO, but as bandwidth is increased, less of the jitter on the input reference clock is removed. All blocks may be affected by their sensitivity to noise sources from both inside the PLL circuit, such as thermal noise and flicker noise, and from outside the PLL block, such as supply and substrate noise. Though all of these noise sources are commonplace and unavoidable, supply and substrate noise will often dominate [*Maneatis*, 2003].

PLLs filter out shorter-term jitter, such as variations in the reference clock period and the noise on the output of the phase detector, but may let longer-term jitter pass through. The amount of long-term jitter that will result depends on the sensitivity of the VCO to noise. VCOs based on LC oscillators typically have high quality factor Q, that can substantially reduces their sensitivity to noise sources, but VCOs based on RC oscillators, such as relaxation or ring oscillators, have low-Q, and thus are very sensitive to noise. This suggests that LC oscillators should be the prefered implementation of the VCO when designing a PLL, but the limited frequency range and the larger chip area requirement of LC oscillators can make an LC VCO implementation impractical or unusable.

There is another trade off: A reduced bandwidth of the low pass filter. The lower the

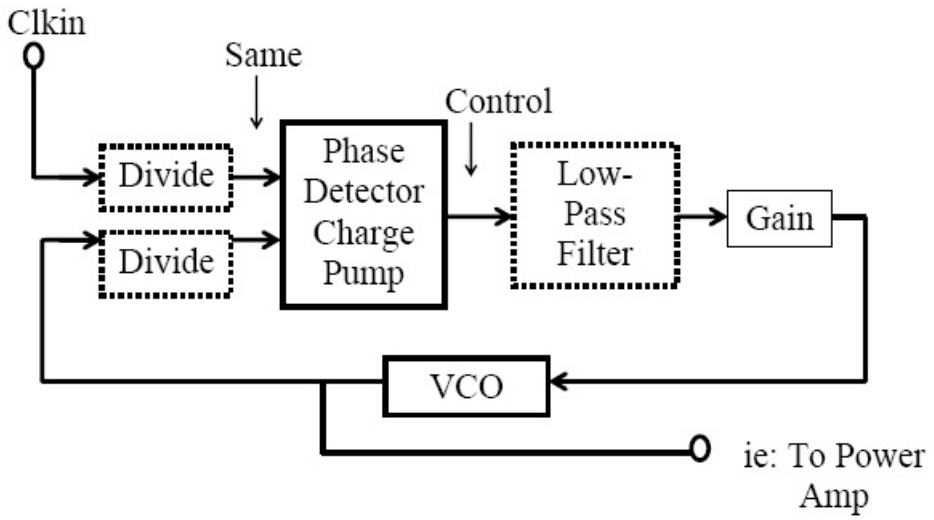


Figure 1.6: The basic structure of a phase-locked loop.

cutoff frequency of the filter, the longer it takes for the PLL to synchronize edges of the output clock to the input clock. The acquisition time is the measure of how quickly the PLL can switch between frequencies. Acquisition time is important in systems where there may be quick switching between frequencies as is required by, for example, the Bluetooth standard. The Bluetooth standard says a system can hop up to $1600 hops/second$ which means that a system may only stay on a frequency for $625 \mu sec$ [*Banerjee*, 2006], [*Bluetooth.org*, 2006]. High-speed switching requires a wider passband, but the wider the passband, the more jitter effects get through from the reference clock.

Another parameter that is often used in PLL design is the acquisition range. This is the frequency range that the PLL can acquire a lock to and is directly proportional to the loop bandwidth. The trade offs in PLL design are shown in Table 1.3.[*Razavi*, 1996], [*Savoj*, 2001], [*Anand*, 2001]

Increase loop bandwidth	Decrease loop bandwidth
Reduces acquisition time	Increases acquisition time
Increases acquisition range	Decreases acquisition range
Reduces VCO phase noise	Reduces reference clock jitter

Table 1.3: Phase locked loop design bandwidth trade offs.

The introduction of a phase-frequency detector before the charge pump block has improved performance. The phase-frequency detector effectively increases lock range and improves acquisition time while still allowing reduction of reference clock jitter. Though this improves acquisition time and range of the PLL, there is still self-generated VCO noise which appears up on the output of the PLL.

1.4.3 Delay-locked loop (DLL)

Delay-locked loops are another option that can be used to synchronize phase, and, compared to PLLs, they produce relatively little self-generated jitter. Delay-locked loops differ from phase-locked loops in that they do not use a voltage controlled oscillator to generate their output. As mentioned, VCOs are usually the main source of self-generated noise in PLLs. Figure 1.6 shows the basic structure of a DLL. The DLL uses a phase detector as does the PLL, but it uses the feedback information to speed up or slow down a delay line instead of control the frequency of a self-generated clock. In the simple case, this delay line is composed of a chain of inverters each current starved such that their output can be adjusted to a slower transition by decreasing their current flow, or sped up by allowing more current to flow [*Zilic*, 2001], [*Rabaey*, 1996]. The input clock is fed directly into the delay chain and as such, any jitter on the input will be delayed but eventually show up at the output. Because of this there is no reduction of the jitter on the input signal, but in the case where phase synchronization is required, a DLL may have less jitter on the output than a PLL simply due to the absence of a VCO.

Table 1.4 compares DLLs and PLLs.

1.4.4 Jitter Attenuation Circuit

Increasing clock speeds in digital circuitry and increasing data rates in communication systems are two trends that require ever faster clocks. As clocks speed up, their jitter must be reduced. For example a $1ns\ pk-pk$ jitter is probably not important on a $1KHz$ clock signal, but it is half the period for a $500MHz$ clock signal. This thesis describes a circuit

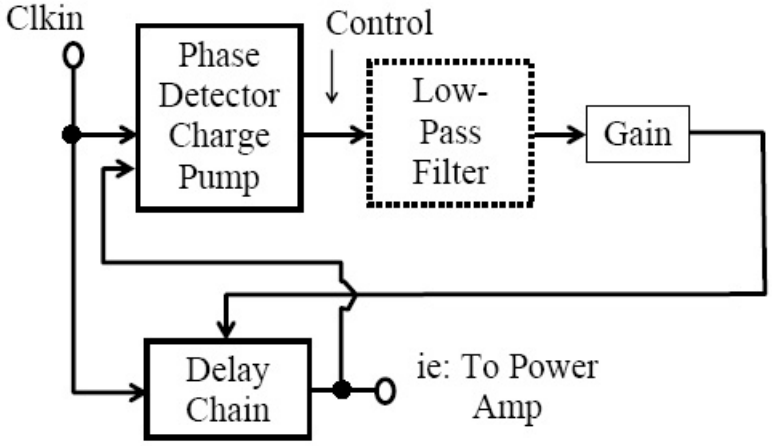


Figure 1.7: The basic structure of a delay-locked loop.

Phase-locked loops	Delay-locked loops
Second/Third order loops (Stability can be an issue)	First order loop (Always stable)
Frequency synthesis possible	Single output frequency
Input jitter is filtered	No VCO noise
Phase error accumulates (Lengthens acquisition time)	Phase error does not accumulate
Limited frequency capture range	Single output frequency
Unlimited phase capture range	Limited phase capture range

Table 1.4: Phase locked loop design bandwidth tradeoffs.

which has as its main function the goal of reducing jitter on high-speed periodic signals. It provides no phase synchronization and therefore has no need for some of the noise producing blocks of the PLL, specifically the phase detection block and the VCO. The argument for the reduction of functionality in the design is that there are places where merely a clean clock is needed and the extra phase-synchronization functionality of the PLL is extraneous. There is also the issue of complexity of design in building the PLL. To build a quality PLL on-chip, design knowledge, testing time, and often large area for the VCO is required. The circuit described in this thesis has been designed with simplicity as a goal such that it may be added to an on-chip design easily and require minimal testing. The underlying theory used in this circuit was originally described in [*Underhill*, 1998], [*Underhill*, 1999], [*Underhill*, 2001], [*Underhill*, 2003], [*Underhill and Brodrick*, 2004].

In Chapter 2 a toplevel block diagram is introduced and the contributions by non-idealities of each block to the jitter of the complete system is examined. In Chapter 3 through 5 design issues, block implementations, causes of block non-idealities, and other related topics are considered for each individual block. Chapter 6 discusses performance of the complete system and presents simulation results, and conclusions.

Chapter 2

High Level Circuit Basics

2.1 Introduction

This chapter describes the the theory behind the jitter attenuation circuit (JAC).

After looking at the theory behind how the circuit works, a top-level block diagram will be introduced and the expected input and output of the block will be described. This will allow investigation of what the performance of the entire circuit will be when there are non-ideal variations in the outputs of individual blocks and what non-idealities are most important to quash in order to guarantee a specified level of performance.

2.2 Basic JAC Theory

The JAC is a multi-staged circuit. The first stage changes the input square wave into another square wave which fits a particular set of requirements. The next stage takes the adjusted square wave and turns it into a sawtooth wave by creating a rising (falling) slope when the square wave is high, and creating a falling (rising) slope when it is low. The final stage is a level sensor which watches when the sawtooth crosses a certain level. The combination of these three blocks can remove 100% of the jitter from a signal if the circuit

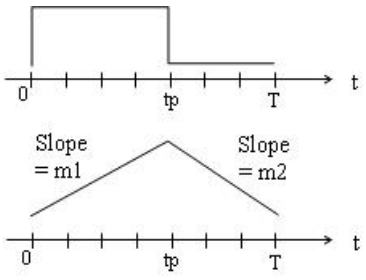


Figure 2.1: Bottom graph is integration of top square pulse around its DC average.

behaves ideally (no intrinsic noise, switching nonidealities, process variations, etc). The ideal case will be looked at in this section.

2.2.1 The Theory

An example of a relationship between a single square pulse and a graph of that pulse integrated around its DC average is shown in Figure 2.1. When the square pulse is high, the value on the graph below it rises with a slope of $m1$. When the square wave is low, the graph below the square wave falls at a slope of $m2$.

Also note that the bottom graph in Figure 2.1 ends at the same value that it starts at. This is not by chance, but by design, and is a requirement for the JAC theory presented in this section. Equation 2.1 puts this information into an equation where $m1$ and $m2$ are the rising and falling slopes respectively, T is the period and t_p is the amount of time the graph is rising per period.

$$t_p \times m1 - (T - t_p) \times m2 = 0 \tag{2.1}$$

This equation can be restated by saying that, if, for some time t_p the graph is rising at a slope of $m1$ and the rest of the time the graph is falling at slope $m2$, after time T, the graph will be at the same height as it was at time zero.

Figure 2.2 shows the integration of another square wave around its DC average value, but in this case the pulse does not rise from time zero, but starts to rise after a delay, t_d.

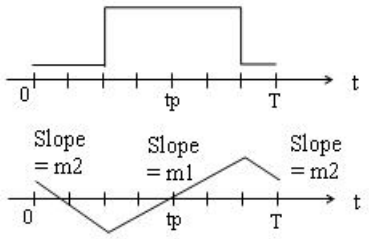


Figure 2.2: Bottom graph is integration of top delayed square pulse around its DC average.

Again, when the square pulse is high, the value on the graph below it rises with a slope of $m1$. When the square wave is low, the graph below the square wave falls at a slope of $m2$. When the pulse goes high, after t_d, the graph starts to rise with a slope $m1$, then, after time t_p, at a total time of $t_d + t_p$, the pulse goes low and the sawtooth falls with a slope of $m2$ again. An important point is that the pulse has the same width as the pulse in Figure 2.1, t_p. Equation 2.2 shows that simplifying the equation for the sawtooth in Figure 2.2 gives the same equation as that in Equation 2.1 which has been defined as equaling zero, and as with the sawtooth in Figure 2.1 that means that at time T, the graph has returned to the same height as it started at at time zero.

$$\begin{aligned} -t_d \times m2 + (t_d + t_p - t_d) \times m1 - (T - (t_d + t_p)) \times m2 \\ = t_p \times m1 - (T + t_d - t_d - t_p) \times m2 \\ = t_p \times m1 - (T - t_p) \times m2 \end{aligned} \tag{2.2}$$

Generalizing, for an $m1$, $m2$, T and t_p that satisfies Equation 2.1, the graph of the integration of any pulse or series of pulses that are high for a total time t_p during a period T, will end at the height that it began at at time zero.

Figure 2.3 shows a square wave with positive jitter on on some of its pulses. The first, third, and sixth pulses start on multiples of T, whereas the rest of the pulses are some amount delayed in their arrival. Below the square wave there is the sawtooth derived by integrating the square wave around its DC average. The first, third, and sixth pulses are

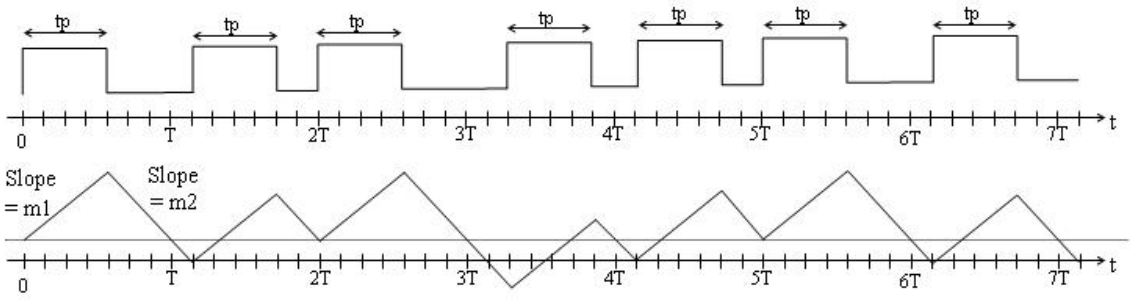


Figure 2.3: Bottom graph is integration of top square pulse around its DC average.

exactly the same as what was described in Figure 2.1; They start at time zero, stay high for some time t_p, and then go low. Also, the tooth of the sawtooth is exactly as shown in 2.1; They start at some height, rise at slope $m1$, fall at some slope $m2$, and end at the same height they began at. The requirement that T, t_p, $m1$, and $m2$ are values that bring the ending value back to whatever the starting value was, is as stated above, central to this JAC theory. The second, forth, fifth, and seventh pulse have positive jitter and arrive later than their ideal arrival timing. The second, forth, fifth, and seventh pulses are exactly the same as what was described in Figure 2.2; They start low, at some time t_d they go high, they stay high for some time t_p, then they go low for the rest of time T. Here, the teeth of the sawtooth are also exactly as shown in Figure 2.2; They start at some value, fall at some slope $m2$ until the corresponding square pulse goes high, then they rise at slope $m1$, then, when the square wave pulse goes low, they fall at some slope $m2$. They finally end at the same height they began at.

The rising sloped side of the sawtooth moves left and right depending on the amount the corresponding square wave moves. But note that in Figure 2.3 the falling slope leg of the sawtooth always hits its starting height at time T. In other words, the height of the falling sloped side of the sawtooth at T, $2T$, $3T$, and so forth, is always the same. Turning that around, you can say that every time a falling slope hits the same height as the sawtooth was at at time zero, time is at a multiple of T. If a pulse is generated every time this condition, *i.e.* the falling sloped leg of the sawtooth hits the height of the graph at time zero, that pulse will be exactly time T after the previous pulse and all jitter will have been

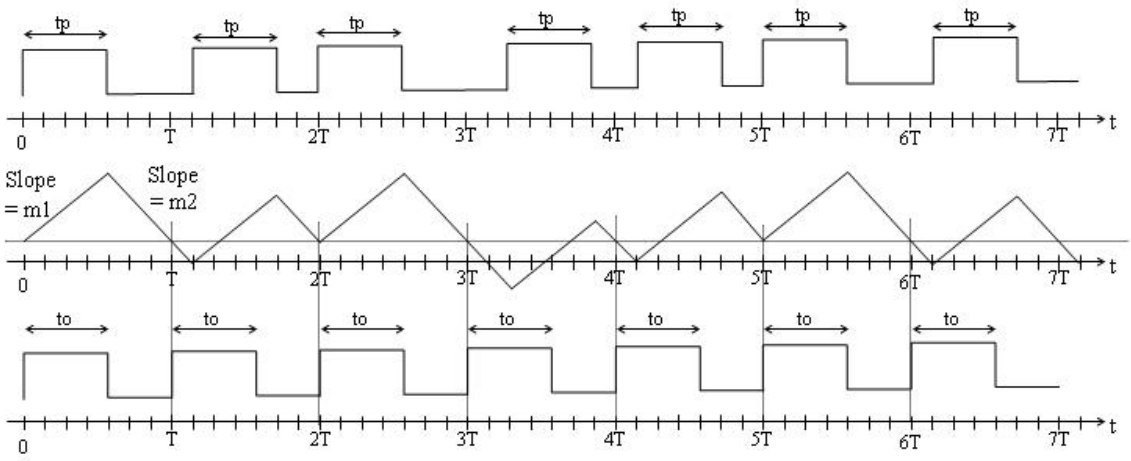


Figure 2.4: Generating a jitter-free square wave.

removed. In Figure 2.3, the original input signal had a large amount of jitter on it, but by doing the transformations described here, a perfect, 100 percent jitter free square wave has been generated. This method of removing jitter is the core of the JAC.

The point where the sawtooth crosses its long-term DC average has been used as the falling edge height where the jitter free pulse is to be generated. This has been used for ease of explanation and is not necessarily a requirement for the system to work. Figure 2.5 shows an elevated threshold that crosses the falling sloped leg of the sawtooth at a different point than the long-term DC average of the wave would. Now more of the graph is below the threshold, but all of the same arguments used to support using the long-term DC average height as a threshold can be applied to this new threshold. The only difference is that the phase is shifted slightly to the left compared to the long-term DC average case. In the same way, the threshold can be moved down, and, though phase shifted to the right, the output pulses will still be generated time T apart. Raising or lowering of the threshold can cause the system to be more likely to fail. The forth pulse is missed because there was too much jitter on the input and the sawtooth never was able to rise above the threshold value during the period $3T$ to $4T$. If the threshold was lowered below the DC average, there would be the chance that the system would fail at time $2T$ and at time $5T$ where there might be a chance that the sawtooth never got above the threshold.

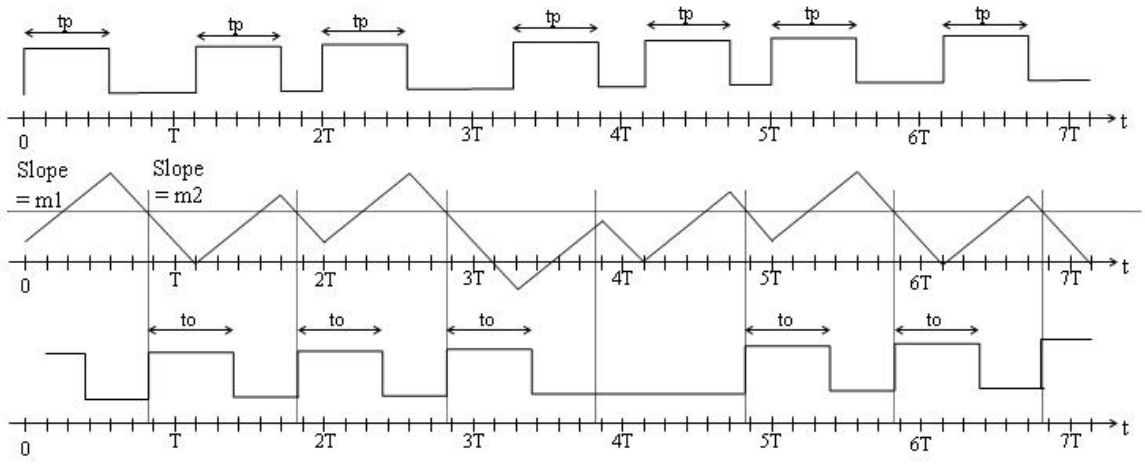


Figure 2.5: Raising the threshold.

Moving the threshold affectively increases or decreases the jitter seen on the input pulse. In Figure 2.5, moving the threshold up affectively added positive jitter to the pulse between time $3T$ and $4T$ and that added jitter was enough to move the end of the pulse into the next period. To avoid missing pulses due to threshold values, the threshold may only be moved from the long-term DC average value a height that keeps the pulse for that period completely contained in the period defined by that new threshold. The maximum jitter allowed in a period is $T - t_p$, and that T starts in different places depending on the value of the threshold. At the beginning of a period in a working system, the slope of the sawtooth will be the falling slope $m2$. A change in threshold will move the beginning of the period by the value shown in Equation 2.3.

$$\Delta T = \frac{\Delta V_{Threshold}}{m2} \tag{2.3}$$

The maximum jitter plus this change in T must be less than the maximum jitter allowed in the system, $T - t_p$. Equation 2.4 shows this relationship, and Equation 2.5 solves for the maximum threshold change that can be allowed.

$$T - T_{MAX-SHIFT} = T - (t_{j_pk-pk} + \Delta T) = T - (t_{j_pk-pk} + \frac{\Delta V_{Threshold}}{m2}) \geq T - t_p \tag{2.4}$$

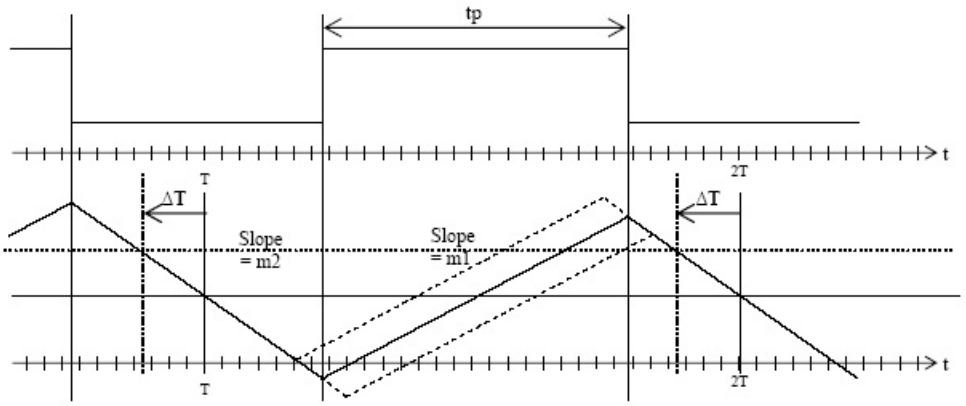


Figure 2.6: Shift in beginning of period for a higher threshold.

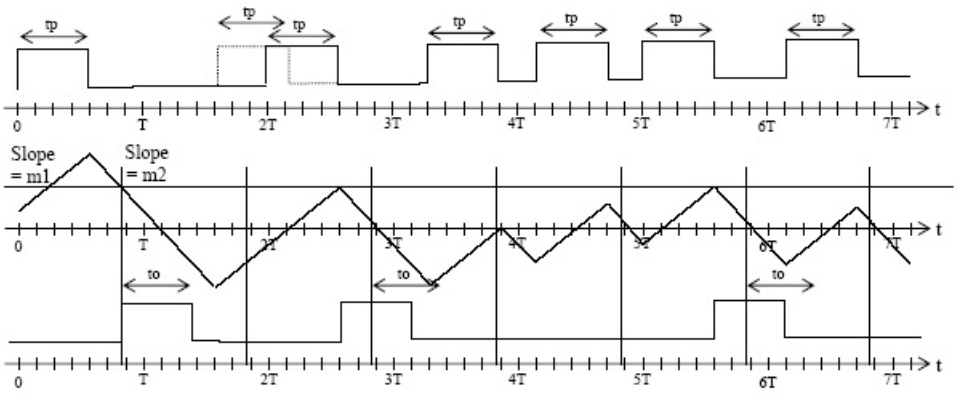


Figure 2.7: Jitter that cannot be accommodated.

$$\Delta V_{Threshold} = m2 \times (t_p - t_{jpk-pk}) \tag{2.5}$$

ΔT for a rise in threshold is shown in Figure 2.6.

Another type of failure can occur is when t_d is too large. For the system to succeed, there must be a complete pulse of width t_p for each period. If t_d is too long, then the chance that the pulse of width t_p for one period may overlap a pulse of another period may occur. Figure 2.7 shows the input pulse for period 2 overlapping with the pulse for period 3. This not only hurts the results for period 2, but all measurements done after that period are shifted and are being sensed incorrectly. Though seven output pulses should have been

generated, only three actually were. To assure no overlapping of t_p pulses, maximum jitter must be kept below $T - t_p$.

2.2.2 An Added Dimension: The Differential JAC System

At multi-GHz clock rates, the single ended system discussed in Section 2.2.1 is more susceptible to supply noise, process variation, and other design challenges than is allowable for the accuracy and performance that may be required. This section introduces the basic diffrential system, which reduces many of the circuit-level problems which occur in a single ended implementation.

Figure 2.8 shows an input square wave and both a positive and a negative sawtooth waves. In the differential system, there will be two square wave inputs; One will be high for t_p, and the other will be an inverted version that is low for t_p. These input waveforms will be called, respectively, $SquareWave_{t_pHIGH}$ and $SquareWave_{t_pLOW}$ and will have the same maximum and minimum amplitude values. Only $SquareWave_{t_pHIGH}$ is shown in Figure 2.8. The sawtooths generates by the $SquareWave_{t_pHIGH}$ and $SquareWave_{t_pLOW}$ will be inverted versions of each other also. $SquareWave_{t_pHIGH}$ will generate a sawtooth waveform which falls for time t_p and rises for $T - t_p$. $SquareWave_{t_pLOW}$ will generate a sawtooth waveform which rises for time t_p and falls for $T - t_p$. The sawtooth waveforms generated by $SquareWave_{t_pHIGH}$ and $SquareWave_{t_pLOW}$ will be called $Sawtooth_{t_pHIGH}$ and $Sawtooth_{t_pLOW}$.

The single ended JAC output was generated by finding when the falling slope of the sawtooth crossed the long-term DC average of the sawtooth or, as in later discussion, another threshold calculated such that system would not fail. In the differential version, an output pulse is generated when the falling slope edge of the negative sawtooth crosses the rising slope edge of the positive sawtooth. Where these two signals cross becomes the differential system's threshold.

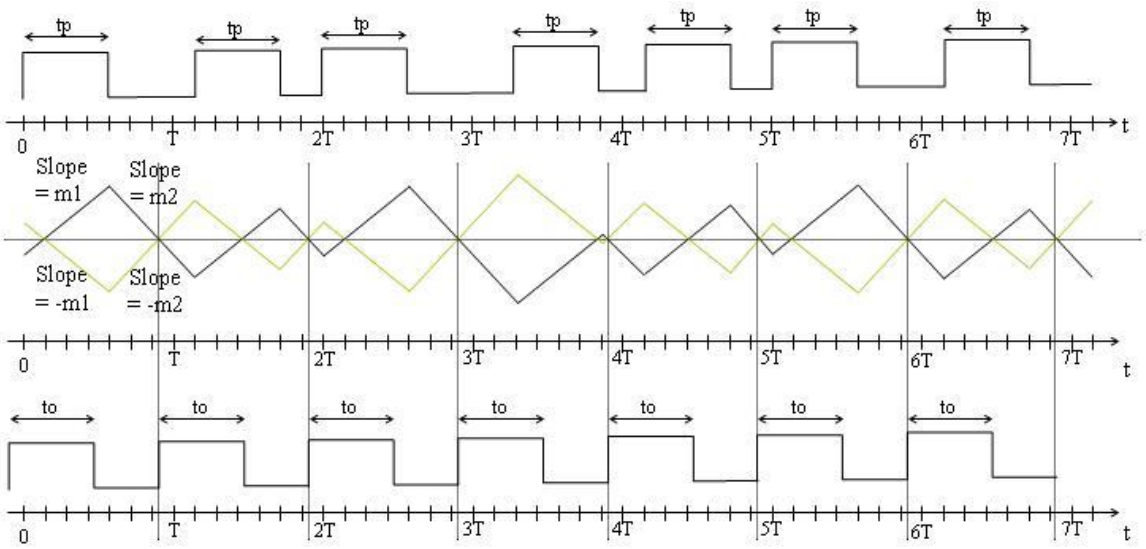


Figure 2.8: A differential system.

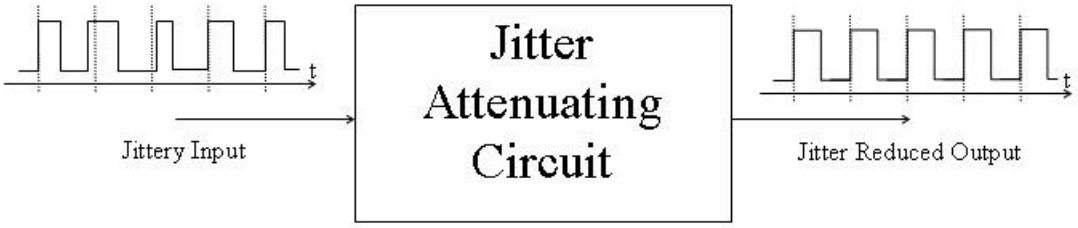


Figure 2.9: Top-level JAC system.

2.3 The JAC System

The JAC system must take a jittery signal as input, and outputs a signal with reduced jitter as illustrated in Figure 2.9. This section will break this high-level picture down into blocks then look at how nonidealities in one block affects the other blocks' performance and ultimately the output of the entire circuit. Discussions in this section will be regarding differential systems unless otherwise noted.

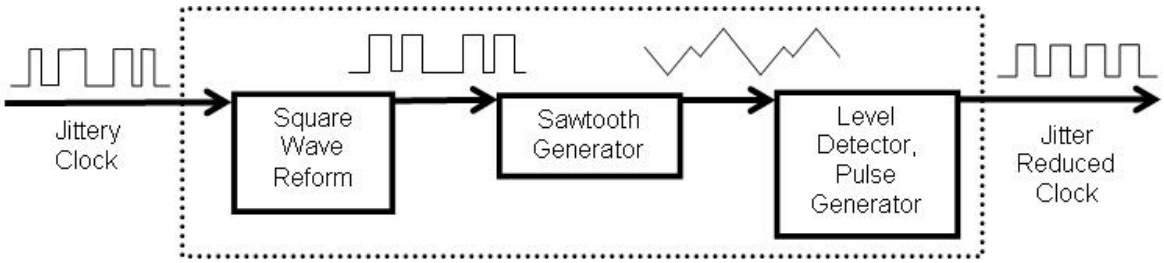


Figure 2.10: Ideal top-level JAC block diagram.

2.3.1 The Toplevel Blocks

Breaking the system down gives the three blocks mentioned in the last section, Section 2.2. The first block takes a jittery input clock signal and outputs a square wave with approximately the same jitter but with all duty cycles of width t_p. In actual implementation the amount of jitter on the input and the maximum and minimum duty cycle of the input pulses will be constrained, but here, as long as jitter is less than $T - t_p$, and t_p is greater than zero and less than T the system may be assumed to be (mathematically) viable. Also, from pulse to pulse, the duty cycle of the jittery input signal may vary freely. The second block generates the sawtooth waveform. It takes the average of the constant duty cycle square wave output from the first block and integrates that square wave around its own average. This is expressed in Equation 2.6. The third block detects when to output a pulse. For a single ended system, this is when the falling slope output ($m2$) from the integrator becomes less than a reference value. For a differential system, the final block watches when the $\pm m2$ sloped legs of differential sawtooth signals cross. When the third block detects a crossing, it generates a pulse which becomes the output of the entire JAC.

$$V_{int}(t) = \int_0^T \begin{cases} -m2\ dt & V_{Squarewave} > V_{AVE} \\ m1\ dt & V_{Squarewave} < V_{AVE} \end{cases} \tag{2.6}$$

2.3.2 Blocks for a Non-ideal System

So far only ideal systems have been discussed. An ideal system is a system whose sawtooth values, T, t_p, $m1$, and $m2$ always satisfy Equation 2.1. The three blocks introduced

in the previous section, Section 2.3.1, are all that is needed for an ideal system. But assuming there will be no supply noise, no process variation, and that biasing will be perfect is not realistic. There, therefore, needs to be a way to correct for incorrect T, t_p, $m1$, or $m2$ values. There are two ways to approach this problem. The first requires that at least one of these values, T, t_p, $m1$, or $m2$, needs to be adjustable so that errors caused by variation of any of the values can be corrected through feedback. The other method, and the one emphasized in this work, is a purely feedforward design which corrects sawtooth value errors at the output of the sawtooth generator. Feedback would correct T, t_p, $m1$, and $m2$ to make the sawtooth generator generate a correct sawtooth, whereas a feedforward system would take an incorrect sawtooth output and make it correct through an additional stage. Possible feedback and feedforward paths to correct T, t_p, $m1$, or $m2$ values are shown in 2.12 and the feedforward system is shown in Figure 2.13.

In theory, it seems that if none of the T, t_p, $m1$, or $m2$ values are adjustable and, say, either T is too short, $m1$ is too steep, $m2$ is not steep enough, or t_p is too long, the sawtooth should diverge as shown in Figure 2.11. This also seems to be supported by Equation 2.1. Equation 2.1 represents numerically the conditions for each period of the sawtooth to start from the same height as that of previous periods. If T, t_p, $m1$, or $m2$ are values which make this equation non-zero, it seems that each subsequent sawtooth should diverge off the value it should have by just exactly that non-zero value. In other words, each subsequent period should start a little more shifted off of the threshold value until it it has been shifted far enough to make the system fail. The theoretical case where the signal has shifted until the system has failed is exactly what is shown in Figure 2.11. Though theoretically this should happen, when sawtooths start to travel too far away from their ideal position, devices switch modes of operation and keep the sawtooths within a certain range of their ideal operating points, though deforming the sawtooths slightly. The trick used in this design is to bias the sawtooth generator as close to its ideal values as possible so it does not enter these other modes, and then have a post-processing block which shifts the sawtooths on top of each other as shown in Figure 2.13. This will be covered in depth in Chapter 5.

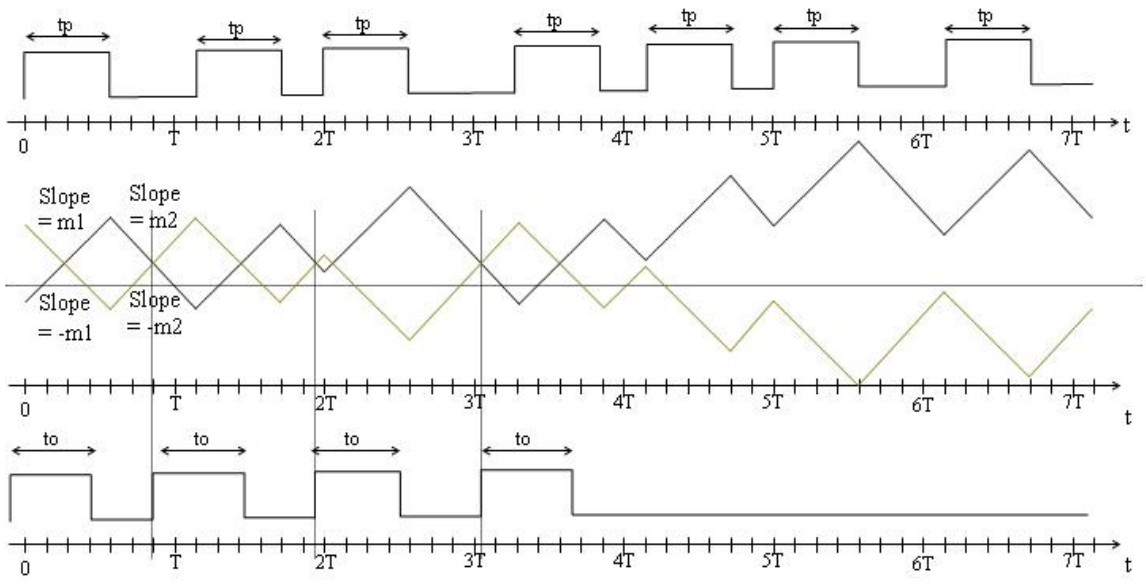


Figure 2.11: Affects of errors in T, t_p, $m1$, or $m2$ values.

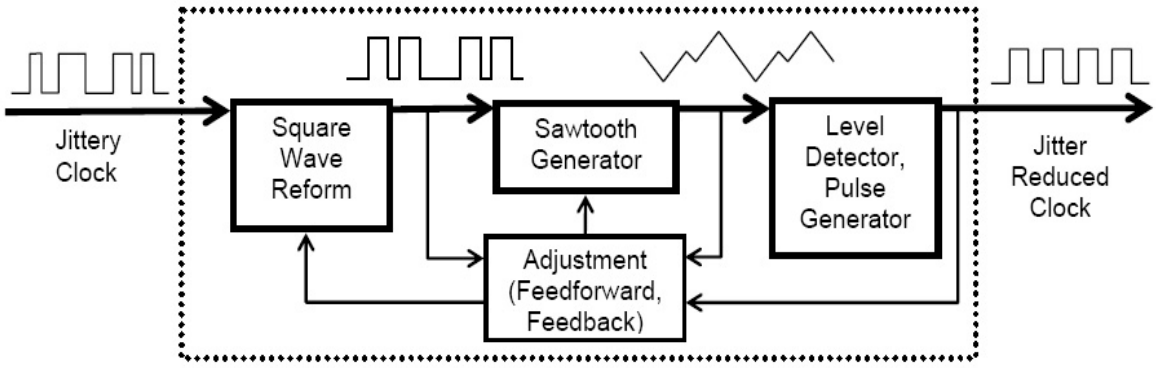


Figure 2.12: Top-level JAC block diagram with all possible feedforward and feedback adjustment paths.

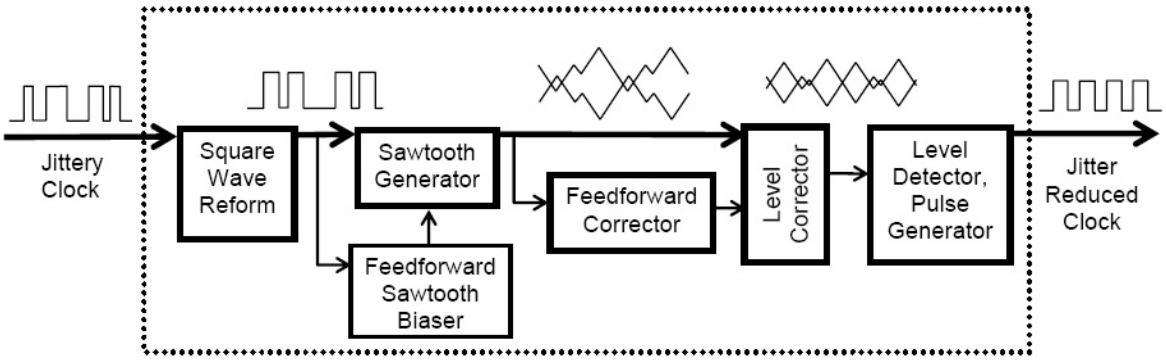


Figure 2.13: Fully feedforward design.

2.3.3 Block Nonidealities' Affect on System

In the ideal system, each block has a well defined output behavior. For example, for each occurrence of a rising edge of the jittery input square wave, the square wave reform block outputs $SquareWave_{t_pHIGH}$, with a duty cycle of width t_p, and $SquareWave_{t_pLOW}$ with a duty cycle of $T-t_p$. The integrator block uses $SquareWave_{t_pHIGH}$ and $SquareWave_{t_pLOW}$ as input and generates signals of slope $\pm m1$ when $SquareWave_{t_pHIGH}$ is high, and increasing and decreasing signals of slope $m2$ when $SquareWave_{t_pHIGH}$ is low. The level detector block senses when $Sawtooth_{t_pHIGH}$ become more than $Sawtooth_{t_pLOW}$ and outputs a high or low value depnding on how the block is implemented. This section looks at the four blocks shown in Figure 2.12, what irregularities or non-idealities could occur on their outputs, and how each output error contributes to the performance of subsequent blocks as well as the circuit as a whole.

"Square Wave Reform" Block (Input Monostable Block)

The "square wave reform" block takes in a square wave signal and "reforms" it into a square wave with constant width pulses at each of the input signal's rising edges, falling edges, or both. A square wave reform block which generates a pulse on the input's rising edge is shown in Figure 2.14. The input to this block is a jittery signal with peak-to-peak jitter of t_{jitter}, and a frequency of $freq$ as shown in Table 2.1. The output of this block will

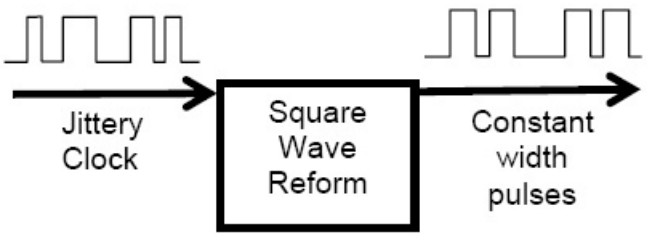


Figure 2.14: Square wave reform block.

be described using t_{ip} or $T - t_{ip}$, V_{iMAX}, V_{iMIN}, t_{iRISE}, t_{iFALL}, V_{iNOISE}, v_{iCROSS}, and t_{jitter} which represent the output waves' duty cycles, maximum or "ON" value, minimum or "OFF" value, rise time, fall time, amplitude noise, crossing height, and jitter after passing the block respectively. These are shown in Table 2.2.

Symbol	Description
t_{jitter}	Input jitter
$freq$	Input signal average frequency

Table 2.1: Parameters used to describe input to "Square Wave Reform" Block.

Symbol	Description
t_p	Ideal $SquareWave_{t_pHIGH}$ duty Cycle
V_{MAX}	Ideal $SquareWave_{t_pHIGH}$ maximum value
V_{MIN}	Ideal $SquareWave_{t_pHIGH}$ minimum value
V_{CROSS}	Ideal crossing height of output square waves
t_{ip}	Actual $SquareWave_{t_pHIGH}$ duty Cycle
$T - t_{ip}$	Actual $SquareWave_{t_pLOW}$ duty Cycle
Δt_{ip}	t_{ip} Variation from t_p
V_{iMAX}	High/ON value of output square waves
V_{iMIN}	Low/OFF value of output square waves
t_{iRISE}	Rise Time of output square waves
t_{iFALL}	Fall Time of output square waves
V_{iNOISE}	Amplitude Noise of output square waves
V_{iCROSS}	Crossing Voltage of output square waves
$t_{ijitter}$	Peak-to-peak jitter on output square waves

Table 2.2: Parameters used to describe output of square wave reform block.

When describing the output of the square wave reform block, a number of irregularities can be imagined. Possible irregularities are listed below.

1. Increased jitter ($t_{ijitter} > t_{jitter}$);
2. Output minimum and maximum voltages (V_{iMAX} and/or V_{iMIN} are not what was expected);
3. Amplitude noise (Supply, substrate and intrinsic);
4. Varying rise and fall times;
5. Non-matching behavior between differential signals (t_{ip} Differential error);
6. Common t_p variation on both positive and negative outputs (t_{ip} common mode error);

Not all the listed signal characteristics above need to be taken into consideration during circuit design, and those that do need to be taken into consideration may be important for different reasons. Some may be important at design time as output levels or bias values, some may be important when analyzing overall jitter of the system, and some others may be important in the way they affect subsequent blocks. A discussion of each signal listed above follows.

Increased jitter on output: Increased jitter, within bounds, does not need to be taken into consideration because it is occurring in the first stage of the system and will be removed by the later jitter removal stage of the circuit (the sawtooth generation block and output pulse generator). But, as noted at the end of Section 2.2.1, there are limitations on how much jitter can be accommodated by the system. If too much jitter is added, the system may fail. When designing, the amount of jitter on the incoming signal and the amount of jitter added by this block must be summed and that amount of jitter must be less than $T - t_p$. If that sum is not less than $T - t_p$, then t_{ip} must be reduced.

Output minimum and maximum voltages (V_{iMIN} and V_{iMAX}): The output minimum and maximum voltages V_{iMIN} and V_{iMAX} are values which cannot be ignored. They are, in effect, digital values which are only used to switch the sawtooth slopes between $m1$ and $m2$, and, if proper care is used in selecting their values, *i.e.* their

difference $V_{iMAX} - V_{iMIN}$ made to be as large as possible, then they will only contribute slightly to non-ideal behavior in the circuit. These values, as noted, switch the sawtooth slopes between $m1$ to $m2$ and control actual switches which steer current. For ideal behavior, switches should be completely on or off between switchings and transition between the two states as quickly as possible. The difference between V_{iMIN} and V_{iMAX} will be directly related to how completely the switches are either open or closed in the block which generates the sawtooth. A large difference will turn off the switches hard that need to be turned off and turn on switches hard that need to be turned on. The closer the values of V_{iMIN} and V_{iMAX} are, the more leakage occurs in the sawtooth generating block's switches and the leakage affects $m1$ and $m2$. Recall that $m1$ to $m2$ are values that need to be well defined for this circuit to work correctly. When they are off, Equation 2.1 becomes non-zero and there is the possibility that the sawtooths will diverge. Errors on V_{iMIN} and V_{iMAX} can be randomly varying or fixed and can be considered as part of the random differential error of the output of this block.

Rise and fall times, amplitude noise, and crossing voltages: Amplitude noise is most important when outputs switch from high to low or from low to high, and less important when the output is at a steady high or low value. Around the switching value, amplitude noise can make the switch be seen earlier or later than expected. Assuming both differential output signals have equal t_{iFALL} and t_{iRISE} values, the slope can be calculated to be approximately $(V_{iMAX} - V_{iMIN})/t_{iRISE}$ for a rising edge and $-(V_{iMAX} - V_{iMIN})/t_{iFALL}$ for a falling edge. If the maximum amplitude noise possible is $\pm V_{MaxNoise}$, then maximum variation in the edge arrival for a falling edge is as shown in Equation 2.7 and the maximum time variation in the edge arrival for a rising edge is as shown in Equation 2.8. This is also shown in Figure 2.15 for a falling edge. From Equation 2.7 and Equation 2.8, it can be seen that the steeper the rise and fall slopes are the less susceptable the output is to errors caused by noise.

$$\Delta t_{p_{AmpNoise}} = \frac{V_{MaxNoise}}{slope} = \frac{V_{MaxNoise}}{((V_{iMAX} - V_{iMIN})/t_{iRISE})} \tag{2.7}$$

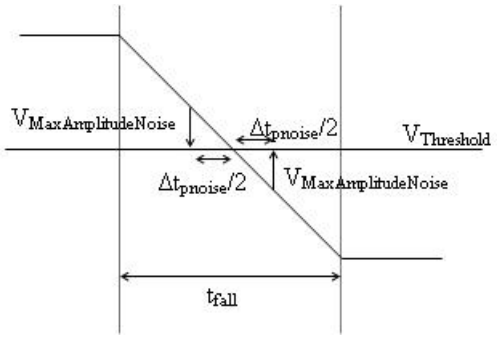


Figure 2.15: Falling edge maximum possible t_p error.

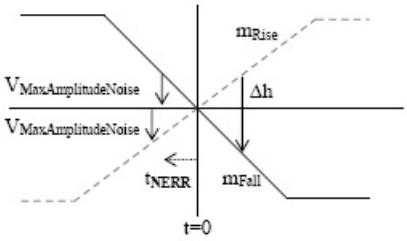


Figure 2.16: Worst case crossing timing error.

$$\Delta t_{p_{AmpNoise}} = V_{MaxNoise}/((V_{iMAX} - V_{iMIN})/t_{iFALL}) \tag{2.8}$$

If the switching point is considered to be at the point where the slope of the falling and rising output square waves cross, then both slopes must be taken into consideration when finding the maximum time error. In Figure 2.16 the total distance between the falling and rising slopes is Δh. For the two graphs to meet due to noise caused errors, Δh must equal the sum of errors for each signal. This is show in Equation 2.9.

$$\Delta h = (t_{NERR} \times m_{Rise}) + (t_{NERR} \times m_{Fall}) = 2 \times V_{MaxNoise} \tag{2.9}$$

Solving for T_{NERR} gives the results shown in Equation 2.10.

$$t_{NERR} = \frac{\Delta h}{m_{Rise} + m_{Fall}} = \frac{2 \times V_{MaxNoise}}{m_{Rise} + m_{Fall}} \tag{2.10}$$

Once again, this result illustrates that sharper slopes will have the affect of reducing timing errors caused by noise, but in this case, and this is the case that is most important, it is the sum of the slopes that is what matters most. The variations on the output of this block due to noise are random and can be considered as part of the differential error of the block. The crossing height can be random or deterministic and can contribute to both the differential and common mode error of the block's output.

Noise on V_{iMAX} and V_{iMIN}: Noise can affect the values of V_{iMAX} and V_{iMIN}, and therefore affect how much leakage is exhibited in the sawtooth generation block. The error caused by a variation in V_{iMAX} or V_{iMIN} is random and can be considered a differential and common mode error. When caused by intrinsic noise sources, it may be considered a differential error and when caused by supply swings, depending on implementation of the circuitry of the block, it may be considered a common mode error.

Variation in t_{ip}: t_{ip}, the duty cycle of $SquareWave_{t_pLOW}$, is one of the four values that must be generated correctly for the circuit to work as desired. Variations from t_p comes from noise in the system which, as described above, can move output crossing points earlier or later and cause a variation in t_p. As such, this will not be considered an error in itself, rather its value will be determined from the errors already discussed above.

V_{iCROSS} values: Noise caused errors on the crossing points of the output square waves were discussed above. The other issue that must be looked at when talking about the crossing point of the output square waves is actual height of the crossing. A crossing value that is too low can cause inefficient switching in the next block, as could a crossing value that is too high. Designing for a mid range, somewhere near $\frac{V_{iMAX}+V_{iMIN}}{2}$, protects against differential errors on the output of this block as well helping to avoid switching inefficiencies in the next block, the sawtooth generation block. Figure 2.17 shows an example of a case where outputs cross at a low value. Equation 2.11 and Equation 2.12 show the height of the crossing.

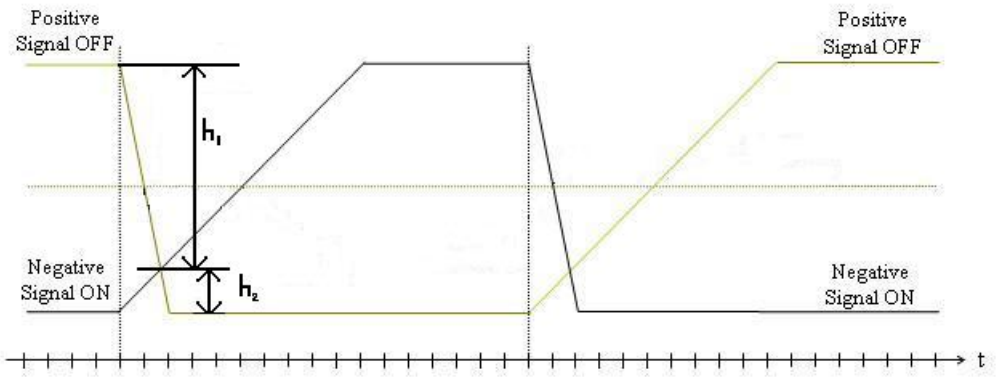


Figure 2.17: A case where the square wave reform block's outputs cross at a point lower than their mid-value.

$$V_{iCROSS} = V_{iMAX} - \frac{(V_{iMAX} - V_{iMIN}) \times t_{iRISE}}{t_{iRISE} + t_{iFALL}} \tag{2.11}$$

$$V_{iCROSS} = V_{iMIN} + \frac{(V_{iMAX} - V_{iMIN}) \times t_{iFALL}}{t_{iRISE} + t_{iFALL}} \tag{2.12}$$

The low crossing point can be explained by the large difference in fall time and rise time of the output square wave signals. If the sawtooth generator block is designed well, a variation in the V_{iCROSS} value will not add jitter to the system, but it is important to consider when design this block.

t_{ip} **value:** It will be shown that increasing t_{ip} could improve jitter reduction peformance of a complete JAC. Increasing of t_{ip} does not come without its costs though. There are two ways to increase t_{ip}; Increase the number of devices in the signal path or increase delay through the devices already there. Increasing the number of devices in the signal path from N to $N + \Delta N$ adds an additional t_{ip} error as show in Equation 2.13.

$$t_{pERR} = \Delta N \times t_{DelayPerDevice} \tag{2.13}$$

This error must be added to original t_{ip} error. If t_{ip} is made longer by slowing the existing devices, the slope at the output of each device will decrease and each output

will become more easily susceptable to noise. During design of the circuit the benefits of increasing t_{ip} must be measured against adding noise to the output of this block.

Three types of information were discussed in this section; Output time errors, output voltage values, and design issues. These are summarized in Table 2.18.

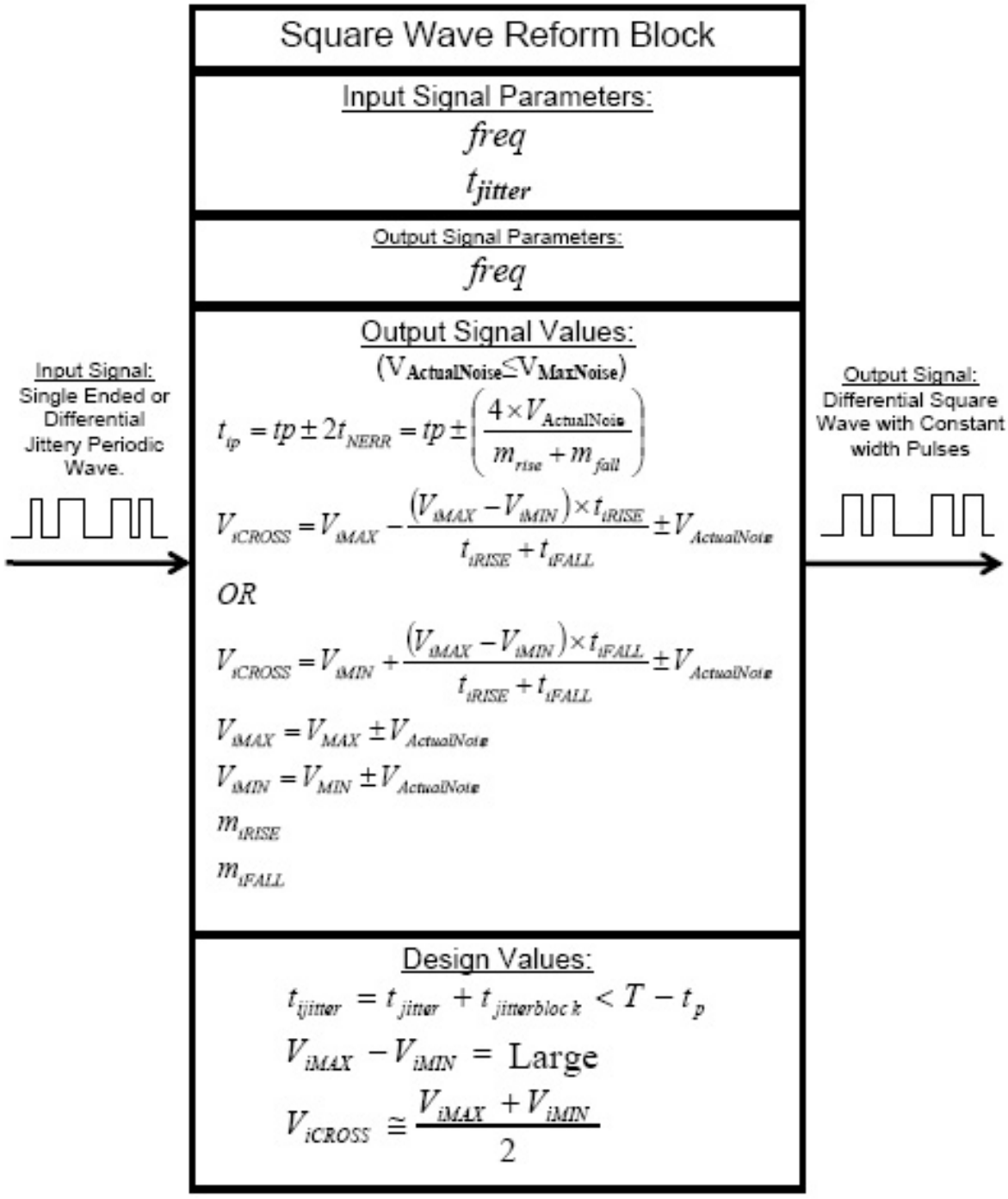


Figure 2.18: A summary of the square wave reform block's inputs, outputs, errors, and design points.

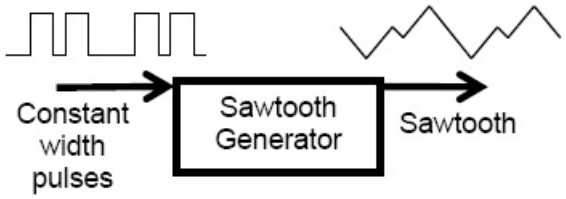


Figure 2.19: Sawtooth generator block.

Sawtooth Generator Block (Integrator Block)

The function of the Sawtooth Generator block is to integrate the square wave generated by the square wave reform block around the square wave's own average to generate a sawtooth signal as output. The inputs are exactly the outputs of the square wave reform block and are listed in Table 2.3. The outputs of this block are described using the values $m1$, $m2$, $t_{sjitter}$, V_{sNOISE}, V_{sCROSS}, and $P_{sTOLERANCE}$ representing, respectively, the non-crossing edge slope (leg of width of t_{ip}), crossing edge slope (crossing which generates output pulse), timing error (jitter) on crossing point, noise on output sawtooth signals, height when signals cross, and worst case process variation as shown in Table 2.4.

Symbol	Description
t_p	Ideal pulse width
Δt_{pi}	Pulse width variation from ideal
V_{iCROSS}	Height where input signals cross
V_{iMAX}	Input maximum value
V_{iMIN}	Input minimum value
m_{rise}	Slope of rising-sloped inputs
m_{fall}	Slope of falling-sloped inputs
t_{iRISE}	Rise Time of output square waves
t_{iFALL}	Fall Time of output square waves
$freq$	Input signal average frequency
$t_{ijitter}$	Peak-to-peak jitter on output square waves

Table 2.3: Parameters used to describe input to sawtooth generator block.

The list of possible irregularities for the sawtooth generator block itself are as listed below.

1. Affects from on irregularities on input;

   (a) Variations on t_p, Δt_{pi}

   (b) Variations on input crossing height

   (c) Noise on V_{iMAX} and V_{iMAX}

   (d) Slopes of output from square wave reform block

2. Amplitude noise;

3. Variation in delay between positive and negative outputs (differential errors);

4. Non-linearity on output sawtooth wave (Variations in $m1$ and $m2$);

Here, as was done for the square wave reform block, an analysis of input signals, output signals and block behavior will be done to highlight important points having to do with selecting design values and understanding ultimate JAC jitter performance.

t_p variations: Variations of the pulse width t_p have a strong effect on the output of the sawtooth generator block. Figure 2.20 shows a case where the pulse width is the ideal width, t_p. A variations to t_p is, of course, simply the shortening or lengthening of the pulse. Figure 2.21 shows the effect on the sawtooth generator block of having a pulse which is too long. In this case it is assumed that the pulse was turned on at the correct time but turned off late and the actual pule width is therefore longer than t_p by some Δt_{pi}. This variation of t_p by Δt_{pi} causes the timing error at the

Symbol	Description
$m1$	Ideal non-crossing edge slope (leg of width t_{ip})
$m2$	Ideal crossing edge slope (crossing which generates output pulse)
$t_{sjitter}$	Time error (jitter) on crossing
V_{sNOISE}	Amplitude Noise of output sawtooth waves
V_{sCROSS}	Height (voltage) on crossing
$P_{sTOLERANCE}$	Worst case process variation
V_{sMAX}	Output maximum allowable value
V_{sMIN}	Ouput minimum allowable value

Table 2.4: Parameters to describe output of sawtooth generator block.

output of the sawtooth generator block calculated in Equation 2.14. Because of the late turn-off, the falling slope edge starts at a higher point than it would otherwise. The total timing error $t_{sjitter}$ seen is the original timing error, Δt_{pi}, plus the extra time it takes for the falling slope side to drop the extra height.

$$t_{sjitter} = \Delta t_{pi} + \frac{\Delta t_p \times m1}{m2} = \Delta t_p \times (1 + \frac{m1}{m2}) \tag{2.14}$$

Another way to make a pulse which is too long is to turn the pulse on early. Figure 2.22 shows the effect on the sawtooth generator block of having a pulse which is turned on early (and turned off at what would have been the expected time) and therefore is longer than t_p by, again, some Δt_p. The output timing error, $t_{sjitter}$, is calculated in Equation 2.15.

$$t_{sjitter} = (t_{err1} \times m1)/m2 = ((\Delta t_{pi} + ((\Delta t_{pi} \times m2)/m1)) \times m1)/m2 = \Delta t_{pi} \times (1 + \frac{m1}{m2}) \tag{2.15}$$

Note that turning the t_p pulse on early causes the same error as turning it off late at the end of the t_p pulse. This makes sense and can be shown using the same proof that was used in Section 2.2.1 to show that moving the t_p pulse around does not alter the ending height of sawtooth period.

For the same reason that the same time error equation was found by lengthening the pulse by either starting the pulse early or ending it late, only one analysis of shortening the pulse is needed. To do the analysis, the pulse may be shortened from either end or from both; the results of starting the pulse late will be the same as ending the pulse early. Figure 2.23 shows a case where the pulse is shortened by turning the pulse off early. The timing error caused by a shortened t_p is shown in Equation 2.16.

$$t_{sjitter} = \Delta t_{pi} + (\Delta t_{pi} \times m1)/m2 \tag{2.16}$$

This is the same result as for the t_p error when t_p is increased by Δt_{pi}, except, in this case, Δt_{pi} is negative.

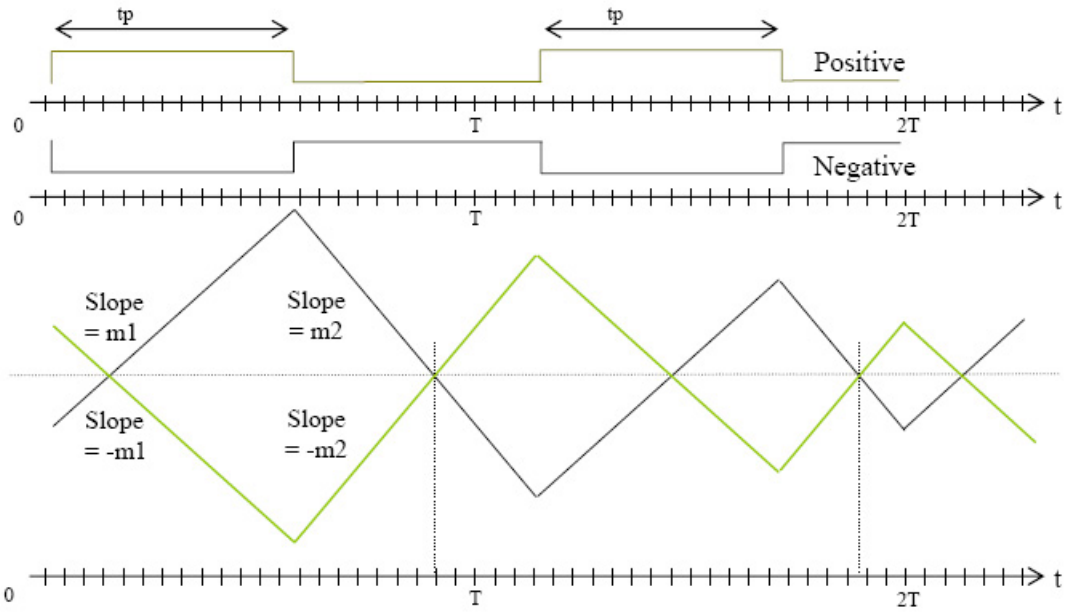


Figure 2.20: Ideal sawtooth input and resulting sawtooth.

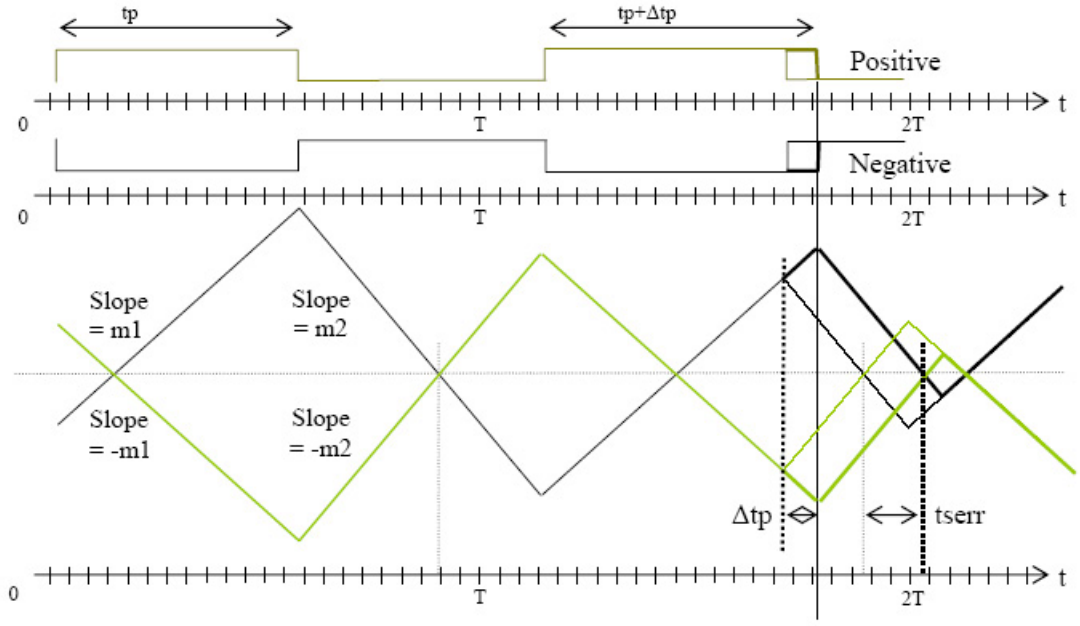


Figure 2.21: Square wave with t_p pulse turned off late.

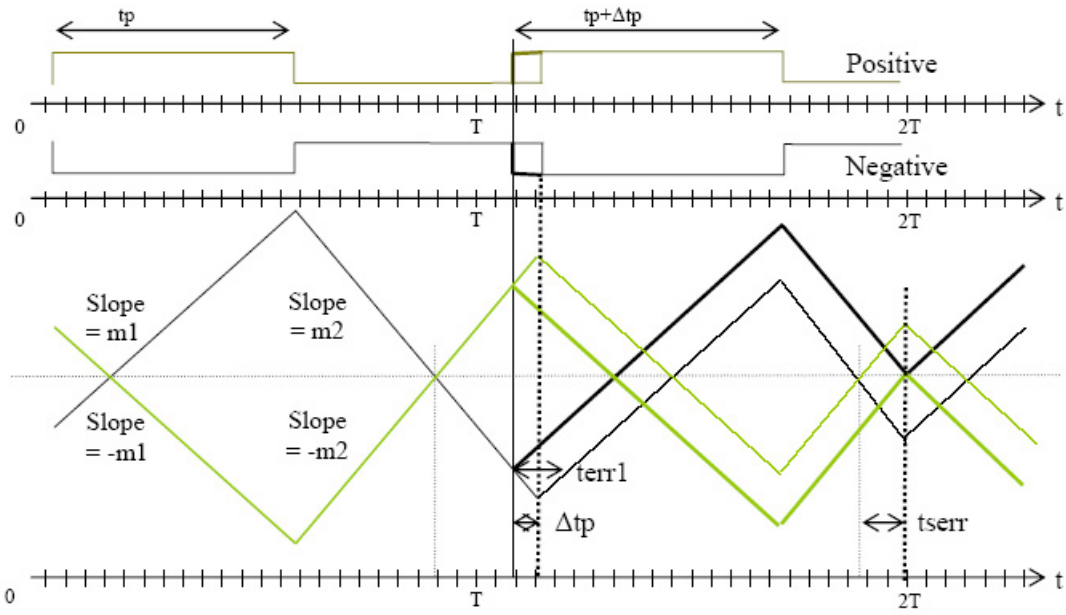


Figure 2.22: Square wave with t_p pulse turned on early.

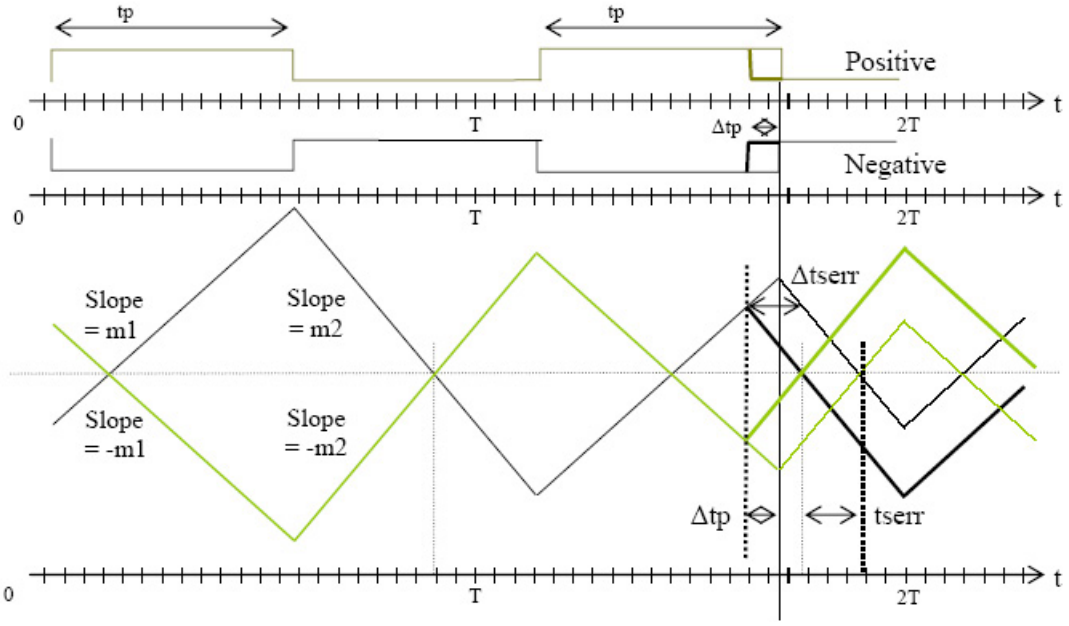


Figure 2.23: Square wave with t_p pulse turned off early.

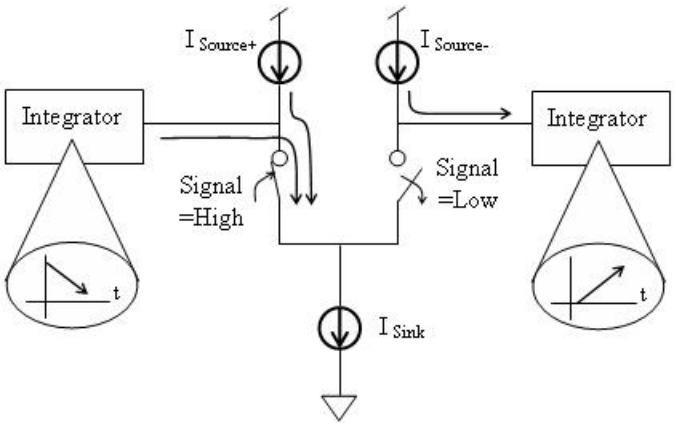


Figure 2.24: Sawtooth generator circuit with one signal from the square wave reform block high and one low (One swithc off and one on).

Before discussing other possible errors, the general implementation of the sawtooth generator must be introduced. The sawtooth generator consists of a differential pair whose outputs are connected to integrators implemented with capacitors. The differential pair steers current on and off of the integrating capacitors to generate voltages which change at the slopes of $m1$ and $m2$. The integrating capacitors generate these slopes according to Equation 2.17.

$$\frac{dV}{dt} = m = \frac{I}{C} \tag{2.17}$$

Figure 2.24 shows the sawtooth generator circuit.

A rising sloped voltage on an integrator happens when the switch on the same side as the integrator is open. This steers all of the current from the source to flow onto the integrating capacitor causing the voltage to rise. A falling sloped voltage on an integrator happens when the switch on the same side as the integrator is closed. When the switch is closed current equal to I_{Sink} is required by the tail current sink. Because the current required by the sink is greater than what the sources can provide, charge is pulled off of the integrating capacitor to make up the difference. The current flowing off the integrator

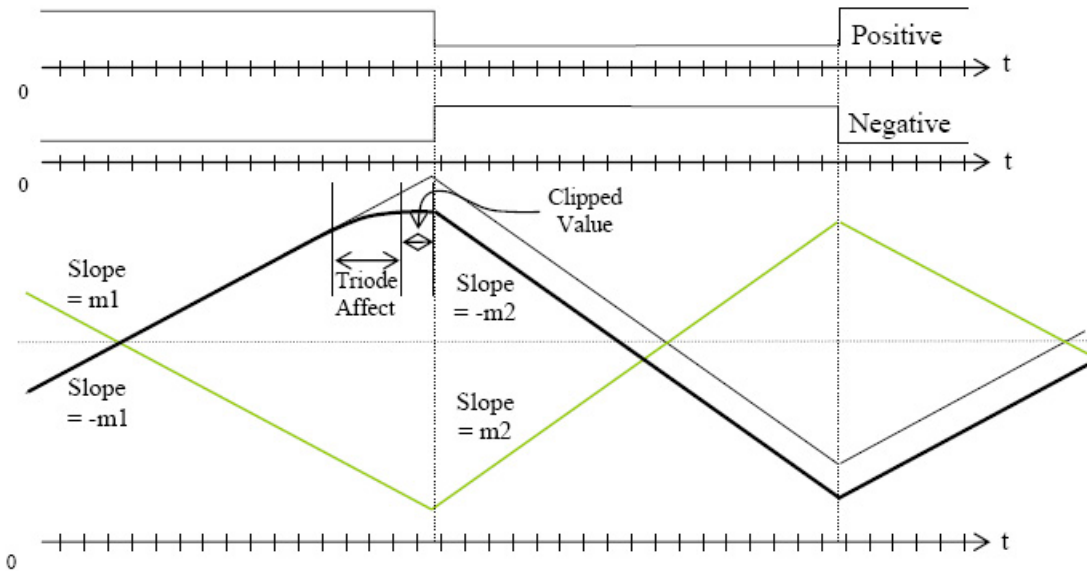


Figure 2.25: Clipped integrator signal.

causes the voltage to drop. Equations for the slopes for the two integrators in Figure 2.24 are shown in Equations 2.18 through 2.21.

$$m1 = \frac{(I_{Source+} - I_{Sink})}{C_+} \tag{2.18}$$

$$m2 = \frac{I_{Source+}}{C_+} \tag{2.19}$$

$$m1 = \frac{I_{Source-}}{C_-} \tag{2.20}$$

$$m2 = \frac{(I_{Source-} - I_{Sink})}{C_-} \tag{2.21}$$

Clipping: In the real world, there is a limit to the range the output of the integrator block can swing. If the output value is forced to go out of this range, the signal will begin clipping as shown in Figure 2.25. Both the positive and negative signals may experience clipping, but only clipping on the negative signal is shown in Figure 2.25.

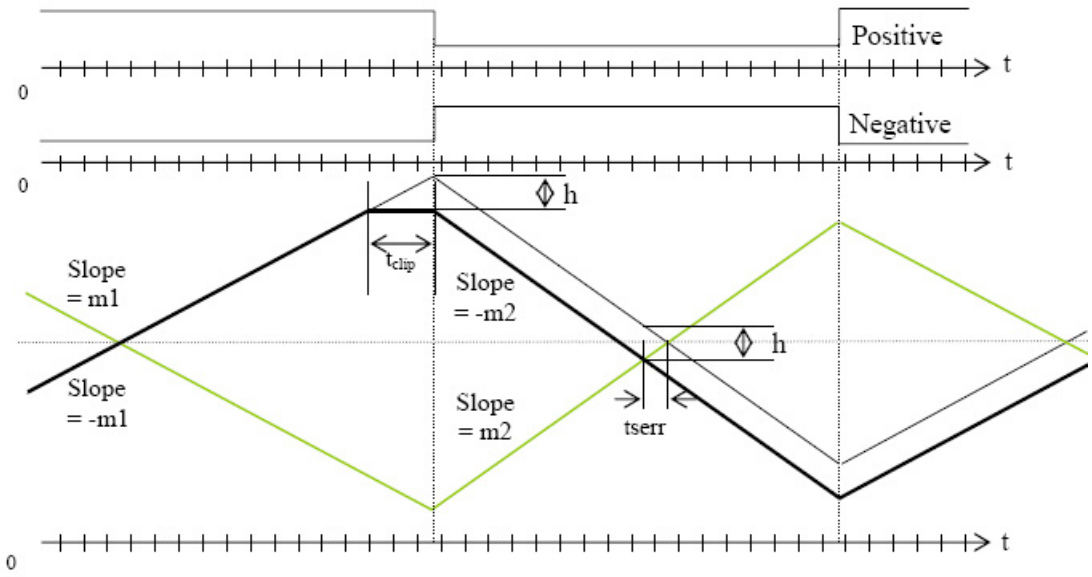


Figure 2.26: Clipping problem simplified.

Figure 2.25 shows a case where, when the output signals of the integrator block switch from $\pm m1$ to $\pm m2$, the negative output of the integrator block clips. Before the signal clips it passes through a non-linear region which is difficult to quantify. To quantify a clipping signal it is most convenient to simplify the problem as shown in Figure 2.26.

The error can be calculated as shown in Equation 2.22, but this requires knowing t_{clip}. To find t_{clip} other values must be known such as V_{top}, how high the signal would go if there was no clipping, and what the voltage, V_{clip}, that the signal will be clipped at.

$$\Delta t_{sjitter} = \frac{\frac{h}{2}}{m2} = \frac{\frac{m1 \times t_{clip}}{2}}{m2} = \frac{m1 \times t_{clip}}{2 \times m2} \tag{2.22}$$

Assuming that V_{clip} and V_{top} are known, the final equation becomes Equation 2.23.

$$\Delta t_{sjitter} = \frac{m1 \times t_{clip}}{2 \times m2} = \frac{m1 \times (\frac{V_{top} - V_{clip}}{m1})}{2 \times m2} = \frac{V_{top} - V_{clip}}{2 \times m2} \tag{2.23}$$

Clipping is discussed here, but, if the circuit is designed correctly, no clipping should occur. Clipping is noted here to make discussion of some of the other possible errors and design decisions clearer.

Affect of C_+ and C_- process variations: The output signal slope of the sawtooth generator block is, in the ideal case, controlled completely by the size of the capacitors used in integration, the tail current sink, and the current sources as shown in Figure 2.24. If the sawtooth generation block receives a square wave with pulse widths of t_p and a period of T, and has current sink and sources such that the integrators generate slopes of $m1$ and $m2$, and these values satisfy the basic JAC equation, $t_p \times m1 - (T - t_p) \times m2 = 0$, it can be shown that varying the value of C_+ and C_- will not affect the crossing point of the output of the sawtooth generator. As a reminder, the crossing point of the output of the sawtooth generator is what generates pulse for the output of the entire circuit and, as such, any error appearing on that crossing translates directly into jitter on the output. To show that the crossing point doesn't change with mismatches in capacitor values, it must be shown that the basic JAC equation is still true with altered capacitor values. Equations 2.18 and 2.19 describe $m1$ and $m2$ for the ideal case. Equation 2.24 has substituted those values into the basic JAC equation

$$t_p \times \frac{(I_{Source+} - I_{Sink})}{C_+} - (T - t_p) \times \frac{I_{Source+}}{C_+} = 0 \tag{2.24}$$

Assuming non-zero values for C_+, Equation 2.24 can be simplified to Equation 2.25.

$$t_p \times (I_{Source+} - I_{Sink}) - (T - t_p) \times I_{Source+} = 0 \tag{2.25}$$

Now, assuming a different value for the integrator capacitor of $C_+ + \Delta C_+$, and substituting it into the basic JAC equation, it can be shown that the equation will still be true. In other words, even with a different value of C_+, the sawtooth will return to the value it started at. Taking Equation 2.25 and dividing both sides by $C_+ + \Delta C_+$ gives the same results calculating new slopes based on a capacitance value of $C_+ + \Delta C_+$ and substituting the new slopes into the basic JAC equation:

$$t_p \times \frac{(I_{Source+} - I_{Sink})}{C_+ + \Delta C_+} - (T - t_p) \times \frac{I_{Source+}}{C_+ + \Delta C_+} = 0 \tag{2.26}$$

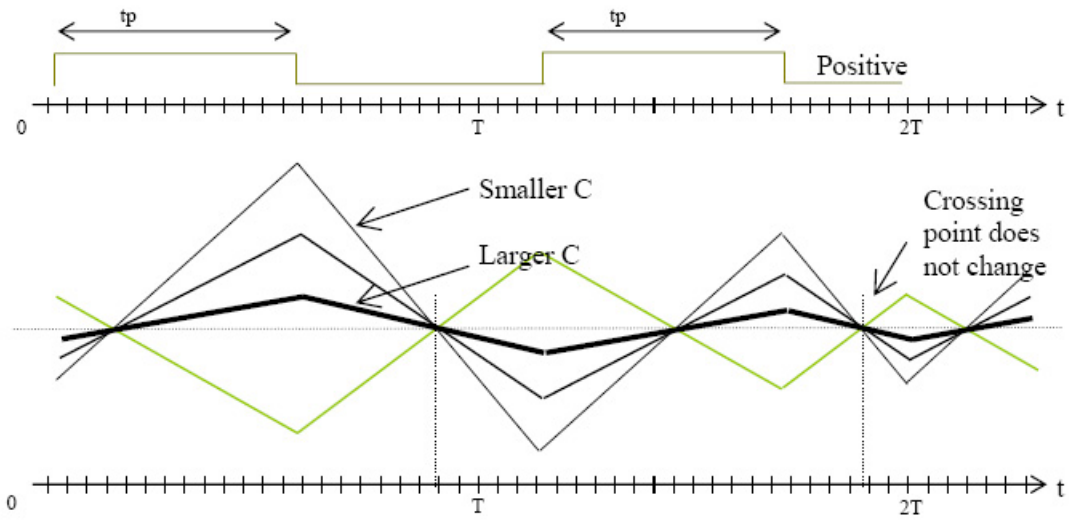


Figure 2.27: Affect of varying integrator capacitor sizes.

A graphic view of a sawtooth generated with different C integrator values is shown in Figure 2.27.

Though, mathematically, variations in the sizes of the integrating capacitors will not cause a circuit with correctly ratioed current sinks and sources to fail, if variations on those capacitors cause an increase in the output swing of the integrator, there is a chance the outputs may clip. As shown in Figure 2.27, the smaller the capacitor, the greater the output swing. At design time, the maximum possible process variation of the capacitors to be used must be taken into account. A capacitor with a 20% possible process variation could have 1.25 times the slope and require the slope aimed for at design time to be reduced to 80% of the design with no process variation possible. If the sawtooth swings around some center voltage, $V_{sCENTER}$, with a maximum allowable voltage to avoid clipping, $V_{sLIMITMAX}$, and maximum input jitter of t_{jitter}, an equation can be set up to relate the slope to the voltage limit and is shown in Equation 2.27.

$$(V_{sCENTER} + \frac{1}{2}t_p \times |m1|) + (t_{jitter} \times |m1|) < V_{sLIMITMAX} \tag{2.27}$$

Solving for the slope gives Equation 2.28.

$$|m1| < \frac{V_{sLIMITMAX} - V_{sCENTER}}{t_{jitter} + \frac{1}{2}t_p} \tag{2.28}$$

Adding in process variation information ($P_{sTOLERANCE} = 1 - Toleranceofdevice$) gives the final equation for maximum slope shown in Equation 2.29.

$$|m1| < P_{sTOLERANCE} \times (\frac{V_{sLIMITMAX} - V_{sCENTER}}{t_{jitter} + \frac{1}{2}t_p}) \tag{2.29}$$

This equation is valid for the rising edge of the negative signal. The positive signal the maximum slope equation is found through the same process and is shown in Equation 2.30.

$$|m2| < P_{sTOLERANCE} \times (\frac{V_{sLIMITMAX} - V_{sCENTER}}{t_{jitter} + \frac{1}{2}(T - t_p)}) \tag{2.30}$$

The allowable slope in the falling direction must also be defined and is in Equation 2.31 for the negative signal and Equation 2.32 for the positive.

$$|m2| < P_{sTOLERANCE} \times (\frac{V_{sCENTER} - V_{sLIMITMIN}}{t_{jitter} + \frac{1}{2}(T - t_p)}) \tag{2.31}$$

$$|m1| < P_{sTOLERANCE} \times (\frac{V_{sCENTER} - V_{sLIMITMIN}}{t_{jitter} + \frac{1}{2}t_p}) \tag{2.32}$$

Output slope: Though the crossing point is not affected by variation in the size of the integration capacitors, the slope of the output sawtooth legs are. As demonstrated in Section 2.3.3, the steeper the slope, the less sensitive the signals to noise, so reduction of slope to accommodate both worst case capacitor process variations and the limits that need to be observed to avoid clipping makes the output of the sawtooth generation block more susceptible to noise. Because any jitter introduced here shows up directly on the output of the JAC, it is important at design time to minimize the amount of

jitter that will be introduced at this stage. Equations 2.33 and 2.34 show the amount of jitter that can be introduced by noise at the output of the sawtooth generator.

$$\Delta h = t_{sjitter} \times m2 \times \frac{C}{C + \Delta C} = 2 \times V_{sMaxNoise} \tag{2.33}$$

$$t_{sjitter} = \frac{2V_{sMaxNoise}}{m2} \times \frac{C + \Delta C}{C} \tag{2.34}$$

Variations in current sources and sinks due to process variation: Equations 2.18 through 2.21 show that a change in any of the current sources or sinks will change either $m1$, $m2$, or both.

$$m1' = \frac{(I_{Source+} + \Delta I_{Source+} - I_{Sink} - \Delta I_{Sink})}{C_+} \tag{2.35}$$

$$m2' = \frac{I_{Source+} + \Delta I_{Source+}}{C_+} \tag{2.36}$$

$$m1' = \frac{I_{Source-} + \Delta I_{Source-}}{C_-} \tag{2.37}$$

$$m2' = \frac{(I_{Source-} + \Delta I_{Source-} - I_{Sink} - \Delta I_{Sink})}{C_-} \tag{2.38}$$

Also, if any one of the slope equations (with a changed current) is substituted into the basic JAC equation, the basic JAC equation will, most likely, no longer be true. The voltage error caused by process variation can be found by substituting the slope equations into the basic JAC equation. Equation 2.39 shows the result of substituting in the slopes of the $I_{Source-}$ side of the integrator into the basic JAC equation.

$$\begin{aligned}
\Delta V &= m1' \times t_p - m2' \times (T - t_P) \\
&= \frac{I_{Source-} + \Delta I_{Source-}}{C_-} \times t_p \\
&\quad - \frac{(I_{Source-} + \Delta I_{Source-} - I_{Sink} - \Delta I_{Sink})}{C_-} \times (T - t_P) \\
&= \frac{\Delta I_{Source-}}{C_-} \times t_p - \frac{(\Delta I_{Source-} - \Delta I_{Sink})}{C_-} \times (T - t_P)
\end{aligned} \tag{2.39}$$

The final jitter error is shown in Equation 2.40.

$$\begin{aligned} t_{Jitter} &= \frac{\Delta V}{m2'} = \frac{\Delta I_{Source-} \times t_p}{m2' \times C_-} - \frac{(\Delta I_{Source-} - \Delta I_{Sink}) \times (T - t_P)}{m2' \times C_-} \\ &= \frac{\Delta I_{Source-} \times t_p}{I_{Source-} + \Delta I_{Source-} - I_{Sink} - \Delta I_{Sink}} \\ &\quad - \frac{(\Delta I_{Source-} - \Delta I_{Sink}) \times (T - t_P)}{I_{Source-} + \Delta I_{Source-} - I_{Sink} - \Delta I_{Sink}} \end{aligned} \tag{2.40}$$

Using the long channel CMOS equation for devices in saturation, the current error can be further defined as shown in Equation 2.41. ΔW_{Dope}, ΔL_{Dope} and ΔV_{TDope} represent variations due to doping inconsistancies and W_{Size} and L_{Size} represent size variations.

$$\Delta I = \frac{1}{2}\frac{W}{L}(V_{GS} - V_T)^2 - \frac{1}{2}\frac{W \times (1 + \Delta W_{Dope}) + \Delta W_{Size}}{L \times (1 + \Delta L_{Dope}) + \Delta L_{Size}}(V_{GS} - V_T \times (1 + \Delta V_{TDope}))^2 \tag{2.41}$$

Assuming that errors on a single process variation can be considered linear in a small variation range [*Lu et al.*, 2004], and that V_{Tvar} is small compared to $V_{GS} - V_T$, Equation 2.41 can be rewritten and simplified as shown in Equation 2.42.

$$\begin{aligned} \Delta I &= \frac{1}{2}\frac{W}{L}(V_{GS} - V_T)^2 - \frac{1}{2}\frac{W \times (1 + \Delta W_{var})}{L \times (1 + \Delta L_{var})}(V_{GS} - V_T \times (1 + \Delta V_{Tvar}))^2 \\ &= \frac{1}{2}\frac{W}{L}(V_{GS} - V_T)^2 \left[1 - \frac{1 + \Delta W_{var}}{1 + \Delta L_{var}}\left(1 - \frac{2V_T \Delta V_{Tvar}}{V_{GS} - V_T}\right)\right] \end{aligned} \tag{2.42}$$

The final timing error is the ΔI from Equation 2.42 sustituted into Equation 2.43.

$$t_{jitter} = \frac{\Delta I}{C \times m2} \tag{2.43}$$

Process variation can cause the circuit to fail by eventually causing the sawtooth generator output to diverge. Ways to prevent this include making sure biasing circuitry is near the sawtooth generator so variations in one block will likely also be seen in the other. Using large devices will also help but to fully address this problem some

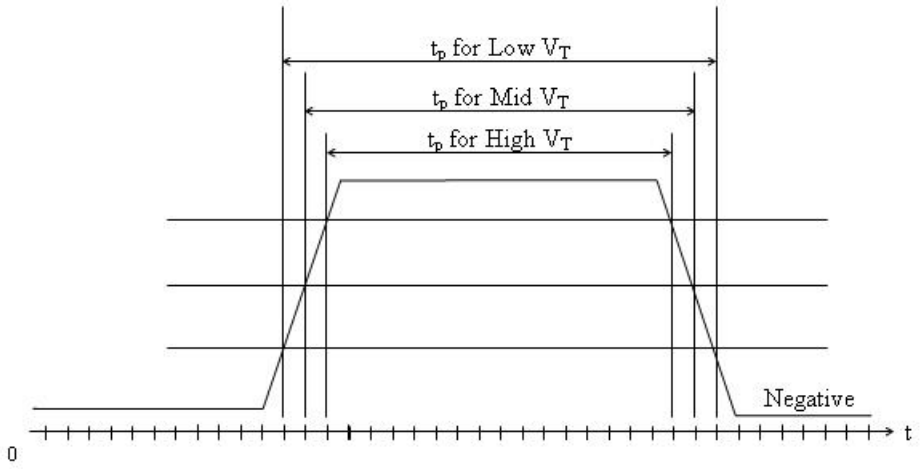


Figure 2.28: Affect of variation of V_T on t_p.

correction is necessary. In this work both feedback and feedforward are discussed as ways to address process variation and other nonidealities which move the sawtooths higher or lower than their ideal position.

Variations in the switches and threshold voltages, V_T: V_T and processing variations in the switching transistors used in the sawtooth generator block may cause additional non-idealities. Variation in the threshold voltage, V_T, can cause perceived variations in t_p as shown in Figure 2.28.

Depending on whether the V_T value is less than or greater than expected, the result could be a perceived shortening or lengthening of t_p. Because a variation in V_T effectively lengthens or shortens t_p, it can be included as a component of Δt_p. The value of Δt_p with consideration of V_T is shown in Equation 2.44.

$$\Delta t_p = t_{ip} - t_p + \frac{V_{Tideal} - V_{Tactual}}{m1} + \frac{V_{Tideal} - V_{Tactual}}{m2} \tag{2.44}$$

The corrected Δt_p from Equation 2.44 can now be substituted in to Equation 2.14 for a more correct calculation of the sawtooth generation block's output jitter. Equation 2.14 is repeated below as Equation 2.45 for convenience.

$$t_{sjitter} = \Delta t_p + \frac{\Delta t_p \times m1}{m2} = \Delta t_p \times (1 + \frac{m1}{m2}) \tag{2.45}$$

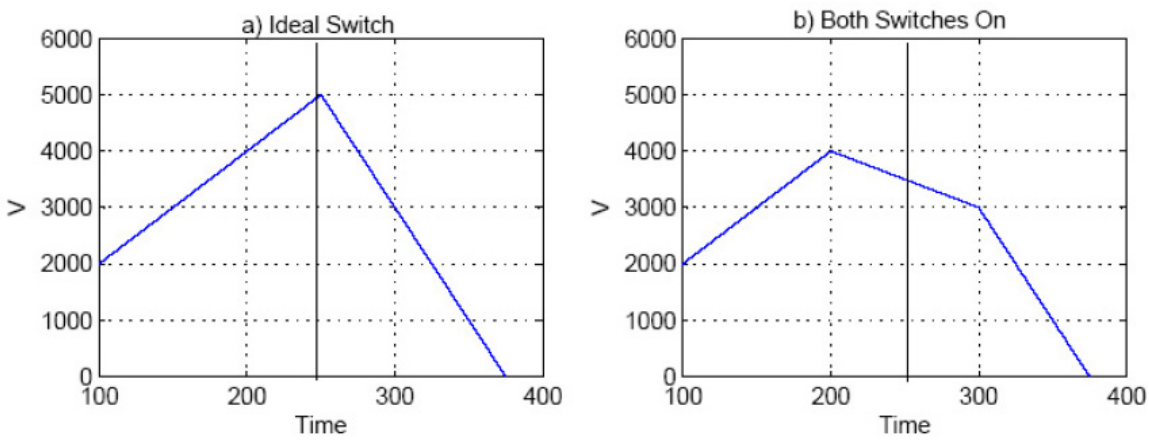


Figure 2.29: a) Ideal switching. b) Linear approximation of switching error.

Because of the steep slope of the output of the square wave reform block, variations in V_T will make a minimal difference in the performance of the sawtooth generator block. But, also, note that small dimension variations in short-channeled devices, such as the devices used in this project, can cause a larger difference in V_T [*Razavi*, 2001] and as such this error should not be disregarded completely.

Switching: Switching from $m1$ to $m2$ or from $m2$ to $m1$ does not happen instantaneously. Closing a switch can cause the transistors involved to go through all of their regions of operation before they arrive at their final state. This is not a linear path and necessarily introduces non-idealities to the signals controlled by the switch. Also, the larger the rise and fall time of the signal which flips the switch, the more time the switch will exist in this non-ideal region. Ideally, the slope should change instantaneously and produce a sharp point as shown in Figure 2.29 a). In the non-ideal case, the signals which switch the switches have a non-zero rise or fall time and cause a rounding of the point where switching occurs.

A simulation of a JAC circuit is shown in Figure 2.30 and shows the rounding. The slopes of the legs of the sawtooth are approximately as desired but the transitions at the peaks and valleys of the sawtooth are rounded, and the time it takes the sawtooth to change from one slope to the other and become linear again is approximately equal

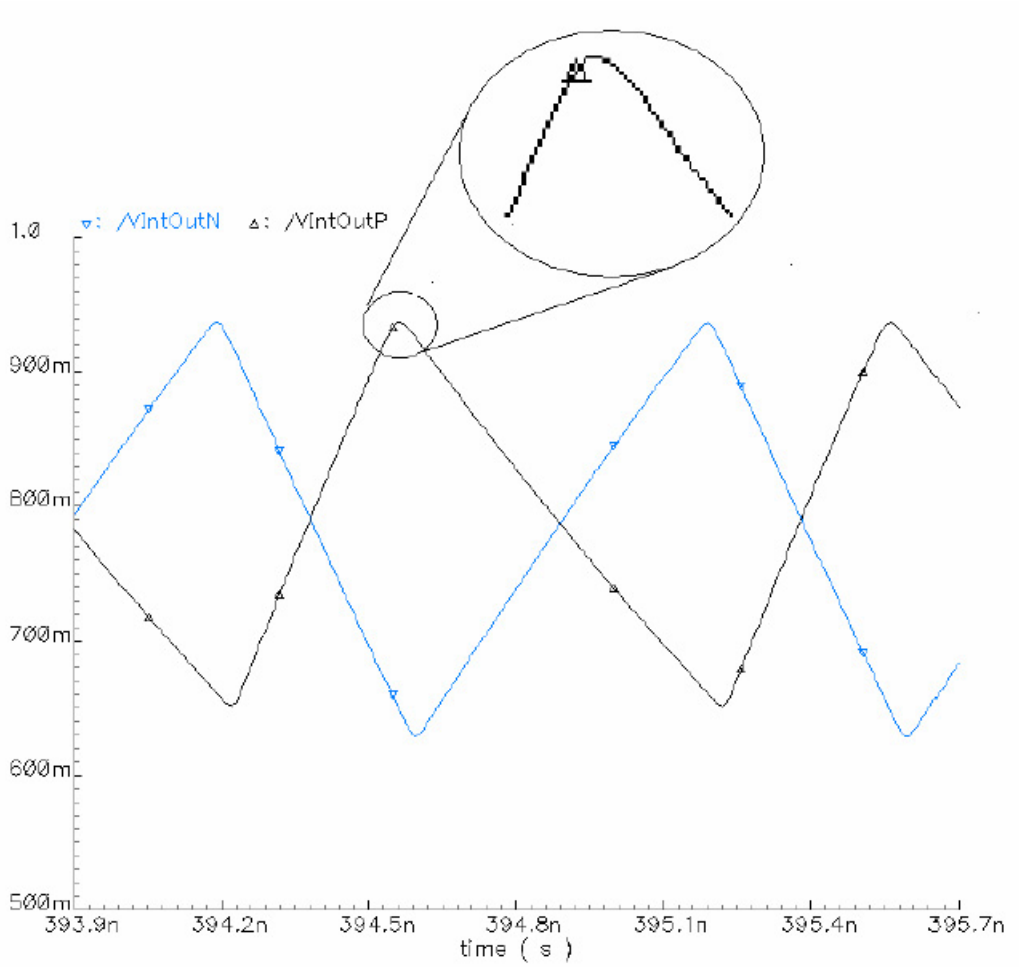


Figure 2.30: Simulation of output of sawtooth generator using inputs with 60ps rise time and 60% duty cycle.

to the rise time of the inputs, *i.e.* the rise and fall times of the signals generated by the square wave reform block. The square wave reform block signal has a duty cycle of 60% and rise and fall times of 60ps.

The next question is how to approximate the error caused during transition. If it is assumed that as soon as the switches begin to switch, in other words, as soon as the input signals start to change, both of the switches shown in Figure 2.24 turn on, and also assuming that both conduct equally ($\frac{I_{Sink}}{2}$) until the inputs to the sawtooth generator reach their final value, then the transition of the sawtooth wave will look

like the transition shown in Figure 2.29 b). The slope of the graph while both switches are conducting can be calculated by substituting $\frac{I_{Sink}}{2}$ for I_{Sink} in the slope equations.

$$\begin{aligned} Slope &= \frac{(I_{Source} - \frac{I_{Sink}}{2})}{C} = \frac{I_{Source}}{C} - \frac{I_{Sink}}{2 \times C} = \frac{I_{Source}}{C} + \frac{C \times m1 - I_{Source}}{2 \times C} \\ &= m2 + \frac{m1}{2} - \frac{m2}{2} = \frac{m1 + m2}{2} \end{aligned} \tag{2.46}$$

The error caused by switching using this approximation depends on whether the segment is centered around the ideal switching point or not. If the segment is centered around where the transition is actually supposed to occur, no error is introduced. But if the segment is not centered, the crossing point of the outputs of the sawtooth generator will be shifted.

Figure 2.32 shows the output of the integrator when both signals are on and $m1$ is greater than $m2$ and Figure 2.33 shows the output of the integrator when both signals are on and $m2$ is greater than $m1$. If the segment is shifted earlier or later by a time of T_{PN}, the difference in the amount of time spent on each side of the ideal switching point changes by twice that amount. Figure 2.31 shows an example of when the segment is shifted later. The error in the output of the integrator will be as shown in Equation 2.47.

$$t_{CrossShift} = 2\frac{h_2}{m2} = 2\frac{h_3 - h_1}{m2} = 2\left[\frac{t_{PN} \times m2}{m2} - t_{PN} \times \left(\frac{m1 + m2}{2m2}\right)\right] = t_{PN}(1 - \frac{m1}{m2}) \tag{2.47}$$

In the case that $m2$ is greater than $m1$ the output of the integrator is as shown in Figure 2.33, the error calculation is shown in Equation 2.48 to be the same as that calculated in Equation 2.47.

$$t_{CrossShift} = 2\frac{h_2}{m2} = 2\frac{h_3 - h_1}{m2} = 2\left[\frac{t_{PN} \times m2}{m2} - t_{PN} \times \left(\frac{m1 + m2}{2m2}\right)\right] = t_{PN}(1 - \frac{m1}{m2}) \tag{2.48}$$

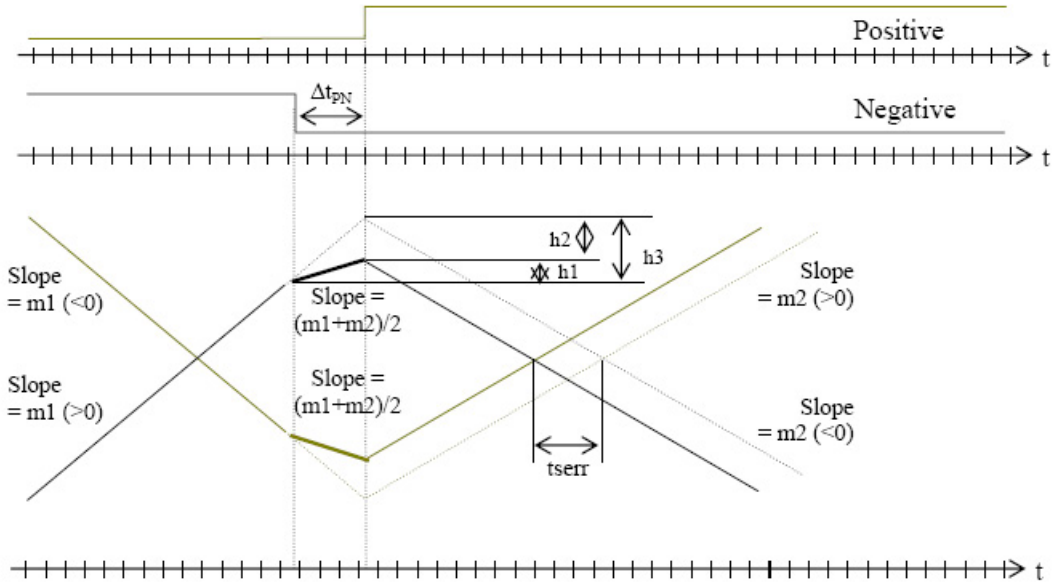


Figure 2.31: More time switching spent on rising slope side than falling slope side.

In Equations 2.47 and 2.48 the signs of $m1$ and $m2$ are opposite and therefore making $m2$ greater than $m1$ reduces the error.

The assumption made in the discussion of switching errors so far is that the switches are both on and conducting the same amount of current throughout the transition of the switch inputs. This is an idealized case. Another method to approximate the switching error is by using the first-order long-channel $V - I$ model to describe the behavior of the switches. The long-channel model for the current of a switch implemented using a CMOS device is shown in Equation 2.49.

$$I_{DS} = \frac{1}{2}\frac{W}{L}\mu_N C_{OX}(V_{GS} - V_T)^2 \tag{2.49}$$

Equations 2.18 ($m1 = \frac{(I_{Source+} - I_{Sink})}{C_+}$) and 2.19 ($m2 = \frac{I_{Source+}}{C_+}$) describe the slopes of the outputs of the sawtooth generator in terms of currents. When in the region of

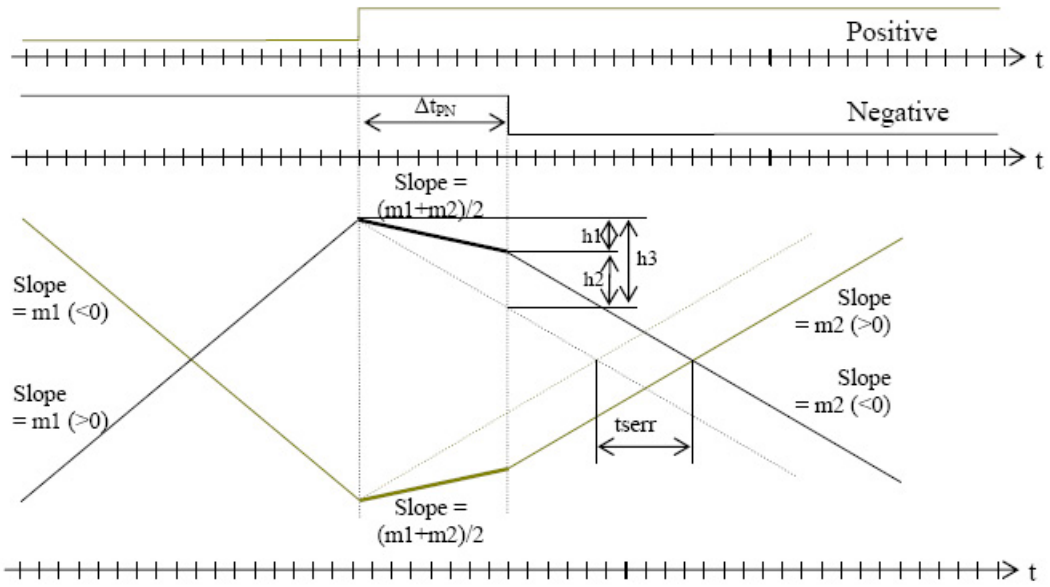


Figure 2.32: Both switches on. $m2 > m1$.

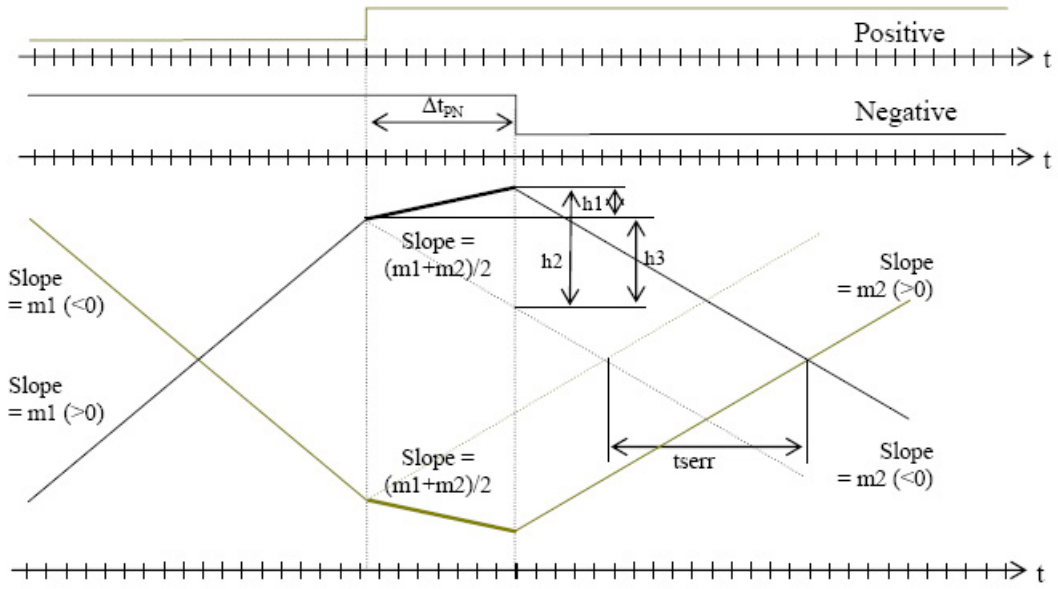


Figure 2.33: Both switches on. $m1 > m2$.

the graph where switching is occuring neither of these equations is applicable. When switching slopes, a more appropriate equation would be

$$m_{Switch} = \frac{(I_{Source} - I_{Switch})}{C} \tag{2.50}$$

where m_{Switch} is the slope of the output of the sawtooth generator block, I_{Switch} is the current through the switch on the side being observed, I_{Source} is either of the current sources shown in Figure 2.24, and C is the capacitance used in the sawtooth generator integrator. When going from a rising slope to a falling slope, I_{Switch} goes from zero to I_{Sink}, and when going from a falling slope to a rising slope, I_{Switch} goes from I_{Sink} to zero.

Before giving a mathematical explanation of the method of approximation using the long-channel equation there are a few points that should be noted. First, the input to the sawtooth generator block from the square wave reform block will be considered linear. This approximation can be made because the output of the square wave reform block seen by the sawtooth generator block is centered in its amplification region and therefore gives somewhat linear behavior.

Second, at the beginning of switching the output of the sawtooth generator block will have the slope of either $m1$ or $m2$, and, at the end of the switching time, the output will have an output with a slope of, respectively, either $m2$ or $m1$. This gives the initial conditions and final conditions of the curve through the transition. Knowing the initial and final conditions of the curve allows simplification of the mathematics. Using Equations 2.18 through 2.21 and the rise and fall times and and the maximum and minimum values of the output of the square wave reform block, an equation can be constucted for the arc which goes between linear legs of the sawtooth generator and represents the behavior of the waveform during transition.

The straight forward method to find an mathematical description which describes the switch between a rising slope and falling slope equation would be to substitute Equation 2.49 into Equation 2.50. Using Equation 2.49 as is requires knowledge of device parameters and makes the equation only useful if these values are known, but,

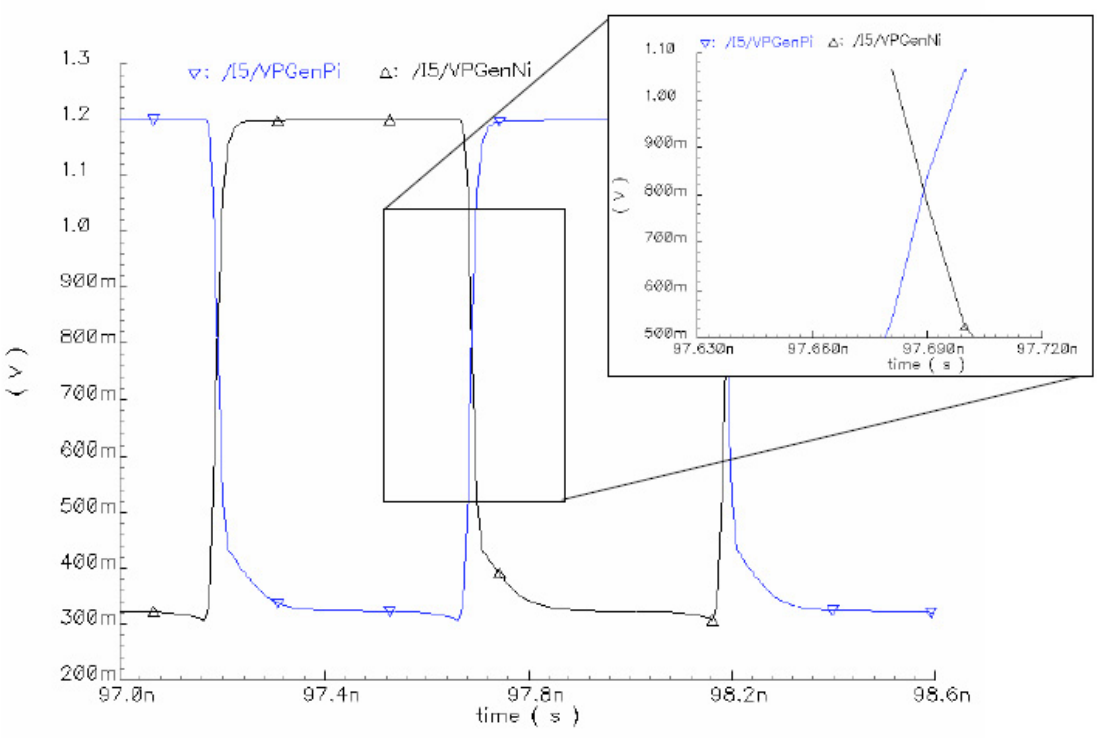


Figure 2.34: Linear characteristics of square wave reform block output (sawtooth generator input).

using the initial and final conditions can be used to simplify the equation and therefore simplify the final mathematical solution. Given the initial and final slopes, the rise and fall times and and the maximum and minimum values of the output of the square wave reform block, Equation 2.49 can be rewritten into Equations 2.51 and 2.52.

$$I_{Switch}(0) = 0 \tag{2.51}$$

$$I_{Switch}(t_m) = I_{Sink} = A_{Long} \times \left(\left(\frac{V_m}{t_m} \right) \times t_m \right)^2 \tag{2.52}$$

In Equation 2.52, I_{Sink} is the final current, A_{Long} is a value which represents the device parameters, V_m is the input voltage swing that takes the current from zero to I_{Sink}, and t_m is the amount of time the sawtooth is in the switching region and can be thought of as the amount of time it takes for the current to go zero to I_{Sink}. Using Equation 2.52 and solving for A_{Long}, gives the value:

$$A_{Long} = \frac{I_{Sink}}{((\frac{V_m}{t_m}) \times t_m)^2} = \frac{I_{Sink}}{(V_m)^2} \tag{2.53}$$

The final current equation for the switch now becomes:

$$I_{Switch}(t) = A_{Long}(\frac{V_m}{t_m}t)^2 = \frac{I_{Sink}}{(V_m)^2} \times ((\frac{V_m}{t_m}) \times t)^2 = \frac{I_{Sink}}{t_m^2}t^2 \tag{2.54}$$

The total current coming off of the capacitor is the difference between the current supplied by the source and current being sunk by he switch, $I_{Source} - I_{Switch}$. Figure 2.35 shows the curves involved in approximating the switching error using the long-channel equation. The topmost curve is the contribution from the current source and has been drawn assuming no current being sunk through the switch. Its slope is $\frac{I_{Source}}{C}$. The bottom-most curve is the contribution from the switch current and is drawn assuming no current being sourced. The graph values are negative because it is sinking current and therefore dropping the voltage on the integrating capacitor. Its slope is $\frac{I_{Sink}}{C \times t_m^2}t^2$. The middle curved line is the sum of the two curves and the actual

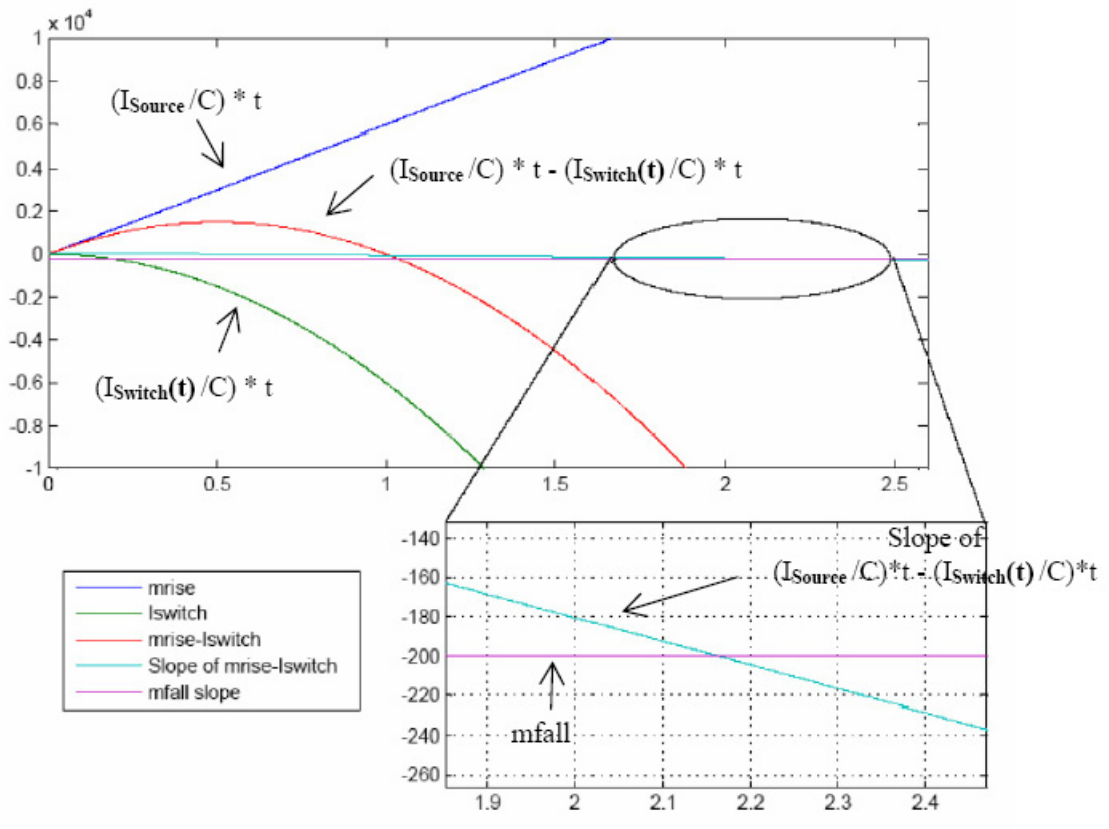


Figure 2.35: Individual curves used in calculating switching error using the first-order long-channel current equation.

amount of current onto the integrating capacitor. The switching current equation has been calculated such that this third curve will, at time t_m after the beginning of the transition, have a slope matching that of slope the sawtooth is transitioning to. In other words, if the sawtooth is transitioning from $m1$ to $m2$, the slope at time t_m of this third graph will be $m2$. The two lines that down the center of Figure 2.35 are the slope of the third curve and the slope that the sawtooth is transitioning to. These two slope lines should be equal at time t_m. The expanded view shows the crossing of these two slope lines.

Substituting the switch current equation into the falling slope equation $m = \frac{(I_{Source} - I_{Switch}(t))}{C}$ and integrating gives an equation for the voltage at time t. Here the equation for switching from $m1$ to $m2$ is shown:

$$V(t) = V_0 + \int_0^t \frac{I_{Source} - I_{Switch}}{C} dt_i = V_0 + \int_0^t \frac{I_{Source}}{C} dt_i - \int_0^t \frac{\frac{I_{Sink}}{t_m^2}}{C} t_i^2 dt_i$$
$$= V_0 + m1 \times t - \frac{I_{Sink}}{3C \times t_m^2} t^3 \quad (2.55)$$

At time t_m the height will be:

$$V(t) = V_0 + m1 \times t_m - \frac{I_{Sink}}{3C \times t_m^2} t_m^3 = V_0 + m1 \times t_m - \frac{I_{Sink}}{3C} t_m \quad (2.56)$$

Figure 2.36 shows an ideal graph (Figure 2.36 a)), a centered version of the first method of approximation (Figure 2.36 c)), and a graph of an approximation using the long-channel current equation for I_{Switch} (Figure 2.36 e)). Figure 2.36 g) shows an overlay of all the graphs and the error present for the approximation using the long-channel current equation. This error is unique to the t_m, $m1$, and $m2$ used and is also dependent on what point on the rising slope the transition starts.

The graphs on the right side of Figure 2.36 are the derivatives of the graphs to their left. Figure 2.36 h) shows an overlay of the slope graphs and shows that all graphs start and end at the same slopes but take different paths to get from one to the other.

The error, V_{error}, in Figure 2.36 g) is the difference between the ideal curve and a curve whose switching has been modeled using the long-channel current equation. For a sawtooth transitioning from $m1$ to $m2$, this error is equal to:

$$V_{error} = V_{LongChannel} - V_{Ideal}(t) = V_0 + m1 \times t_m - \frac{I_{Sink}}{3C} t_m - (V_0 + \frac{m1 + m2}{2} t_m)$$
$$= t_m(m1 - \frac{I_{Sink}}{3C} - \frac{m1 + m2}{2}) \quad (2.57)$$

Note that this is the error seen if the curve is started at $\frac{t_m}{2}$ before the ideal switching point. Equation 2.58 is a more general solution for the switching error with Δt_{LShift} the amount of time the starting point is moved from $t_{IdealSwitchingPoint} - \frac{t_m}{2}$.

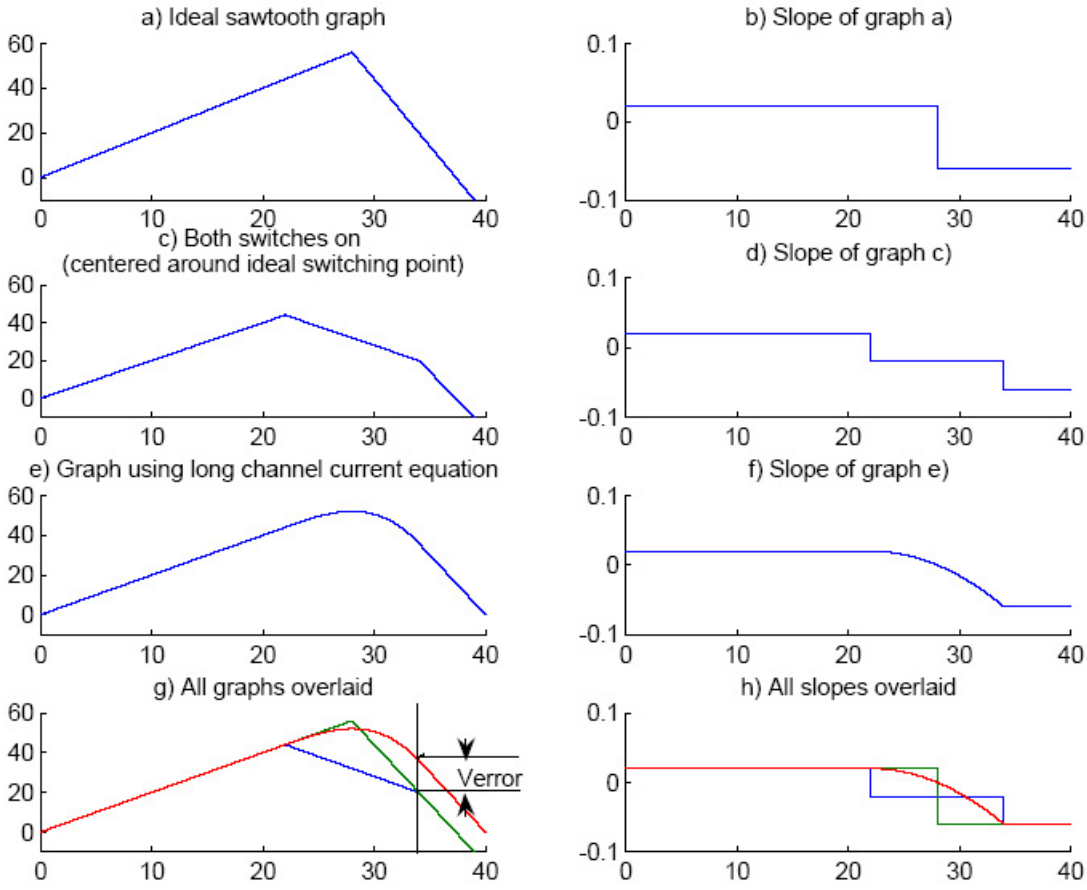


Figure 2.36: Switching error pproximation methods and slopes of graph made using the approximation methods.

$$
\begin{aligned}
V_{error} &= V_0 + \Delta t_{LShift} \times m1 + m1 \times t_m - \frac{I_{Sink}}{3C} t_m \\
&\quad -(V_0 + \Delta t_{LShift} \times m2 + \frac{m1+m2}{2} t_m) \\
&= (m1 - \frac{I_{Sink}}{3C} - \frac{m1+m2}{2}) t_m + \Delta t_{LShift}(m1 - m2) \\
&= (\frac{m1-m2}{2} - \frac{I_{Sink}}{3C}) t_m + \Delta t_{LShift}(m1 - m2) \qquad (2.58)
\end{aligned}
$$

And, for the case that a rising slope $m2$ is transitioning to a falling slope $m1$, the error becomes:

$$
V_{error} = (\frac{m2-m1}{2} - \frac{I_{Sink}}{3C}) t_m + (m2 - m1)\Delta t_{LShift} \qquad (2.59)
$$

Short-channel device have a more linear input voltage *vs* current relationship [*Razavi*, 2001]. In the extreme, the current equation becomes:

$$
I_{DS} = A_{Short}(V_{GS} - V_T) \qquad (2.60)
$$

Going through the same procedure as was done for the long-channel equation, first the variable A_{Short} is solved for:

$$
A_{Short} = \frac{I_{Sink}}{V_m} \qquad (2.61)
$$

Constructing an equation for switch current:

$$
I_{Switch}(t) = A_{Short} \frac{V_m}{t_m} t = \frac{I_{Sink}}{V_m} \times \frac{V_m}{t_m} \times t = \frac{I_{Sink}}{t_m} t \qquad (2.62)
$$

Constructing an equation for voltage along the curve:

$$
\begin{aligned}
V(t) &= V_0 + \int_0^t \frac{I_{Source} - I_{Switch}}{C} dt_i \\
&= V_0 + \int_0^t \frac{I_{Source}}{C} dt_i - \int_0^t \frac{\frac{I_{Sink}}{t_m}}{C} t_i dt_i \\
&= V_0 + m1 \times t - \frac{I_{Sink}}{2C \times t_m} t^2 \qquad (2.63)
\end{aligned}
$$

Finding an equation for the error when $m1$ is the rising slope and $m2$ is the falling slope:

$$\begin{aligned} V_{error} &= V_0 + \Delta t_{LShift} \times m1 + m1 \times t_m - \frac{I_{Sink}}{2C} t_m \\ &\quad -(V_0 + \Delta t_{LShift} \times m2 + \frac{m1+m2}{2} t_m) \\ &= (m1 - \frac{I_{Sink}}{2C} - \frac{m1+m2}{2}) t_m + (m1 - m2)\Delta t_{LShift} \\ &= (\frac{m1-m2}{2} - \frac{I_{Sink}}{2C}) t_m + (m1 - m2)\Delta t_{LShift} \end{aligned} \tag{2.64}$$

Finding an equation for the error when $m2$ is the rising slope and $m1$ is the falling slope:

$$V_{error} = (\frac{m2-m1}{2} - \frac{I_{Sink}}{2C}) t_m + (m2 - m1)\Delta t_{LShift} \tag{2.65}$$

To find the timing error or shift in the sawtooth crossing, V_{error} should be divided by the falling slope. In the case where $m1$ is transitioning to $m2$, the timing error is:

$$t_{error} = \frac{V_{error}}{m2} \tag{2.66}$$

and, the timing error when $m2$ is transitioning to $m1$ is:

$$t_{error} = \frac{V_{error}}{m1} \tag{2.67}$$

Small devices are used in the switches to reduce the load on the square wave reform block, reduce the variable capacitance on the integrator node, and to increase speed of switching and, as such, the short-channel error approximation is more appropriate, but a completely solution probably lies somewhere between the long-channel and short-channel approximations.

Note that the above equation still contains an unknown, t_m. t_m is less than or equal to t_{iRISE} so replacement of t_m by t_{iRISE} would multiply the error by some value. Another value that t_m could be replaced with is two times the threshold voltage of

the CMOS device being used as a switch, but this will also inflate the error calculated. ($2V_T$ is considered a "Rule of thumb" difference between input voltages to a differential pair to have one side or switch effectively "OFF" and the other side effectively "ON").

The switching errors discussed so far have been for the transition from a rising slope and a falling slope. Error can also be introduced on the falling slope to a rising slope transition. Again, a method assuming that both switches are on for the entire transition, a method based on the first-order long-channel $V - I$ equation, and a method based on the first-order short-channel $V - I$ equation will be examined.

The derivation of the error for the first method, the method which assumes that both switches are on for the entire transition, is identical to the case of a rising slope transitioning to a falling slope, though signs are opposite. Going through the same derivation gives the same error as was derived in Equation 2.68:

$$t_{CrossShift} = 2\frac{h_2}{m2} = 2\frac{h_3 - h_1}{m2} = 2(\frac{t_{PN} \times m2}{m2} - t_{PN} \times (\frac{m1 + m2}{2m2}) = t_{PN}(1 - \frac{m1}{m2}) \tag{2.68}$$

Other equations derived for the rising slope to falling slope transition are also identical.

Finding error models for the method based on the first-order long-channel and short-channel $V - I$ equations for the falling slope to rising slope transition error is more complicated. In the case of a transition between a rising slope to a falling slope, the switch devices could be assumed to be in saturation, but, in the falling slope to rising slope transition, the switch devices will be in their triode region throughout the switching or, depending on the amount of jitter and the height of the sawtooth's swing, may start in triode and end in saturation. Because most of the transition should occur in the presence of a small V_{DS}, calculations will be done assuming the device stays in triode throughout the transition. Also, though there were two separate solutions for long-channel and short-channel devices when calculating the rising to falling slope transition error, due to the likeness of the behavior of both types of devices in triode, only one error equation is needed here.

The first order triode region current equation to be used to describe the current through the switch is [*Rabaey*, 1996]:

$$I_{DS} = A_{Triode1}\left[(V_{GS} - V_T)V_{DS} - \frac{V_{DS}^2}{2}\right] \tag{2.69}$$

When switching from a falling slope to a rising slope, the output will start at its minimum and go to its maximum. The output, on the other hand, will only change minimally, held in place by the charge on the integrating capacitors. As such Equation 2.69 can be rewritten with the output dependent term, V_{DS}, a constant, and the input dependent term $V_{GS} - V_T$ written as a function of time:

$$I_{DS}(t) = A_{Triode2}(V_{In}(t) - \frac{V_{DS}}{2}) \tag{2.70}$$

Where V_{In} is equal to $V_G - V_S - V_T$, V_{DS} is equal to $V_{Out} - V_S$, and $A_{Triode2}$ is a coefficient which will make the current at time zero equal to I_{Sink} and the current at time t_m equal to zero. Setting Equation 2.70 equal to I_{Sink} for a time equal to zero, $A_{Triode2}$ can be found:

$$A_{Triode2} = \frac{I_{Sink}}{V_m - \frac{V_{DS}}{2}} \tag{2.71}$$

v_m is the maximum value of $V_{In}(t)$. To find the voltage error on the output, the ideal behavior voltage must be subtracted from the voltage on the output generated by the current through the switch. Equation 2.72 shows the derivation of the voltage equation for a switch in triode and Equation 2.73 shows the final voltage error. Equation 2.73 assumes the slope is changing from a falling slope of $m1$ to a rising slope of $m2$.

$$
\begin{aligned}
V(t) &= V_0 + \Delta t_{LShift} \times m1 + \int_0^t \frac{I_{Source} - I_{Switch}}{C} dt \\
&= V_0 + \Delta t_{LShift} \times m1 + \frac{I_{Source} t}{C} - \frac{I_{Sink}}{v_m - \frac{V_{DS}}{2}} \int_0^t \frac{v_m}{t_m} t - \frac{V_{DS}}{2} dti \\
&= V_0 + \Delta t_{LShift} \times m1 + \frac{I_{Source}}{C} t - \frac{I_{Sink}}{v_m - \frac{V_{DS}}{2}} (\frac{v_m}{t_m} \frac{t^2}{2} - \frac{V_{DS}}{2} t) \quad (2.72)
\end{aligned}
$$

$$
\begin{aligned}
V_{error} &= V(t_m) - V_{Ideal}(t_m) = V_0 + \Delta t_{LShift} \times m1 + m1 \times t_m \\
&\quad - \frac{I_{Sink}}{v_m - \frac{V_{DS}}{2}} (\frac{v_m}{t_m} \frac{t_m^2}{2} - \frac{V_{DS}}{2} t_m) - (V_0 + \Delta t_{LShift} m2 + \frac{m1 + m2}{2} t_m) \\
&= \left[\frac{m2 - m1}{2} - \frac{I_{Sink}}{v_m - \frac{V_{DS}}{2}} (\frac{v_m - V_{DS}}{2}) \right] t_m + (m1 - m2) \Delta t_{LShift} \quad (2.73)
\end{aligned}
$$

The voltage error for a falling slope of $m2$ changing to a rising slope of $m1$ is Equation 2.73 with $m1$ and $m2$ switched. The time error for $m1$ switching to $m2$ is:

$$t_{error} = \frac{V_{error}}{m2} \quad (2.74)$$

and, the timing error when $m2$ is transitioning to $m1$ is:

$$t_{error} = \frac{V_{error}}{m1} \quad (2.75)$$

The affect of leakage and subthreshold conduction on $m1$ and $m2$: The swing of the output of the square wave reform block determines how "hard" the switches in the sawtooth generation block are turned on and off. In the ideal case, all of the current goes from the sources to the capacitors when the switches in Figure 2.24 are off or non-conducting. But, if the difference between V_{iMAX} and V_{iMIN} is not large enough, the switch will leak and change the slopes of the sawtooth generator outputs. Figure 2.37 shows a basic sawtooth generator circuit with leakage. Note that the current sink pulls a fixed amount of current so the amount of current that is leaked through

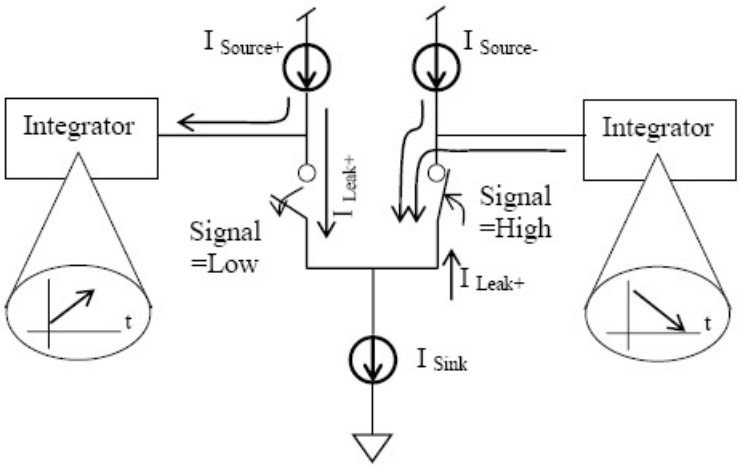


Figure 2.37: Sawtooth generator circuit with leakage.

the switch on the positive output side reduces the current pulled from the negative output side.

Equations 2.76 through 2.79 show how the currents and slopes for a sawtooth generator with leakage.

$$m1 = \frac{(I_{Source+} - I_{Sink} + I_{Leak-})}{C_+} \tag{2.76}$$

$$m2 = \frac{I_{Source+} - I_{Leak+}}{C_+} \tag{2.77}$$

$$m1 = \frac{I_{Source-} - I_{Leak-}}{C_-} \tag{2.78}$$

$$m2 = \frac{(I_{Source-} - I_{Sink} + I_{Leak+})}{C_-} \tag{2.79}$$

Plugging the positive output values into the basic JAC equation gives an error as shown in Equation 2.80.

$$
\begin{aligned}
& t_p \times \frac{(I_{Source+} - I_{Sink} + I_{Leak-})}{C_+} + (T - t_p) \times \frac{I_{Source+} - I_{Leak+}}{C_+} \\
= \; & t_p \times \frac{(I_{Source+} - I_{Sink})}{C_+} + (T - t_p)\frac{I_{Source+}}{C_+} + t_p \times \frac{I_{Leak-}}{C_+} + (T - t_p)\frac{-I_{Leak+}}{C_+} \\
= \; & t_p \times \frac{I_{Leak-}}{C_+} - (T - t_p) \times \frac{I_{Leak+}}{C_+} \qquad (2.80)
\end{aligned}
$$

This error can clearly be made smaller by making the leakage currents small. Or this error can be made to be zero by making the ratios of the currents equal to

$$\frac{t_p}{T - t_p} = \frac{I_{Leak+}}{I_{Leak-}}. \qquad (2.81)$$

Controlling the leakage currents this exactly is not a trivial task and as such the simplest method to correct this error is to provide some sort of correction. Without correction, the amount of jitter allowable on the input may be limited and results may have larger jitter than the ideal case.

The 'closed' switch leakage can be calculated using the CMOS subthreshold conduction equation ([*Van Zeghbroeck*, 2004], [*Meyer et al.*, 2001])

$$I_{DS}(t) = I_0 e^{\frac{V_{IN} - V_S - V_{Th}}{\zeta V_T}} (1 - e^{\frac{-(V_{OUT}(t) - V_S)}{V_T}}) \qquad (2.82)$$

where I_0 is a constant used to describe current in subthreshold region, V_T equals $\frac{kT}{q}$, V_{Th} is the device's threshold voltage, V_{IN} is the input voltage, V_S is the source voltage of the device under consideration, and ζ is a nonideality factor:

$$\zeta = 1 + \frac{1}{2C_{OX}} \sqrt{\frac{q\epsilon_S N_a}{\phi_F}} \qquad (2.83)$$

Equation 2.84 is the result of substituting in values determined during the design stage of the circuit such as sawtooth generator output swing values.

$$I_{DS}(t) = I_0 e^{\frac{V_{iMIN} - V_S}{\zeta V_T}} (1 - e^{-(\frac{\frac{I_{Source}}{C} + V_{sMIN} - V_S}{V_T})}) \qquad (2.84)$$

Because leakage on one side reduces the amount of current being sunk on the other, the amount of charge leaked is the error for both the positive output and the negative output, but in opposite directions. The leaking side will end up with a deficit of charge and the side with the "ON" switch will end up having an equal amount of excess charge. For a 50% duty cycle and no jitter, the leakage will not cause an error. But, because of leakage, sytems where the duty cycle of the signal coming from the square wave reform block is not 50% or systems that have jitter, will have an imbalance in the amount of time that the switches on each side are open or closed. Having one switch open or closed for more time over the course of a single period will mean that the output signal controlled by that switch will diverge by a rate determined by the difference in the "ON" and "OFF" time. In Equation 2.84, the imbalance between "ON" and "OFF" time is taken into consideration through the value of t_{rise}.

At the end of a single period, the error caused by leakage is the difference in the errors caused by leakage on each side.

$$\begin{aligned}
\Delta Q &= \int_0^{T-t_p} I_{DS^+}(t)dt - \int_0^{t_p} I_{DS-}(t)dt \\
&= I_0 e^{\frac{V_{iMIN}-V_S}{\zeta V_T}} \times \\
&\quad \left(\int_0^{T-t_p} 1 - e^{-(\frac{\frac{I_{Source+}}{C^+}+V_{sMIN}-V_S}{V_T})}dt - \int_0^{T-t_p} 1 - e^{-(\frac{\frac{I_{Source-}}{C^-}+V_{sMIN}-V_S}{V_T})}dt \right) \\
&= I_0 e^{\frac{V_{iMIN}-V_S}{\zeta V_T}} \times \\
&\quad \left[T - 2t_p + V_T e^{-\frac{V_{sMIN}-V_S}{V_T}} \left\{ \frac{1 - e^{\frac{I_{Source-}t_p}{C^-}}}{\frac{I_{Source-}}{C^-}} - \frac{1 - e^{\frac{I_{Source-}(T-t_p)}{C^+}}}{\frac{I_{Source+}}{C^+}} \right\} \right] \quad (2.85)
\end{aligned}$$

Substituting $m2$ for $I_{Source+}/C+$ and $m1$ for $I_{Source-}/C-$ and rearranging gives Equation 2.86.

$$\begin{aligned}
\Delta Q &= I_0 e^{\frac{V_{iMIN}-V_S}{\zeta V_T}} \times \\
&\quad \left[T - 2t_p + V_T e^{-\frac{V_{sMIN}-V_S}{V_T}} \left\{ \frac{(1 - e^{\frac{-m1t_p}{V_T}})}{m1} - \frac{(1 - e^{\frac{-m2(T-t_p)}{V_T}})}{m2} \right\} \right] \quad (2.86)
\end{aligned}$$

This means the error on the positive side will be

$$V_{ERROR+} = +\frac{\Delta Q}{C^+} \tag{2.87}$$

and the error on the negative side will be

$$V_{ERROR-} = -\frac{\Delta Q}{C^-}. \tag{2.88}$$

There are a number of ways to minimize this error. A wide swing on the input square wave will reduce V_{iMIN} to a value close to V_S. A 50% duty cycle ($T = 2t_p$) eliminates error ($\Delta Q = 0$), and in the case where the duty cycle is not equal to 50%, using large values of $m1$ and $m2$ can reduce the leakage error.

Non-linear affects: As the output of the sawtooth generator changes, the voltages across some of the devices also change. The CMOS devices used as current sources and switches experience a swing in their V_{DS} values because of the changing output. The most important of these CMOS devices are the ones used in the $I_{Source+}$ and $I_{Source-}$. These are biased in their saturation region and must supply a constant current throughout operation. The long-channel device model is shown in Equation 2.89:

$$I_{DS} = \frac{1}{2}\frac{W}{L}\mu_P C_{OX}(V_{GS} - V_T)^2(1 + \lambda V_{DS}) \tag{2.89}$$

where λ is the channel length modulation coefficient. The final term, $(1 + \lambda V_{DS})$, shows that an change in V_{DS} will change the current through the device. Channel length modulation is related to the ratio of the change in length over the actual length: $\frac{\Delta L}{L} = \lambda V_{DS}$ and λ can be reduced by increasing L. The most important devices to reduce channel length modulation are the current sources and sinks because they are directly responsible for the output slopes $m1$ and $m2$. The length of the devices is a design issue, and therefore will be addressed in the chapter discussing the sawtooth generator block, Chapter 4, but because the non-linearity caused by channel length

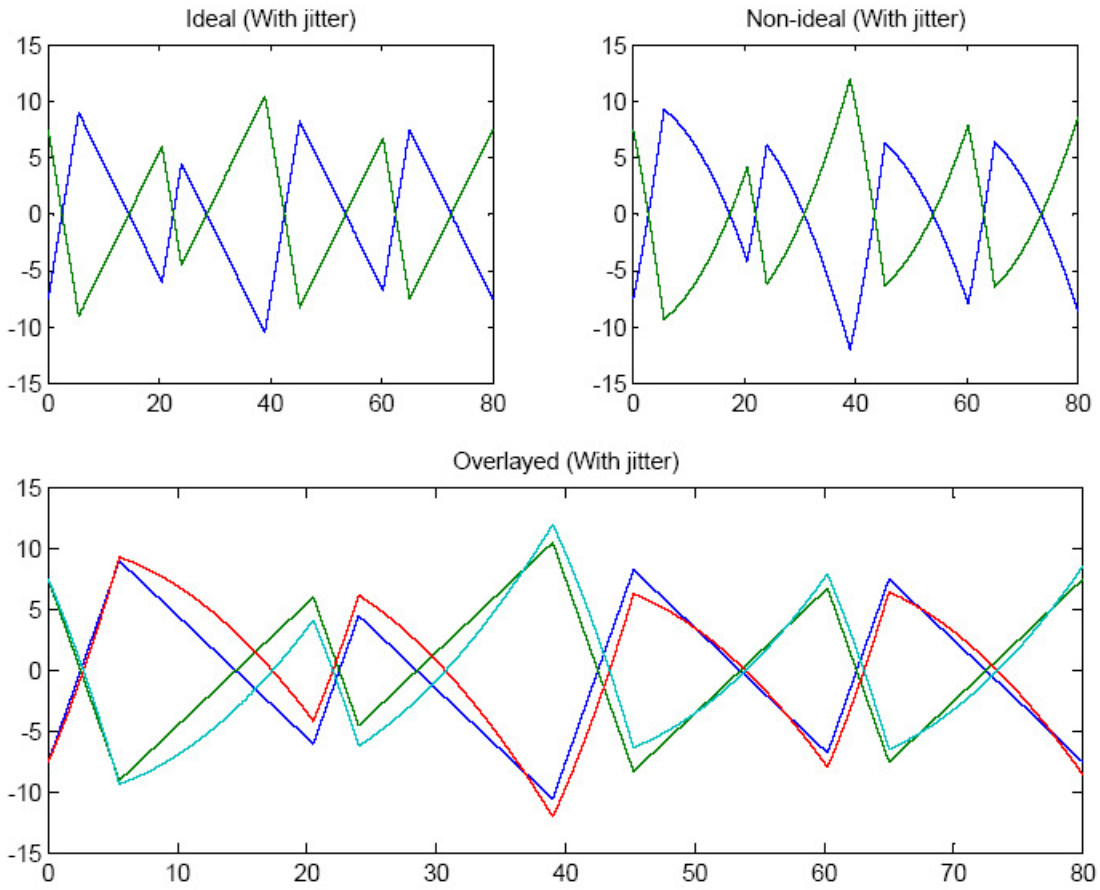


Figure 2.38: Affects of V_{DS} variation across swing range.

modulation contributes to output jitter, it must be included here in the behavioral model.

As the voltage on the integrators rise, the voltage between the supply and the integrator decreases, and this means that the voltage across the devices, V_{DS}, also decreases. Figure 2.38 shows an ideal sawtooth wave and the same sawtooth taking V_{DS} variation into account.

The one saving grace of this type of error is that, to the first order in long-channel devices, it is not random and a general calculation can be made using the long channel equation. Calculations of $V_{OUT}(t)$, the output of the value on the integrator capacitors when channel length modulation is ignored is:

$$
\begin{aligned}
V(t) &= V_0 + \int_0^t \frac{I(t_i)}{C} dt_i \\
&= V_0 + \int_0^t \frac{\frac{1}{2}\frac{W}{L}\mu_P C_{OX}(V_{GS} - V_{Tp})^2}{C} dt \\
&= V_0 + \left[\frac{\frac{1}{2}\frac{W}{L}\mu_P C_{OX}(V_{GS} - V_{Tp})^2}{C}\right] t \\
&= V_0 + At
\end{aligned} \tag{2.90}
$$

where A equals

$A = \frac{\frac{1}{2}\frac{W}{L}\mu_P C_{OX}(V_{GS}-V_{Tp})^2}{C}$.

The output of the integrator depends on its starting value, V_0, and how much current is put onto the integrating capacitor over the time being watched.

When the effects of V_{DS} are taken into account, the change in the slope as V_{OUT} changes must be taken into account. Equation 2.92 gives the value of V_{OUT} at time t along a rising slope. The calculation assumes V_{GS}, V_T, and V_{dd} are constant values.

$$
\begin{aligned}
V_{OUT}(t) &= V_0 + \int_0^t \frac{I(t_i)}{C} dt_i \\
&= V_0 + \int_0^t \frac{\frac{1}{2}\frac{W}{L}\mu_P C_{OX}(V_{GS} - V_{Tp})^2(1 + \lambda V_{DS}(t_i))}{C} dt \\
&= V_0 + \int_0^t \frac{\frac{1}{2}\frac{W}{L}\mu_P C_{OX}(V_{GS} - V_{Tp})^2(1 + \lambda(V_{dd} - V_{OUT}(t_i)))}{C} dt
\end{aligned} \tag{2.91}
$$

(2.92)

Taking the derivative of both sides of Equation 2.93.

$$
\frac{\partial V_{OUT}(t)}{\partial t} = \frac{\frac{1}{2}\frac{W}{L}\mu_P C_{OX}(V_{GS} - V_{Tp})^2 \left[1 + \lambda(V_{dd} - V_{OUT}(t))\right]}{C} \tag{2.93}
$$

Solving the differential equation in Equation 2.93 gives Equation 2.94.

$$V_{OUT}(t) = \frac{1 + \lambda V_{dd}}{\lambda} + Be^{-A\lambda t} = \tag{2.94}$$

where, A equals as defined above and B is a constant created during integration. At time equals zero the voltage equals V_0, and using that to solve for B gives:

$$\begin{aligned} V_{OUT}(0) &= \frac{1 + \lambda V_{dd}}{\lambda} + Be^{-A\lambda 0} = \frac{1 + \lambda V_{dd})}{\lambda} + B = V_0 && (2.95) \\ B &= V_0 - \frac{1 + \lambda V_{dd}}{\lambda} && (2.96) \end{aligned}$$

The error between the ideal case shown in Equation 2.90 and the case where channel length modulation is taken into account (Equation 2.94)is as shown in Equation2.97.

$$V_{Error}(t) = V_{OUTclm} - V_{OUTideal} = (\frac{1 + \lambda V_{dd}}{\lambda} + Be^{-A\lambda t}) - (V_0 + At) \tag{2.97}$$

Equation 2.97 is the voltage error on the output of the sawtooth generator block when the signal starts with no error and is rising. The falling slope is the sum of the current which causes the rising slope plus the sink current, I_{Sink} as shown in Equations 2.18 and 2.21. The V_{DS} of the devices which generate I_{Sink} is shielded from variation by the switches and additional capacitance on nodes important nodes. Redoing the above calculations for the falling slope,

$$\begin{aligned} V_{OUT}(t) &= V_0 + \int_0^t \frac{I(t_i)}{C} dt_i \\ &= V_0 + \int_0^t \frac{\frac{1}{2}\frac{W}{L}\mu_P C_{OX}(V_{GS} - V_{Tp})^2(1 + \lambda V_{DS}(t_i))}{C} - I_{Sink}dt \\ &= V_0 + \\ &\quad \int_0^t \frac{\frac{1}{2}\frac{W}{L}\mu_P C_{OX}(V_{GS} - V_{Tp})^2(1 + \lambda(V_{dd} - V_{OUT}(t_i))) - I_{Sink}}{C} dt \end{aligned} \tag{2.98}$$

Taking the derivative of both sides of Equation 2.98.

$$\frac{\partial V_{OUT}(t)}{\partial t} = \frac{\frac{1}{2}\frac{W}{L}\mu_P C_{OX}(V_{GS} - V_{Tp})^2 \left[1 + \lambda(V_{dd} - V_{OUT}(t))\right] - I_{Sink}}{C} \tag{2.99}$$

Solving the differential equation in Equation 2.99 gives Equation 2.100.

$$V_{OUT}(t) = \frac{A(1 + \lambda V_{dd}) - \frac{I_{Sink}}{C}}{A\lambda} + Be^{-A\lambda t} \tag{2.100}$$

where, again, A equals

$\frac{\frac{1}{2}\frac{W}{L}\mu_P C_{OX}(V_{GS}-V_{Tp})^2}{C}$,

and B is a constant created during integration. Assuming that V_{OUT} is V_{dd} at time equals zero, the constant B can be solved for:

$$\begin{aligned} V_{OUT}(0) &= \frac{A(1 + \lambda V_{dd}) - \frac{I_{Sink}}{C}}{A\lambda} + Be^{-A\lambda 0} \\ &= \frac{A(1 + \lambda V_{dd}) - \frac{I_{Sink}}{C}}{A\lambda} + B = V_{dd} \qquad (2.101) \\ B &= V_{dd} - \frac{A(1 + \lambda V_{dd}) - \frac{I_{Sink}}{C}}{A\lambda} \qquad (2.102) \end{aligned}$$

The error between the ideal case shown in Equation 2.90 and the case where channel length modulation is taken into account (Equation 2.94)is as shown in Equation2.97.

$$V_{Error}(t) = V_{OUTclm} - V_{OUTideal} = \left(\frac{A(1 + \lambda V_{dd}) - \frac{I_{Sink}}{C}}{A\lambda} + Be^{-A\lambda t}\right) - (V_0 + At) \tag{2.103}$$

Equation 2.103 is the voltage error on the output of the sawtooth generator block when the signal starts with no error and is falling. To find the time error, the voltage error must be divided by the slope. For the error on a segment with slope $m2$ will be:

$$t_{error} = \frac{V_{error}}{m2} \tag{2.104}$$

and, the timing error when the segment has a slope of $m1$ is:

$$t_{error} = \frac{V_{error}}{m1} \tag{2.105}$$

The total error that is seen at the output of the sawtooth generator is summarized in Table 2.39. The jitter error is caused by error on the input, Δt_{pi}, threshold voltage error (second half of t_{ps}), process variation (t_{j2}), switching error (t_{j3}), leakage error (t_{j4}), and non-linear error (t_{j5}). Three methods of approximating the switching error are shown. t_{3ja} is based on a linear error, t_{3jb} is based on the first-order CMOS long channel equation, and t_{3jc} is based on the CMOS short-channel current equation. All three of these error calculations are based on a rising slope of $m1$ switching to $m2$. In the case that a risng slope of $m2$ is switching to $m1$, $m1$ and $m2$ must be switched in the equations. t_{3jd} is based on the CMOS long-channel model and describes the error on a falling $m2$ slope switching to a rising $m1$ slope. Again, for the case where a falling slope of $m1$ switches to a rising slope of $m2$, $m1$ and $m2$ must be switched in the equations.

A number of things can be understood from these equations. It is clear that the t_p error is important in that it shows up as jitter at the output of the sawtooth generator block multiplied by a factor determined by the output slopes of the sawtooth generator. The larger $m2$ is the smaller the affect of the t_p error. A larger $m2$ also helps in reducing non-linear and leakage effects. Errors due to channel length modulation and process variation can be reduced by using larger devices. Larger currents, in other words, larger W/L values improve non-linear performance. Here there is a trade off between power and jitter removal performance. In this thesis, jitter removal performance is central and power use is secondary.

Output Generation Block

The function of the Output Generation block is to form a pulse at every crossing of the $m2$ sloped sawtooth generator outputs. Depending on the application where the clock signal is to be used, the pulse generated may have different requirements on its duty cycle, amplitudes, amplitude noise, jitter, and whether the generated output will be a single ended

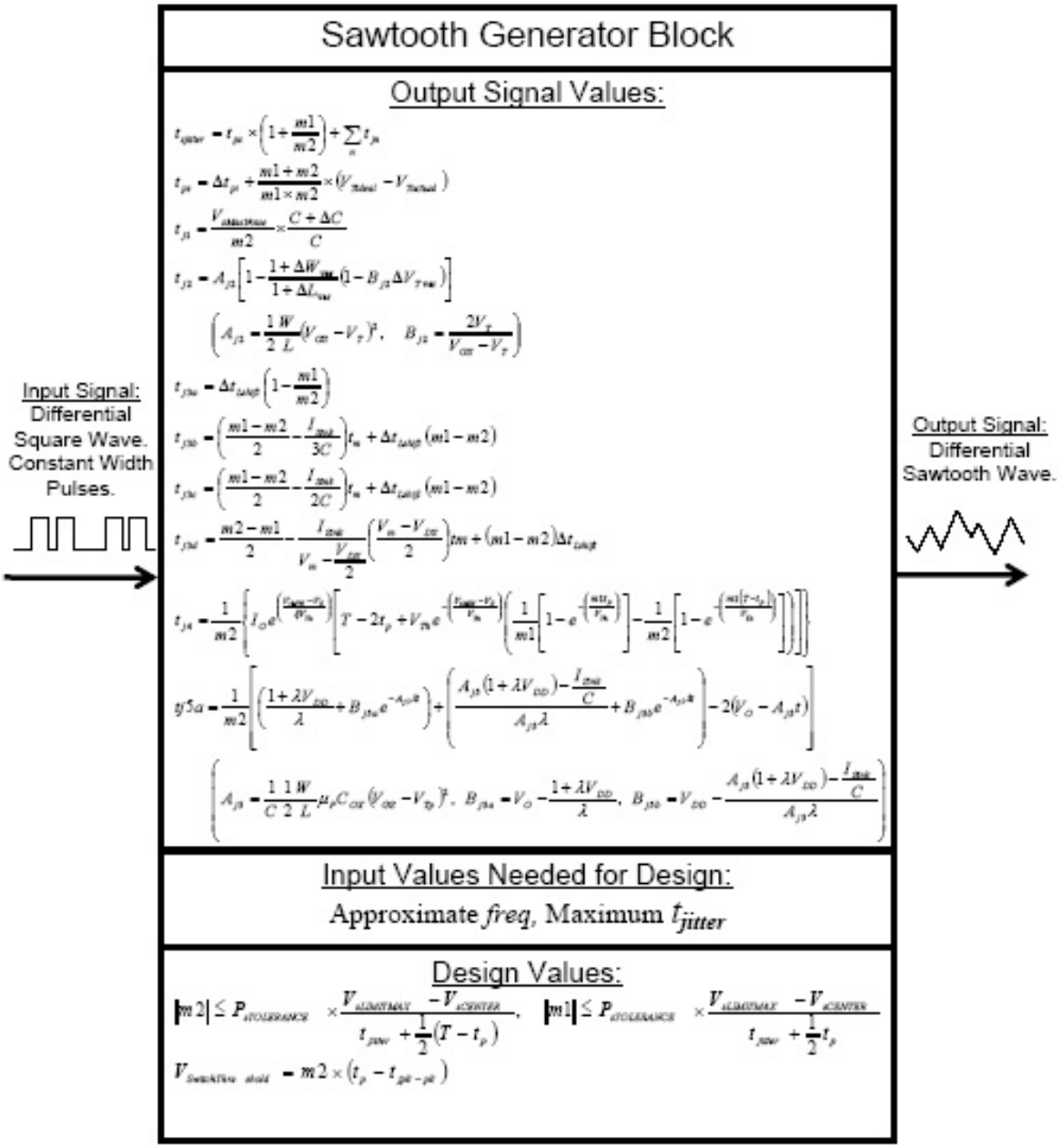


Figure 2.39: Summary of jitter sources through Sawtooth generation block.

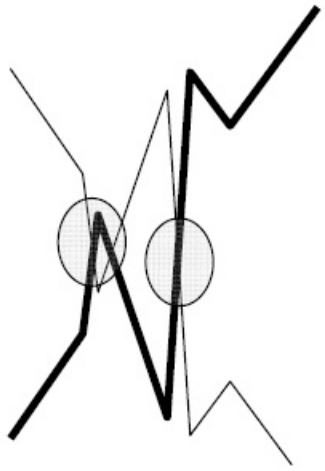


Figure 2.40: Two ideally constant slope signals with glithes from noise. Two crossings sensed though only one should exist.

or differential signal. Though this block will be very different for different systems, it will always have some type of amplifier to sense the crossing of the sawtooth $m2$ sloped signal crossing. The requirements for this sensing block is that it has high-gain, has Schmitt trigger like qualities, and that it does not affect the performance of the sawtooth generator block. High-gain is necessary because the slope of $m2$, in most cases, will not be as steep as, say, the digital transitions in the input monostable block. Having a reduced slope allows for more errors due to noise as well as making the crossing point more difficult to sense. A high-gain comparator in the output block will give a sharper edged output and therefore a clearer clock edge for whatever application is using this circuit. The Schmitt trigger behavior is necessary in noisy systems to protect against double sensing a crossing as shown in Figure 2.40.

The final requirement is that the sensing block does not affect the sawtooth generator itself. The sawtooth generator block's output voltage is controlled through management of charge on to and off of capacitors. Any additional capacitance or possible current paths added by the sensing block would be detrimental to the operation of the sawtooth generator. To this end, a sensor with high input impedance and low input capacitance is required. The input and output requirements for this block are shown in Tables 2.5 and 2.6.

Symbol	Description
$m1$	Ideal non-crossing edge slope (leg of width t_{ip})
$m2$	Ideal crossing edge slope (crossing which generates output pulse)
$\Delta m2$	Variation from ideal crossing edge slope
$t_{sjitter}$	Time error (jitter) on crossing
V_{sNOISE}	Amplitude Noise of output sawtooth waves
V_{sCROSS}	Height (voltage) on crossing
V_{sMAX}	Output maximum allowable value
V_{sMIN}	Ouput minimum allowable value

Table 2.5: Parameters to describe input of Output Generation Block.

Symbol	Description
$t_{oJitter}$	Final jitter value of output square wave
V_{oNOISE}	Amplitude Noise of sensor output
t_{oRise}	Rise time of sensor output
t_{oFall}	Fall time of sensor output
V_{oMAX}	Output maximum value
V_{oMIN}	Ouput minimum value

Table 2.6: Parameters to describe output of Output Generation Block.

Effects of noise on level sensor output: As discussed above, the circuitry to generate the final waveform is dependent on the application, and, as such, is not addressed here. Discussions of noise for this block will only cover the level sensor. As shown in the square wave reform block and sawtooth generation block sections, the error introduced by noise is as shown in Equation 2.106 and Equation 2.107, and illustrated in Figure 2.41.

$$t_{oJitter}\frac{V_{oMAX}-V_{oMIN}}{t_{oRise}}+t_{sjitter}\frac{V_{oMAX}-V_{oMIN}}{t_{oFall}}=2V_{oNoise} \tag{2.106}$$

$$t_{oJitter}=\frac{2V_{oNoise}}{(V_{oMAX}-V_{oMIN})\left(\frac{1}{t_{oRise}}+\frac{1}{t_{oRise}}\right)} \tag{2.107}$$

Other contributions to output block jitter: Other contributions to the final output jitter are dependent on how the rest of the output block is implemented. Because this block is dependent on the system that the circuit is to be used in, it is impossible

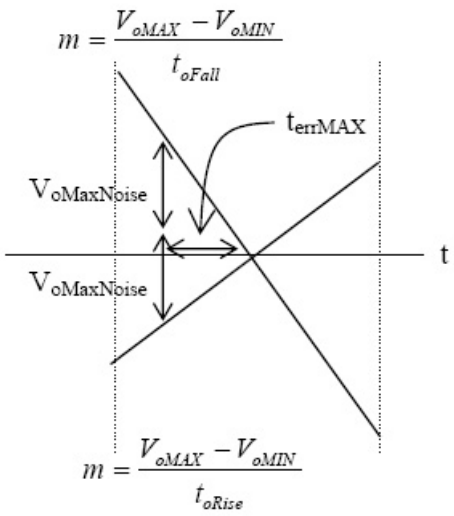


Figure 2.41: Noise error on level sensor block.

to defined an equation for the jitter the block may introduce. Therefore, here, a "'dummy"' jitter function will be introduced as a placeholder.

$$Additional\ t_{oJitter} = f_{oJitter}(OutputBlock) \tag{2.108}$$

If a block similar to the input monostable block is used as the output block this function would be replaced with values such as those in Table 2.18.

The summary of the output generation block's jitter information is shown in Table 2.42.

Feedback

The non-idealities such as leakage, process variation, short-channel effects, and non-linear inputs can cause the output of the sawtooth generator to diverge and even cause the circuit to fail. To keep this from happening, feedforward and/or feedback circuitry can be added. The feedforward and feedback blocks' job is to keep the average of the sawtooth generator's outputs constant and on top of each other.

Both feedback and feedforward are discussed in Chapter 5, but the design emphasized in this work is a purely feedforwad design. Feedback influences the performance of each

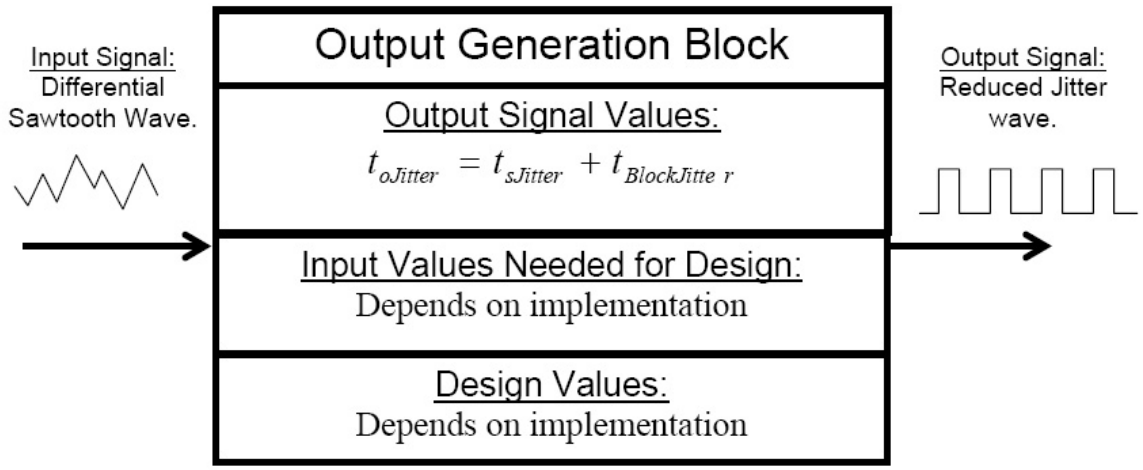


Figure 2.42: Output block jitter summary.

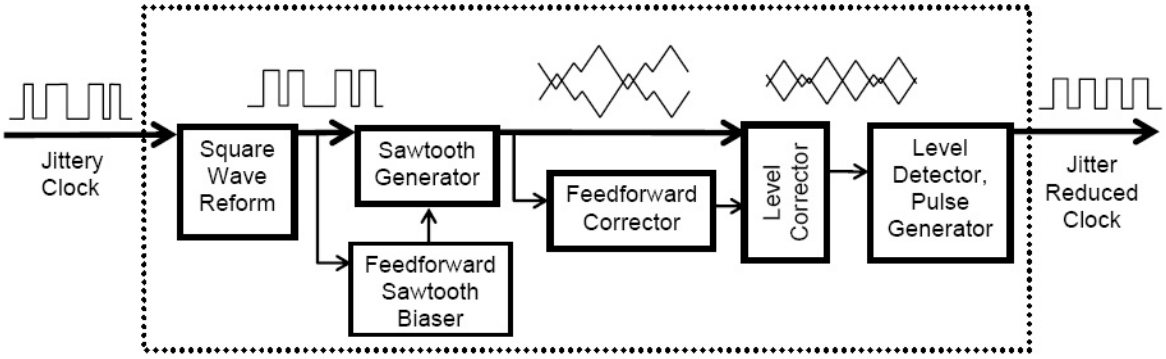


Figure 2.43: Fully feedforward design.

block in its loop and, because of this, analysis of jitter feedback may cause is complicated. Feedforward, on the other hand, has no influence on the other blocks in the circuit except as it may be generating a signal that another block will use. The feedback block can be thought of as an independent block and can be analyzed just for the noise or jitter it adds to the signal it recieves. The feedforward design is shown in Figure 2.43.

The feedforward sawtooth biaser uses information from the output of the squarewave reform block to bias the sawtooth generator. The feedforward sawtooth biaser block uses a low-pass filter which affects the start-up and acquisition time of the circuit. Until the low-

pass filter used in the feedforward sawtooth biaser settles, the sawtooth generator will have meaningless outputs. Once it settles, if the filter is set to a low enough cutoff frequency, this block will not contribute to jitter directly. That is not to say it has no effect on jitter though. If the biasing values are far enough off of the ideal values, one or both of the sawtooths could be forced to swing to values which put some of the sawtooth generator devices into their triode region. Once a device has been forced into triode region, its $V - I$ characteristics change and, as such, the crossing point of the sawtooths may be changed. But, if the variation of the output of the feedforward sawtooth generator block can be considered at design time such that devices are forced to stay in saturation through the circuit's operation, crossing points will only be effected by the channel-length modulation affects and velocity saturation affects. To minimize these affects, large devices can be used as the current sources and sinks. Since the biasing values are DC values when the circuit is settled, large devices cause no harm in this block.

When discussing the feedforward biasing block, it is useful to also mention the feedforward corrector block. The corrector determines how much each sawtooth is off the ideal average and calculates how much each needs to be shifted to be on top of each other. Then that value is sent to shifters that reposition the two sawtooths around the idea average (and, therefore on top of each other). One interesting thing to note here is that, if, say, one sawtooth is too high, it will effectively reduce the current through its I_{Source} through channel-length modulation and have it swing reduced. When it comes to shifting, it will need to be shifted more than the other sawtooth and therefore the shifter will have a larger V_{DS} and produce a larger current and, though unintentional at design time, will increase the sawtooth's swing.

Because the contributions by these two blocks is expressed in the results as channel-length modulation affects on the sawtooth generator, no new jitter will be considered as being generated by these blocks.

Feedback is a much more complicated animal. Feedback does its correction by adjusting $m1$, $m2$, T, or t_p change such that the basic JAC equation, $m1 \times t_p + m2 \times (T - t_p) = 0$, stays true. Making any one of $m1$, $m2$, T, or t_p adjustable will mean the equation can,

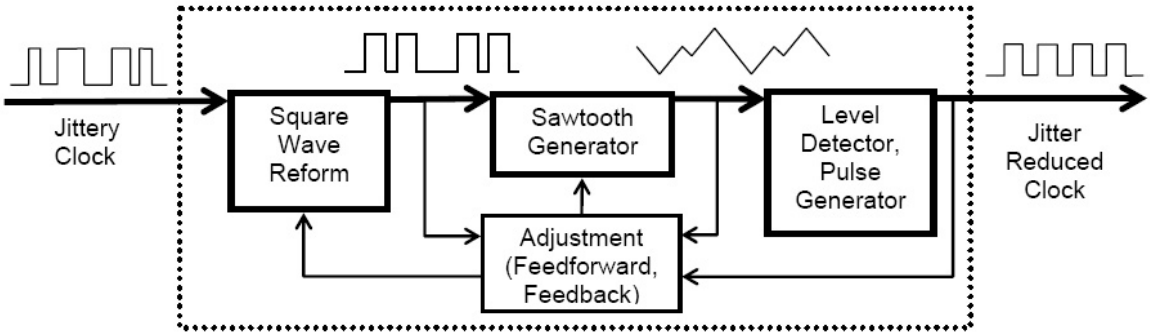


Figure 2.44: Possible feedforward and feedback path options.

mathematically, be made true, but adjustment in the actual implementation using just one variable may not be possible. Path options for possible adjustments to $m1$, $m2$, T, or t_p are shown in Figure 2.44. T is determined by the incoming signal and can not be adjusted, but adjustment to t_p can be done through a feedback path to the input monostable block. $m1$ and $m2$ can be adjusted by the feedback path to the sawtooth generator block. The feedback path from the output of the sawtooth generator is useful in recovering the average value of the sawtooth waveform and deducing if the waveform is diverging or not. The feedback path from the output of the entire JAC is useful because it will have less jitter than the input signal and therefore is easier to use to retrieve a value for T. But the challenge there is that the output signal is not useful until the system is already successfully working.

If feedback was to be used, the decision on which path or paths to implement could be based on a process of elimination. The first option for a feedback path is from the output of the entire JAC system. Though the feedback path from the output of the entire JAC may be useful for fine tuning in future work, the length of the path and the fact that the system must work before it is useful, removes it from the candidates. The path from the output of the squarewave reform block does not give information on whether the sawtooths are diverging and is not useful on its own, but may be useful if used in conjunction with other "feedback" information. The path from the output of the sawtooth generator gives exactly the information necessary to test for divergence and is the shortest possible feedback path. Of these three paths, clearly the last gives the most useful and timely information. One

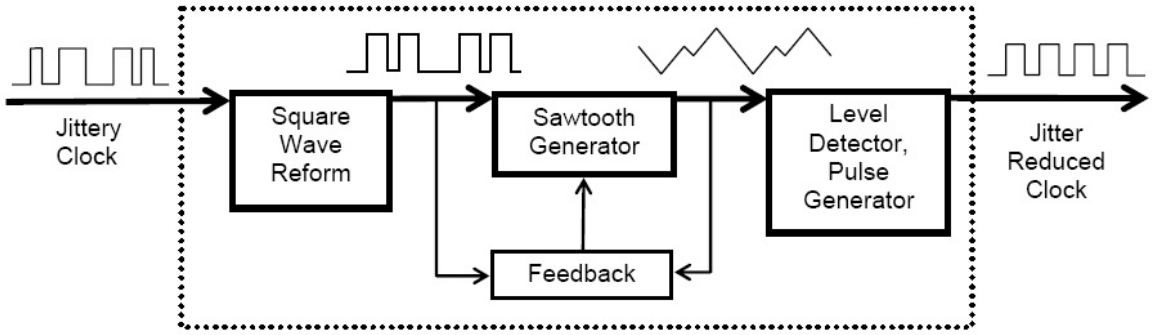


Figure 2.45: Final feedforward and feedback paths.

note here: Using the feedforward biasing block can give a quick approximation of what the biasing values should be. Using the feedforward biasing block for preliminary biasing and feedback for detailed adjustment is another option and the method used while exploring feedback biasing for the JAC for this thesis (though eventually the purely feedforward design was selected).

The next question would be whether to adjust t_p, $m1$ and $m2$, or both. Adjustment to both creates loops and a complex stability challenge. Errors on the width of t_p are multiplied and show up at the output of the JAC as jitter. The width of t_p is also related to the bias values given to the sawtooth generator. Adjusting t_p not only requires a longer and more complicated feedback path, but it also affects more parts of the circuit including biasing (if the first feedforward block is used) and switching. Using the output of the sawtooth generator to adjust the average of each side of the sawtooth waveform to match a reference value requires a short feedback path that affects a localized part of the circuit. It is also much easier to compare two voltages as is done when taking feedback from the output of the sawtooth generator, than to compare times or edges as required if the feedback is based on the width of t_p. The final feedforward and feedback paths are shown in Figure 2.45.

The feedback block used to explore possible adjustment methods in this work used a very low frequency low-pass filter to take averages of the sawtooths. Just as with the feedforward biasing block, the filter slows response time and transitions between frequencies. It also means that jitter and noise on the input signal, as long as it is random and not long-term

sinusoidal, will not show up in the biasing values. This is important because the JAC is sensitive to any changes in biasing. Changes in biasing move the sawtooths up and down and any movement of the sawtooths results in jitter.

Analysis of the jitter introduced to the system by a feedback system is not included here. There are any number of ways to adjust the biasing of the integrator with feedback and each has its own effect. If the feedforward biasing block is used, the feedback can adjust biasing values inside the feedforward biasing block which, in effect, adjusts the ratio of the I_{Source} currents, or it could adjust the biasing values after the block, which effectively is just and adjustment of each branch of the sawtooth generator independently. If the feedforward block is not used, then the feedback goes directly to the I_{Source} generating devices. All of these have different costs to the output jitter. The last option, direct biasing to the I_{Source} generating devices, is probably the best option, though the other two options were what were explored for this thesis.

When all options were examined, the purely feedforward was selected. Simplicity of design in terms of number of devices and actual implementation was a factor, but the importance of a stable as possible sawtooth generator output was the deciding factor.

Chapter 3

Input Monostable Block

3.1 Introduction

The first stage of the jitter attenuation circuit is a one-shot multivibrator or monostable. It's job is to produce a constant width pulse each time it sees specific edge events on the square-wave signal input to the circuit. Edge events seen by this block could be only rising edges, only falling edges or both. Introduction of jitter in this block does not matter since it will be removed by the next stage, and the width of the generated pulse does not matter as long as it is constant between clock cycles and is less than a value determined by the amount of input jitter and the frequency of the signal being cleaned. Though unaffected by jitter added at this point and flexible in the width of the generated pulse, this block can have a great affect on the final performance of the complete JAC circuit if the widths of the pulses between cycles vary. Any variation in the width of the pulse generated by this block is multiplied through the jitter removal block and shows up as jitter on the output of the complete circuit. As such, the central challenge in designing this block is to make it as robust against supply and intrinsic noise which contributes to pulse width variation. This chapter gives a high-level description of the block then describes the effect on the monostable of supply noise and intrinsic noise and the steps taken to reduce their affect. The final section describes the complete proposed monostable circuit.

3.2 Block Level Design

The one-shot multivibrator block, or monostable block as it will be called here, generates a constant width square pulse. The main component of this block, clearly is a monostable circuit. Older monostable circuits were often based on RC time constants [*Schultheiss et al.*, 1951][*Lo*, 1952][*Microelectronics*, 2003][*Lappalainen*, 1974][*Ashok et al.*, 1975][*Bigongiari et al.*, 1995], but at the clock speeds being targeted here, these types of designs can not generate a constant width pulse with the accuracy needed. To achieve the small width pulse which is needed when working with a high-speed clock, a short RC time constant would be needed. Using a large value of C and a small value of (a large-area) R would reduce the sensitivity of C to noise on its current supplies, but the area cost could not be ignored. Using a small values for C make C more susceptible to noise in the system. Improvements to these RC based monostables incorporated operation amplifiers [*Bigongiari et al.*, 1995][*Chung et al.*, 2002][*Tuwanut et al.*, 2005][*Lo and Chien*, 2006] to get sharp transitions and reduce slope-noise based errors. The sharpened edges improved pulse width accuracy somewhat, but the delay through the amplifiers in the signal path defined a minimum pulse width greater than what is required for the speeds needed here. For example, a pulse of 400ps may be required for cleaning up a clock of 2.5GHz, whereas the documented amplifier-based design examined provided pulse widthes in the microseconds and required input pulses greater than $2ns$.

There were three designs the author created specifically as possible monostables for this project. The first was a pulse-mode asynchronous logic finite state machine (FSM) which went through a number of states, each state adding in a delay, such that the output was discretely adjustable to N-states, *i.e.* N-delays. This was attractive because of its reuse of hardware as outputs are fed back through the FSM to achieve the next state. As well as using less area, this design would have given discrete delay options. The problems with this design had to do with the low supply voltage. To implement the logic required to produce outputs and next-states for the FSM, transistors needed to be stacked. For slower clock speeds, more states were need, and as more states were needed, the number of stacked

transistors that would be needed to achieve the desired delay became prohibitive. In an effort to make this JAC design applicable to clocks over a wide range, this design was not used. Please note that this design did show promise in low power monostable design and deserves more investigation.

The second design which was specifically designed and investigated for the monostable circuit was basically a 1-bit memory cell that is set by a pulse at an input square wave event, and reset on another pulse which is generated according to signal from a timer. The timer is implemented with a current steering circuit and capacitors, just as the next JAC block, the sawtooth generator block, is. The current is steered on or off of a capacitor according to the value in the memory cell. A differential design minimizes the affects of supply noise. This design also showed promise, but the complexity needed to make the design work accurately defeats the main goal of a simple, low power design. An example of a design issue which made this design less attractive was the slopes on the timing part of the block. The timer consisted of a charging and discharging capacitor. When an edge was seen, the 1-bit memory cell would be set. Once set, a current source would steer current on to (or a current sink steered current off of) the timing capacitor. When the value on the capacitor reached a certain value, the 1-bit memory cell would be reset. If a long pulse was needed, the slope of the voltage on the capacitor would have to be reduced. Reducing the slope of the voltage makes the design more sensitive to noise. Also, the capacitor, in the process variation information used as reference, showed best case variations of $\pm 10\%$ for MOS capacitors. If there is a large process variation, extra circuitry would need to be included to adjust currents so that the slope would be the value desired. To increase the integrity of the value on the capacitor, the capacitor should be large and the current on to or off of the capacitor should also be large. This introduces power and area issues. For all of these reasons and a few more, this design was not used.

The third design created was a mixture of a traditional digital design and second idea above. The core of the monostable used is the well-known monostable shown in Figure 3.1. This monostable works well as long as the input square wave's duty cycle is longer than the output pulse width. When the input square wave's duty cycle is shorter than the desired

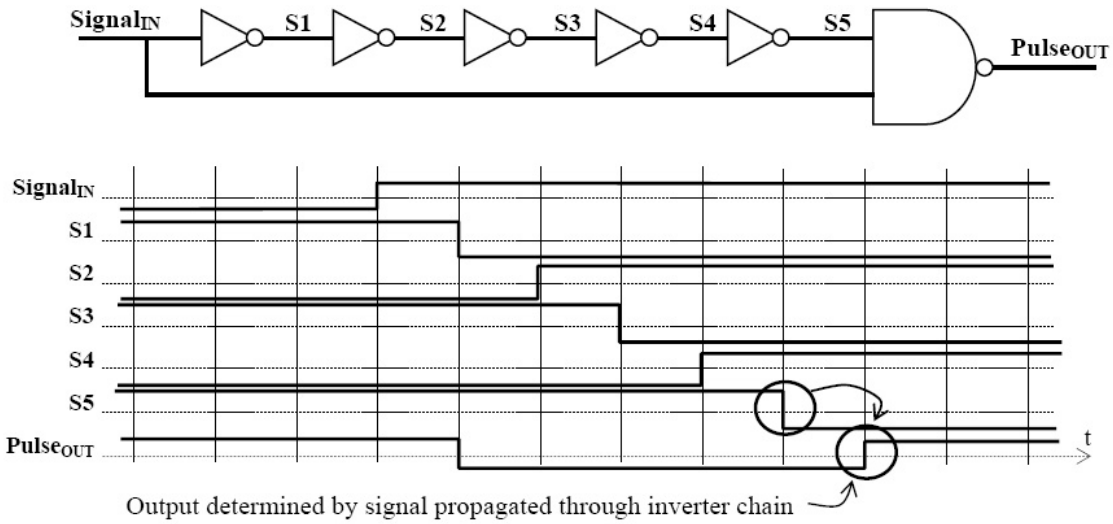


Figure 3.1: Pulse generation using an inverter chain delay.

output pulse, the width of the output pulse will be the width of the input pulse, not of the desired output pulse. This is shown in Figure 3.2.

In order to remove the influence of the duty cycle of the input signal on the output, a flip-flop can be added to the input of the block. This block can be set when the SET edge event is observed on the input and $RESET$ when the output transitions at the end of the output pulse. This is shown in Figure 3.3.

While the circuit in Figure 3.3 solves the problem of a input signal duty cycle that is too short, there remains a problem when the next input signal arrives before the flip-flop $RESET$ value has had time to propagate through the inverter delay chain. Figure 3.4 shows an example of this case.

A comment on the flip-flop needs to be made before discussing a solution to the case shown in Figure 3.4. To make the flip-flop completely safe from SET and $RESET$ conflicts caused by attempts to reset while input is still high, and, also, to make it independent of the duty cycle of the input, a short pulse is used to SET and $RESET$ the flip-flop instead of the actual input signal. This pulse is generated by a pulse generating block designed as shown in Figure 3.1. The input generates a narrow pulse to SET the flip-flop, and the

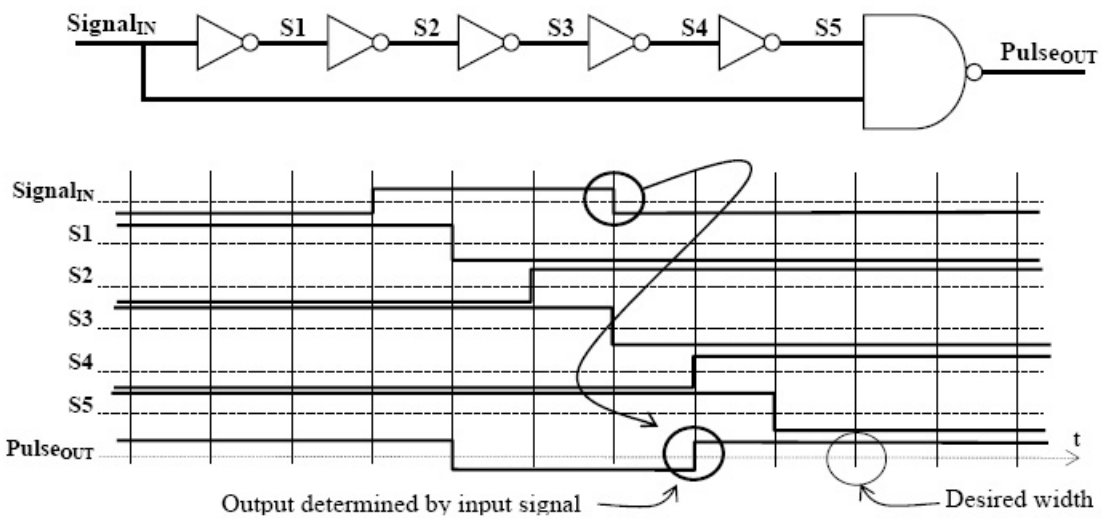


Figure 3.2: Incorrect pulse generation due to insufficient width of incoming square wave pulses.

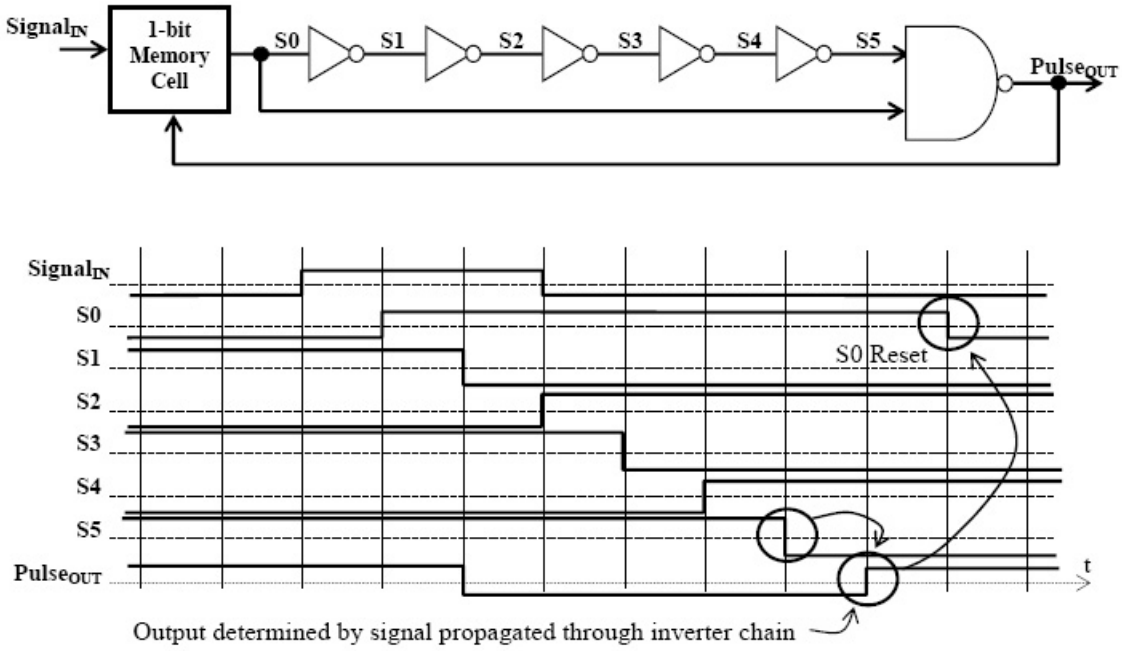


Figure 3.3: Monostable block diagram with protection against input duty cycle length.

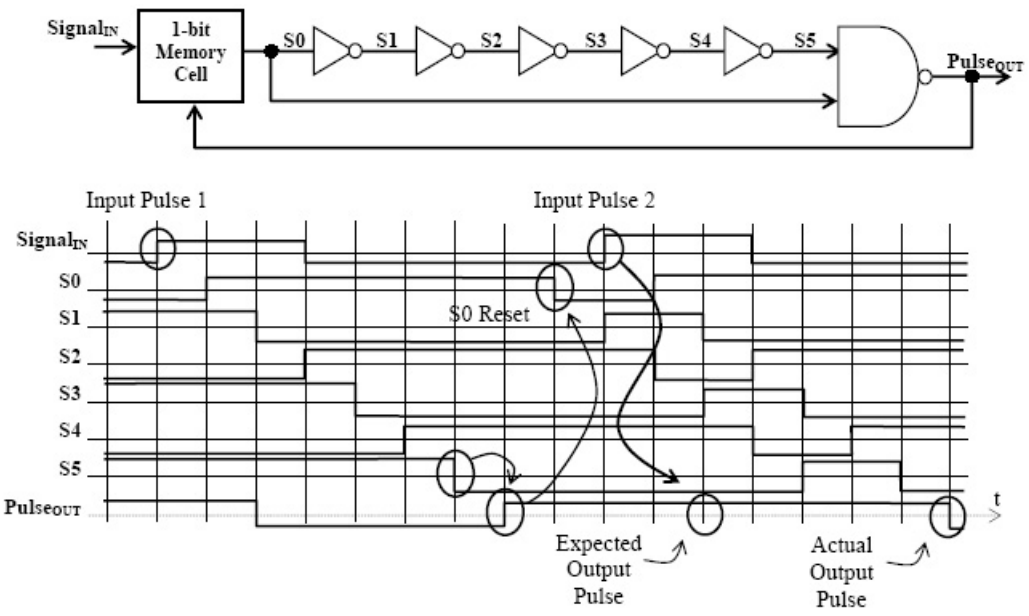


Figure 3.4: Monostable block diagram with protection against input duty cycle length, but problems with signal propagation time through the inverter delay chain.

output generates a pulse to *RESET* the flip-flop as shown in Figure 3.5 and timing as shown in Figure 3.6.

Using the pulse controlled flip-flops the problem described in Figure 3.4 can now be solved. By replacing the inverters with pulse generation block / flip-flop pairs, intermediate values can be reset when they are no longer needed instead of waiting for the signal to

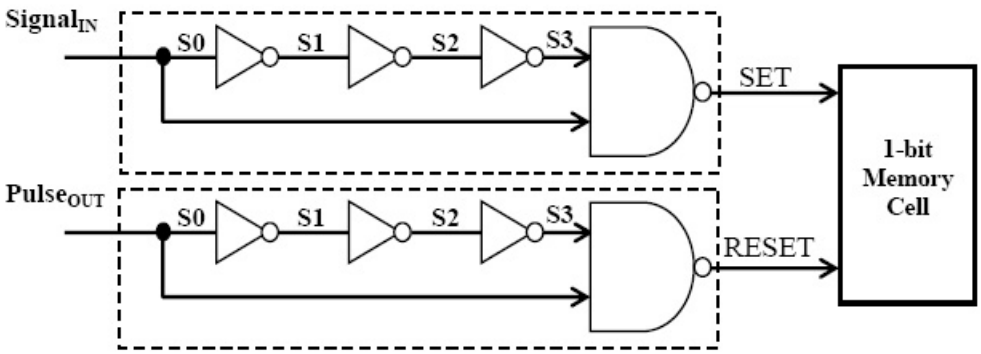


Figure 3.5: Flip-flop *SET* and *RESET* circuitry.

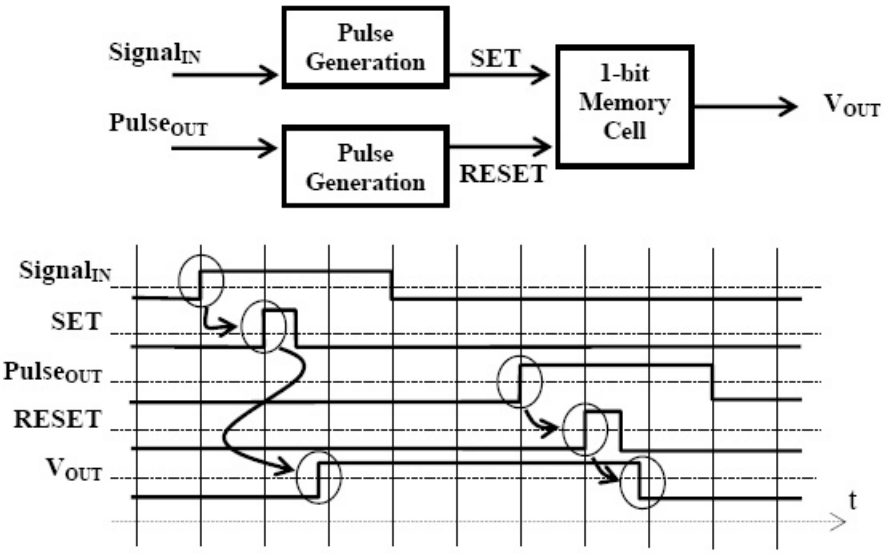


Figure 3.6: Flip-flop *SET* and *RESET* timing.

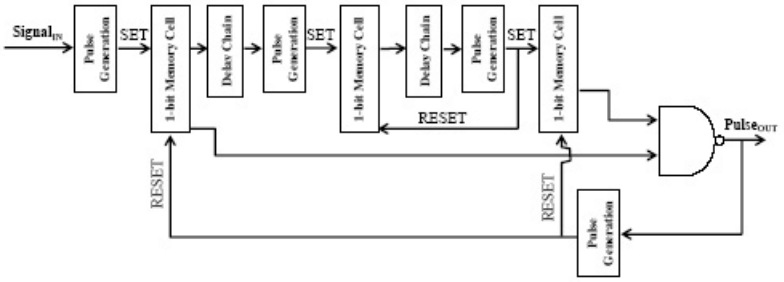


Figure 3.7: Delay chain implemented with flip-flops and pulse generation blocks.

propagate through the entire chain. Figure 3.7 shows a short chain. The pulse generation blocks introduce delay, and the flip-flops allow reseting of intermediate values.

The main goal of this pipelined delay chain is to reduce the amount of time needed between input pulses. This is done by reseting intermediate delay chains as they are not needed and ending up, at the time that generation of the pulse is done, with only having to reset the first segment's delay chain. This delay can be reduced further by removing the delay chain from the first segment and adding it on as an additional delay chain / pulse generation block / flip-flop segment later in the pipeline. This effectively means that, now, the delay required between pulses is merely whatever the pulse generation block's delay

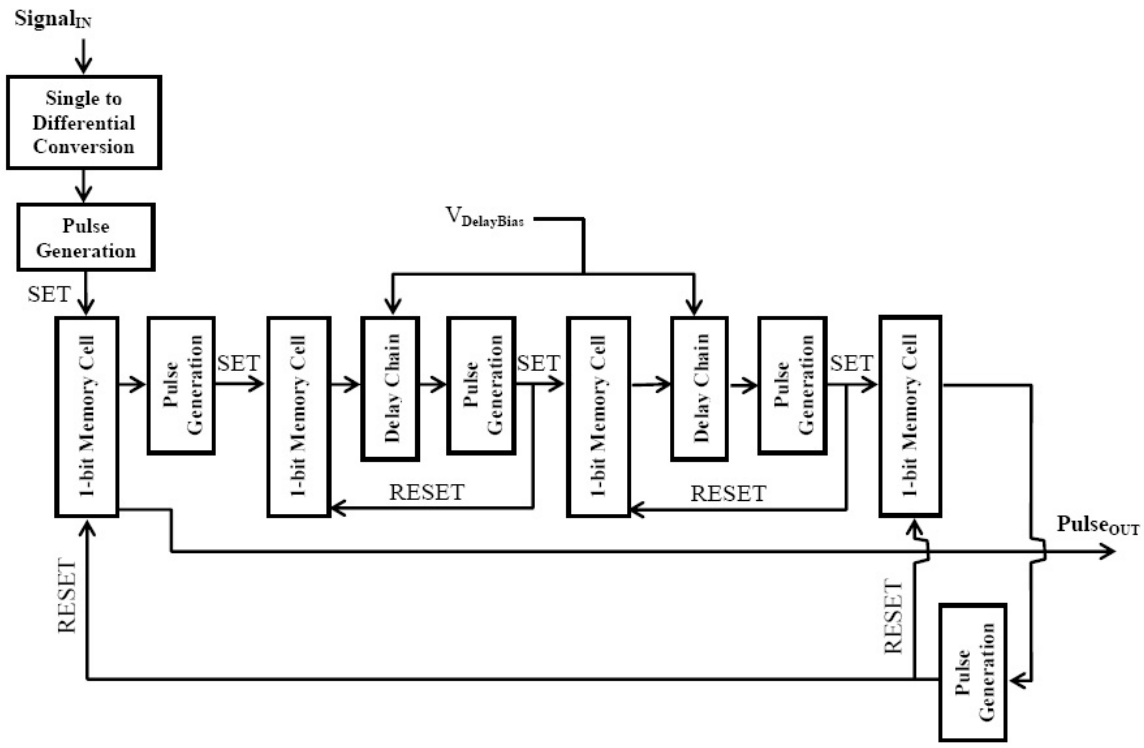


Figure 3.8: Input monostable requiring minimal time between input pulses.

chain length is. The reduction of the required time between pulses means that higher jitter input signals can be processed, and, at design time, less attention needs to be paid to accommodating said delay requirement, making the system easier to design as well as more flexible. A input monostable block set up to require minimal time between input pulses is shown in Figure 3.8.

The delay through the chain is important in that it determines the duty cycle of the output pulse, or, as it has been referred to in previous chapters, t_p. It was shown that the longer t_p is, the steeper the crossing slope of the sawtooth generator will be and the less susceptible that output will be to noise. So, in theory, a longer t_p would be better for the JAC. The trade off is that, in order to make t_p longer, the delay chain must be made longer or the slope of the signal through the delay chain must be reduced, both of which increase the chance for noise to influence the width of t_p. As noted, having having as close to zero variation in t_p is more important for the success of this system than the absolute width of t_p. From this fact the decision to "hard wire" the biasing values to the delay chains was

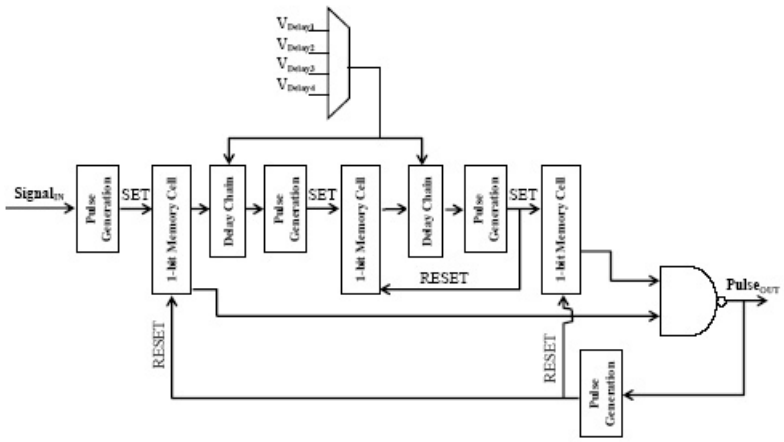


Figure 3.9: Pulse generation block's delay biasing using a multiplexer.

made. But, if the system is required to work across a band of frequencies, the ability to adjust the delay may be important. A monostable that generates an output with a pulse width of $300ps$ is appropriate for a $2GHz$ input signal because it is providing a 60% duty cycle and reasonable protection against noise. But if that same system's input changed to a $500MHz$ signal, the system's $300ps$ pulse width would ony be a 15% duty cycle and the sawtooth generator would not be as robust in the presence of supply and intrinsic noise. In the monostable presented here, the delay is adjustable by current starving the delay chains, but not adjusted automatically with frequency. Automatic adjustment of the delay value will be left as a future topic. For this design, the delay biasing value is either set externally or is one of some number of values selected through a multiplexer to adjust the delay through the delay chain. There are two advantages to having more devices in the delay chain; The more devices in the delay chain, the larger the range of delays that can be produced, and the less effect there will be on the output slopes of each device and noise sensitivity will be more constant across the range of operation. The trade off is, of course, the noise per device is multiplied by the number of devices in the chain. If a frequency is too low, it may be multiplied by using the XNOR option on the input, and then dividing again at the JAC output. In Figure 3.9, the delay biasing is set through a multiplexor.

The final additions to the input monostable block involve the input signal. Because the JAC is a differential circuit, if the input is a single ended clock, a single to differential

converter becomes necessary. Because introduction of jitter in this block does not matter, a simple, easy to build circuit will suffice. Once a differential signal has been generated, the next step is to decide the type of input signal event that is to be watched. If a pulse generation block such as the one shown in Figure 3.1 is used, a rising edge of the input will be watched. By switching the inputs to the NAND gate (effectively inverting them), the falling edge of the input signal will become the pulse generating event. If both edges are to be used, the NAND gate can be replaced with an XNOR gate. An XNOR gate will generate a pulse on both rising and falling edges effectively doubling the frequency of signals given to the JAC. A combination of NAND gates, XNOR gates, flip-flops, or daisy-chained JACs, can provide 2^N multiplication and any division of the input signal. Multiplication can also allow processing of clock signals too slow for the design of the integrator or monostable. To retrieve the original clock, a toggle flip-flop can be used as the output generation block. For the most flexibility, the user can be given the option of selecting between the NAND and XNOR options by use of a multiplexer. The multiplexer option is shown in Figure 3.10.

The monostable shown in Figure 3.10 protects against too-short input signal pulses and too-long propagation delay through the delay chain as well as providing adjustability to the output pulse width. But, if the input waveform is well formed and an output t_p of less than a 50% duty cycle is acceptable to the system being built, then there is no reason why the monostable shown in Figure 3.1 can not be used. Here, the performance of the design in Figure 3.11 will be evaluated. Conveniently, the final $NAND$ gate can be removed and the output of the first flip-flop can be used as the output of the entire monostable block.

3.3 Circuit Design

As noted in Chapter 2, noise, supply or intrinsic, can greatly affect the performance of the input monostable. In this block, the most important value to protect is the output pulse width, t_p, and the path which has the greatest chance of influencing the consistency of t_p is the delay path in Figure 3.11. The delay path includes inverters (delay chain), pulse generation blocks (inverters and $NAND$ gates), and flip-flops, and its function is to time

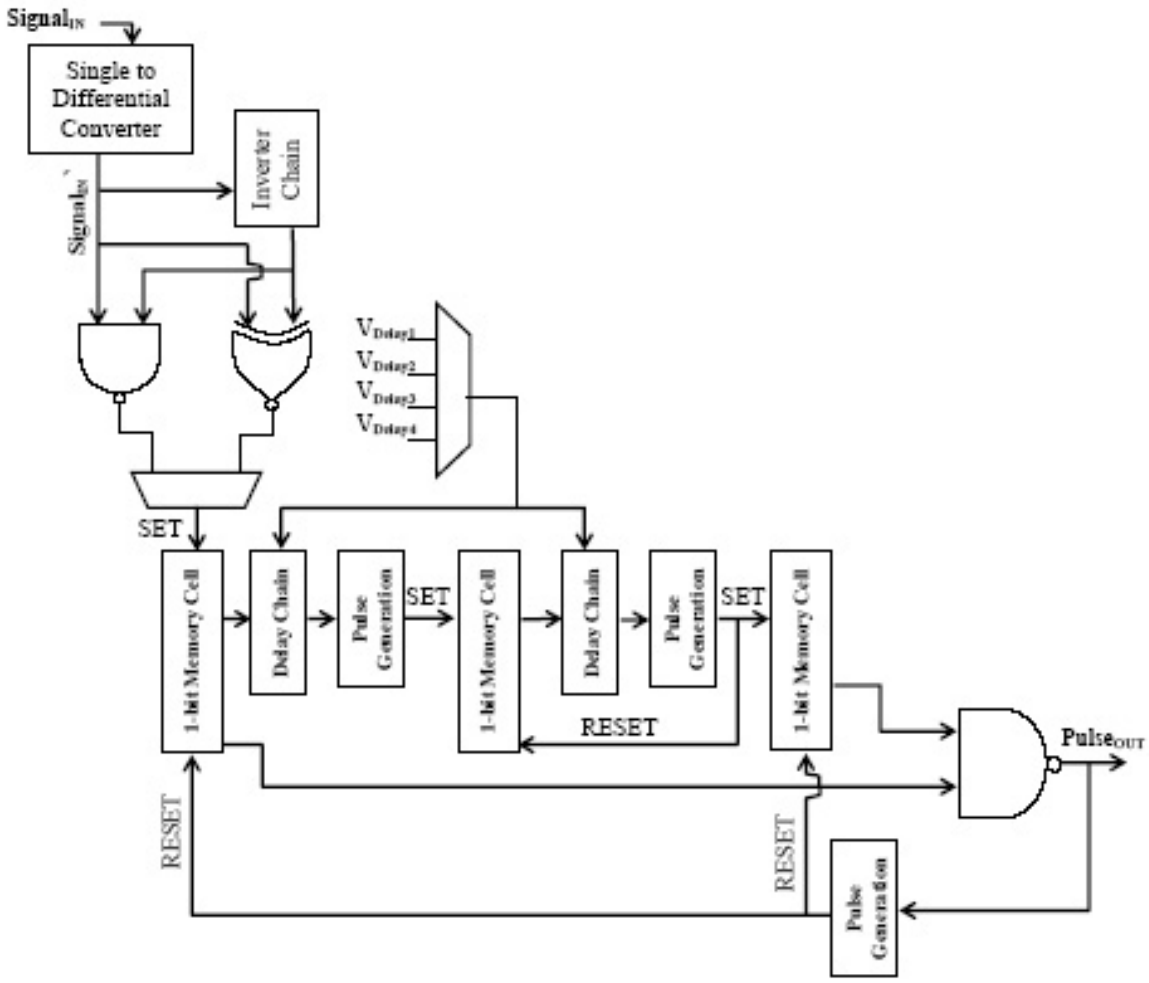


Figure 3.10: Complete input monostable block diagram with all options.

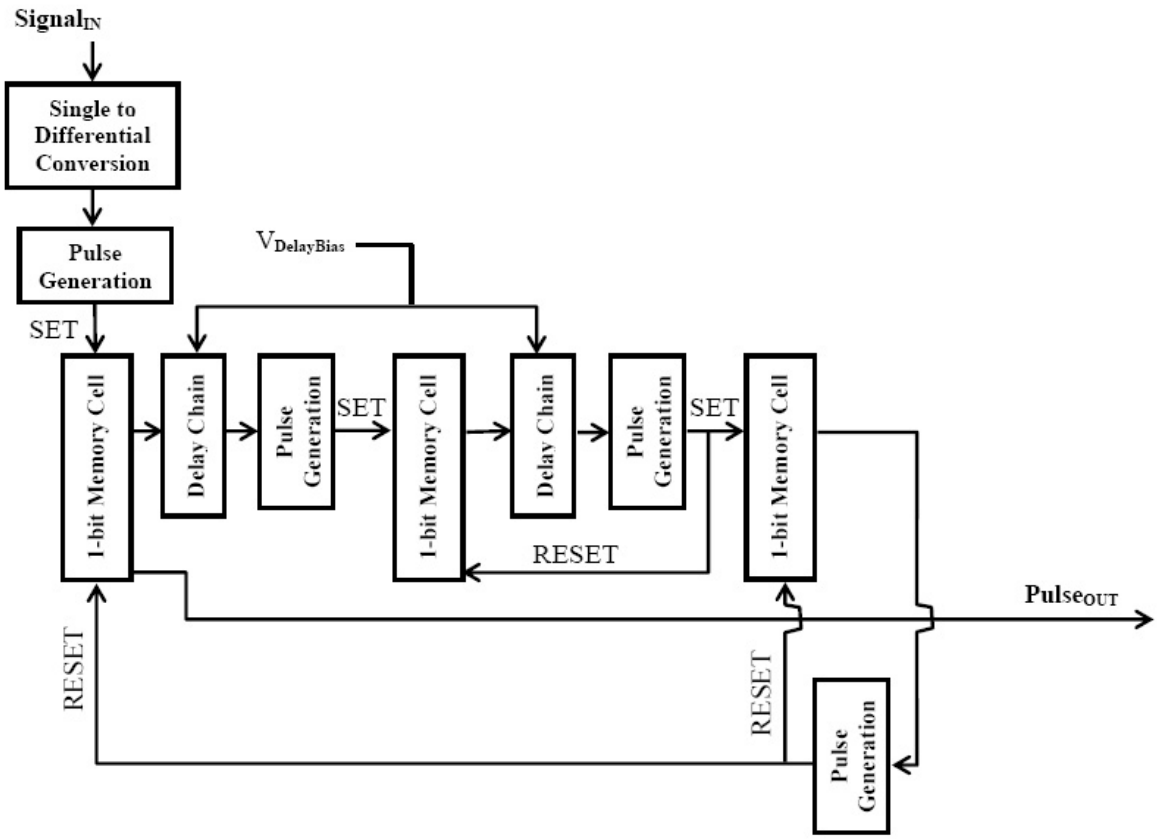


Figure 3.11: Input monostable block diagram of design analyzed.

the amount of time from the time the flip-flop controlling the output value is set until it is reset. To reduce sensitivity to supply noise and GND bounce, the MOS current-mode logic (MCML) family was selected for implementation of the devices in this block. MCML, being differential, has a high power supply rejection ratio (PSRR) and, as such, much of common mode supply noise is removed ([*Razavi*, 2001]). In [*Allstot et al.*, 1993], reduction of noise sensitivity up to 65% over complementary CMOS were documented. In other works ([*Musicer*, 2000], [*Toda et al.*, 2006], [*Heydari*, 2006], [*Mizuno et al.*, 1996]), further analysis of the common mode rejection are done as well as analysis of the power saving ability of MCML at high frequencies. Power reduction is achieved by reducing output swings and the use of a constant current through the circuits irregardless of the frequency the circuit is being run at. Unlike complementary CMOS circuits, whose power is related to the number of times the circuit's nodes must be charged and discharged (and therefore is related to the frequency the circuit is being run at), using a constant current means there is a constant power requirement, and, at high-frequencies, with proper circuit design, this can be turned into power savings.

3.3.1 Sizing

Smaller sized transistors have the ability to run at high speeds due to their reduced capacitance. In a circuit to be used at relatively high speeds such as the JAC being proposed here, this characteristic is something that needs to be taken advantage of. Unfortunately, unlike long channel devices, a complete and convenient set of mathematical equations describing short-channel device behavior was not to be found. Though the channel lengths being used were clearly in the short-channel range, the design process followed in the early stages of this project was to use the long-channel device equations to find initial sizes of devices, and then tweak until the circuit simulated correctly. That process was time consuming and unsatisfying. To make the design process less random, the fact that most of the JAC is constructed of differential pairs was used. A short-channel differential pair with acceptable with current values (10μm to 40μm), output voltages (maximum: VDD, Minimum: $VDD - 2V_T$), and rise and fall times of approximately 40ps was used as a base

design. See Table 3.1 for base sizing, Figure 3.14 and 3.15 for a simulation of rising and falling edges, and see Figure 3.16 for affect of using wider devices. The region of operation of the active load PMOS, output voltage of the differential pair, and the sink and load $PMOS$ current of two model cases were then observed as only the W of the differential pair were incrementally multiplied up, and as both W and L were incrementally multiplied up. The gate voltage to the load $PMOS$ was fixed at $350mV$, and the voltage to the sink $NMOS$ was determined by a variable swing controller circuit. The grid formed by sweeping these two multipliers gives a way to select a differential pair size for a set of requirements. Figures 3.12 and 3.13, the X-axis represents the W and L multiplication factor. All transistors' W and L in the differential pair are multiplied by this number. Each individual line represents the result of sweeping the width of all transistors in the differential pair. As example of how this might be used, say a differential pair with an output voltage low value of $870mV$, and a current of $60\mu A$ was needed. Referring to Figure 3.12, the multiplier for both W and L can be found. In this case it is around 3 or 4. Once this value has been found, Figure 3.13 can be used to find the multiplier of W. In this case it looks to be approximately 2. These graphs can be adjusted to accommodate smaller or larger swings and smaller and larger currents. Also note that, because the differential pairs in the delay chain are being used as digital components, process variation which changes just W or just L (Figure 3.13), or both (a combination of Figures 3.13 and 3.12) may change swing minimum voltage and/or the drive by some amount, but, unless the process variation is quite excessive, the resulting differential pair will still be a functioning inverter and behave as needed in the delay chain.

Transistor	Width	Length
$PMOS$	0.38	0.26
$NMOSSwitch$	0.13	0.13
$NMOSSink$	2.08	1.04

Table 3.1: Base differential pair size with input $PMOS$ voltage of $350mV$.

Sizing of the differential pair base design took into account input capacitance, PSRR, and gain. To reduce capacitance seen by the driving circuit, minimum sized switching transistors were used. PSRR was calculated using the long-channel small signal model

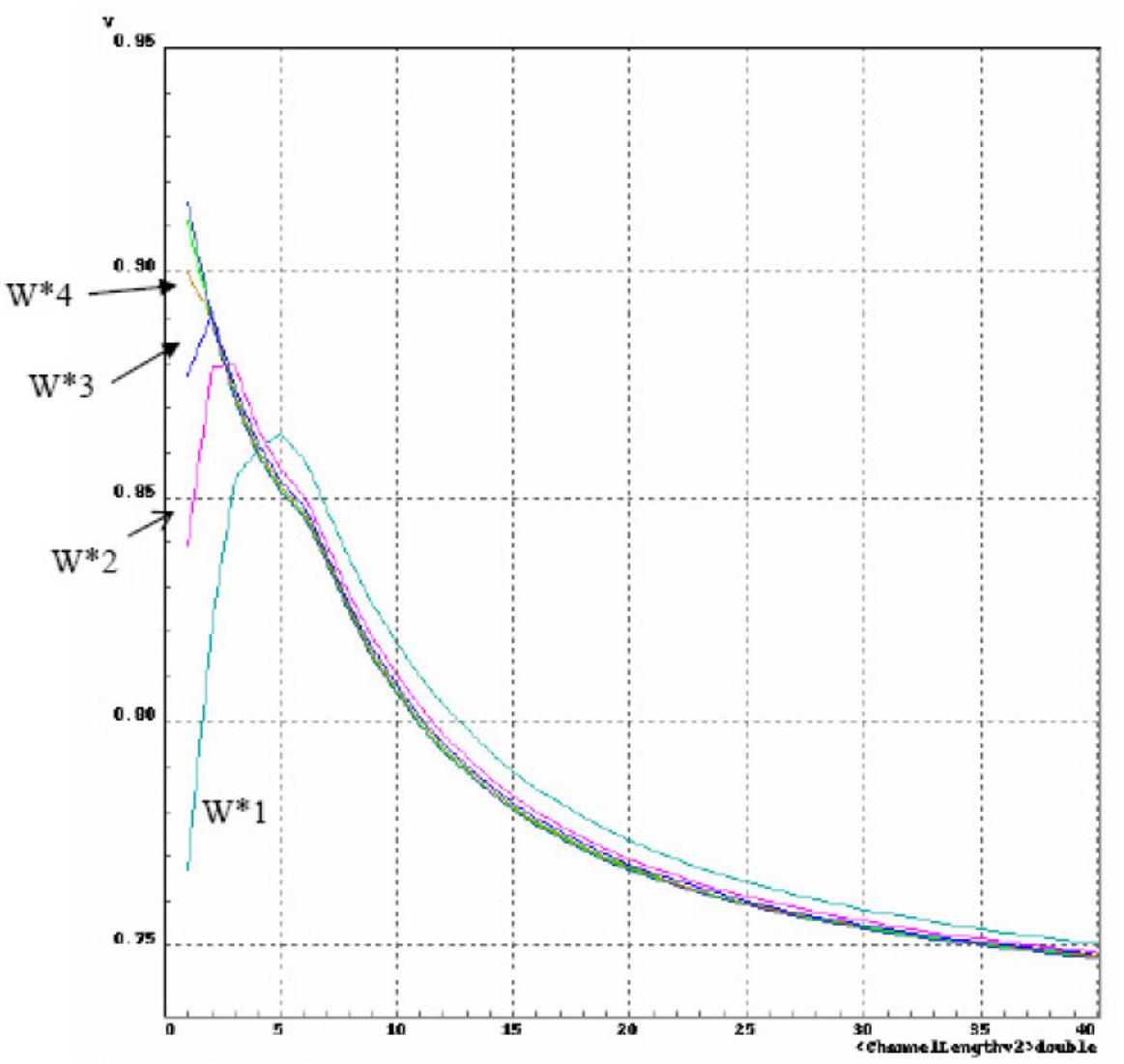


Figure 3.12: Both W and L multiplied.

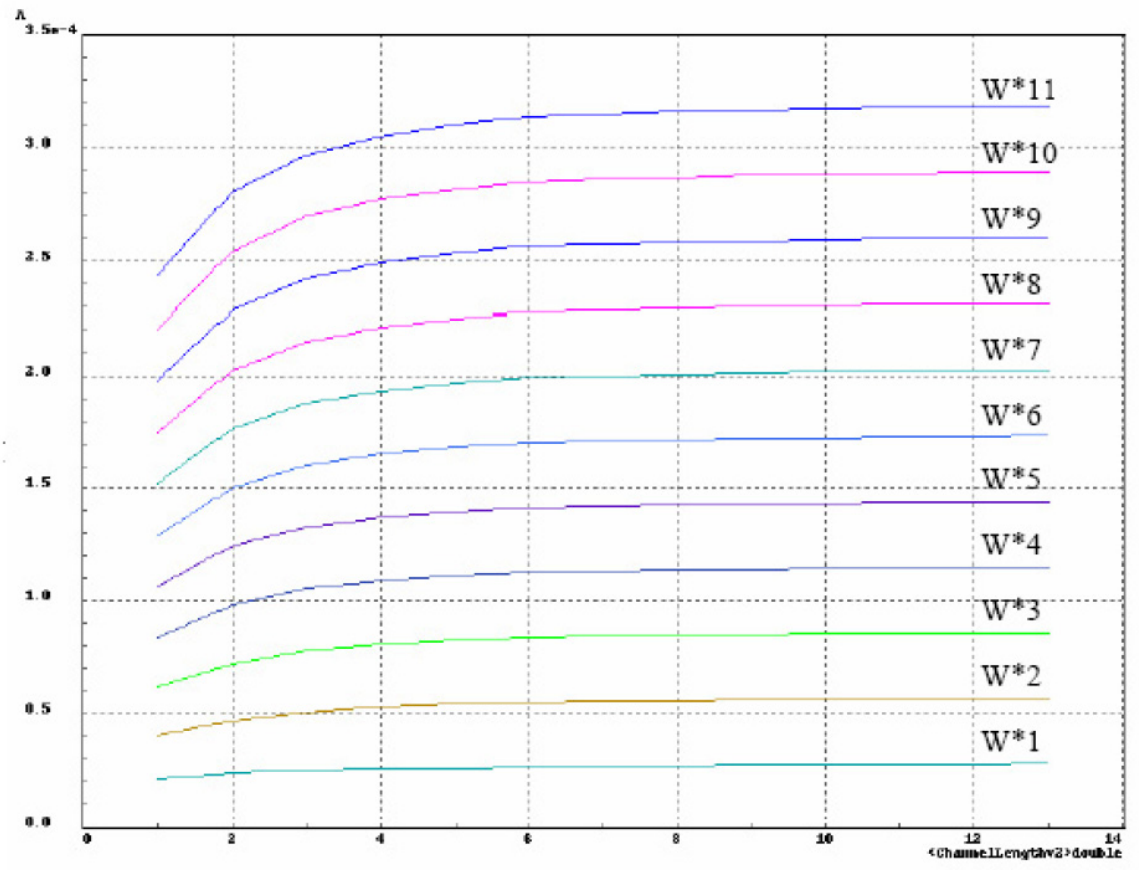


Figure 3.13: Only W multiplied.

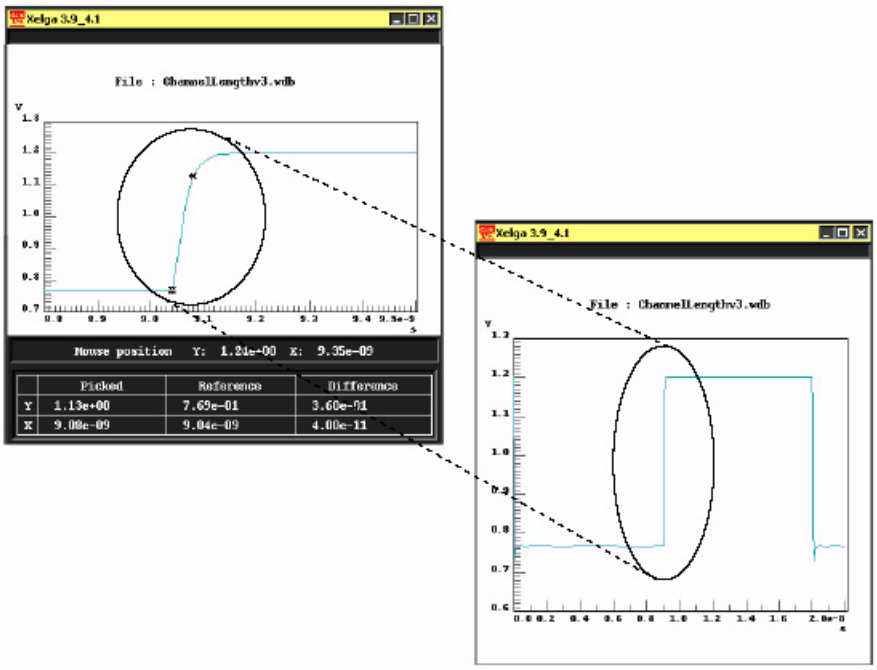


Figure 3.14: Rise and fall times of approximately 40ps.

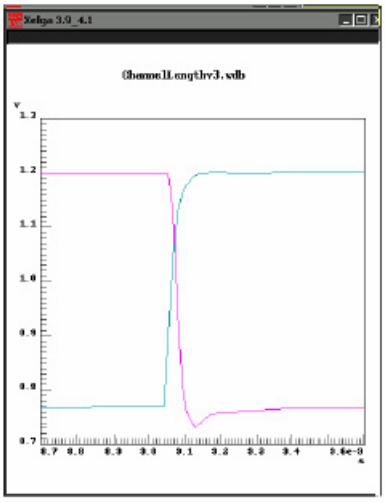


Figure 3.15: Fall time is less than rise time for base differential pair circuit.

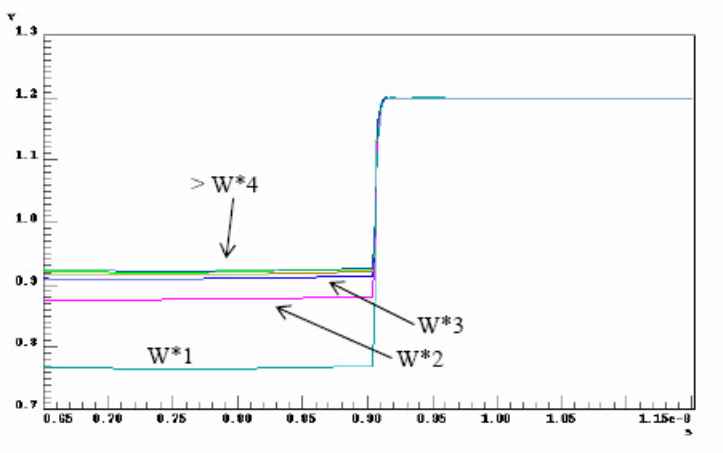


Figure 3.16: Affect on transition voltage of multiplying W.

though, of all the transistors used in the inverter, only the the tail current source transistor had a length over $1um$. Figure 3.17 shows the transistor level circuit diagram for the inverter and Figure 3.18 shows the small-signal model. Because the small-signal model is being used to calculate a common mode value, the half-circuit model is used. Also, the modes of operation of the individual transistors is taken into account and unimportant capacitance is removed from the model, and the source-bulk g_m generator is excluded from calculations. The contribution of supply noise to output noise is calculated separately from ground-bounce noise. Figure 3.19 shows the model used to calculate the effect of supply noise on the output and Figure 3.20 shows the model used to calculate the effect of ground bounce on the output.

As defined in [*Meyer et al.*, 2001], the small-signal output voltage of a circuit is

$$v_o = A_{dm} v_{id} + A^+ v_{VDD} + A^- v_{GND} \tag{3.1}$$

where A^+ and A^- are the small-signal gains from positive and negative power supplies to the output, respectively. Power supply rejection ratio is then defined

$$v_o = A_{dm}\left(v_{id} + \frac{A^+}{A_{dm}} v_{VDD} + \frac{A^-}{A_{dm}} v_{GND}\right) = A_{dm}\left(v_{id} + \frac{v_{VDD}}{PSRR^+} + \frac{v_{GND}}{PSRR^-}\right) \tag{3.2}$$

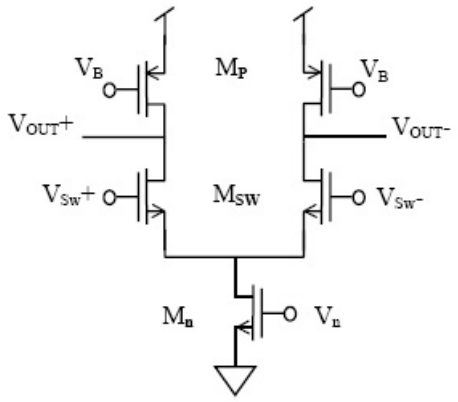


Figure 3.17: Transistor level diagram of inverter.

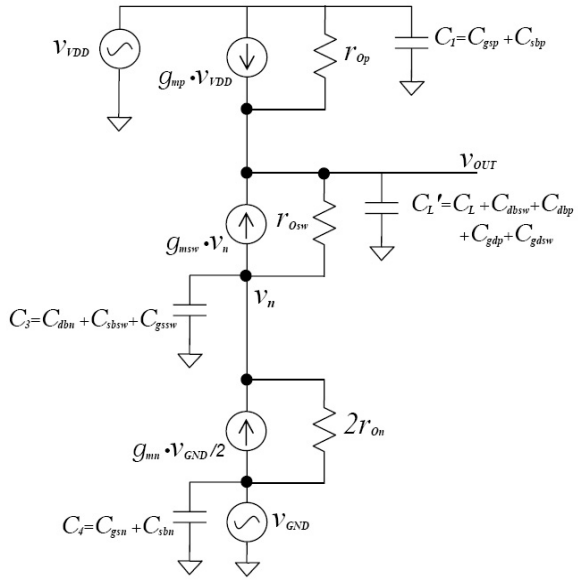


Figure 3.18: Small-signal model including supply and GND noise.

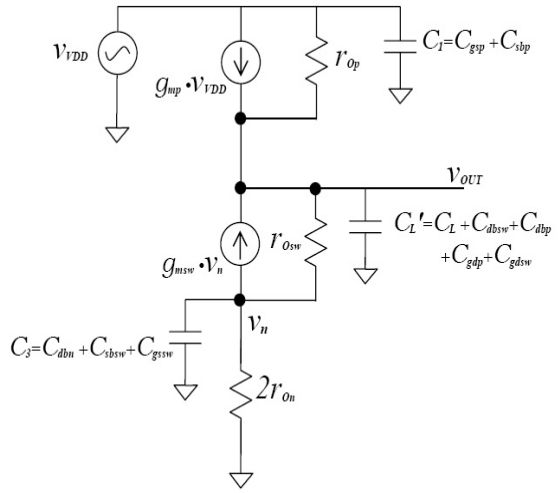


Figure 3.19: Small-signal model including only supply noise.

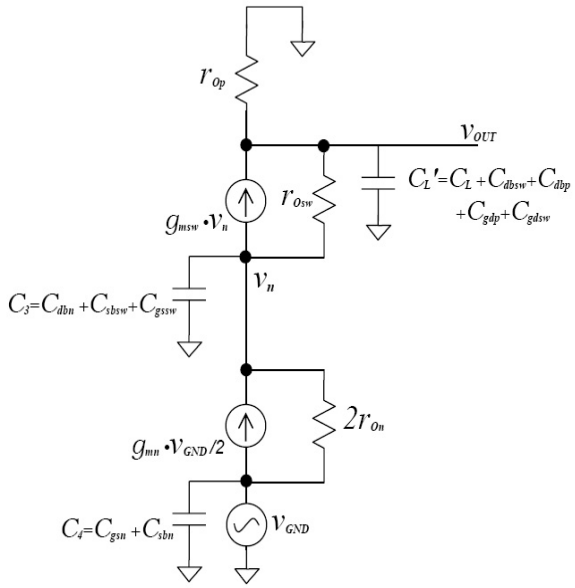


Figure 3.20: Small-signal model including only GND noise.

where

$$PSRR^{+} = \frac{A_{dm}}{A^{+}} \tag{3.3}$$

and

$$PSRR^{-} = \frac{A_{dm}}{A^{-}} \tag{3.4}$$

To determine A^{+}, $\frac{v_{out}}{v_{VDD}}$, the following KCL node equations were used:

$$\begin{aligned} g_{mp}v_{VDD} + \frac{v_{VDD} - v_{out}}{r_{Op}} + g_{msw}v_n + \frac{v_n - v_{out}}{r_{Osw}} - v_{out}sC_L\prime = 0 \\ -g_{msw}v_n + \frac{v_{out} - v_n}{r_{Osw}} - v_n sC_3 - \frac{v_n}{2r_{On}} = 0 \end{aligned} \tag{3.5}$$

To determine A^{-}, $\frac{v_{out}}{v_{GND}}$, the following KCL node equations were used:

$$\begin{aligned} \frac{-v_{out}}{r_{Op}} - v_{out}sC_L\prime + g_{msw}v_n + \frac{v_n - v_{out}}{r_{Osw}} = 0 \\ -g_{msw}v_n - v_n sC_3 + \frac{g_{mn}v_{GND}}{2} + \frac{v_{GND} - v_n}{2r_{On}} + \frac{v_{out} - v_n}{r_{Osw}} = 0 \end{aligned}$$

Solving for v_{VDD} using the equations listed in Equation 3.5 and then simplifying gave the following results:

$$v_{VDD} \approx v_{out}\left[\frac{sC_L\prime r_{Op}\left(A_{sw} + sC_3 r_{Osw}\right) + sC_3\left(r_{Osw} + r_{Op}\right) + A_{sw}}{A_p(A_{sw} + sC_3 r_{Osw})}\right] \tag{3.6}$$

where $A = g_m r_O$ of device. Solving for v_{GND} using the equations listed in Equation 3.6 and then simplifying gave the following results:

$$v_{GND} \approx v_{out}\left[\frac{2sC_L\prime r_{On}r_{Op}\left(A_{sw} + sC_3 r_{Osw}\right) + 2sC_3 r_{On}\left(r_{Osw} + r_{Op}\right) + 2A_{sw}r_{On}}{r_{Op}A_{sw}A_n}\right] \tag{3.7}$$

Note that the low-frequency contribution to noise on the output by V_{VDD} is

$$v_{VDD} \approx v_{out} \frac{1}{A_p} \tag{3.8}$$

and that the low-frequency contribution to noise on the output by V_{GND} is

$$v_{GND} \approx v_{out} \frac{2r_{On}}{r_{Op}A_n} \tag{3.9}$$

The results in Equations 3.8 and 3.9 make sense in that, as shown by these equations, if gain of the device which the supply or GND noise must cross to get to the output is large, the effect on the output will be large.

Finally, the total low-frequency contribution to output noise from noise on the supply and GND will be

$$v_{VDD} + v_{GND} \approx v_{out} \left(\frac{1}{A_p} + \frac{2r_{On}}{r_{Op}A_n} \right) \tag{3.10}$$

and rearranging gives

$$\frac{v_{out}}{v_{VDD} + v_{GND}} \approx \frac{1}{\frac{1}{A_p} + \frac{2r_{On}}{r_{Op}A_n}} = \frac{A_p A_n r_{Op}}{A_n r_{Op} + A_p} \approx A_p \tag{3.11}$$

Also, the high-frequency contribution to noise on the output by V_{VDD} is

$$v_{VDD} \approx v_{out} \frac{sC_L\prime r_{Op} r_{Osw} + r_{Osw} + r_{Op}}{A_p r_{Osw}} \tag{3.12}$$

and the high-frequency contribution to noise on the output by V_{GND} is

$$v_{GND} \approx v_{out} \left[\frac{2s^2 C_L\prime r_{On} r_{Op} C_3 r_{Osw} + 2sC_3 r_{On} \left(r_{Osw} + r_{Op} \right)}{r_{Op} A_n A_{sw}} \right] \tag{3.13}$$

Equation 3.12 says that there is a zero at approximately

$$s \approx \frac{r_{Osw} + r_{Op}}{C_L\prime r_{Op} r_{Osw}} \tag{3.14}$$

after which the affects of noise on the supply diminish.

Equation 3.13 says that there is a zero at zero frequency, and a second at

$$s \approx \frac{r_{Osw} + r_{Op}}{C_L\prime r_{Op} r_{Osw}} \tag{3.15}$$

These results suggest that, if possible, moving the zeros calculated in Equations 3.14 and 3.15 as low as possible will decrease GND and supply affects on the output. Of course, this must be balanced with the overall frequency response of the differential pair so that the circuit can run at the desired operating frequency. As derived in [*Razavi*, 2001], the first pole of a differential pair occurs at

$$\omega_{p1} \approx \frac{1}{(r_{Osw} || r_{Op}) C_L\prime} = \frac{r_{Osw} + r_{Op}}{C_L\prime r_{Op} r_{Osw}} \tag{3.16}$$

Note that the zero for the supply noise, the second zero for the GND noise, and the first pole of the differential pair are all equal. The conclusion reached here is that, to get the best high-frequency supply and GND noise attenuation on a differential pair that will work at a particular frequency, the first pole should be moved as close as possible to the operating frequency. Using the $10x$ rule of thumb, ω_{p1} should equal $10 \times \omega_{OperatingFrequency}$.

As an aside, the results derived here did not agree with a [*Pialis and Phang*, 2003], though zero values agreed with [*Razavi*, 2001], among others.

To find the total high-frequency supply and GND noise transfered to the output, Equations 3.12 and 3.13 can be added as shown in Equation 3.17.

$$v_{VDD} + v_{GND} \approx v_{out}\left[\frac{sC_L\prime r_{Op} r_{Osw} + r_{Osw} + r_{Op}}{A_p r_{Osw}}\right] + v_{out}\left[\frac{2s^2 C_L\prime r_{On} r_{Op} C_3 r_{Osw} + 2sC_3 r_{On}\left(r_{Osw} + r_{Op}\right)}{r_{Op} A_n A_{sw}}\right] \quad (3.17)$$

Simplifying Equation 3.17 does not give additional insight and is therefore not done here.

Revisiting the differential pair sizing using the supply and GND noise results suggests that minimal gain tail current source and load transistors should be used. Both the tail current source and the load transistors are in the common gate configuration but the tail current source is in saturation and the load transistors are in triode. The gain for the tail current source is shown in Equation 3.18 and the gain for the load transistors is shown in Equation 3.19 and 3.22.

$$A_v = G_{mn} R_{On} = \left(\frac{1}{r_{On}} + g_{mn}\right)\left[r_{On} || \left(\frac{1}{g_{msw}} + r_{Op}\right)\right] \approx \frac{\left(1 + g_{mn} r_{On}\right) r_{Op}}{r_{On} + r_{Op}} \quad (3.18)$$

$$A_v = G_{mp} R_{Op} \approx \left(k_p\prime \frac{W}{L} V_{DSp}\right)\left[r_{Op} || \left(r_{Osw} + r_{On} || r_{Op}\right)\right] \quad (3.19)$$

Because the load transistor is in triode, its r_{Op} is equal to

$$r_{Op} = \frac{V_{DSp}}{I_{DSp} + \frac{V_{DSp}^2}{2}} \approx \frac{2}{V_{DSp}} \quad (3.20)$$

r_{Op} can be assumed to be much less than r_{On} or r_{Op} and when substituted into Equation 3.19 gives

$$A_v = G_{mp} R_{Op} \approx \left(k_p\prime \frac{W}{L} V_{DSp}\right)\left[r_{Op} || \left(r_{Osw} + r_{On} || r_{Op}\right)\right] \quad (3.21)$$
$$\left(k_p\prime \frac{W}{L} V_{DSp}\right) r_{Op} \approx \left(k_p\prime \frac{W}{L} V_{DSp}\right) \frac{2}{V_{DSp}} = 2k_p\prime \frac{W}{L}$$

To reduce gain of the tail current sink device, r_{On} needs to be reduced, and the simplest way to reduce r_{On} is to increase the current through the device. For the tail current source device, which is in saturation, this will reduce the gain by a factor of $\frac{1}{\sqrt{I_{DS}}}$. Note that the equation for g_m for devices in saturation is

$$g_m = \sqrt{\frac{2k_n{}'\frac{W}{L}}{I_{DS}}} \tag{3.22}$$

and therefore changing the value of $\frac{W}{L}$ will not result in a change in gain. To reduce gain, V_{in} must be reduced, not the value of $\frac{W}{L}$.

To reduce the gain of the active load device, $\frac{W}{L}$ should be reduced, and therefore the current through the device. The current can then be increased by increasing the device's bias voltage, but, in general the requirement that the current must be reduced for the tail current sink and decreased for the loads presents a design challenge. Knowledge of the environment where the circuit is to be used can give hints on whether reducing the gain of the tail current sink or the gain of the load devices is more important, but, in this thesis, neither is given priority and a balanced design was targeted.

The calculations in this section were done using the long-channel model assuming that, though the equations would not totally agree with the behavior of the devices, in general, the trends of the short-channel devices would still be described by the equations. Also, the calculations were for the case where both switches were "ON", which is only the case during transition. Luckily, this is the case that is most important in that the transition is actually where information is passed on from one inverter to the next. An error here matters whereas an error while an inverter is at a steady-state value can be ignored.

3.3.2 Sub-block Design

This section describes the individual gates at the transistor level and talks about design decisions.

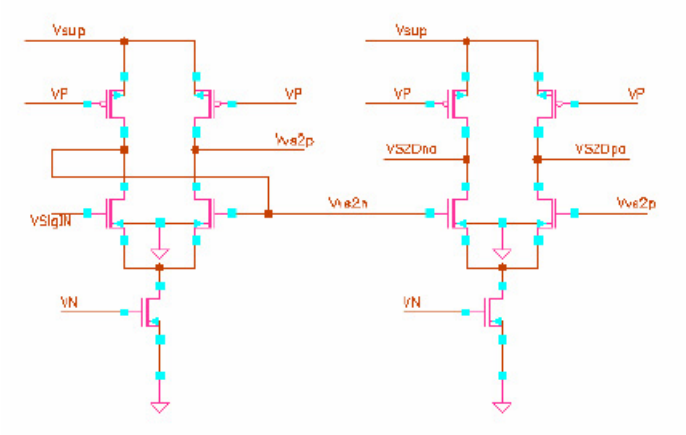


Figure 3.21: Single-to-differential converter cell.

Single to Differential Conversion

Because internal signals of the JAC are differential, single-ended input signals must be converted to differential before they can be used in the circuit. Addition of jitter at this point is removed by subsequent blocks and does not need to be a consideration in designing this block. This block is implemented in a two-stage circuit of two cascaded inverter cells. The device sizes are that shown in Table 3.1. Figure 3.22 shows a simulation of the single-to-differential block.

MOS Current-Mode Logic Inverter

An MCML inverter is simply a differential pair (See Figure 3.23). Unlike some implementations of MCML circuits seen in publications, the inverters in this design were designed to have a larger output swing; They are designed to swing to the supply voltage as a maximum value, and down to approximately $VDD - 2V_T$ as a minimum value. The design decision to use a wider swing was made after observing the negative effect on slope and drive power when the OFF switch was not completely OFF. The wider swing uses more power, but results in steeper slope and therefore less susceptibility to noise.

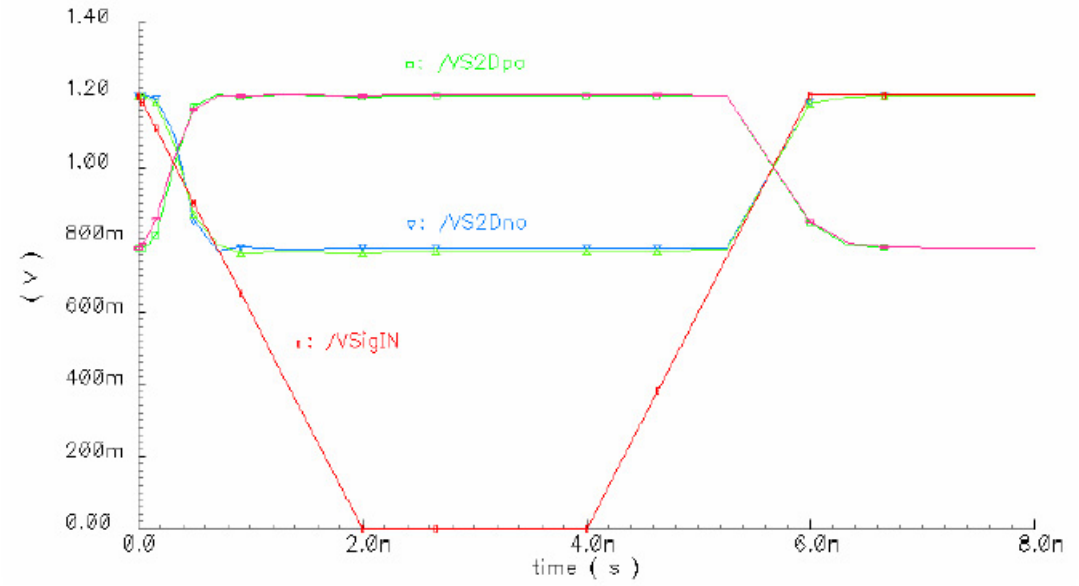


Figure 3.22: Single-to-differential converter cell output.

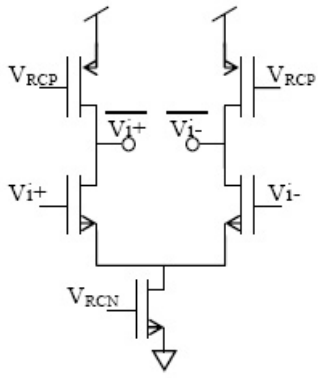


Figure 3.23: Inverter MCML cell.

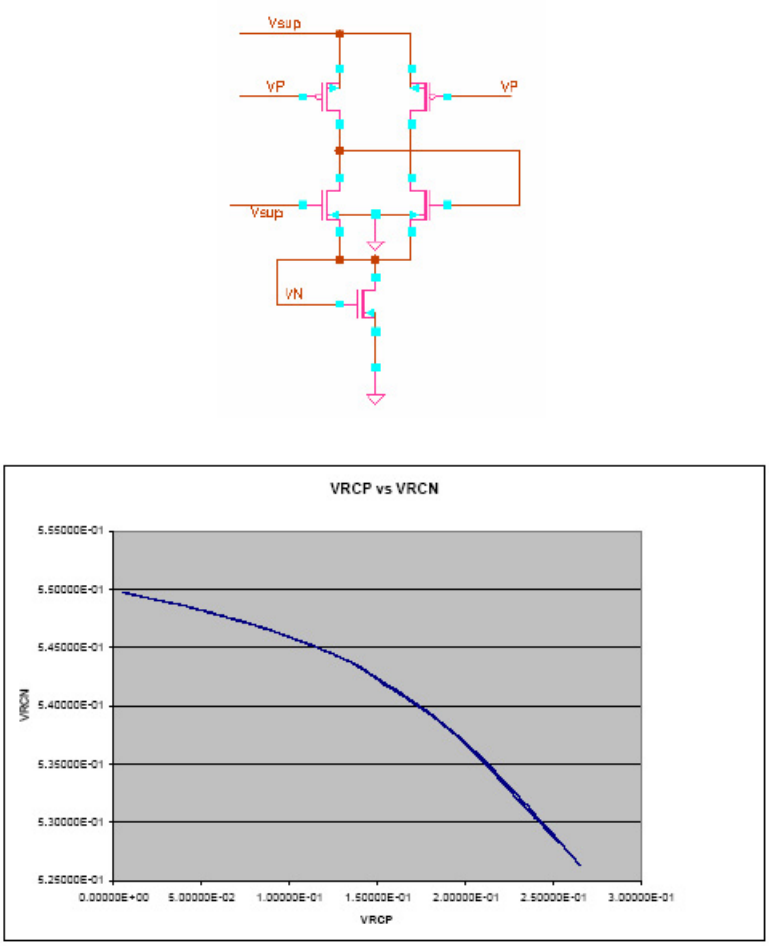


Figure 3.24: VRCN generated using VRCP.

MOS Current-Mode Logic Variable Delay Controller

The variable delay control block generates the bias value for the tail current sink given a voltage bias value for the active load PMOS devices. Figure 3.3.2 shows the circuit diagram for the block and Figures 3.24 and 3.25 show, respectively, when given a bias value for the load devices, the tail current sink bias value, and the output delay for a four-inverter chain.

MOS Current-Mode Logic $NAND$ and $XNOR$ Gates

The $NAND$ and $XNOR$ gates, unlike the simple inverter, require the stacking of switches. Because, simply stated, the high and low output voltage is decided by voltage

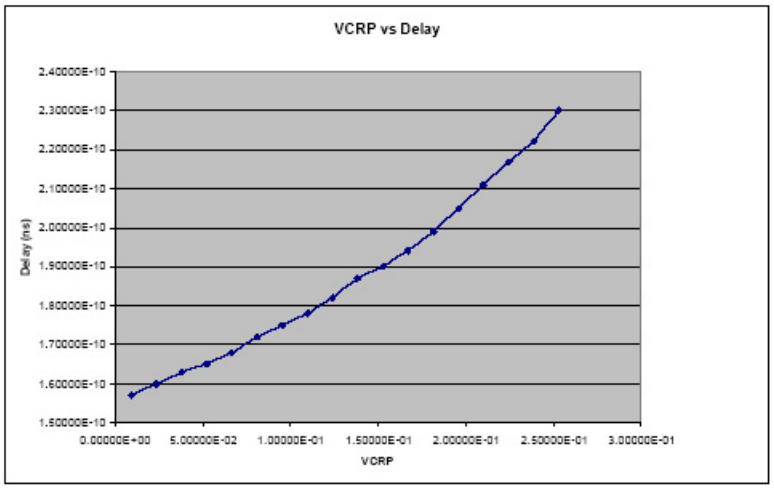


Figure 3.25: Delay through four-inverter chain by varying VRCP.

division, stacking switch transistors of the same size as the inverter will move the output voltage above that of the inverter. In order to get the same slope and swing on the output of the $NAND$ or $XNOR$ gate, the resistance of the switches must be halved. As shown in Figure 3.13, doubling the width of the devices in the inverter doubles the current, essentially halving the resistance. Using this as proof of a linear W verses resistance relationship allows the assumption that doubling the width of the stacked transistors should give the same resistance through them as a single transistor. As shown in Figure 3.26, some of the transistors are stacked (V_A, V_B, and $V_{\overline{B}}$), and one is not stacked (*i.e.* $V_{\overline{A}}$). To keep a delay close to the inverters, double the drive is needed to the stacked transistors, and the driving inverter of the $NAND$ gate should have, optimally, twice the current. [*Rabaey*, 1996] states that in order to drive larger loads, a chain of inverters may be used, each with a larger driving capability, and the optimal (maximal) scaling factor for such a chain of inverters is derived to be e or 2.7182. Because only twice the drive is needed, only a single scaled up inverter is needed before the $NAND$ gate, and that inverter should have only its W multiplied by the scaling factor. As shown in Figure 3.28, doubling the W of the driving inverter and sizing the $NAND$ gate as described above, the $NAND$ gate maintains the swing height and slope of the driving inverters.

Also, because the stacked switching transistors must be in triode, they do not have the

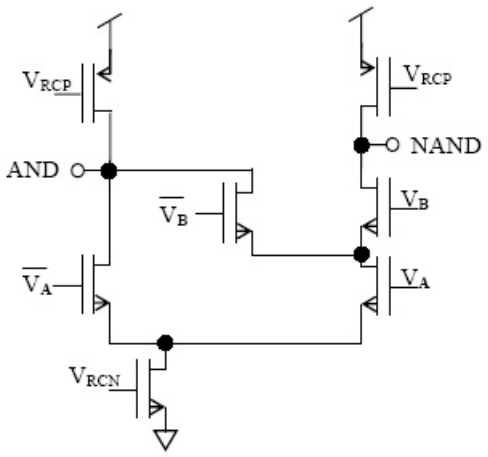


Figure 3.26: NAND MCML cell.

pull-down power of the single switching transistor on the other side of the $NAND$ gate. Because the side with the single switching transistor pulls down better than the stacked side, and the stacked side is better at pulling up, attention should be made to how inputs are connected to the gate.

The $XNOR$ gate, if used, also has stacked transistors and needs the same considerations as the $NAND$ gate in its own sizing and the sizing of the inverter driving it. The sizes used in this simulation are shown in Table 3.2.

Transistor	Width	Length
$PMOS$	0.38	0.26
$NMOS$ (Stacked - Top transistors)	0.26	0.13
$NMOS$ (Stacked - Bottom transistors)	0.26	0.13
$NMOS$ (Single transistor)	0.13	0.13
$NMOSSink$	2.08	1.04

Table 3.2: NAND device sizes.

MOS Current-Mode Logic Flip-Flop

As described in earlier sections, monostables based solely on delay blocks and a $NAND$ gates cannot generate a pulse width of greater than a 50% duty cycle and cannot have an

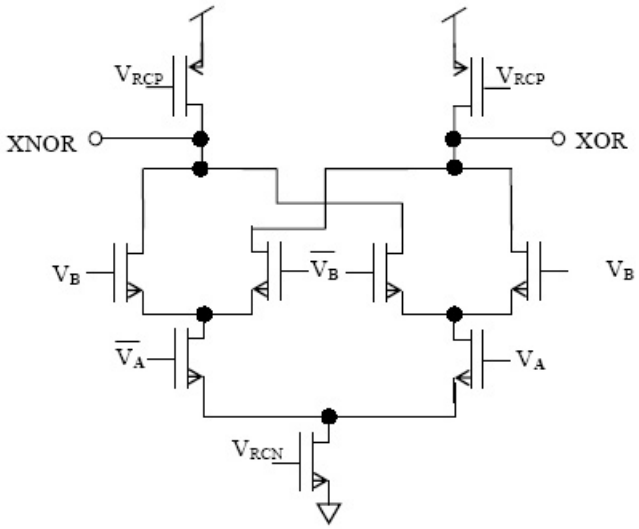


Figure 3.27: XNOR MCML cell.

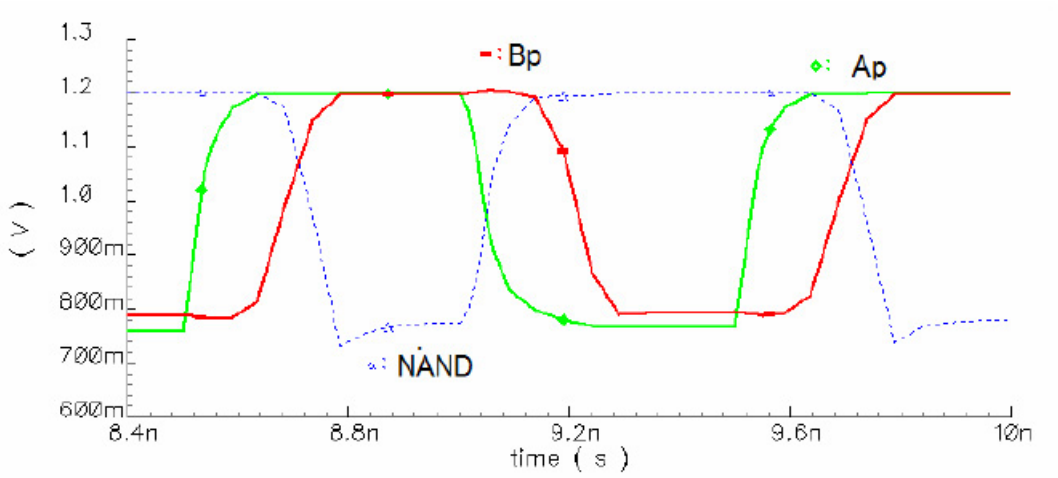


Figure 3.28: Single input pair and output of $NAND$ gate.

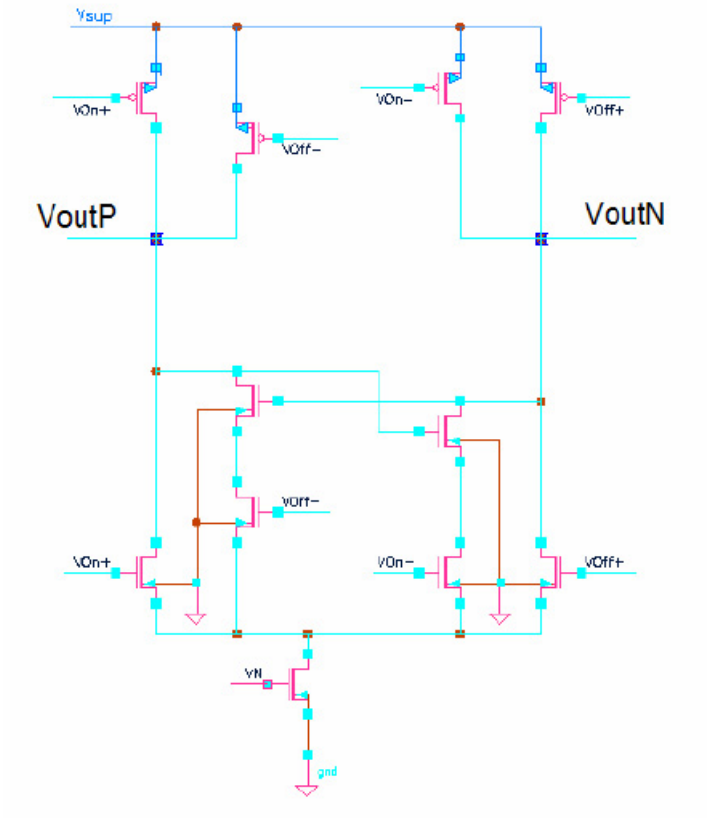


Figure 3.29: Latch circuit.

output pulse wider than their input pulses. Pulse mode flip-flops can be used to implement a monostable that does not have these contraints.

Figure 3.29 shows the latch circuit. The latch circuit has four possible modes affecting V_{outN} (V_{outP}); Hold low (high), transition from low to high (high to low), hold high (low), and transition high to low (low to high). Because V_{outN} and V_{outP} have exactly the same behavior except their modes of operations are 'opposite' (high-to-low transition for one is the low-to-high transition. Hold high for one is a hold low for the other), only a description of one, V_{outN}, will be done here. All descriptions of V_{outN} are applicable to V_{outP}.

Phase 1 is shown in Figure 3.30 for V_{outN}. During phase 1, V_{outN} is low and is not in transition. The value of V_{outN} is held constant by voltage division between the PMOS pull-up transistor and the NMOS pull-down transistors. This allows the swing to be controlled and kept from going all the way to GND. The top transistor of the two stacked transistors

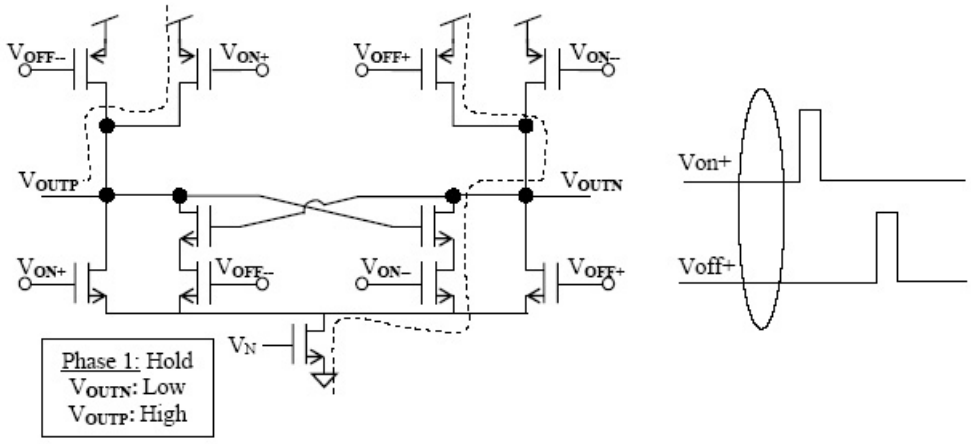


Figure 3.30: Latch holding value with V_{outN} (V_{outP}) low (high).

is cross coupled to the opposite (held at high) output, V_{outP}, and is therefore on. The bottom transistor is on or conducting during holding on non-transitioning time due to its high input. The lower transistor on the V_{outP} pull-down network is actually on at this time, but because the top, cross-coupled NMOS is controlled by the value of V_{outN}, which is low, it is off and the network does not pull-down the V_{outP} node value.

In phase 2 (Figure 3.31), V_{outN} transitions from low to high. V_{ON-} goes low and shuts off the path through the pull-down network and, in addition to the pull-up transistor that was conducting in phase 1, an additional parallel transistor is turned on. The additional transistor is only used during transitions and can be sized to control switching speed. The trade off using a large transistor here for a fast pull-up is the loading to the driving circuit. In this design, the driving circuit is an inverter which may be sized up as described in the sizing section of this chapter.

Phase 3 (Figure 3.32) is again a holding phase. In this phase, only the pull-up network is activated and V_{outN} is held high. Note that V_{outP} is being held at a low value set by voltage division of the pull-up and pull-down networks on its side now.

The last phase, phase 4 (Figure 3.33), shows the V_{outN} transition from high to low. As with the transition from low to high, the opposing (pull-up) circuitry is turned off so only the pull-down network is active. At first glance it looks as though there is a danger of

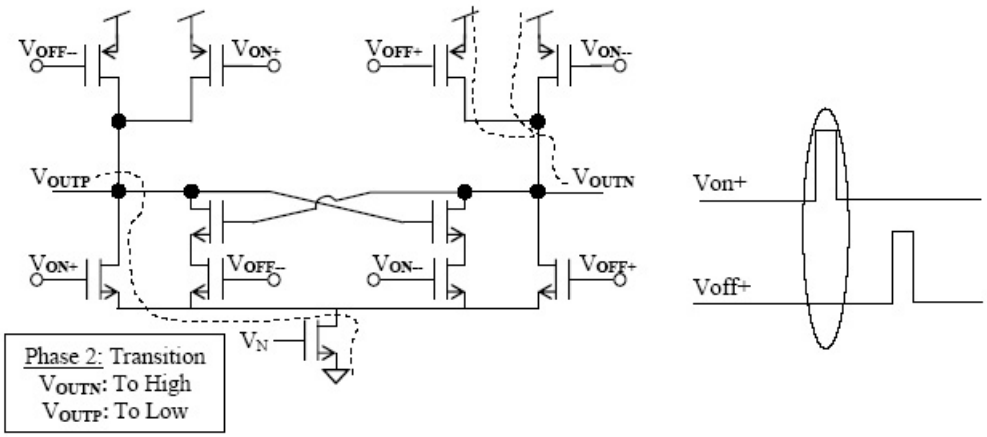


Figure 3.31: Latch with V_{outN} (V_{outP}) transitioning from low to high (high to low).

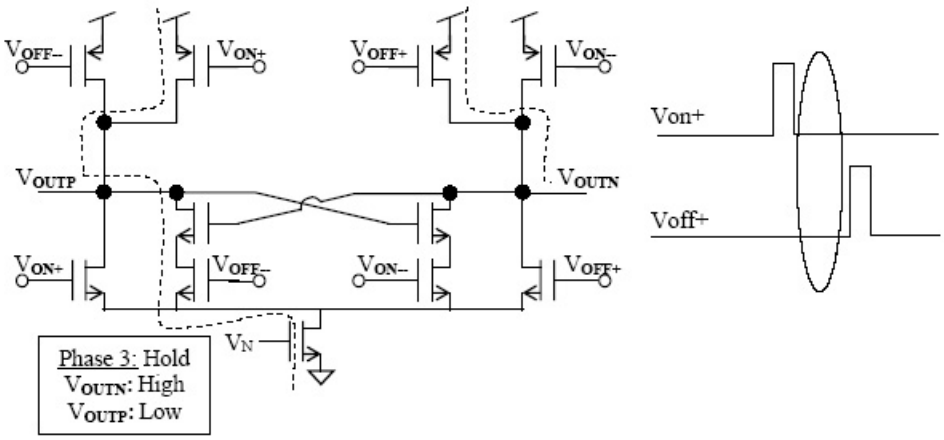


Figure 3.32: Latch holding value with V_{outN} (V_{outP}) high (low).

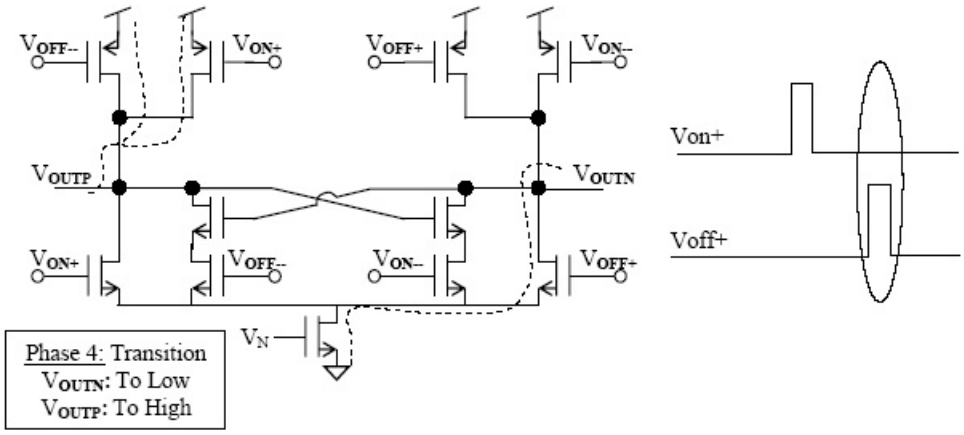


Figure 3.33: Latch with V_{outN} (V_{outP}) transitioning from high to low (low to high).

V_{outN} being pulled down to GND. Fortunately and unfortunately, if V_{outN} drops below a certain voltage, the source voltage of the tail current source transistor also drops and the V_{outP} pulldown network starts conducting. This is fortunate in that it keeps V_{outN} from dropping to GND, but unfortunate in that having the pulldown V_{outP} network active slows the pull-up of V_{outP}.

Figure 3.34 shows a simulation of a transition on the proposed flip-flop. The output low hold value is approximately $750mV$ and the high value, $1.2V$. The simulation shows a very high transition speed, reaching the desired output in about $6ps$ for the high to low transition, and taking less than $40ps$ for the low to high transition. A higher low to high transition speed can be reached using larger pull-up transistors. The transition transistors are sized larger than the hold transistors to give approximately twice the current. The reason being that the hold transistors only need to supply enough current to hold a value, whereas the transistors used during a transition not only need to supply internal nodes, but also drive whatever devices are on the flip-flops output. Also, as with the inverters used in the delay chain, process variation could make a difference in output swing and/or drive of the flip-flop, but the variation it could cause would not hurt the basic functionality of this block.

Figure 3.35 shows a simulation of a latch receiving an input pulse with a height of

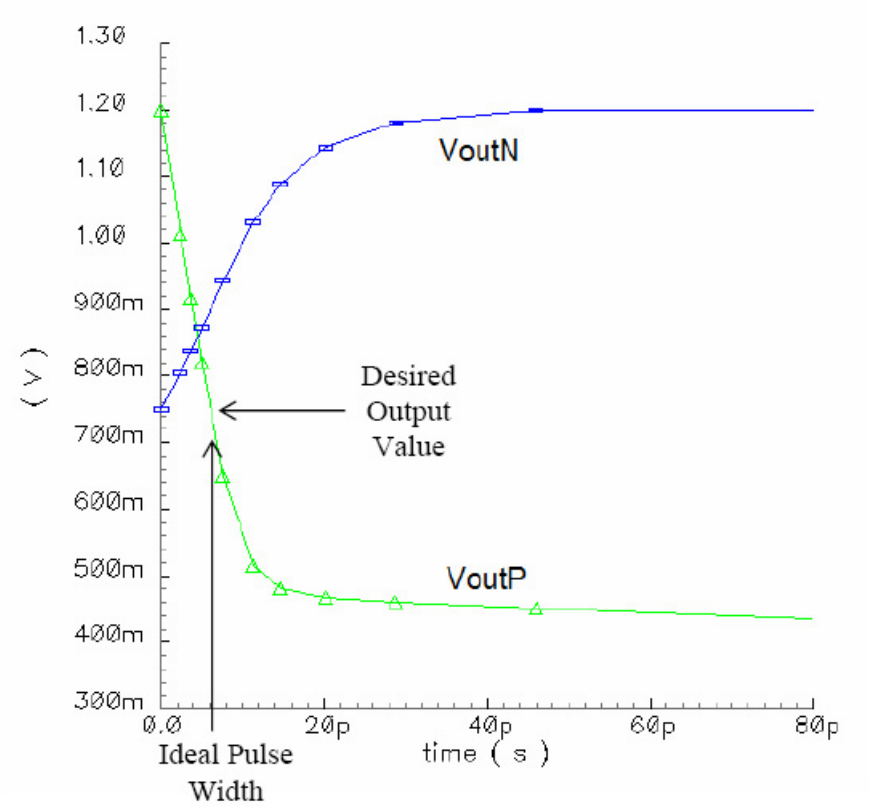


Figure 3.34: Simulation of latch with V_{outN} (V_{outP}) transitioning from low to high (high to low).

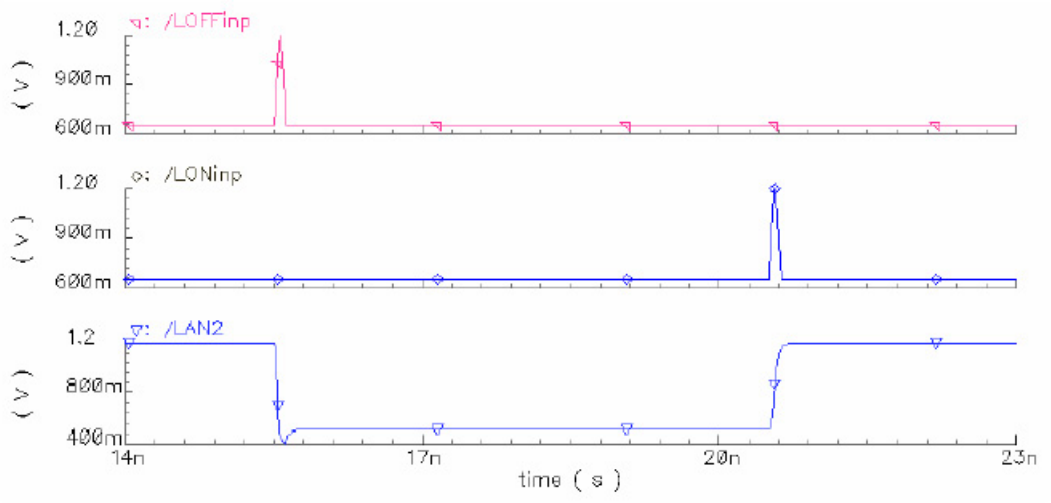


Figure 3.35: Simulation of latch with $10ps$ input pulse width with $60ps$ rise and fall times.

$550mV$, a pulse width of $10ps$, and rise and fall times of approximately $60ps$. The current in the flip-flop in this simulation was approximately $10uA$.

To guarantee efficient operation of this flip-flop, the input swing should be larger than the output swing. To show this, first assume the converse; that the lowest voltage seen on the input is equal to the lowest voltage seen on the output. If this is the case then the low inputs into the flip-flop will equal the low outputs out of the flip-flop. This low value will be called V_{Min} here. As an example, in the V_{outN} 'hold low' state (phase 1), the value of the output V_{outN} and the input to V_{ON+} will both be the same value, V_{Min}. The output voltage is determined by voltage division between the pull-up and the pull-down transistors, so current must be flowing through the pulldown network of V_{outN}. On the V_{outP} side, the lower left transistor with V_{ON+} as its input must be off in order for V_{outP} to be efficiently pulled up. For the transistor being controlled by V_{ON+} to be in cutoff, its source voltage must be greater than $V_{Min} - V_{TN}$. With V_{outN} equal to V_{Min}, and the source voltage of the stacked transistors at a value greater than $V_{Min} - V_{TN}$, there is very little voltage across the stacked transistors, and, to get the current desired, it may be necessary to oversize them which in turn would affect how the driving circuitry was designed and sized. The solution to this problem is to use a smaller output swing than input swing. With V_{ON+} lower than the minimum output swing value, the source voltage of the the transistor being controlled

by V_{ON+} and stacked transistors will still be $V_{InMin} - V_{TN}$ or greater, but the voltage across the stacked transistors will now be increased to $V_{OutMin} - (V_{InMin} - V_{TN})$ instead of less than V_{TN}.

In this design, the input to the flip-flop comes from a $NAND$ gate. To design the $NAND$ gate, Figures 3.13 and 3.12 were used. The sizes of the devices in the wide swing $NAND$ gate are shown in Table 3.3.

Transistor	Width	Length
$PMOS$	0.2	0.26
$NMOS$ (Stacked - Top transistors)	0.26	0.13
$NMOS$ (Stacked - Bottom transistors)	0.26	0.13
$NMOS$ (Single transistor)	0.13	0.13
$NMOSSink$	2.08	1.04

Table 3.3: NAND device sizes for $550mV$ output swing.

The final challenge is the behavior at start-up. If the flip-flops are not in the reset state at start-up, the pulse generators will not work as desired. The pulse generators in this design are designed to receive a rising edge which starts a signal through a loop. This loop involves a chain of three component blocks as shown in Figure 3.11. The three components are a delay chain, a pulse generator, and a flip-flop. The signal propagates through this chain and as subsequent flip-flops are set, flip-flops that hold information no longer needed are reset. The signal which resets earlier flip-flops is generated when a flip-flop goes from its $RESET$ state to the SET state and if this transition does not occur no reset signal is generated. And, clearly, if a flip-flop is not in its $RESET$ state, it cannot transition to its SET state. That means that at start-up the flip-flops must be set to their $RESET$ state for the pulse generator to work. The reset signal which feeds back from the output, resets two of the three flip-flops as shown in Figure 3.11, but this signal is important in keeping t_p constant and needs to be as clean and steep as possible. Adverse effects to that signal can be minimized by adding an additional NOR gate to the least sensitive part of the pulse generation block, its internal delay chain. The important trait of the pulse generator block is the steepness of front edge of its pulse, not the width of the pulse. Adding the NOR gate to the delay chain does not effect the quality of the front edge of the pulse but allows

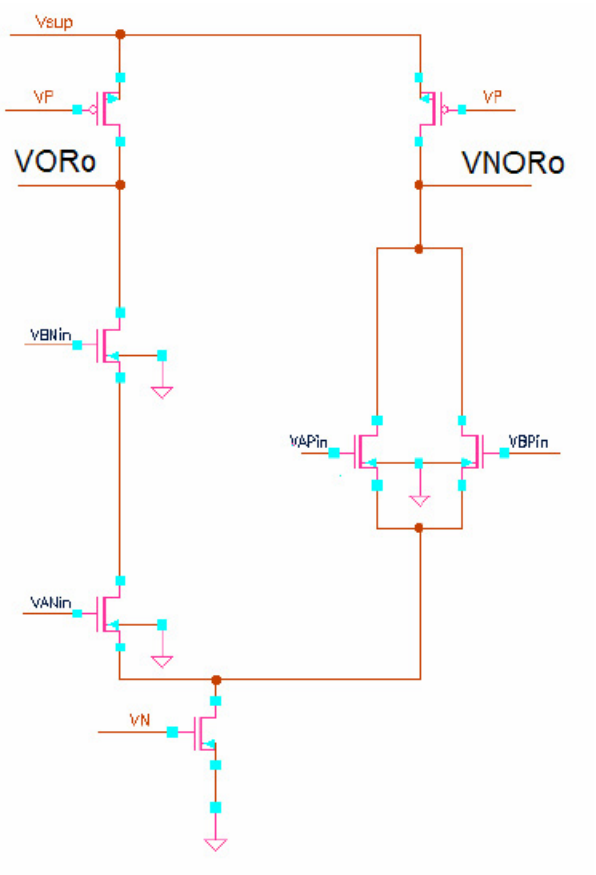


Figure 3.36: MCML NOR gate.

generation of a $RESET$ pulse. The signaling is shown in Figure 3.37 for Figure 3.26. The NOR gate (Figure 3.36) will OR an external $RESET$ signal with the signal going through the delay chain and produce a pulse with minimal disturbing the sensitive part of the signal.

A final note on operation; During hold phases all transistors except the pull-up transistor will be in their triode mode of operation. The pull-up PMOS will be in saturation if $V_{DD} - V_{InMin} - |V_{TP}| < V_{DD} - V_{OutMin}$, or, more simply, $V_{InMin} + |V_{TP}| > V_{OutMin}$. During transitions, the devices that pull-up or pull-down begin in saturation and transition into triode. The tail current source is diode connected and in saturation through all phases.

The sizes of the devices used in this simulation are listed in Table 3.4.

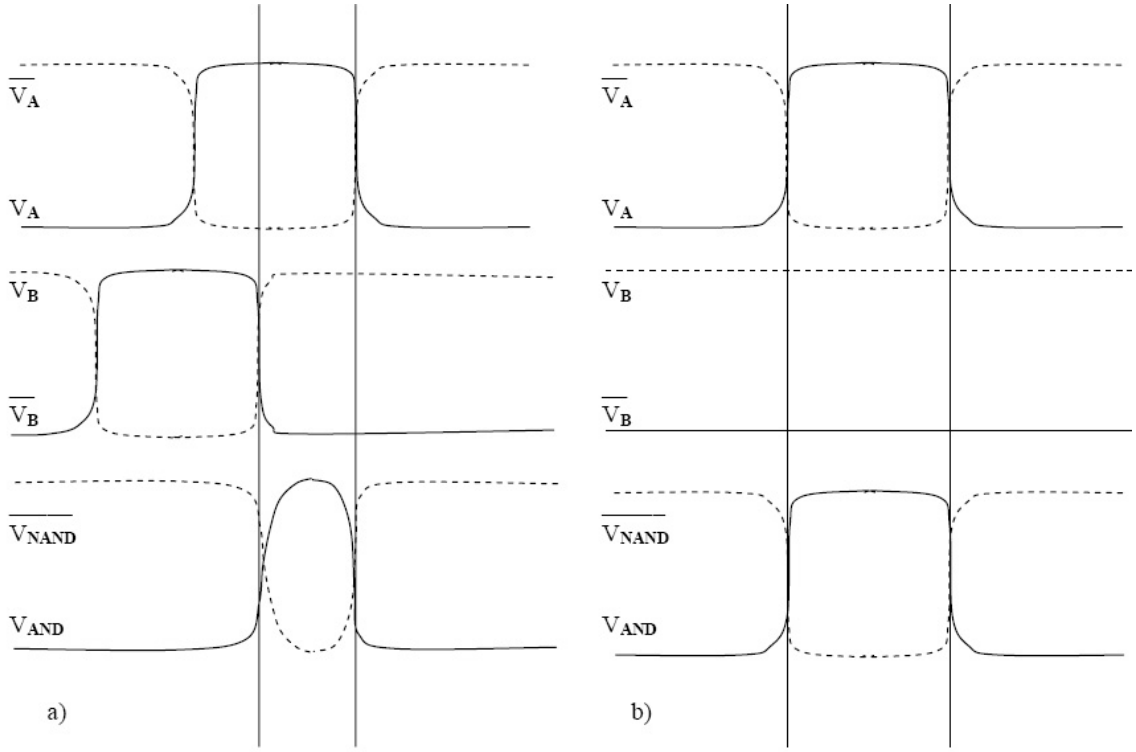


Figure 3.37: a: Usual pulse from delay chain. b: Pulse induced through *RESET* signal.

Transistor	Width	Length
PMOS (Hold)	0.3	0.13
PMOS (Pull-up)	0.6	0.13
NMOS (Hold - Top)	0.16	0.13
NMOS (Hold - Bottom)	0.15	0.13
NMOS (Pull-down)	1	0.13
NMOS (Sink)	2.08	1.04

Table 3.4: Latch device sizes.

3.4 Performance

In this section the performance of input monostable is examined through simulation. Intrinsic noise as well as supply and GND noise will be evaluated alone and with process variation. Monte Carlo simulations are done with and without supply, GND, and intrinsic noise, as well as worst case analysis. Simulations were done using ST Microelectronic $90nm$ technology and the Eldo simulator. GND and supply simulations were done with noise between $\pm 100mV$. Noise files were generated from a self-written tool which generates piecewise linear files for inclusion in Eldo decks. Intrinsic noise values were found using the $.NOISETRAN$ Eldo command which finds intrinsic noise during transient response simulations.

3.4.1 Process variation

Process variation did not adversely affect the performance of this block. It affected the width of t_p but the width stayed constant between cycles and therefore was not an issue. "Worst case" values were obtained by doing doing a statistical analysis, a Monte Carlo with worst case reporting and all parameter values set to be independently assigned on a normal distribution. With all devices allowed to move $\pm 10nm$, t_p varied between $770ps$ to $800ps$. Repeating the simulation with a $\pm 10\%$ variation allowed on, again, all widths and length of the devices in the design gave a value of t_p between $780ps$ and $820ps$. This is duty cycle appropriate for frequencies at about $1GHz$ and below.

3.4.2 Intrinsic Noise

Intrinsic noise was analyzed during transient response simulations using Eldo (Mentor Graphics). As shown in Figure 3.38, the output of the differential pair with the lower value experiences more intrinsic noise. This makes sense since that there is current flowing through the node as it is held low which will increase flicker and shot noise. Also, thermal noise will be greater because g_m of a conducting $PMOS$ will clearly be greater than the pull-up device which will be in cutoff. Also, note that there is a larger noise spike on

the high-to-low transitions, where the circuit, as it tries to pull-down the node's value, momentarily conducts a larger current than the side pulling-up a node.

The noise values in Figure 3.38 are *rms* values. The average *rms* value for the lower output of the differential pair is, excluding transitions, is approximately $2.85mVrms$. The average high value is approximately $0.89mVrms$. The intrinsic noise can be attributed in part to the small devices used in this design, and the number of devices chained together. Using larger devices with higher currents would reduce this noise. But, as discussed in earlier sections, the noise on the transitions is much more important. The rising slope of the transition is approximately $5.81 \times 10^9 V/s$, and the falling slope is approximately $10.89 \times 10^9 V/s$. The maximum noise on a falling transition is approximately $25mV$, and a rising transition $18mV$. This means that an error of $0.11ps$ of jitter could be introduced on to the signal from intrinsic noise. This calculation is shown in Equation 3.23.

$$\Delta t_p = sqrt2 \frac{43mVrms}{(5.81 \times 10^9 V/s + 10.89 \times 10^9 V/s)} \approx 2.54pspk - pk \tag{3.23}$$

3.4.3 Supply Noise

Noise from the supply has different effects on an output of the differential pair depending on whether the output is high, low, or transitioning. When an output node is high, there is relatively little current flowing through it and it is tied to the supply through a triode or cutoff region $PMOS$. As discussed in the section on differential pair sizing, the larger the size of the $PMOS$, the more the noise is filtered. Figure 3.39 shows a slight filtering of a "high" output of a differential pair. The "low" output signal experiences the same filtering effect, but also has the noise reduced by the voltage divider formed by the $PMOS$ and the $NMOS$ switch and tail current source. Figure 3.40 shows this reduction due to voltage division. Though the noise on a "high" output can be somewhat reduced by making the $PMOS$ of the circuit larger, the reduction due to voltage division is not as easily taken advantage of. If the output "low" voltage is a value that can be changed freely, the closer the output "low" voltage is to GND (when there is no noise on GND), the less noise will

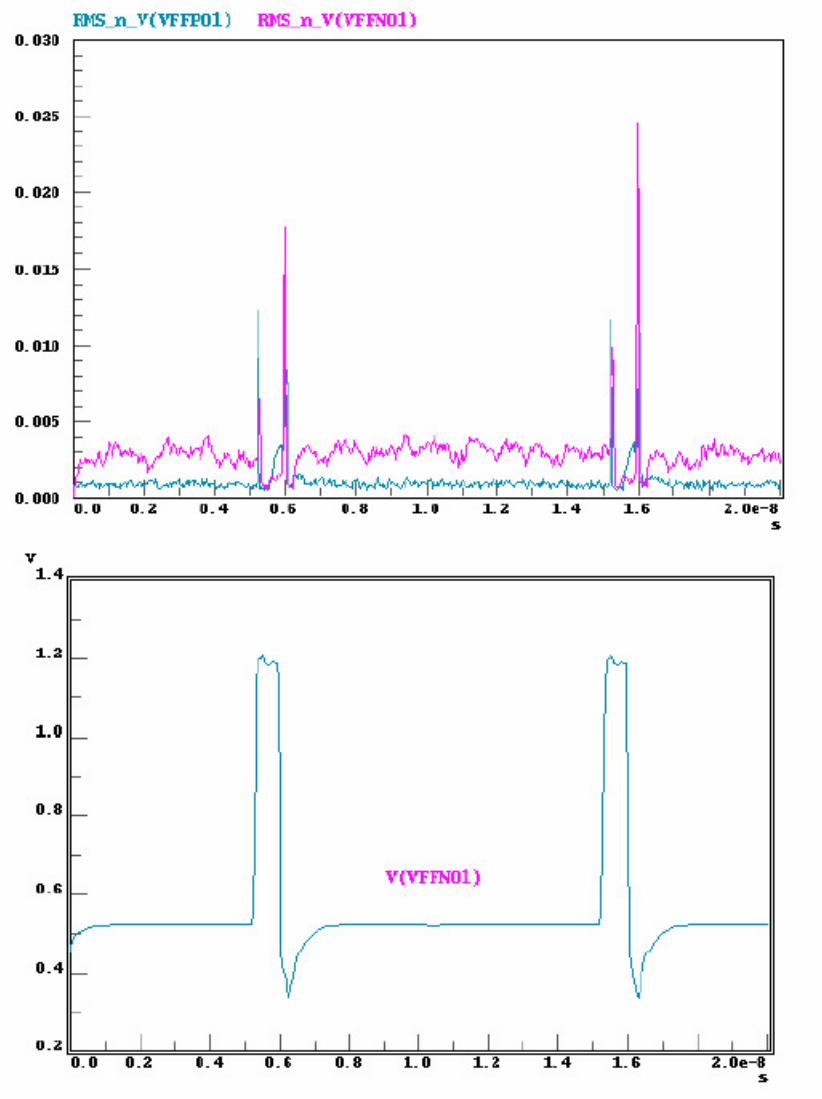


Figure 3.38: Intrinsic noise of input monostable block.

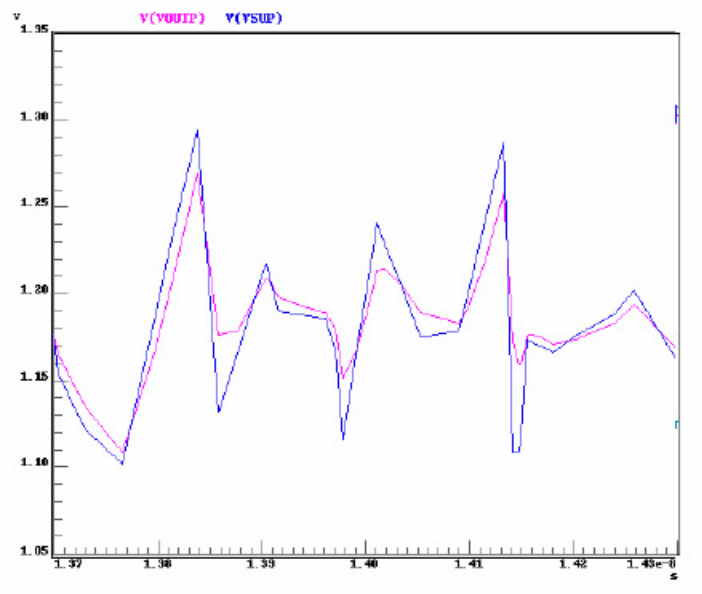


Figure 3.39: "High" output of differential pair is slightly filtered by device capacitance.

be seen on the single-ended output. An important point to realize here is that it is not advantageous to remove the noise from only one output of the differential pair. Differential pairs strength in noisy environments is its ability to remove common mode noise. If noise is removed from one output, conversely, there will be more noise seen on the differential output. It is, therefore, advantageous to have the "low" output as high as possible so that it sees as much of the noise as possible. Reducing the difference between the "high" output and the "low" output has its trade offs though. First, the output of one inverter is the input to the next. If the outputs are near one another, the inverter being controlled by the signals will leak more current through its "off" switch than it would if there was a larger swing between inputs. Also, a smaller swing may be more susceptible to noise than a larger swing. A swing of at least one V_T was selected for this design, and two V_T was designed for. This was used to guarantee the "OFF" switch was relatively strongly off, and the maximum slope was attained on the output. Care was taken to keep the swing at the minimum value which sufficiently kept the "OFF" switch off.

Assuming the maximum differential output noise on the transition is less than or equal to the maximum noise when the output signals are not transitioning (*i.e.* where the common

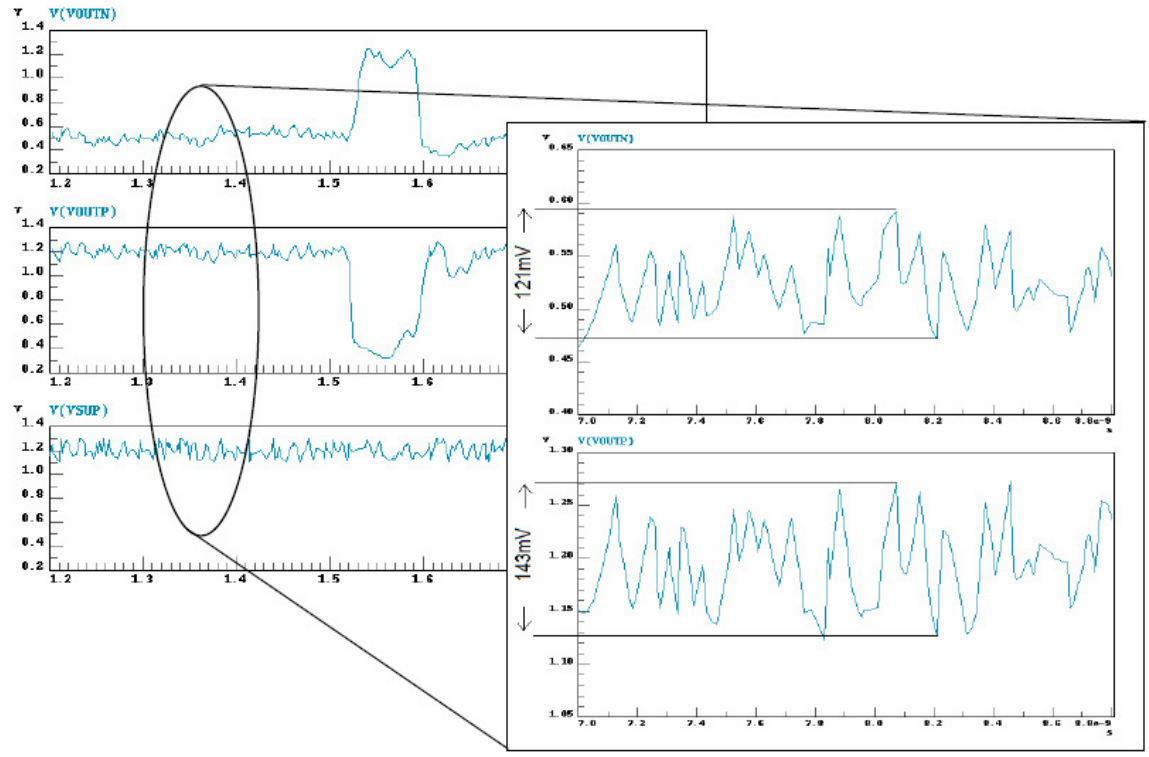


Figure 3.40: Supply noise division on “low” output of differential pair.

mode rejection is worst), the worst case t_p variation can again be calculated. The worst case differential noise on the non-transitioning signals was approximately $126mV$, and, again, the rising slope of the transition is approximately $5.81 \times 10^9 V/s$, and the falling slope is approximately $10.89 \times 10^9 V/s$. This gives an approximate maximum variation on t_p of $7.54ps$. (The simulations shown here have uniformly distributed noise with both duration and amplitude varying. The minimum width noise is of a duration of $1ps$ and maximum, $50ps$. The voltage limits are $\pm 100mV$ around an average value of $1.2V$).

$$\Delta t_p = \frac{0.126V}{(5.81 \times 10^9 V/s + 10.89 \times 10^9 V/s)} \approx 7.54ps \tag{3.24}$$

3.4.4 "GND" Noise

Original calculations done on the input monostable block only looked at supply noise. This was done assuming that differentially, at least, increasing the supply noise was equivalent to adding GND bounce. Realization that, just as supply noise effected the "high" output in isolation, *i.e.* the switch to GND, was "OFF", inspired some recalculations and simulations done separately on supply noise and GND bounce.

In the case of supply noise, the "high" output was influenced in relative isolation by the supply noise. In the case of GND bounce, the "high" value is relatively protected, but the "low" value is vulnerable. Theoretically this should cause a relatively larger error on the output than supply noise. Assuming the maximum differential output noise on the transition is less than or equal to the maximum noise when the output signals are not transitioning (*i.e.* where the common mode rejection is worst), the worst case t_p variation can again be calculated. The worst case differential noise on the non-transitioning signals was approximately $54mV$, and, again, the rising slope of the transition is approximately $5.81 \times 10^9 V/s$, and the falling slope is approximately $10.89 \times 10^9 V/s$. This gives an approximate maximum variation on t_p of $3.23ps$. (The simulations shown here have uniformly distributed noise with both duration and amplitude varying. The minimum width noise

is of a duration of $1ps$ and maximum, $50ps$. The voltage limits are $\pm 100mV$ around an average value of $0V$).

$$\Delta t_p = \frac{0.054V}{(5.81 \times 10^9 V/s + 10.89 \times 10^9 V/s)} \approx 3.23ps \tag{3.25}$$

3.4.5 Supply and "GND" Noise

The stimulation specifications used in the individual supply and GND noise simulations were then used together to simulate a circuit experiencing both supply and GND noise at the same time. The worst case differential noise on the non-transitioning signals was approximately $195.2mV$, and this gives an approximate maximum variation on t_p of $11.69ps$.

$$\Delta t_p = \frac{0.1952V}{(5.81 \times 10^9 V/s + 10.89 \times 10^9 V/s)} \approx 11.69ps \tag{3.26}$$

3.4.6 Intrinsic, Supply and "GND" Noise

The stimulation specifications used in the individual supply and GND noise simulations were then used together to simulate a circuit experiencing both supply, GND and intrinsic noise at the same time. The worst case differential noise on the non-transitioning signals was approximately $195.2mV$, and this gives an approximate maximum variation on t_p of $11.69ps$.

$$\Delta t_p = \frac{0.1952V}{(5.81 \times 10^9 V/s + 10.89 \times 10^9 V/s)} \approx 11.69ps \tag{3.27}$$

3.4.7 Analysis of Results

The simulations in this section are summarized in Table 3.5. The GND and supply noise was uniformly distributed over $0V \pm 100mV$ and $1.2V \pm 100mV$ respectively. The design simulated is that shown in Figure 3.11, and the size of the inverters, $NAND$ gates and latches are shown in Tables 3.1, 3.3, and 3.4 respectively. The only exception is the

$2x$ and $4x$ inverters used to increase drive of signals controlling the $NAND$ gates. Those inverters have their widths multiplied by $2x$ and $4x$ respectively.

Intrinsic	Supply	GND	t_p variation (pk-pk)	t_p variation (rms)
			$0ps$	$0ps$
√			$2.54ps$	$1.8ps$
	√		$7.54ps$	$5.33ps$
		√	$3.23ps$	$2.28ps$
	√	√	$11.69ps$	$8.27ps$
√	√	√	$14.23ps$ (approximation)	$10.62ps$

Table 3.5: t_p variation due to various noise sources.

There were two observances of glitching that could effect the performance of the input monostable. Both occurred when the noise on the supply stayed near the upper limit, $1.2V + 100mV$, for some amount of time more than $60ps$, and the GND value is near average or less. The increase in current brought on by the increased V_{GS} of the $PMOS$ requires the switches to conduct more. As the source voltage of the switch transistor on the "ON" side drops to accommodate the increased current requirement, the V_{GS} of the switch transistor on the "OFF" side also increases and eventually turns on. The result is a drop in the pulled-up node's voltage value. Values down to half of the swing were observed for the sizes used.

As discussed in Chapter 2, one component of the jitter on the output of the entire JAC block is the t_p variation from this block multiplied by a value determined by the output slopes of the subsequent "Sawtooth Generator" block:

$$\Delta t_p = \Delta t_{pi} \times (1 + \frac{m1}{m2}) \tag{3.28}$$

Larger devices, higher currents, and biasing circuitry that is less sensitive to supply, GND, and intrinsic noise would reduce the variation on t_p, but the sizes selected for the monostable being discussed here were chosen to be "middle-of-the-road", in other words, the design was not optimized for power or jitter, but designed to address both aspects equally. The power dissipated in the input monostable is approximately $3.89mW$, a value four-times the original design goal. Current was increased and larger devices were used

when initial noise figures were above $20ps$ of jitter. Further sizing and current adjustment would continue to decrease the t_p error, but would come at a power cost. The sizing used here was selected to give a total JAC power figure under $8mW$ for the three main blocks, a design constraint that also constrained the jitter performance.

3.5 Summary

This Chapter introduced the input monostable block of the JAC. Starting from the block level, the function of each component of the input monostable was described. Transistor level descriptions of each component allowed discussion of important performance and design considerations. Simulations of the input monostable were done with intrinsic, supply, and GND noise. The supply and GND noise varied independently by $\pm 100mV$ around their nominal value of $1.2V$ and $0V$ respectively. A $780ps$ pulse width was generated which is appropriate for low-jitter a $1GHz$ signals (78% duty cycle) to a $640MHz$ (50% duty cycle). For large changes in the length of the pulse width, removal of a delay chain - pulse generator - flip-flop section from the block diagram shown in Figure 3.11 will halve the width of the output pulse, and reducing or increasing the length of the delay chains in that same diagram will accordingly reduce or increase the pulse width. For smaller adjustments, the $PMOS$ gate voltage can be raised or lowered to increase or decrease the delay through the existing delay chains. The worst case t_p variation for a pulse width of $780ps$ was approximated at $12.75ps$ using these simulation results. Higher frequencies will require a shorter pulse and therefore have proportionally less t_p variation. For example, tripling the targetted frequency would reduce the pulse width and t_p variation by $\frac{1}{3}$. A 50% duty cycle at $1GHz$ would require $\frac{2}{3}$ of t_p variation or approximately $8ps$. The t_p variation is multiplied by a value determined by the output slopes of the sawtooth generator block and comprises one part of the JAC output jitter. Jitter can be reduced further in this block, but at a power consumption cost. Adding the design constraint for the entire JAC of $8mW$ limits the amount that the t_p variation can be reduced. This is because reduction methods include sizing up of components which, with increased currents, add to power consumption.

Chapter 4

Integrator and Output Pulse Generation Block

4.1 Introduction

This section examines two blocks; The central block to the JAC, the integrator or sawtooth generator block, and the block responsible for taking the output of the integrator block and turning it into a jitter free waveform, the output pulse generation block. The sawtooth generator block is responsible for generating a differential sawtooth wave which follows the basic equations $m1 \times t_p + m2 \times (T - t_p) = 0$, where $m1$ and $m2$ are the slopes of the two legs of the sawtooth wave, T is the period of the target frequency, and t_p is the length of one leg or segment of the sawtooth wave. The crossing of the "non-t_p" length leg determines edges of the output of the entire JAC. The job of the output pulse generation block is to accurately and clearly identify the crossing of the $m2$ sloped segments, and generate the output waveform. This chapter describes the design and performance issues controlling of these two blocks.

4.2 The Integrator Block

This section gives a high-level description of the integrator block, and then discusses design decisions.

4.2.1 Block Level Design

As described in Chapter 2, the integrator circuit can be modeled as shown in Figure 4.1a. The switches of the integrator are controlled by the differential outputs of the input monostable, and therefore, ideally, only one switch will be open at a time. Opening and closing a switch steers the current on and off of a the integrators. When a switch is open and not conducting, all of the current from the I_{source} of the same side of the integrator is delivered to the integrator, which, as shown in Figure 4.1b, is implemented simply with a capacitor. Figure 4.2 shows an alternative single capacitor implementation. The design in Figure 4.1 was used in the final JAC tested in this thesis. The single capacitor design has the advantage of needing only a single capacitor of half the value of the capacitors in the two-capacitor design. It also should improve noise performance in the presence of GND bounce. As noted in the discussion of the input monostable, when a switch is off, the output on that side does not see noise on the GND supply, but the other side, the side with the ON switch sees it directly. By having a capacitor between output nodes, the noise from the ON switch side is also communicated to the OFF switch side improving common mode agreement. Both capacitor architectures function in the same way. As the current is steered on to the capacitors, the voltage rises at a slope equal to:

$$m_{+/-} = \frac{I_{source+/-}}{C^{+/-}} \tag{4.1}$$

When the switch is closed, a current of $I_{source+/-}$ is still coming from the current source, but there will also be a current equal to I_{Sink} being pulled off the $V_{OUT+/-}$ node. The net

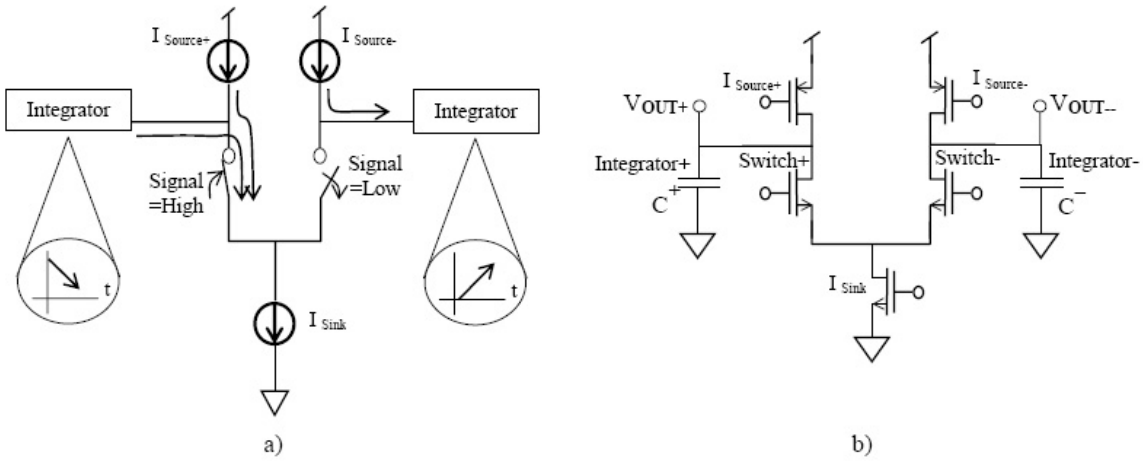


Figure 4.1: a. Integrator circuit model. b. Integrator circuit implementation.

current to the integrating capacitor will therefore be the sum of the source current and the sink current, $I_{source+/-} - I_{Sink}$ and the slope of the voltage at the output node will be:

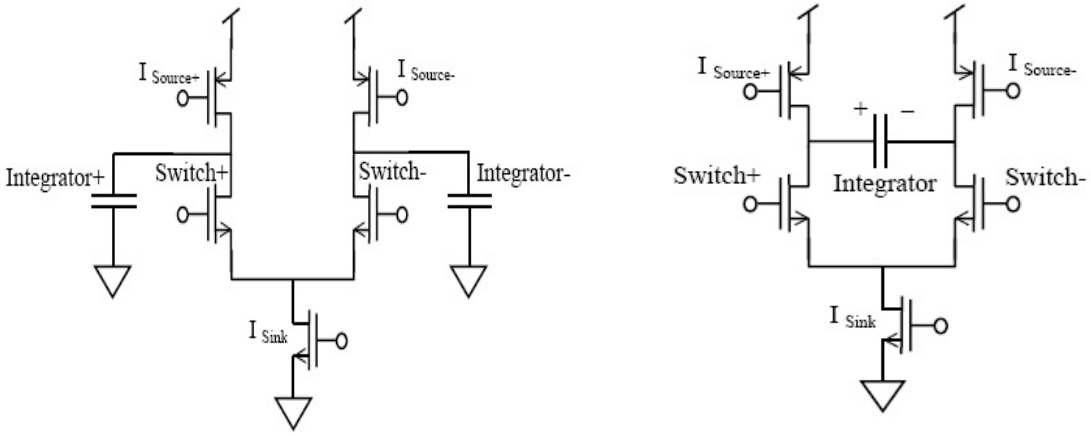
$$m_{+/-} = \frac{I_{source+/-} - I_{Sink}}{C^{+/-}} \tag{4.2}$$

In this design I_{Sink} is greater than $I_{source+/-}$ and causes the slope in Equation 4.2 will be negative. The biasing of the current sources, $I_{source+}$, $I_{source-}$, and I_{sink} will be described in Chapter 5, and will not be discussed here, but, as a reminder, these slopes need to be the $m1$ and $m2$ in the basic equations $m1 \times t_p + m2 \times (T - t_p) = 0$ and $m2 \times t_p + m1 \times (T - t_p) = 0$.

As noted, the charging and discharging of the capacitance on the output node is what controls the output voltage of the integrator. The capacitances that exists on the output node are the capacitance from the integrator circuit transistor's themselves, the load capacitance contributed by the circuit being driven by the output signal, and the ideal capacitance added to slow the swing to the slope desired.

4.2.2 Circuit Design

Figure 4.1b shows the implementation of the integrator block. The challenge in implementing this block is keeping the $I_{source+}$, $I_{source-}$, and I_{sink} generating devices in saturation



(a) Capacitors to GND

(b) Single capacitors between outputs

Figure 4.2: Capacitor placement options for integrator.

and acting as close as possible to perfect current sources. Making devices larger makes them less susceptible to V_{DS} variations and intrinsic noise, but load the driving circuit and slow switching of the integrator. In Chapter 2 approximations for both the error introduced during switching and the error due to channel length modulation were proposed but both were just that, approximations, and, as such, using them to find an accurate, quantitative comparison of the importance of each of these errors is difficult. Though accurate quantitative comparison of design decisions is difficult, trends can be identified and used for guidance. Table 4.1 lists up some of the design trade offs and trends.

Design strategy	Positive effect	Negative effect
Higher current through integrator	Less sensitivity to intrinsic noise	1. Larger power budget 2. Longer width switching transistors needed
Longer channel devices	Less sensitivity to V_{DS} (Channel length modulation)	Larger load to driving circuit - slower switching
Stronger driver	Shorter switching time	Larger power budget

Table 4.1: Design trade offs for integrator.

The design strategy taken here is to increase channel lengths of all devices and take advantage the items in the "Positive effect" column of Table 4.1. As also listed in Table 4.1,

this will require a stronger driver. To achieve this, the last gate in the input monostable can be sized-up with no effect to the rest of the monostable. Though not done in this design, capacitance can be added between the gate and the source of the source devices to keep the current sources' gate-source voltages as constant as possible.

The second item in Table 4.1 is probably the biggest reason for the decision to design using larger, long-channel devices. Because the drain voltage of the $I_{source+}$, $I_{source-}$, and switches the sawtooth output voltage, variation in V_{DS} is unavoidable. After various attempts to work with short-channel devices and somehow 'deal' with the channel-length modulation effects, the effort to design this block using small devices was abandon and saner heads prevailed. By using long-channel devices, the current variation due to the varying V_{DS} was effortlessly reduced to a manageable level. Channel-length modulation is represented by the channel-length modulation coefficient, λ, in the first-order long-channel device equation (Equation 4.3) for devices in their saturation region.

$$I_{DS} = \frac{\mu C_{OX}}{2}\frac{W}{L}(|V_{GS}| - |V_T|)^2(1 + \lambda V_{DS}) \tag{4.3}$$

The channel-length modulation coefficient is directly related to the effective channel length, and the effective channel length varies with V_{DS}. Rewriting the effective channel length as L', then

$$L' = L - \Delta L, \tag{4.4}$$

a first order relationship between λ, V_{DS}, and the channel lengths can be derived ([*Razavi*, 2001], [*Howe and Sodini*, 1997]):

$$\frac{\Delta L}{L} = \lambda V_{DS}. \tag{4.5}$$

The channel-length modulation coefficient describes the relative variation in length for a given V_{DS} and becomes smaller for larger L values. Taking the partial derivative of Equation 4.3 with respect to V_{DS}, gives an equation for the slope of ΔV_{DS} vs ΔI_{DS}, Equation 4.6.

This makes it clear that the smaller λ is, the more constant the current will be when V_{DS} varies. The most straight forward way to reduce the effects of channel-length modulation is, therefore, to increase the channel length of the PMOS devices, though, from Equation 4.6 it is clear that making $|V_{GS}| - |V_{TP}|$ small is also an option.

$$\frac{\partial I_{DS}}{\partial V_{DS}} = \lambda \frac{\mu C_{OX}}{2} \frac{W}{L} (V_{GS} - V_{Tp})^2 \tag{4.6}$$

The region of operation of the source devices is also important. For them to behave as current sources, they must be solidly in their saturation region. For a device to be operating in its saturation mode, the voltage across the gate and source (V_{GS}) and drain and source (V_{DS}) must be satisfy the equations:

$$\begin{aligned} V_{GS} &> V_T \\ &and \\ V_{GS} - V_T &< V_{DS}. \end{aligned} \tag{4.7}$$

For $I_{source+}$ and $I_{source-}$ that means that:

$$\begin{aligned} V_{DD} - V_G - |V_{Tp}| &< V_{DD} - V_{OUT} \\ &or \\ V_G + |V_{Tp}| &> V_{OUT}. \end{aligned} \tag{4.8}$$

This defines the absolute minimum that V_G can be and defines an important constraint on the output swing of the integrator. It is desirable to have a V_G that is not right at the bare minimum value due to the fact that V_{DS} begins to have more of an affect on a device's behavior as the device nears the saturation-triode region border, which, in this case, is where

$$V_G + |V_{Tp}| = V_{OUT}. \tag{4.9}$$

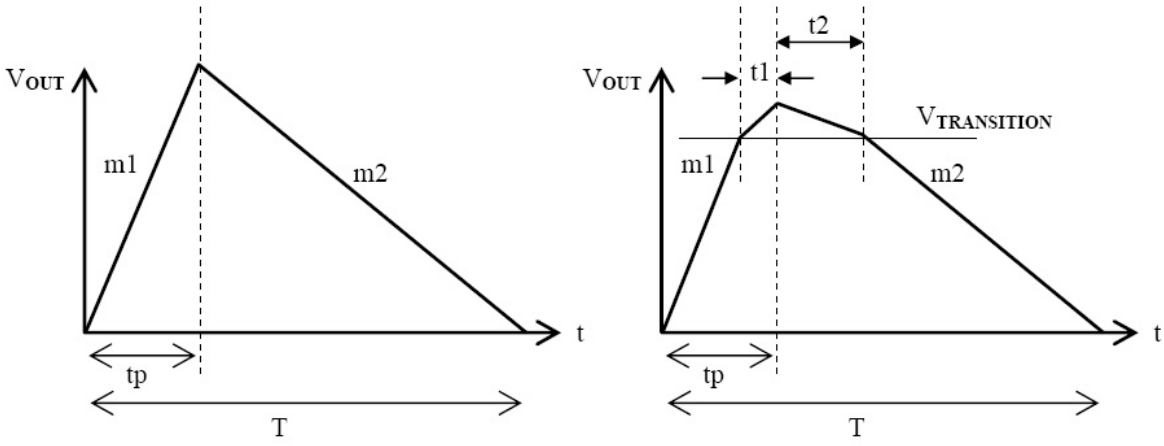


Figure 4.3: Complete cycle of output voltages under constant currents of I_{Source} and $I_{Source} - I_{Sink}$.

There is one case where the PMOS supplying I_{Source} can go into triode and the JAC will still perform correctly; That is when a 50% duty cycle is being used and exactly the same amount of time is spent in triode during the rising and falling of the output. If the duty cycle is not equal to 50%, as is suggested as optimal for this system, there will not be equal time in triode during the rising and falling of the output and the amount of charge delivered to the integrating capacitors will not be equal.

Though it is clear that the PMOS supplying I_{Source} cannot go into triode and still perform as a current source unless the system is using a 50% duty cycle, it is still informative to examine the effect of capacitance on the output when the sources pass through both saturation and triode regions. To make the illustration clearer, constant sink and source currents will be used in this example. The capacitance at the node where integration occurs determines the slope of the output voltage and is important in the correct operation of the JAC. The capacitances which are important to the slope of the output are C_{GD} and C_{DB}. In saturation and in cutoff the C_{GD} capacitance contributed to the output node by the I_{source} PMOS is equal to only the overlap capacitance, but in triode, there is an additional capacitance of $C_{OX}WL_{eff}/2$, where C_{OX} is the gate capacitance per unit area of the device. A complete output cycle is shown in Figure 4.3.

Equations to describe the capacitive effects on the output can now be constructed. $V_{TRANSITION}$ is the voltage where the PMOS current sources enter triode.

$$\begin{aligned}
\frac{I_{Source}}{C_1}(t_p - t_1) &= V_{TRANSITION} \\
\frac{I_{Source}}{C_1 + C_2}t_1 &= V_{Max} \\
V_{Max} + \frac{I_{Source} - I_{Sink}}{C_1 + C_2}t_2 &= V_{TRANSITION} \\
V_{TRANSITION} + \frac{I_{Source} - I_{Sink}}{C_1}(T - t_P - t_2) &= V_{ERROR} \\
\frac{I_{Source}}{C_1}t_p + \frac{I_{Source} - I_{Sink}}{C_1}(T - t_P) &= 0
\end{aligned} \tag{4.10}$$

Solving this system of equations results in V_{ERROR} being equal to zero. In other words, if the current is constant through the rising part of the cycle and also, possibly at a different value, constant through the falling part of the cycle, the capacitance change caused by entering and leaving the triode region of operation does not change the ending point of the cycle. To clarify, just as the integrator had done with constant capacitance, the output returned to the same voltage as it started at. This fact is useful when it is noted that the switching transistors, unlike the sources, must be allowed to go into triode to accommodate the minimum output swing voltage, and, as with the example above, will contribute a varying amount of capacitance to the output node over a single output cycle. As in the example, the capacitance will not upset the results of the integrator, but care should be taken, if possible, to make sure the differential signals cross while devices are in saturation so as to take advantage of the steeper signal slope. Because the gate voltage to the switching devices is V_{DD} when conducting, the switching devices will go into triode when

$$\begin{aligned}
V_{DD} - V_N - V_{Tn} &> V_{OUT} - V_N \\
&orwhen \\
V_{DD} - V_{Tn} &> V_{OUT}.
\end{aligned} \tag{4.11}$$

The farther the switching device goes into triode, the larger the device needs to be to to conduct the current it needs to conduct. A trade off needs to be made between a larger swing, which has the advantage of steeper slopes, and a large input capacitance as the switch transistor is made larger to make the large swing possible. Note that the current through the switch is determined by the tail current source and is equal to I_{Sink}.

The final transistor in the integrator circuit is the NMOS used in the tail current source. The function of this transistor is to supply a constant, noise free current sink. As with the PMOS devices which generate $I_{Source+}$ and $I_{Source-}$, a long-channel device was used not only to reduce sensitivity to variations in V_{DS}, but also to reduce sensitivity to GND bounce and intrinsic noise.

Sizing of the capacitors depend on the jitter on the input and the width of the output swing. As noted in Chapter 2, the equations that define constraints for the rising edge of the negative signal are:

$$|m1| < P_{sTOLERANCE} \times (\frac{V_{sLIMITMAX} - V_{sCENTER}}{t_{jitter} + \frac{1}{2}t_p}) \tag{4.12}$$

The equation for the falling edge of the negative signal is:

$$|m2| < P_{sTOLERANCE} \times (\frac{V_{sLIMITMAX} - V_{sCENTER}}{t_{jitter} + \frac{1}{2}(T - t_p)}) \tag{4.13}$$

The the equations that define constraints for the falling edge of the positive signal are:

$$|m2| < P_{sTOLERANCE} \times (\frac{V_{sCENTER} - V_{sLIMITMIN}}{t_{jitter} + \frac{1}{2}(T - t_p)}) \tag{4.14}$$

And the equation that is valid for the rising edge of the positive signal is:

$$|m1| < P_{sTOLERANCE} \times (\frac{V_{sCENTER} - V_{sLIMITMIN}}{t_{jitter} + \frac{1}{2}t_p}) \tag{4.15}$$

Where $V_{sCENTER}$ is the value the output swings around, $P_{sTOLERANCE}$ is $1 - Toleranceofdevice$, $V_{sLIMITMIN}$ is the lower limit of the output swing, $V_{sLIMITMAX}$ is

the upper limit of the output swing, and t_{jitter} is the amount of jitter on the signal at the output of the integrator block. Also, as shown in Equations 4.1 and 4.2, $m1$ and $m2$ are calculated from the current onto the capacitor divided by the capacitance of the capacitor. The value of this capacitance is equal to the intentionally added load capacitance, the capacitance on the integrator transistors, and the capacitance of the circuit that the integrator is driving. The intentionally added capacitance is controllable, and the capacitance contributed by the current source and switch capacitors does not affect the crossing point (See Equations 4.10 and Figure 4.3). Capacitance contributed by the circuit driven by the integrator, though, cannot be ignored. It will be described in detail in the level detector section of this chapter, but using a level shifter as the first stage of the level detector, the load capacitance can be held relatively constant. The gate voltage of the input transistor of the level shifter is the output of the integrator, and that input transistor is kept in saturation. Keeping the input transistor in saturation means that gate-drain and gate-bulk capacitance can essentially be ignored and the load capacitance to the integrator stays relatively constant.

4.3 The Output Generation Block

The function of the final stage of the JAC is to use the output of the sawtooth generator block to generate an output waveform appropriate to the system being designed. Depending on the system the JAC is being used in, the JAC output requirements may be different. For example, the output may need to:

- Be differential output or a single ended
- Have a duty cycle of 50% or the duty cycle may not be important
- Have a well controlled output swing or there may just be a lower limit

The attributes listed above are specific to particular systems and not common to all JAC systems. The part of this output block that is common to all JACs is the level sensing

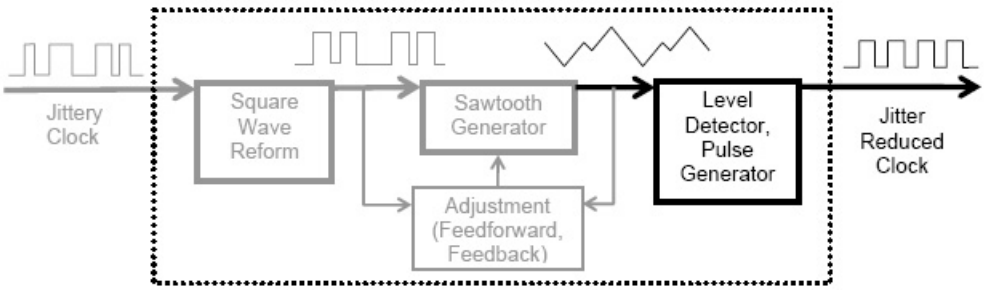


Figure 4.4: JAC block diagram.

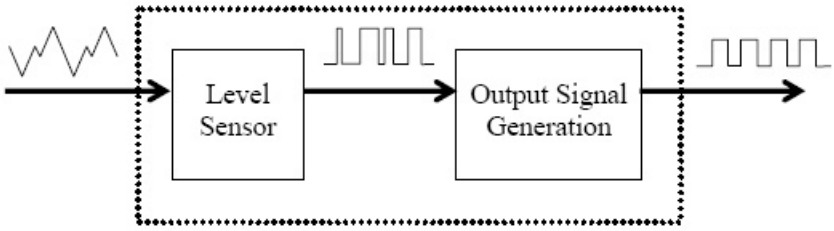


Figure 4.5: Output block block diagram.

circuitry which senses when the differential output signals of the sawtooth generator block cross. Because the level sensing circuitry is common to all JAC systems, this chapter will concentrate on its design and leave the system-specific parts for future theses.

The output generation block is shown in Figure 4.4. It stands alone in that it takes input from other blocks in the system but it is not in a feedback path and does not supply input to other JAC blocks.

4.3.1 Block Level Design

The block consists of two main blocks; The level sensor and the part of the block unique to the application using the JAC. This is shown in Figure 4.5.

As noted in Chapter 3, the relevant parameters in determining the jitter introduced by

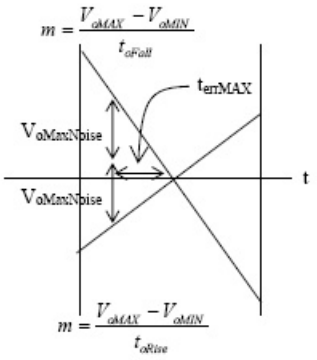


Figure 4.6: Earliest possible point where crossing may be sensed due to noise on signal.

the level sensing block are are the slopes of level sensor's outputs and the amount of noise on those signals.

$$t_{oJitter} = \frac{2V_{oNoise}}{(V_{oMAX} - V_{oMIN})\left(\frac{1}{t_{oRise}} + \frac{1}{t_{oRise}}\right)} \tag{4.16}$$

It is clear from Equation 4.16 that the steeper the slope of the output of the level-sensing block, the less jitter will be introduced by noise. There are any number of ways to implement the high-gain comparator needed here, and some may be preferable to others depending on what type of output signal generation circuitry follows the level sensing block, but, though more complicated circuits may give better gain and possibly some other advantages, the comparator used here follows the "simple is best" theme followed throughout the rest of this circuit's design.

When watching for the crossing point of the sawtooth generator outputs, there is the chance that noise may cause two crossings where there only should be one. Figure 4.7 shows such a case. To avoid double crossings, or, in other words, provide noise immunity, a Schmitt (or Schmidt) trigger can be used.

Schmitt triggers reduce noise sensitivity by having a different threshold for a low-to-high transition and a high-to-low transition. Various Schmitt triggers were examined ([*Chen and Ker*, 2005], [*Wang*, 1991], [*Dejhan et al.*, 2004], [*Zhang et al.*, 2003], [*Al-Sarawi*, 2002], [*Pfister*, 1992], etc), but in order to minimally load the driving circuit, the integrator block,

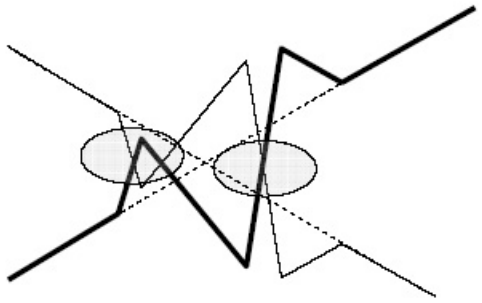


Figure 4.7: The dotted lines represent an ideal crossing. The solid lines show a noisy (double) crossing.

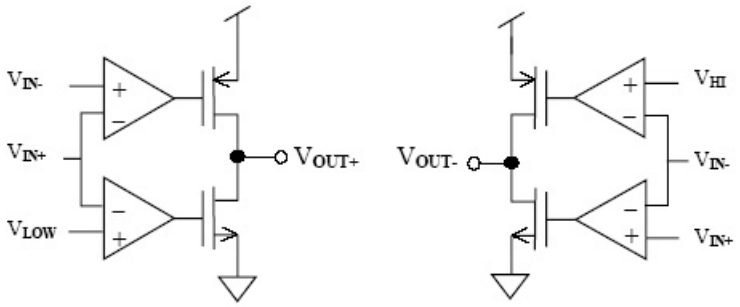


Figure 4.8: Schmitt trigger high-level diagram.

and give a strong, steep slope on its output, a design inspired by [*Pedroni*, 2005] was decided upon. Figure 4.8 shows a high-level representation, and Figures 4.9 and 4.10 show the behavior of the Schmitt trigger used in the design.

The final circuit used in this JAC design includes an input level shifting stage (Figure 4.11) to reduce loading of the integrator block and to provide a mechanism to control V_{HI} and V_{LOW}. This version, $M7$ and $M8$ are large and were added to absorb some of the supply noise. Using $M7$ and $M8$ restricts the usable input values and were removed in versions where a larger input value swing was being observed. Control of V_{HI} and V_{LOW} can be done by either adjusting the size of V_P or by adjusting the size of the $M1$. Both adjust the current through, and therefore the V_{GS} of the transistors that take the input signals, V_{SigN} and V_{SigP}, $M5$ and $M6$. Care must be taken that all devices stay in saturation. $M5$ and $M6$ must stay in saturation to keep the shift, V_{GS}, constant. Short-channel devices

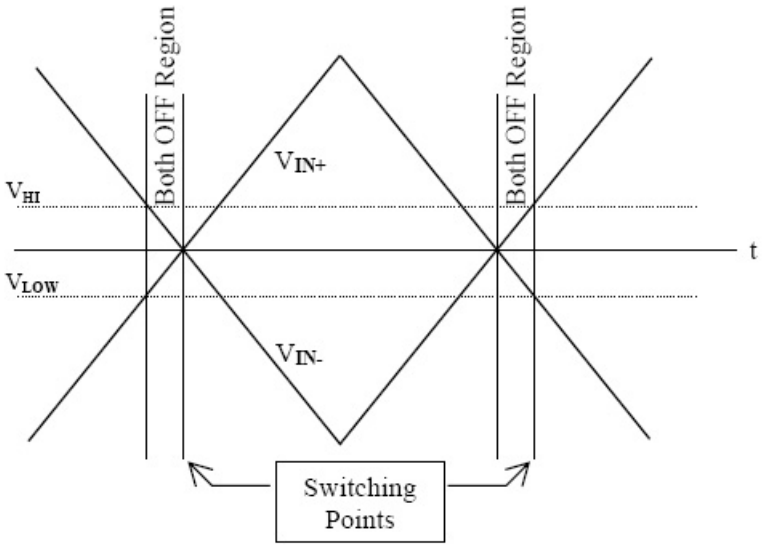


Figure 4.9: Schmitt trigger behavior diagram.

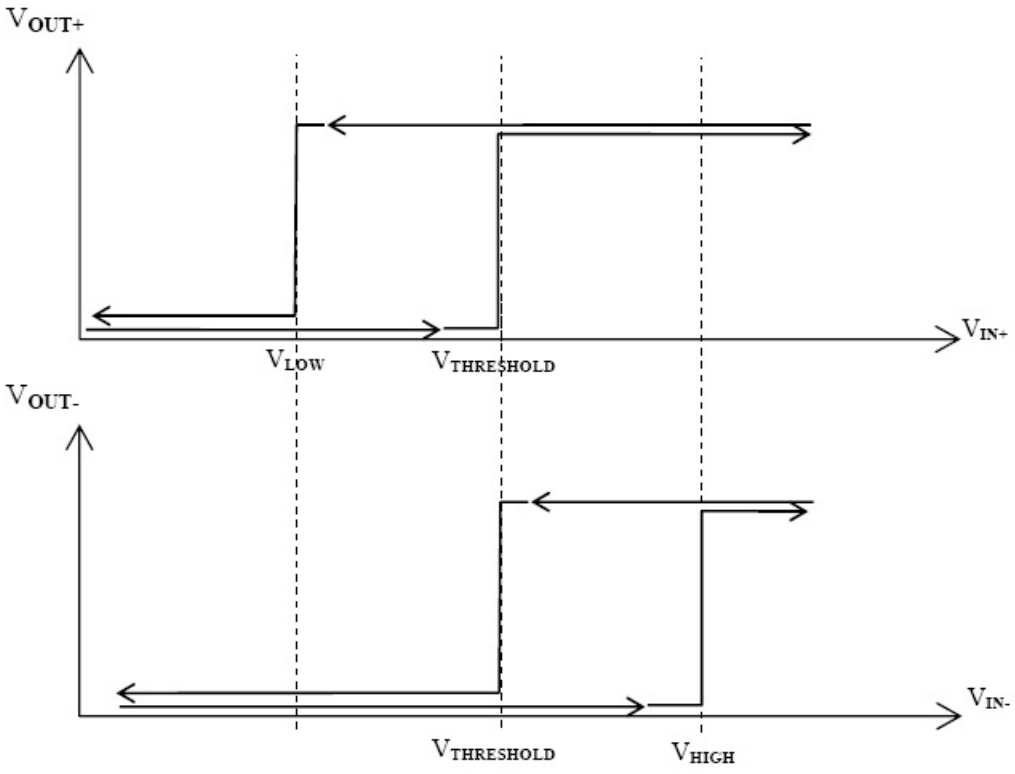


Figure 4.10: Schmitt trigger V_{IN} V_{OUT}.

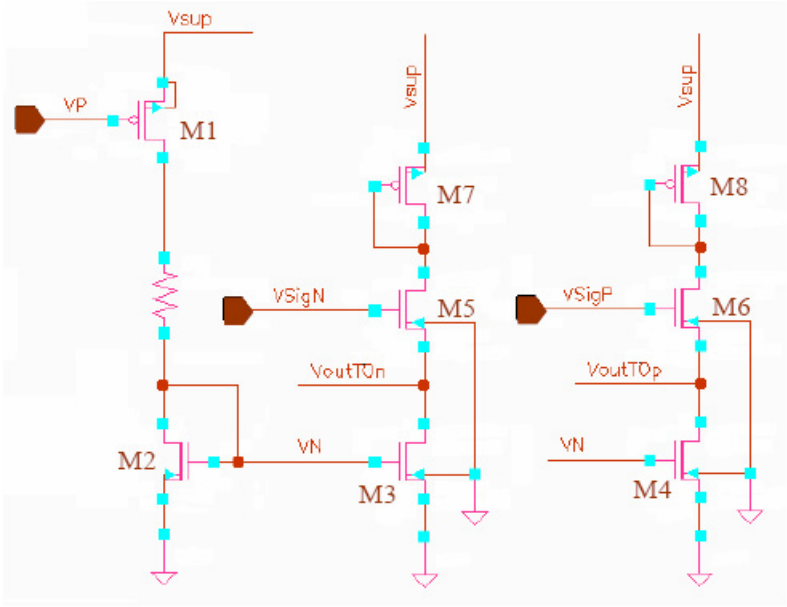


Figure 4.11: Level shifting input stage of level detector.

are more susceptible to channel-length modulation effects so a larger L on $M5$ and $M6$ will give more linear performance than a smaller device, but that will affect the load capacitance the integrator block sees. If kept in saturation, the capacitance to the channel from the gate should be negligible since the source moves with the gate voltage and other channel capacitances are close to zero. Capacitance contributed by overlap is related to width of the device but also is between gate and source so can be ignored. Diffusion capacitances of the devices in the level shifter may affect the speed which the output of the level shifter changes, but does not load the integrator block directly. To reduce channel-length modulation affects, the length of the device is kept over 1μ.

The output of the level shifters are used as differential inputs to comparators. The two level shifters in Figure 4.11 should be placed near each other on the same ground and supply so, when they are used as differential inputs to the amplifier, they have the same common mode characteristics. The comparator used needs to have a wide output swing and low noise. The low noise is, of course, to reduce jitter introduced at this point. The wide output swing is required to make sure that when the transistors in the CMOS inverter are turned off, they are strongly in cutoff. To avoid leakage and increase switching speed, the

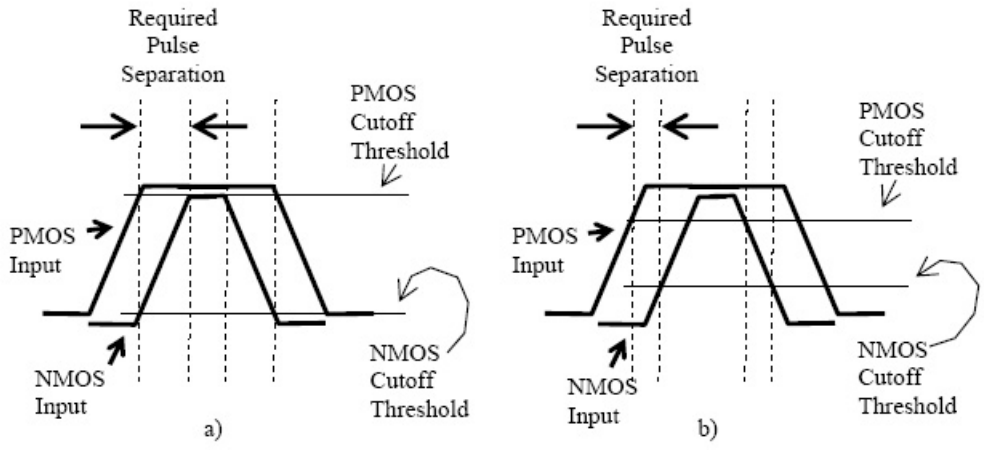


Figure 4.12: a. Pulse separation for high $PMOS$ and low $NMOS$ cutoff thresholds. b. Pulse separation for lower $PMOS$ and higher $NMOS$ cutoff thresholds.

opposing transistor during a switch (the $PMOS$ in a high-to-low transition, and the $NMOS$ in the low-to-high transition), must be switched off before the other transistor is switched on. For example, the inverter in Figure 4.8 would require the $PMOS$ to be switched off before the $NMOS$ is switched on as shown in Figure 4.12a. The inverter shown in Figure 4.8 requires a swing to above $V_{DD} - V_{TP}$ on the high swing and a swing to below V_{TN} on the low side to turn off the $PMOS$ and $NMOS$ respectively. Figure 4.12b shows that there is an added advantage to loosening the threshold levels and that is that pulses that turn on and off the transistors can be closer together. The same effect can be achieved by using steeper sloped inputs to the inverter. A close-up of a single transition is shown in Figure 4.13.

To generate the required wider $PMOS$ and the narrower $NMOS$ inputs, a combination of shifting amounts was used. Figure 4.14 shows the input differential signals, V_{SIGP} and V_{SIGN}. V_{SIGP} and V_{SIGN} are shifted to generate V_{SIGP_SHIFT}, V_{SIGN_SHIFT}, $V_{SIGP\_SHIFT\_LARGE}$ and $V_{SIGN\_SHIFT\_SMALL}$. V_{SIGP_SHIFT} and V_{SIGN_SHIFT} are versions of V_{SIGP} and V_{SIGN} shifted by the same amount. $V_{SIGP\_SHIFT\_LARGE}$ is V_{SIGP} signal shifted by more than V_{SIGP_SHIFT}, and $V_{SIGN\_SHIFT\_SMALL}$ is V_{SIGN} signal shifted by less than V_{SIGN_SHIFT}. The crossing of $V_{SIGP\_SHIFT\_LARGE}$ and $V_{SIGN\_SHIFT\_SMALL}$

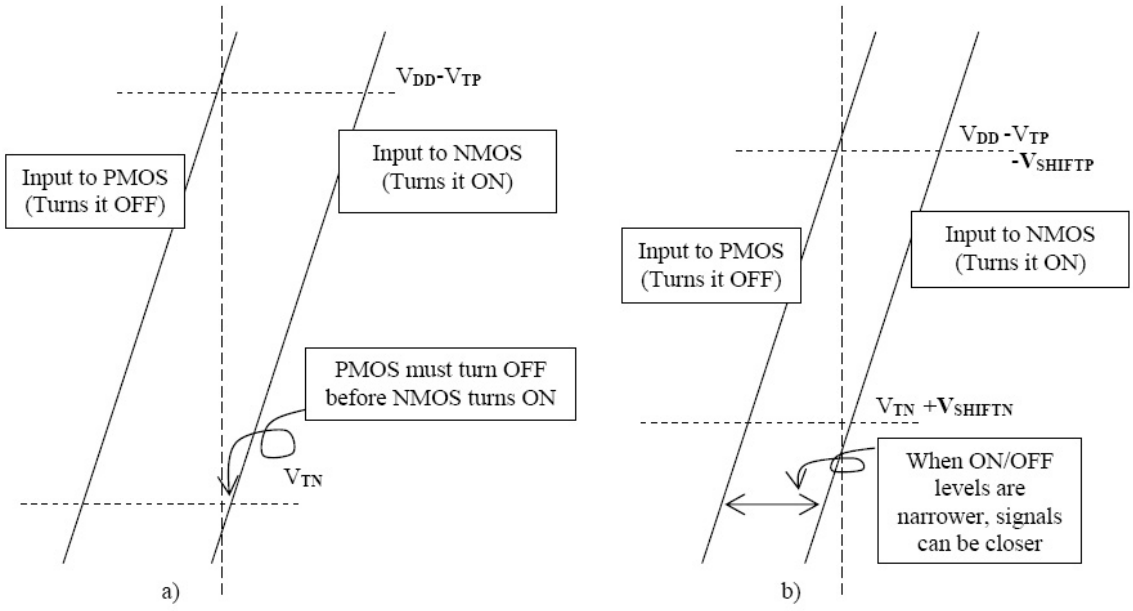


Figure 4.13: V_P and V_N signals can be closer if ON/OFF voltages are farther from the rails.

is used to turn off the presently ON transistor shown in Figure 4.9 as the beginning of the both OFF region on the left side of the diagram. The V_{SIGP_SHIFT} and V_{SIGN_SHIFT} crossing turns on the pull-up or pull-down transistor in the inverter required to perform the actual output switch.

The V_{SIGP_SHIFT} and V_{SIGN_SHIFT} crossing, and the $V_{SIGP\_SHIFT\_LARGE}$ and $V_{SIGN\_SHIFT\_SMALL}$ crossing is input to a comparator. The comparator must have the largest gain possible in order to require the smallest shift possible.

The goal of this section was to design and evaluate level detection circuitry appropriate for the JAC. Proof of concept was shown and an implementation based on basic current sources and level shifters was shown. The performance up to the output of the integrator block is considered central to the performance of the JAC and, therefore, the performance of the level detector The performance of this circuitry is not considered in the performance of the JAC system Further simulations of the output level detection block were not done.

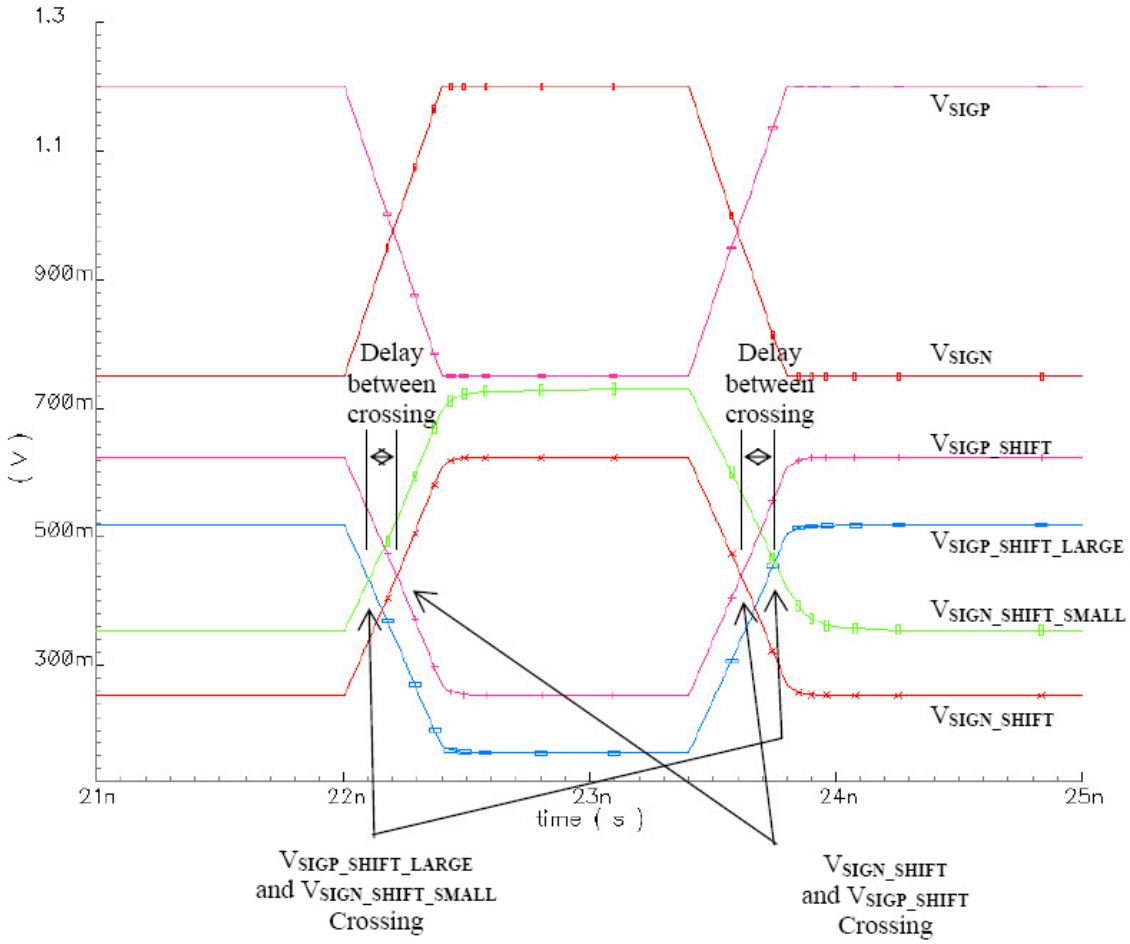


Figure 4.14: Generating V_{HI} and V_{LOW} by adjusting shift amount.

4.3.2 Circuit Design

The three blocks involved in the output stage is the level shifter, the comparator, and the $CMOS$ inverter. The level shifter was shown in Figure 4.11. The current source in the diagram consisting of $M1$, $M2$ and the resistor, can, of course, be replaced with sources that are less sensitive to temperature, noise, and process variation.

The comparator and $CMOS$ inverter need to be designed together. If the $CMOS$ inverter used is a simple inverter as shown in Figure 4.8, the output of the comparator must be near rail-to-rail. If the threshold of the $CMOS$ inverter is somehow adjusted such that the thresholds are greater than the V_T of each transistor, the output swing of the comparator can be smaller.

Figure 4.15 shows the circuitry and signal flow of the final design. The flow through the circuit starts in the upper left corner of Figure 4.15. The output of the integrator is shifted by four shifters, two of which shift the positive and negative output of the integrator some intermediate amount, one which shifts the negative integrator output by a greater amount, and one that shifts the positive integrator output by a smaller amount. The crossing of the last two signals generates the outer pulse in Figure 4.12. After shifting the signals, they are amplified. The steeper the slope, the better, and any number of amplifiers were tried including traditional comparators ([*Lotfi et al.*, 2003]), and traditional amplifiers (folded and telescopic amplifiers, gain-boosting architectures, and various multi-stage amplifiers). Though each amplifier had its advantages and disadvantages, high-gain amplifiers, in general, gave better results because they required a smaller delay between shifted signals and therefore smaller shifts. In Figure 4.15, a differential pair is used as an amplifier because of its simplicity though it is not the optimal option in terms of performance, but quite clearly one of the simplest options for a designer. Two types of differential pairs are used as amplifiers; Those with $NMOS$ active devices and those with $PMOS$ as their active devices. As mentioned previously, increasing the voltage needed to turn the $NMOS$ and $PMOS$ devices in the final $CMOS$ inverter from $V_{DD} - V_{TP}$ and V_{TN} to $V_{DD} - V_{TP} - V_{\Delta P}$ to $V_{TN} + V_{\Delta N}$ can make the design of the amplifier easier. Instead of building amplifiers that swing rail-to-

rail (or, more correctly at least $V_{DD} - V_{TP}$ to V_{TN}), they can be built to a smaller output swing. Instead of taking this approach of changing the $CMOS$ inverter thresholds, the method chosen for this design is to control the actual inverter $NMOS$ and $PMOS$ inputs. The amplifiers with $NMOS$ active devices swing from V_{DD} to $V_{DD} - (V_{TP} + 200mV)$ and the amplifiers with $PMOS$ active devices swing from $0V$ to $V_{TN} + 200mV$. The output of the amplifiers with $NMOS$ active devices are then used as input to the $PMOS$ device in the $CMOS$ inverter and the output of the amplifiers with $PMOS$ active devices are then used as input to the $NMOS$ device in the $CMOS$ inverter.

Simulations of the output of the amplifiers are shown in Figure 4.16. The graph on the left shows important points in causing the CMOS inverter to pulldown its output. Note that the $PMOS$ is off well before the $NMOS$ is turned on. The graph on the right shows important points in causing the CMOS inverter to pullup its output. Note that the $NMOS$ is off well before the $PMOS$ is turned on.

Simulations of the output of the CMOS inverter are shown in Figure 4.17. There are a few important points to note. The most important is that the inverter output value is preserved throughout the regions where both transistors are off. Early versions were the victim of leakage. The inspiration for the separate $NMOS$ and $PMOS$ controlling amplifiers was to reduce this leakage to an acceptable level (*i.e.* to where the value is preserved when both transistors are off), which it has done. The input signal from the integrator has a slope of $881V/\mu s$ and the output of the inverter has a slope of $14176V/\mu s$. This gain can be increased by using higher gain amplifiers, sizing up the devices in the CMOS inverter, and increasing the output swing of the amplifiers. The slew rates of the transistors as well as the switching speed of the inputs are also limiting factors here. Figure 4.17 shows a steeper slope on the rising edge than on the falling edge. This was a design choice made assuming that the rising edge, in this case, is the only important edge. If the output of the JAC is going into a monostable such as the one used as an input monostable in this design, only one edge is important.

The differential level detector described in this section can be greatly improved using less sensitive current sources and with more attention to sizing detail. It will be used as a

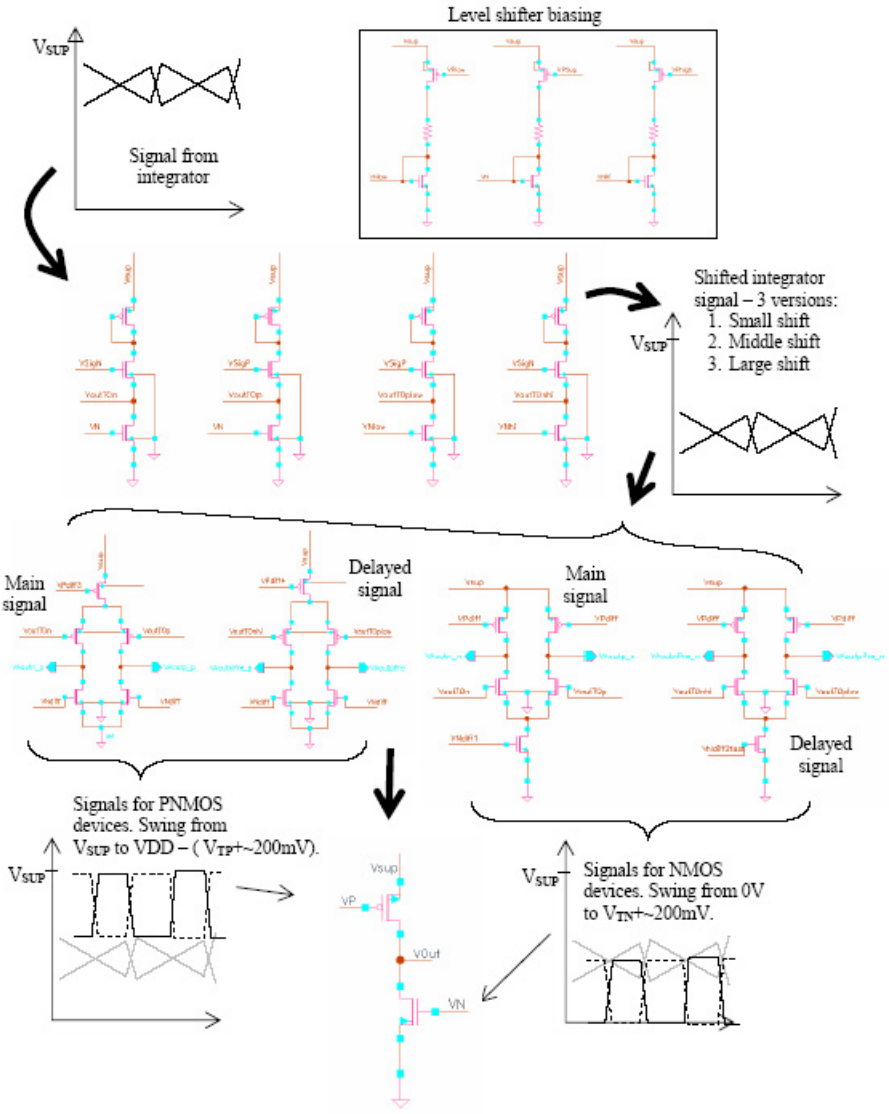


Figure 4.15: Final level shifter, amplifier, Schmitt trigger flow.

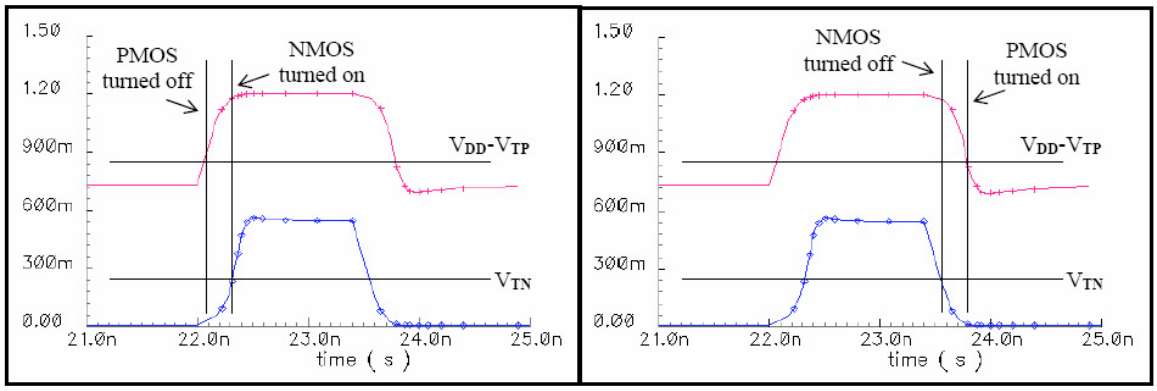


Figure 4.16: Simulation of output of amplifiers and important points in operation of CMOS inverter.

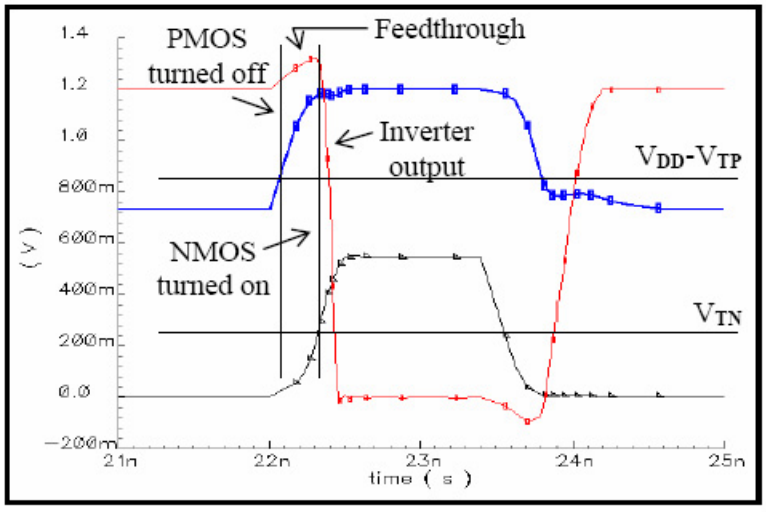


Figure 4.17: Simulation of output of CMOS inverter.

load for the integrator block, but its performance will not be further investigated in this work.

4.4 Integrator Performance

In this section the performance of integrator is examined through simulation. Assuming that feedback and feedforward circuitry do adjustments which correct process variation affects on circuit, only intrinsic, supply and GND noise will be evaluated. Carlo simulations are done with and without supply, GND, and intrinsic noise, as well as worst case analysis.

4.4.1 Jitter

The integrator circuit used for simulations in this section is built for large currents and integration capacitance, and duty cycle not equal to 50%.

Item	Description
Tail current	160μ
Integration Capacitance	$80fF$
Frequency	$1GHz$
Duty Cycle	65%

Table 4.2: Design parameters of integrator.

Using Equations 4.12 through 4.15, the maximum swing of the integrator output can be calculated. Because of the large device sizes, it is appropriate to use the first-order long-channel device equation. The equation for V_{DSAT} is as shown in Equation 4.17.

$$V_{DSAT} = \sqrt{\frac{I_{DS}}{\frac{k_P'}{2}\frac{W}{L}}} \approx 0.25V \tag{4.17}$$

And

$$V_{sLIMITMAX} = V_{DD} - V_{DSAT} \approx 1.2V - 0.25V = 0.95V \tag{4.18}$$

The $V_{sCENTER}$ value can also be found:

$$V_{sCENTER} = \left(V_{sLIMITMAX} + V_{sCENTER}\right)/2 \approx 0.68V \tag{4.19}$$

With a period of $1ns$, and a duty cycle of 65% gives a slope of:

$$|m1| \leq P_{sTOLERANCE} \times \frac{0.95V - 0.68V}{150ps + \frac{1}{2}650ps} = P_{sTOLERANCE} \times 5.6 \times 10^8 \tag{4.20}$$

The other slope is:

$$|m2| \leq P_{sTOLERANCE} \times \frac{0.95V - 0.68V}{150ps + \frac{1}{2}\left(1ns - 650ps\right)} = P_{sTOLERANCE} \times 8.3 \times 10^8 \tag{4.21}$$

The simulations done in this section are done with ideal parts and process variation of sizes is ignored. In the case that devices were not ideal, their tolerances would have to be included in the slope calculations. Because process variation is ignored, the value of $P_{sTOLERANCE}$ can be considered as 1 for the following simulations.

Figures 4.18 through 4.21 show simulations with $\pm 150ps$ of jitter on the input to the integrator. There is no supply noise, GND bounce, or intrinsic noise included, and biasing is fixed. Figure 4.18 shows a sawtooth with jitter and slope requirements met. This case gave a $-35.5dB$ (a $59.6x$ improvement) reduction in jitter with dB being defined as $20log_{10}\left(\frac{t_{InputJitterRMS}}{t_{IouputJitterRMS}}\right)$. Figure 4.19 shows a single integrator output to illustrate the movement of the output signal in the presences of jitter. Figure 4.20 shows an output of an integrator that has too large a slope for the jitter present on the inputs. As the output increases beyond $V_{sLIMITMAX}$ the $PMOS$ are forced into triode (or as the output goes below $V_{sLIMITMIN}$ and reduces the current through the switch transistors or tail current source), V_{DS} begins to affect the shape of the signal and the output becomes non-linear. Even with such non-linearities, a reduction of jitter of $-32dB$ (a $39.8x$ improvement) was achieved. The final diagram, Figure 4.21, shows what happens when there is more than $T - t_p$ jitter on the input. At about $14.5ns$ the signals do not cross and will therefore not generate an output pulse.

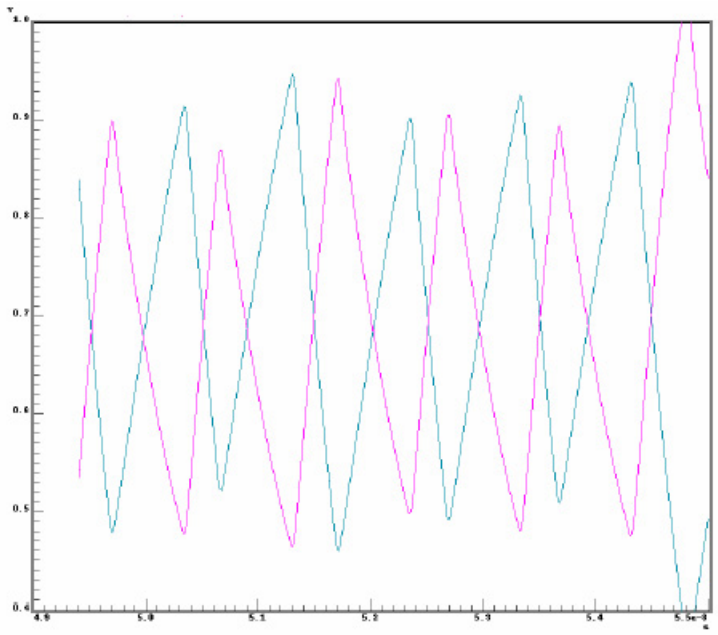


Figure 4.18: Output of differential integrator. $-35.5dB$ jitter reduction. (A $59.6x$ improvement)

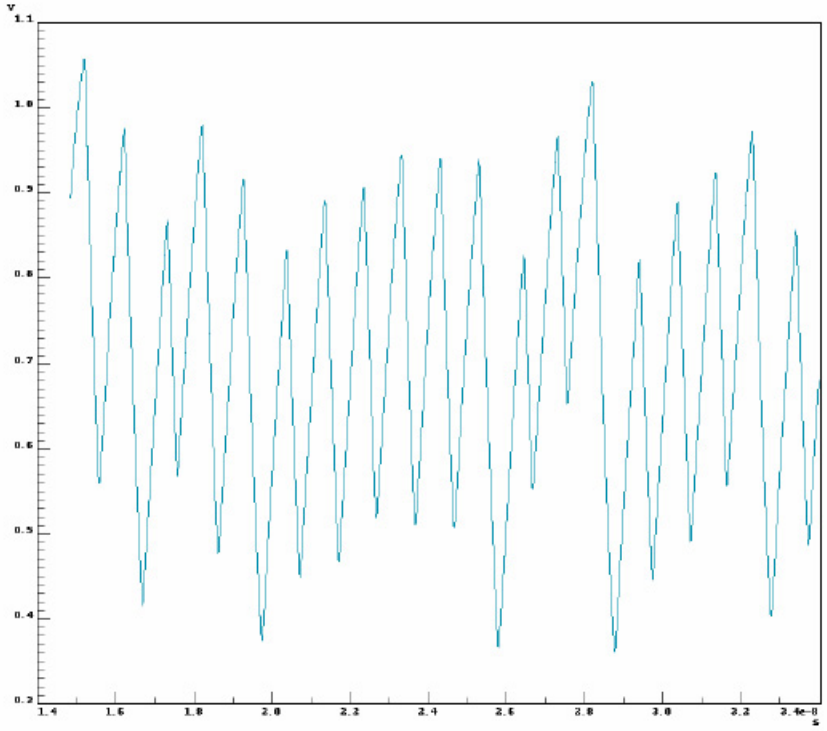


Figure 4.19: Single output of differential integrator.

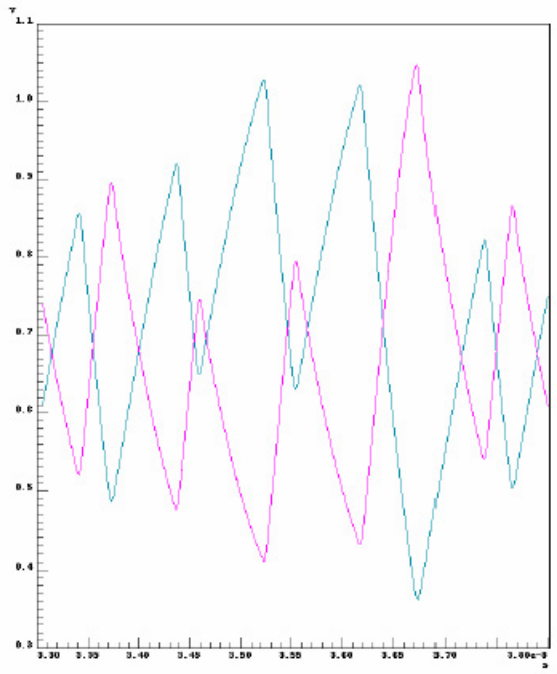


Figure 4.20: Output swing that forces $PMOS$ in triode. $-32dB$ jitter reduction. (A $39.8x$ improvement)

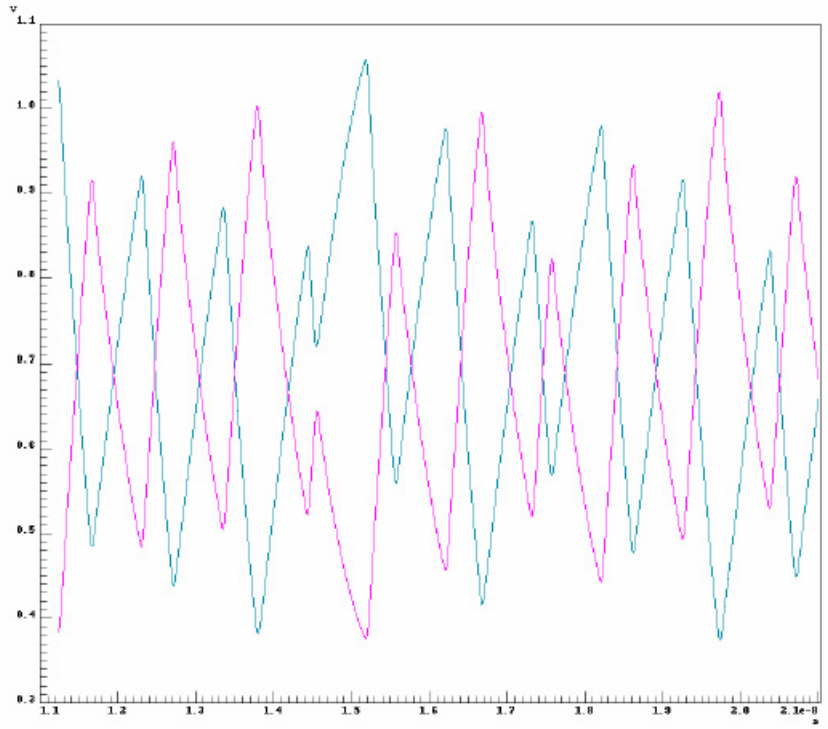


Figure 4.21: Jitter which exceeds $T - t_p$.

These results can be improved by decreasing the integrator's output swing (decreasing the size ratio of the $PMOS$ or increasing the size of the integrating capacitors), increasing the swing of the input signal (keeps tail and switch transistors in saturation longer), and increase transistor size. The sizes used here were not optimized. They were selected for a high current so the load capacitance would affect the integrator's behavior minimally, and a small switch size so as to not load the driving circuit. Previous tests with a more highly optimized integrator gave jitter reduction numbers near $-40dB$ (a $100x$ improvement) jitter reduction, but they could not drive the level detection circuitry sufficiently.

4.4.2 Intrinsic, Supply and GND Noise

Intrinsic noise was analyzed during transient response simulations using Eldo (Mentor Graphics). RMS Jitter reduction was only reduced by about $0.5dB$ to $-35dB$ (a $56.2x$ improvement). Supply and GND noise was done with noise randomly from $+200mV$ to $-200mV$ and with continuous noise pulses varying randomly from $1ps$ to $50ps$. 230 to 240 crossings were observed. Table 4.3 summarizes the results.

Noise Source	dB Jitter Reduction	With Intrinsic Noise
None	$-35.5\,(59.6x)$ dB	$-35\,(56.2x)$ dB
With large o/p range	$-32\,(39.8x)$ dB	$-27\,(22.4x)$ dB
Supply $(200mV)$	$-14.04\,(5.0x)$ dB	$-13.35\,(4.7x)$ dB
GND $(200mV)$	$-22.03\,(12.6x)$ dB	$-20.39\,(10.5x)$ dB

Table 4.3: Summary of affects of noise on integrator block jitter reduction.

4.4.3 Analysis of Results

Chapter 2 discussed the errors important to the integrator block's contribution to the output jitter. Many of the variables, for example, $m1$ and $m2$, the input and output swing, and t_p width, were fixed during the simulations done in this chapter. λ, the channel length modulation coefficient, was minimized by increasing lengths of transistors which function as current sources or sinks. Capacitances and currents were large to reduce the effects of

loads on the integrator. In order to increase switching speeds, the switching transistors were minimized.

The results in Table 4.3 show results of supply and GND noise on integrator with large output range. Supply noise is clearly the most significant source of error. The error found with no noise sources taken into account is from channel length modulation of $PMOS$ current sources and error introduced during switching. Switching errors, as discussed in Chapter 2, are reduced as the transition is sped up. This can be done by reducing the size of the switching transistors of the integrator and/or increasing the drive power and reducing the rise/fall time of the circuit generating the signal which controls those switching transistors. Ideal performance will occur when switching time equals zero, clearly an unachievable performance goal.

In this case, the both the reduction of intrinsic noise and of channel length modulation effects can be done by increasing the size of the $PMOS$ current source devices. The area hit from increasing the size of the $PMOS$ transistors may be offset by the required reduction in the size integrating capacitors. As the size of the $PMOS$ is increased, their diffusion capacitance will reduce the slope of the integrator's output. To keep the output slope constant as size of the $PMOS$ are increased, the size of the integrating capacitors must be decreased. Taken to its limit, eventually integrating capacitors may not even be necessary, but no investigation of where this point is or whether it would actually provide the performance required has been done.

Chapter 5

Adjustment Block (Feedforward and Feedback Blocks)

5.1 Introduction

The adjustment (feedforward and feedback) blocks are responsible for biasing the integrator block or adjusting the integrator's output such that the sawtooths are positioned on top of each other and that each individual sawtooth follows the basic equation ($m1 \times t_p + m2 \times (T - t_p) = 0$). Putting the two sawtooths on top of each other allows for maximum peak-to-peak input jitter when the input jitter has a Gaussian distribution. In cases where input jitter is non-Gaussian, the two sawtooths may be adjusted to cover outlying cases, though, because this is a solution for special case jitter spectra, the robustness of the circuit against noise may be reduced. A purely feedforward design was selected as the final design, but the possibility of using feedback was also examined. This chapter will examine methods of feedforward and feedback control to keep the circuit working correctly and the circuitry proposed to carry this out.

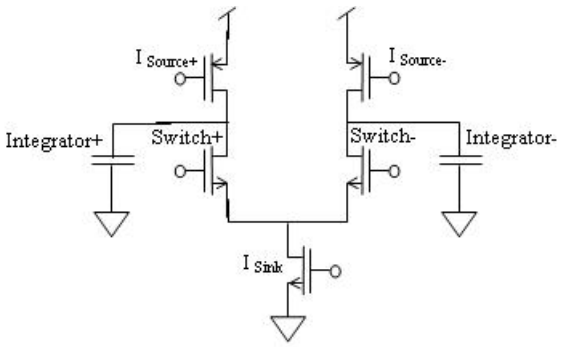


Figure 5.1: Transistor level diagram.

5.2 Integrator Block Biasing

The devices which require biasing are the $PMOS$ and tail current source $NMOS$ in the integrator or sawtooth generator block (Figure 5.1).

The switches of the integrator are controlled by the output of the input monostable. The output of the input monostable is differential and as such, only one of the $NMOS$ switches of the integrator is open at a time. When a switch is open, the current from the $I_{source+/-}$ charges the integrating capacitor and the value on the capacitor rises. When a switch is closed, there is still the current on to the capacitor from $I_{source+/-}$, but there is also an amount of current being sunk, namely I_{sink}, from the integrating capacitor. In this circuit, I_{sink} is greater than $I_{source+/-}$ and accordingly there is a negative net amount of charge being delivered to the capacitor, and therefore a falling slope on the voltage on that capacitor. The values on the capacitors are the outputs of this block and they give the sawtooth waveform. The slopes $m1$ and $m2$ in the basic equation, $m1 \times t_p + m2 \times (T - t_p) = 0$, come from the current on to, and off of the capacitors. The other two values in the equation, t_p and T, come from the duty cycle of the signal controlling the switches, the output of the input monostable. Assuming the signal from the input monostable is fixed and t_p and T are not adjustable, then $m1$ and $m2$ must be set appropriately for the equation to stay true, and those values depend directly upon the the currents $I_{source+}$, $I_{source-}$, and I_{sink}. If they are not set correctly, then the sawtooths will will not cross and no output will be generated. The relevant equations for determining proper biasing are listed in Table 5.2.

The ideal case assumes instantaneous switching, devices with currents not affected by varying V_{DS}, no noise, and properly ratioed current sources $I_{source+}$, $I_{source-}$, and I_{sink}. The capacitors are not in this list because even if they vary, say, due to process variation, they will not affect the position of the crossings. If C_+ does not equal C_-, it is true $m1_-$ will not equal $m1_+$, and m_2- will not equal $m2_+$. What is important is that the ratios $\frac{m1_-}{m1_+}$ will equal $\frac{m2_-}{m2_+}$, and the ratio will be the same as the ratio of $\frac{C_+}{C_-}$. As long as the integrator block's $PMOS$ and sink device do not go into triode, one sawtooth will merely look like an inverted and scaled version of the other and the voltage where the signals cross should not change.

The first important equation needed in biasing the integrator correctly is a relationship between $I_{source+}$, $I_{source-}$, and I_{sink}. If the ratio $\frac{C_+}{C_-}$ is equal to α, then α will cancel out as shown in Equation 5.1. Multiplying both sides by C_+ and rearranging gives Equation 5.2, a relationship relating the three currents, $I_{source+}$, $I_{source-}$, and I_{sink}. This relationship states that the tail current sink of the integrator must be equal to the sum of the $PMOS$ current sources.

$$\frac{I_{source+}}{C_+} = -\frac{(I_{source-} - I_{sink})}{\alpha \times C_-} = -\frac{(I_{source-} - I_{sink})}{\alpha \times \frac{C_+}{\alpha}} = -\frac{(I_{source-} - I_{sink})}{C_+} \tag{5.1}$$

$$I_{source+} = -I_{source-} + I_{sink} \Rightarrow I_{sink} = I_{source+} + I_{source-} \tag{5.2}$$

The second relationship needed to bias the integrator comes from the constraint equations in Table 5.2. In Equation 5.3, a relationship for $I_{source+}$ and I_{sink} is derived. This says that the ratio of $I_{source-}$ to I_{sink} is the duty cycle of the signal from the input monostable.

Integrator Output	Switch Open (Slope)	Switch Closed (Slope)	Constraint
V^+	$m1^+ = \frac{I_{source+}}{C^+}$	$m2^+ = \frac{(I_{source+} - I_{sink})}{C^+}$	$m1^+ \times t_p + m2^+ \times (T - t_p) = 0$
V^-	$m2^- = \frac{I_{source-}}{C^-}$	$m1^- = \frac{(I_{source-} - I_{sink})}{C^-}$	$m2^- \times t_p + m1^- \times (T - t_p) = 0$

Table 5.1: Slopes and constraints of integrator block.

$$\begin{aligned} m2 \times t_p + m1 \times (T - t_p) &= \frac{I_{source-} - I_{sink}}{C_-} \times t_p + \frac{I_{source-}}{C_-} \times (T - t_p) = 0 \\ &\Rightarrow \; I_{source-} \times T - I_{sink} \times t_p = 0 \\ &\Rightarrow \; \frac{I_{source-}}{I_{sink}} = \frac{t_p}{T} \end{aligned} \tag{5.3}$$

Using the same math as used in Equation 5.3, Equation 5.4 shows that the relationship between $I_{source-}$ and I_{sink} is again related to the duty cycle of the signal coming from the input monostable. In this case, the duty cycle is that of the pair of the differential for $I_{source-}$, and therefore sees a duty cycle of $1 - \frac{t_p}{T}$ or $\frac{(T-t_p)}{T}$.

$$\begin{aligned} m1 \times t_p + m2 \times (T - t_p) &= \frac{I_{source+}}{C_+} \times t_p + \frac{I_{source+} - I_{sink}}{C_+} \times (T - t_p) = 0 \\ &\Rightarrow \; _{source+} \times T - I_{sink} \times T + I_{sink} \times t_p = 0 \\ &\Rightarrow \; \frac{I_{source+}}{I_{sink}} = \frac{T - t_p}{T} \end{aligned} \tag{5.4}$$

This gives the important relationship given in Equation 5.5 that says that the ratio of the currents, $I_{source+}$ and $I_{source-}$, are exactly the ratio of the duty cycles of the output signals of the input monostable.

$$\frac{\frac{I_{source-}}{I_{sink}}}{\frac{I_{source+}}{I_{sink}}} = \frac{I_{source-}}{I_{source+}} = \frac{\frac{t_p}{T}}{\frac{T-t_p}{T}} = \frac{t_p}{T - t_p} \tag{5.5}$$

5.3 Pre-Integrator Feedforward Block

The purely feedforward design and the design that uses feedback both use a feedforward block before the integrator to set initial biasing values to the integrator. This provides a quick start-up biasing method for the design which uses feedback, and, in the case of the purely feedforward design, makes the purely feedforward design possible. A feedback design

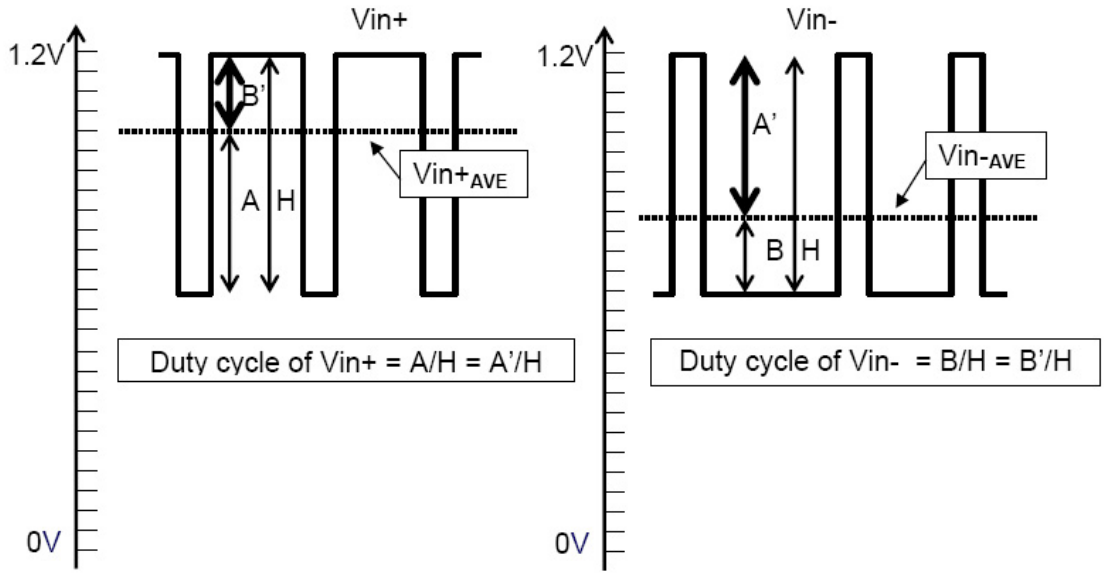


Figure 5.2: Generating information on the duty cycle of a square wave.

would be possible without the pre-integrator feedforward block, but this design assumes its existence and uses it as part of the feedback loop. In the case of the purely feedforward design, nothing is added to this block but the design which uses feedback would require a slight alteration.

For the ideal integrator with no leakage, no channel-length modulation effects, no velocity saturation, and no environmental influences, Equations 5.2 and 5.5 give all the information necessary to perfectly bias the integrator. The integrator, of course, will not behave ideally, but initially biasing according to these equations will serve a multiple purposes. As mentioned, it is the key to making a purely feedforward system possible. It will also allow the feedback to only have to deal with small adjustments, rather than biasing from scratch. It also provides a mechanism to use feedback to adjust the biasing.

5.3.1 Block Level Design

The flow used here to create the pre-integrator, feedforward biasing starts with converting the output of the input monostable into a voltage which represents the duty cycle.

Figure 5.2 shows a possible output of the input monostable. In this example, the square wave does not swing rail to rail, but swings from a maximum value of V_{DD} to a value around 700mV, V_{min}. In Figure 5.2 there are a number of values marked including the average of the square waves. The average of a square wave can be found by integrating its value over a period and then dividing the result by the length of the period. An important point to note here is that the values being looked at here are relative to V_{min}, not GND and, as such, V_{min} must be subtracted from the value being integrated. Equation 5.6 integrates the signal values, and then calculates what percentage this value is of the full swing by dividing by the of the square wave gives the duty cycle. This is true for the opposing differential signal as well and is shown in Equation 5.7.

$$\frac{1}{T\left(V_{DD}-V_{min}\right)}\left[\int_0^{t_p}\left(V_{DD}-V_{min}\right)dt+\int_{t_p}^{T}\left(V_{min}-V_{min}\right)dt\right]=\frac{t_p\left(V_{DD}-V_{min}\right)}{T\left(V_{DD}-V_{min}\right)}=\frac{t_p}{T} \tag{5.6}$$

$$\frac{1}{T\left(V_{DD}-V_{min}\right)}\left[\int_0^{t_p}\left(V_{min}-V_{min}\right)dt+\int_{t_p}^{T}\left(V_{DD}-V_{min}\right)dt\right]=\frac{T-t_p}{T} \tag{5.7}$$

In this scenario, all values, A, B, and H are measured from V_{min}, but the bottom edge of the square wave is often not flat with mismatching affecting its slope, transitions causing bounce, and corners that are often rounded. V_{min} can be removed from the equation by noting that A, the ΔV from V_{min} to V_{in+AVE} in Figure 5.2, is the same as A', the ΔV from V_{in-AVE} to V_{DD}, and that B, the ΔV from V_{min} to V_{in-AVE}, is the same as B', the ΔV from V_{in+AVE} to V_{DD}. Using this, an equation for the ratios of duty cycles can be written only using V_{in+AVE}, V_{in-AVE}, and V_{DD} as shown in Equation 5.8. The average values V_{in+AVE} and V_{in-AVE} also suffer from a noisy V_{min}, but, unlike duty cycle calculations, a DC value for the V_{min} edge is not required, and slopes, noise and rounded corners on the V_{min} edge tend to somewhat average themselves out and the variation between periods is relatively small. Even though the 'noise' on V_{in+AVE} and V_{in-AVE} somewhat average itself out, the imperfections it introduces do affect the value of the averages, and correcting

circuitry to the integrator current sources must be available. The error caused by imperfect V_{min} edges is one error that the feedback or the later feedforward circuitry addresses.

$$\frac{V_{in+}}{V_{in-}} = \frac{A'/H}{B'/H} = \frac{A'}{B'} = \frac{V_{DD} - V_{in-AVE}}{V_{DD} - V_{in+AVE}} \tag{5.8}$$

Once the duty cycles are obtained, they must somehow be converted into biasing for the current sources. Though Equation 5.8 is a relatively simple, in general, subtracting voltages is difficult. But, as shown in Equation 5.9, if proportional currents were available, the problem would become the sum and difference of currents, a much simpler problem.

$$\frac{V_{in+}}{V_{in-}} = \frac{V_{DD} - V_{in-AVE}}{V_{DD} - V_{in+AVE}} = \frac{I_{V_{DD}} - I_{V_{in-AVE}}}{I_{V_{DD}} - I_{V_{in+AVE}}} \tag{5.9}$$

To carry out this mathematics, the voltages were converted into current. Figure 5.3 shows a simulation of the voltage to current converter taking values as input in the range of possible monostable averages. The next step is to find the appropriate currents that correspond to average voltage inputs. The dotted vertical lines in Figure 5.3 cross the voltage curves at 100%, 75%, and 50% duty cycles. At a 100% duty cycle the average of one of the input monostable outputs will be V_{DD} and the average of the other will be V_{min}. At a 50% duty cycle the averages will be the same and can be seen in Figure 5.3 where the voltage lines cross. The 75% duty cycle shows a higher V_{in+AVE} of approximately $1.1V$, and lower voltage for the V_{in-AVE} of approximately $880mV$. Following the dotted lines down to the current graph, the currents that correspond to the voltages are marked. These two values are the $I_{V_{in-AVE}}$ and $I_{V_{in+AVE}}$ values from Equation 5.9. The current that corresponds to V_{DD} must also be found and is marked in Figure 5.3 at center bottom of the graph. The final step is to subtract $I_{V_{in-AVE}}$ and $I_{V_{in+AVE}}$ from $I_{V_{DD}}$ to get the currents needed for the integrator. These currents are shown by arrows, also, in Figure 5.3.

Ideally, the feedforward block should provide biasing that sets currents such that no other adjustments are needed, but, among other reasons, mismatch in the feedforward circuitry, errors from the averager, and integrator switches that do not turn off completely will move one or both of the sawtooths off of their ideal position. Errors in the averager will

give incorrect t_p and $T - t_p$ values and mismatch and switching issues will affect $I_{Source+}$, $I_{Source-}$, and I_{sink}.

Assuming that these errors are a fixed shift for a given input and not random varying values, the values used in determining biases are as shown in Table 5.3.1.

Value Description	Value No Error	Value With Error
Monostable + Pulse Width	t_p	$t_p + \Delta t_p$
Monostable - Pulse Width	$T - t_p$	$T - t_p + \Delta T t_p$
Current Source (PMOS+)	$I_{source+}$	$I_{source+} + \Delta I_{AVE+}$
Current Source (PMOS-)	$I_{source-}$	$I_{source-} + \Delta I_{AVE-}$
Current Sink (NMOS)	I_{sink}	$I_{source+} + I_{source-} + \Delta I_{AVE+} + \Delta I_{AVE-} + \Delta I_{sink}$
Slope (V^+) Switch open	$\frac{I_{source+}}{C^+}$	$\frac{I_{source+}+\Delta I_{AVE+}-\Delta I_{Leak}}{C^+}$
Slope (V^-) Switch open	$\frac{I_{source-}}{C^-}$	$\frac{I_{source-}+\Delta I_{AVE-}-\Delta I_{Leak}}{C^-}$
Slope (+) Switch closed	$\frac{I_{source+}-I_{sink}}{C^+}$	$\frac{(I_{source+}+\Delta I_{AVE+}-I_{sink})}{C^+}$
Slope (-) Switch closed	$\frac{I_{source-}-I_{sink}}{C^-}$	$\frac{(I_{source-}+\Delta I_{AVE-}-I_{sink})}{C^-}$

Table 5.2: Slopes and constraints of integrator block with errors.

Rewriting the constraint equation using the values from Table 5.3.1, give Equations 5.10 and Equation 5.11.

$$\begin{aligned}&(I_{source+} + \Delta I_{AVE+} - \Delta I_{Leak}) \times (t_p + \Delta t_p)\\ &+ (I_{source+} + \Delta I_{AVE+} - I_{sink} - \Delta I_{sink}) \times (T - t_p + \Delta T t_p) = 0\end{aligned} \tag{5.10}$$

$$\begin{aligned}&(I_{source-} + \Delta I_{AVE-} - I_{sink} - \Delta I_{sink}) \times (t_p + \Delta t_p)\\ &+ (I_{source-} + \Delta I_{AVE-} - \Delta I_{Leak}) \times (T - t_p + \Delta T t_p) = 0\end{aligned} \tag{5.11}$$

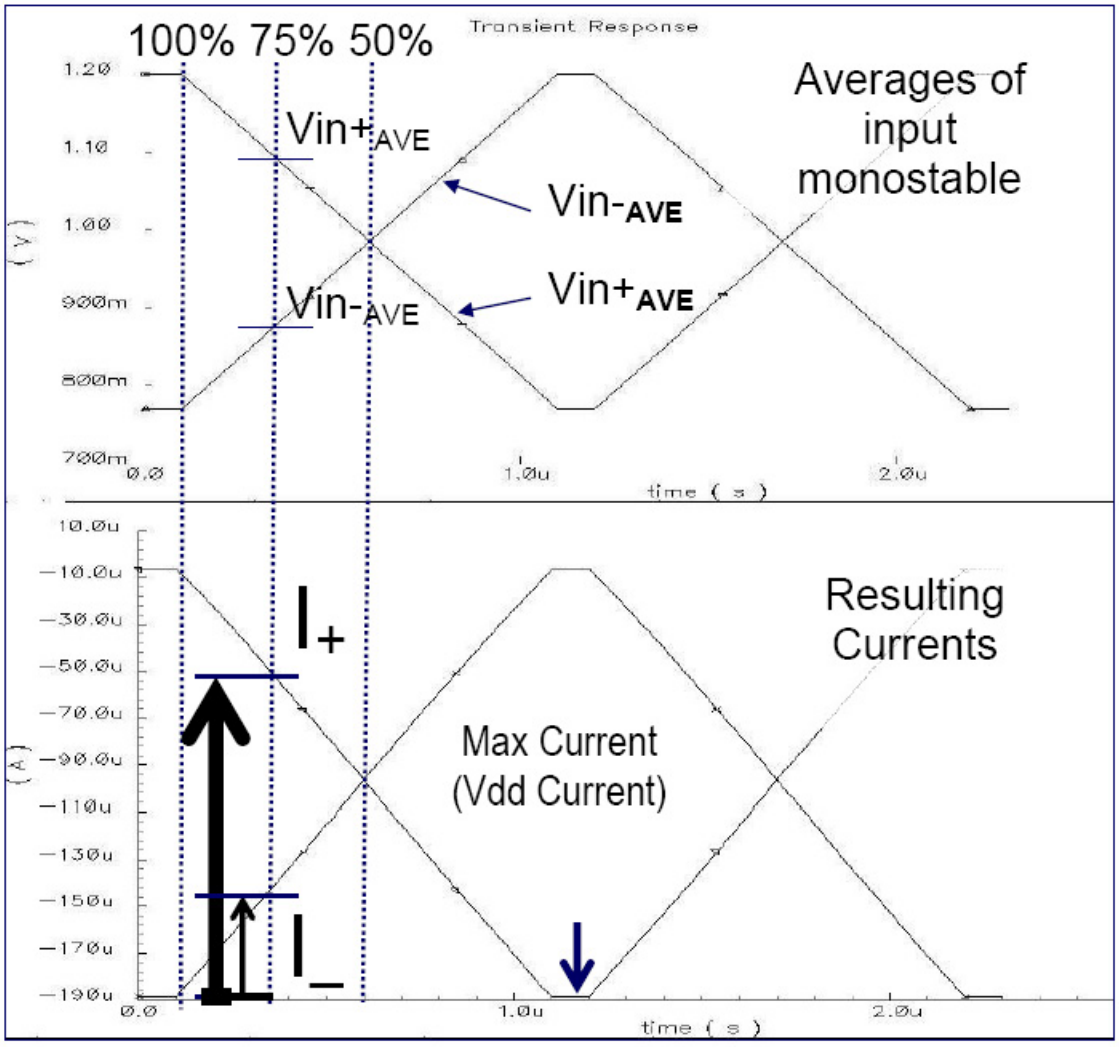


Figure 5.3: Voltage to current converter simulation.

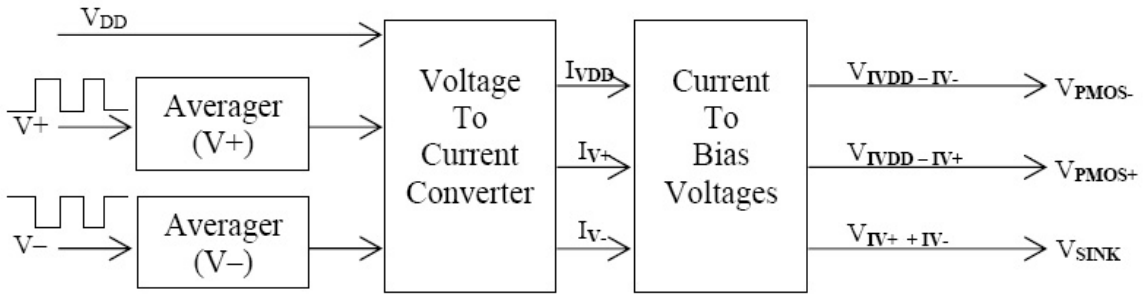


Figure 5.4: Feedforward block diagram.

Substituting the values from Table 5.3.1 for I_{sink} into Equations 5.10 and Equation 5.11, multiplying one equation by -1, and setting the equations equal to each other result in the equality in Equations 5.12.

$$\Delta I_{sink} = -\Delta I_{Leak} \tag{5.12}$$

In other words, if the errors are considered as fixed shifts for a given input and not random varying values, the necessary adjustment to repair integrator current and duty cycle errors is merely an adjustment to I_{sink}. This is interesting in that Δt_p, $\Delta T t_p$, ΔI_{AVE+}, and ΔI_{AVE-} can be ignored when addressing fixed feedforward biasing errors. This changes Equation 5.2 to Equation 5.13. Though this is an interesting observation, note that simulations did not support that this was the most important source of error. Upon simulation, the effects V_{DS} on the $PMOS$ currents, values which varies as the output changes, affected the position of the sawtooth waves the most and is not fixed as was the assumption for deriving Equation 5.13.

$$I_{sink} = I_{source+} + I_{source-} + \Delta I_{leak} \tag{5.13}$$

The block diagram of the current derivation is shown in Figure 5.4. $V+$ and $V-$ are the signals from the input monotable, and V_{DD} is the supply voltage, but more importantly, the maximum value of $V+$ and $V-$.

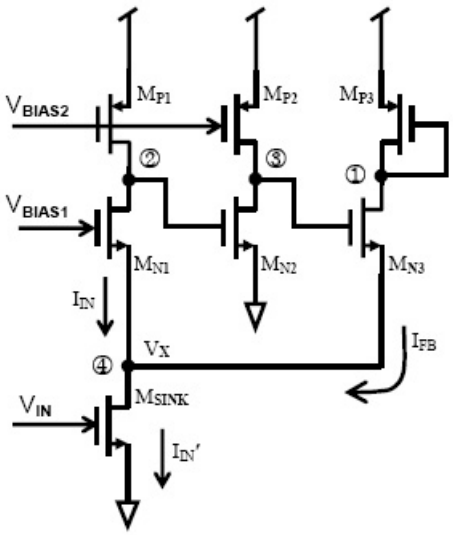


Figure 5.5: Circuit diagram for voltage to current converter.

5.3.2 Circuit Design

This section describes the feedforward circuitry, the general operation of the feedforward circuitry, its non-dealities and how those non-idealities affect the performance of the entire JAC circuit.

In order to bias the $PMOS$ of the integrator, the voltage values in Equation 5.9 are changed to current values, and subtracted as needed. The circuit used to convert voltage to current is shown in Figure 5.5. All devices are biased in active region except M_{SINK} which is biased in deep linear region. The top half of the circuit controls the voltage at the drain of M_{SINK}, V_X, and keeps it as close as possible to a constant value. Holding constant the V_{DS} of triode biased M_{SINK} means that the device will exhibit linear behavior, like a resistor, as V_{GS} varies.

Ideal long channel current equations can be used to show how the loop composed of all transistors except M_{SINK} holds V_X constant even while V_{IN} varies. Note that the current through M_{SINK} is the sum of the currents through M_{N1} and M_{N3}, $I_{M_{SINK}} = I_{M_{N1}} + I_{M_{N3}}$. Replacing each current with the equation for the proper region gives the

equation in Equation 5.14. The V_{G3} term is a function of V_X and is therefore written as $V_{G3f(V_X)}$.

$$k_{N}\prime\frac{W_{SINK}}{L_{SINK}}V_X\left(V_{IN}-V_{TN}-\frac{V_X}{2}\right)= \\ \frac{k_{N}\prime}{2}\frac{W_1}{L_1}\left(V_{BIAS1}-V_X-V_{TN}\right)^2+\frac{k_{N}\prime}{2}\frac{W_3}{L_3}\left(V_{G3f(V_X)}-V_X-V_{TN}\right)^2 \tag{5.14}$$

Solving for V_{IN} Equation 5.14 becomes Equation 5.15.

$$V_{IN}=\frac{1}{\frac{W_{SINK}}{L_{SINK}}V_X}\left[\frac{1}{2}\frac{W_1}{L_1}\left(V_{BIAS1}-V_X-V_{TN}\right)^2+\frac{1}{2}\frac{W_3}{L_3}\left(V_{G3f(V_X)}-V_X-V_{TN}\right)^2\right] \\ +\frac{1}{\frac{W_{SINK}}{L_{SINK}}V_X}\left[\frac{W_{SINK}}{L_{SINK}}\left(V_XV_{TN}+\frac{V_X^2}{2}\right)\right] \tag{5.15}$$

Taking the derivative of V_{IN} with respect to V_X gives the results in Equation 5.16.

$$\begin{aligned}\frac{\partial V_{IN}}{\partial V_X} &= \frac{1}{2}\frac{\frac{W_1}{L_1}}{\frac{W_{SINK}}{L_{SINK}}}\left[\frac{-1}{V_X^2}\left(V_{BIAS1}-V_X-V_{TN}\right)^2+\frac{-2}{V_X}\left(V_{BIAS1}-V_X-V_{TN}\right)\right] \\ &+\frac{1}{2}+\frac{1}{2}\frac{\frac{W_3}{L_3}}{\frac{W_{SINK}}{L_{SINK}}}\left[\frac{-1}{V_X^2}\left(V_{G3f(V_X)}-V_X-V_{TN}\right)^2\right] \\ &+\frac{1}{2}\frac{\frac{W_3}{L_3}}{\frac{W_{SINK}}{L_{SINK}}}\left[\frac{2}{V_X}\left(V_{G3f(V_X)}-V_X-V_{TN}\right)\left(\frac{\partial V_{G3f(V_X)}}{\partial V_X}-1\right)\right]\end{aligned} \tag{5.16}$$

V_{G3} is a function of V_X, and replacing the value for $\frac{\partial V_{G3f(V_X)}}{\partial V_X}$ in Equation 5.16, gives Equation 5.17.

$$\begin{aligned}\frac{\partial V_{IN}}{\partial V_X} &\approx \frac{1}{2}\frac{\frac{W_1}{L_1}}{\frac{W_{SINK}}{L_{SINK}}}\left[\frac{-1}{V_X^2}\left(V_{BIAS1}-V_X-V_{TN}\right)^2+\frac{-2}{V_X}\left(V_{BIAS1}-V_X-V_{TN}\right)\right] \\ &+\frac{1}{2}-\frac{1}{2}\frac{\frac{W_3}{L_3}}{\frac{W_{SINK}}{L_{SINK}}}\left[\frac{\left(V_{G3f(V_X)}-V_X-V_{TN}\right)^2}{V_X^2}\right]+\frac{1}{2} \\ &-\frac{1}{2}\frac{\frac{W_3}{L_3}}{\frac{W_{SINK}}{L_{SINK}}}\left[\frac{-2\left(V_{G3f(V_X)}-V_X-V_{TN}\right)\left(gm_{M_{N1}}r_{O_{N1}}gm_{M_{N2}}r_{O_{N2}}-1\right)}{V_X}\right]\end{aligned} \tag{5.17}$$

The large forward gain through M_{N1} and M_{N2}, $gm_{M_{N1}}r_{O_{N1}}gm_{M_{N2}}r_{O_{N2}}$, makes the third term much larger than the other terms and Equation 5.17 can be simplified to Equation 5.18.

$$\frac{\partial V_{IN}}{\partial V_X} \approx \frac{2}{V_X}\left(V_{G3f(V_X)} - V_X - V_{TN}\right)\left(gm_{M_{N1}}r_{O_{N1}}gm_{M_{N2}}r_{O_{N2}} - 1\right) \tag{5.18}$$

Because M_{N2} is in saturation through its operation, $\left(V_{G3f(V_X)} - V_X - V_{TN}\right)$ may be replaced with $V_{dsat\_M_{N3}}$ of M_{N3}, and V_X may be replaced with $V_{BIAS1} - V_{dsat\_M_{N1}}$, both of which represent values which give Equation 5.18 its minimum value. Flipping this result gives the relation for $\frac{\partial V_X}{\partial V_{IN}}$, how much V_X changes for a change in V_{IN}. This result is shown in Equation 5.19.

$$\frac{\partial V_X}{\partial V_{IN}} \approx \frac{1}{\frac{2}{V_{BIAS1} - V_{dsat\_M_{M1}}}V_{dsat\_M_{N3}}\left(gm_{M_{N1}}r_{O_{N1}}gm_{M_{N2}}r_{O_{N2}} - 1\right)} \tag{5.19}$$

The larger the open-loop gain ($gm_{M_{N1}}r_{O_{N1}}gm_{M_{N2}}r_{O_{N2}}$), the less V_X moves. Optimizing of this circuit for linearity involves reducing $\frac{\partial V_X}{\partial V_{IN}}$ as much as possible. Using the fact that V_X moves very little for a change in V_{IN}, it can be shown that the current through the tail current source, M_{SINK}, is linearly related to the input voltage, V_{IN}. Assuming V_{DS} of V_{SINK} (V_X) is held constant, the long-channel current equation as shown in Equation 5.20 can be used to show the linear relationship between V_{IN} and $I_{IN}\prime$.

$$I_{DSSINK} = k_N\prime\frac{W}{L}V_{DS}\left(V_{IN} - V_{TN} - \frac{V_{DS}}{2}\right) \tag{5.20}$$

Because V_X, the voltage at the drain of M_{SINK}, is constant, the V_{DS} terms can be replaced in Equation 5.20 and Equation 5.21 results where $C1$ equals $k_N\prime\frac{W}{L}V_{DS}$ and $C2$ equals $k_N\prime\frac{W}{L}V_{DS}\left(V_{TN} + \frac{V_{DS}}{2}\right)$.

$$I_{DSSINK} = k_N\prime\frac{W}{L}V_{DS}\left(V_{IN} - V_{TN} - \frac{V_{DS}}{2}\right) = C1V_{IN} - C2 \tag{5.21}$$

From Equation 5.21 it is clear that, in the ideal, long-channel case with the drain of

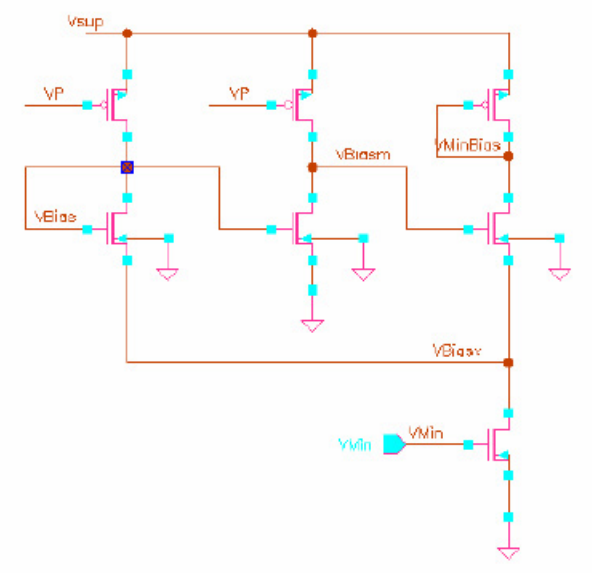


Figure 5.6: Circuit used to generate current to be subtracted.

M_{SINK}, V_X, held constant, the current I_{DSSINK} is linearly related to the input voltage V_{IN}.

The mechanism used here to fix V_X is gain boosting, and the gain-boosting method used here is well documented in [*Torralba et al.*, 2002].

The feedforward circuit uses the voltage to current converter to convert the duty cycle expressed in voltage to a duty cycle expressed in current. To do this, the base current must be subtracted. The current that needs to be subtracted is the current representative of a 100% duty cycle. This current is found by converting the voltage of the low (V_{min}) value of the input monostable block output square wave. The circuit used to generate this minimum current is shown in Figure 5.6. The right-most branch conducts a current representative of the minimum possible current and generates a $PMOS$ biasing value, $V_{MinBias}$.

To subtract this current a branch was added to the basic voltage to current converter. The circuit shown in Figure 5.7 has a branch added to the right-most side of the circuit shown in Figure 5.5. The right-most branch, through a current mirror, conducts the same amount of current as generated by the circuit shown in Figure 5.6. In the circuit shown in Figure 5.5, the current of interest is the current through M_{P3}. By supplying current

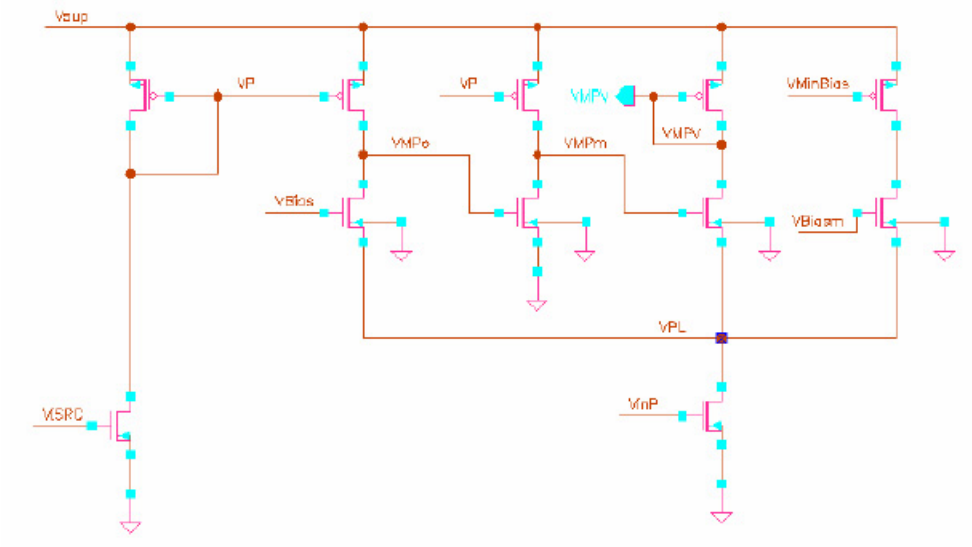


Figure 5.7: Voltage to current converter with current subtraction branch added.

through the added branch, the amount of current required to go through M_{P3} is effectively reduced by the supplied amount. In other words, the current through the extra branch is subtracted from what the current would be through M_{P3} without that branch.

Subtraction of the minimum current is carried out on conversion of both of the output signals of the input monostable block. The resulting currents must then be summed to generate the biasing value for the integrator $NMOS$ current sink. This is done using the circuit shown in Figure 5.8.

Feedforward gave simulated results within 1.5% of correct currents for tests done between $800MHz$ to $1.25GHz$. This was determined by sending a square wave of a known frequency through a differential inverter chain, averaging the outputs, and using the averages generated as the input to the voltage to current converter. The resulting currents were of the results mentioned above. Unfortunately 'close' is not good enough and the resulting sawtooths all were misaligned in a way similar to that in Figure 5.9. Sources of error were, as mentioned in previous chapters, largely due to switching errors, leakage, and channel length modulation effects. Simulations with process variation and environmental influences performanced similarly. The biasing values generated by the feedforward block were good enough to produce sawtooths that crossed or nearly crossed for all simulations, but none

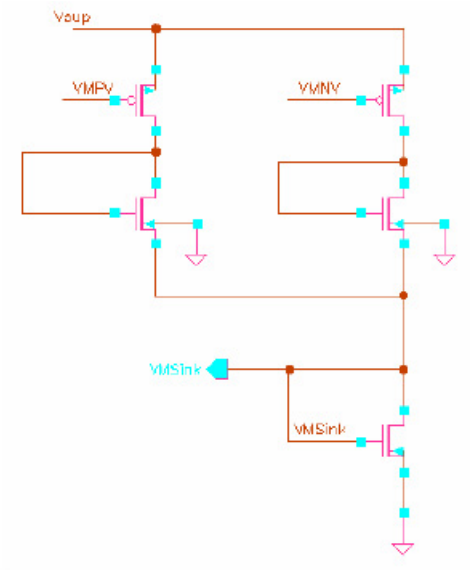


Figure 5.8: Current summing circuit.

produced sawtooths that were positioned directly on top of each other. What this means is that for input Gaussian jitter to the JAC circuit with maximum values above a certain amount, without further correction, this circuit will fail. The amount of total jitter allowable is $T - t_p$, $\frac{T-t_p}{2}$ in each direction. The total amount of jitter allowable is still $T - t_p$, but the jitter allowable in either direction is not equal. Figure 5.10 show the sawtooth crossing of a signal with no jitter shifted due to the sawtooths not oscillating around the same average. The ideal crossing is marked at $\frac{T-t_p}{2}$ between the maximum and minimum values of the jitterless sawtooth. With the unequal averages, Figure 5.10 shows that the resulting crossing is no longer at $\frac{T-t_p}{2}$, but at the value shown in Equation5.22, and the allowable jitter in the other direction will be as shown in Equation 5.23.

$$t_{JitterFF\_P} = \frac{\frac{V_{P\_AVE} - V_{N\_AVE}}{2}}{m2} \tag{5.22}$$

$$t_{JitterFF\_N} = (T - t_p) - \left(\frac{\frac{V_{P\_AVE} - V_{N\_AVE}}{2}}{m2} \right) \tag{5.23}$$

The pre-integrator feedforward block provides a quick way to get close to correct biasing upon start-up, no instability or build-up problems such as those possible with feedback

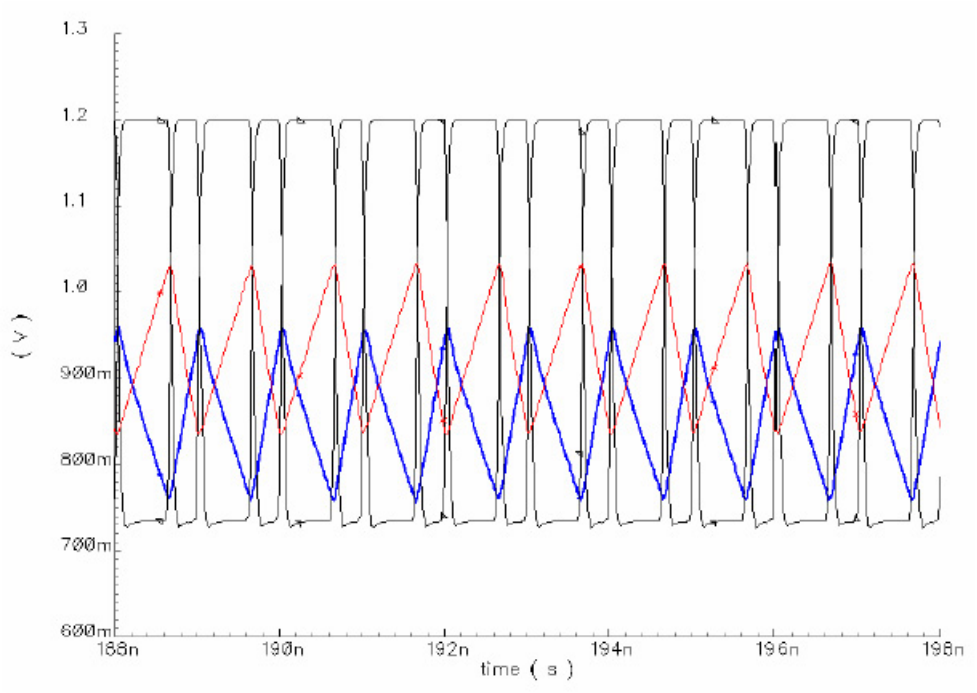


Figure 5.9: Output of integrator block with only feedforward biasing.

systems, and a simplicity of design. To improve upon the amount of jitter the circuit can process, an additional feedback block or feedforward block is necessary. Only one of these two blocks is necessary, but both will be introduced in the next sections.

5.4 Post-Integrator Feedforward and Feedback Blocks

Two methods of sawtooth signal correction, feedforward and feedback, are discussed in this section.

5.4.1 Post-Integrator Feedforward Block

Block-Level Design

The post-integrator feedforward processing, simply put, does nothing to correct the different averages of the sawtooth waveforms in the actual integrator block, but takes the two output sawtooth signals from the integrator and shifts them such that they are positioned

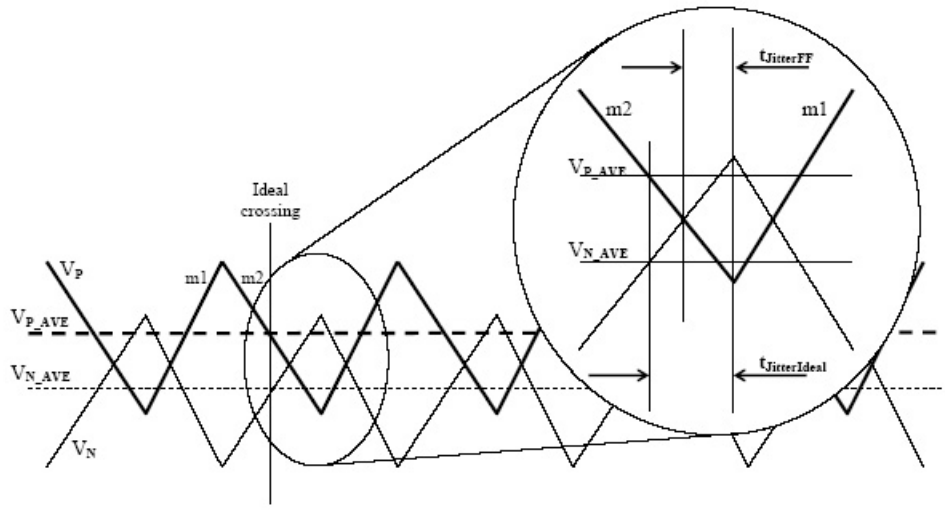


Figure 5.10: Output error of integrator block with only feedforward biasing.

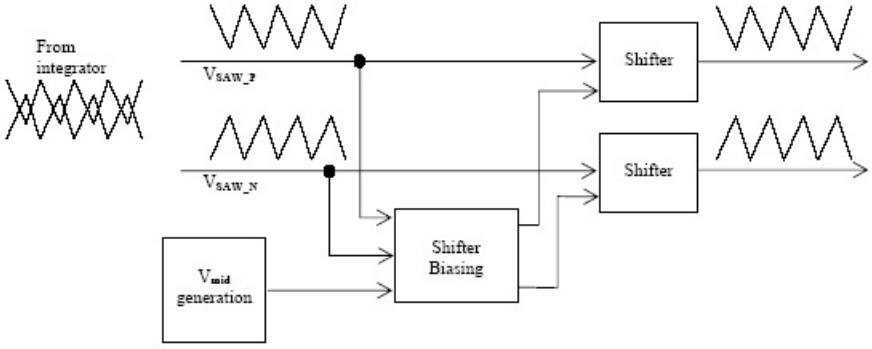


Figure 5.11: Post-integrator feedforward processing.

on top of each other. The two shifted signals are then the inputs to the level detector. Figure 5.11 shows this flow. The sawtooth signals from the integrator do not have the same average. Each signal goes to its own shifter and each shifter's shift amount is controlled by a control block. One of the inputs to the control block is V_{mid}, the value the integrator sawtooth would be oscillating around in the ideal situation.

The shifters used to shift signals in the JAC designed here have two inputs. One is the signal to be shifted and the other specifies the amount to be shifted. The output of the shifter is a version of the input signal which has been shifted and has a slightly reduced

peak-to-peak amplitude. The reduced amplitude does not greatly effect what the average of the signals would be if there was a perfect shift with no reduction in amplitude and is due, most notably, to channel-length modulation. The amount the signal is shifted is the voltage from the biasing value to GND if the $NMOS$ based shifter is being used, or the biasing value to VDD if the $PMOS$ based shifter is being used. The $NMOS$ shifter shifts signals down and the $PMOS$ shifter shifts signals up.

Figure 5.12 breaks down the biasing block. The shifter shifts V_{mid} (a DC value) down an amount equal to its output. The design of the shifter used here results in the shift amount being equal to approximately $\frac{V_{mid}}{2}$. The shift down of V_{mid} is shown in the leftmost column, *a.*, in Figure 5.13. The other signals are also shifted down the same amount as shown in column *b.*, in Figure 5.13. The signals in column *b.* are the signals out of the shifters shown in Figure 5.12. The output of these shifters is then averaged to get the signals in column *c.* of Figure 5.13. To understand the final column *d.*, it may be helpful to restate the meaning of the signals in column *c.*. $V_{P\_SH\_AVE}$ is the average of the shifted V_P signal. The distance between V_{mid_SH} and $V_{P\_SH\_AVE}$ is the amount that V_P would need to be shifted down more than the shift value $\frac{V_{mid\_SH}}{2}$ for the shifted signal to have its average on top of V_{mid_SH}. In other words, if V_P was shifted down by V_{mid_SH} plus the difference between $V_{P\_SH\_AVE}$ and V_{mid_SH}, its average would be on top of V_{mid_SH}. The trick here is to see that the necessary shift value is:

$$V_{mid\_SH} + (V_{P\_SH\_AVE} - V_{mid\_SH}) = V_{P\_SH\_AVE} \tag{5.24}$$

So, to get the average of the shifted value of V_P to be on top of V_{mid_SH}, V_P must be shifted by $V_{P\_SH\_AVE}$. Once the average of the shifted V_P signal is generated, that value, $V_{P\_SH\_AVE}$, can be used as input to the final set of shifters shown in Figure 5.11. To generalize, if the sawtooth average is too high, it will be shifted more, and if the sawtooth average is too low, a smaller shift value will be generated because the average will be lower.

As an aside, this design may also be implemented as a feedback block as shown in Figure 5.14.

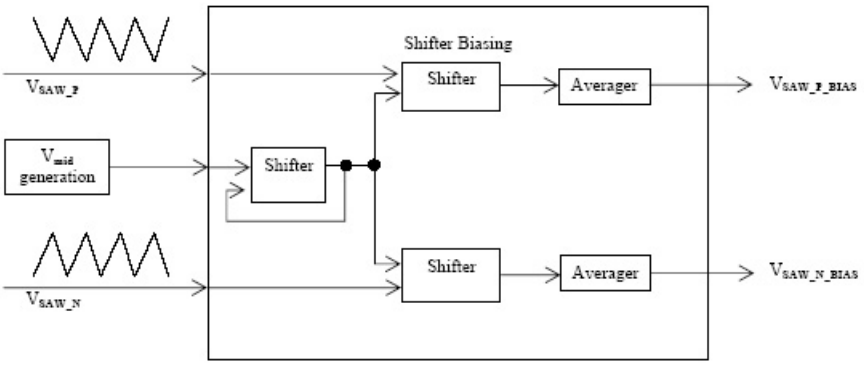


Figure 5.12: Post-integrator feedforward biasing block.

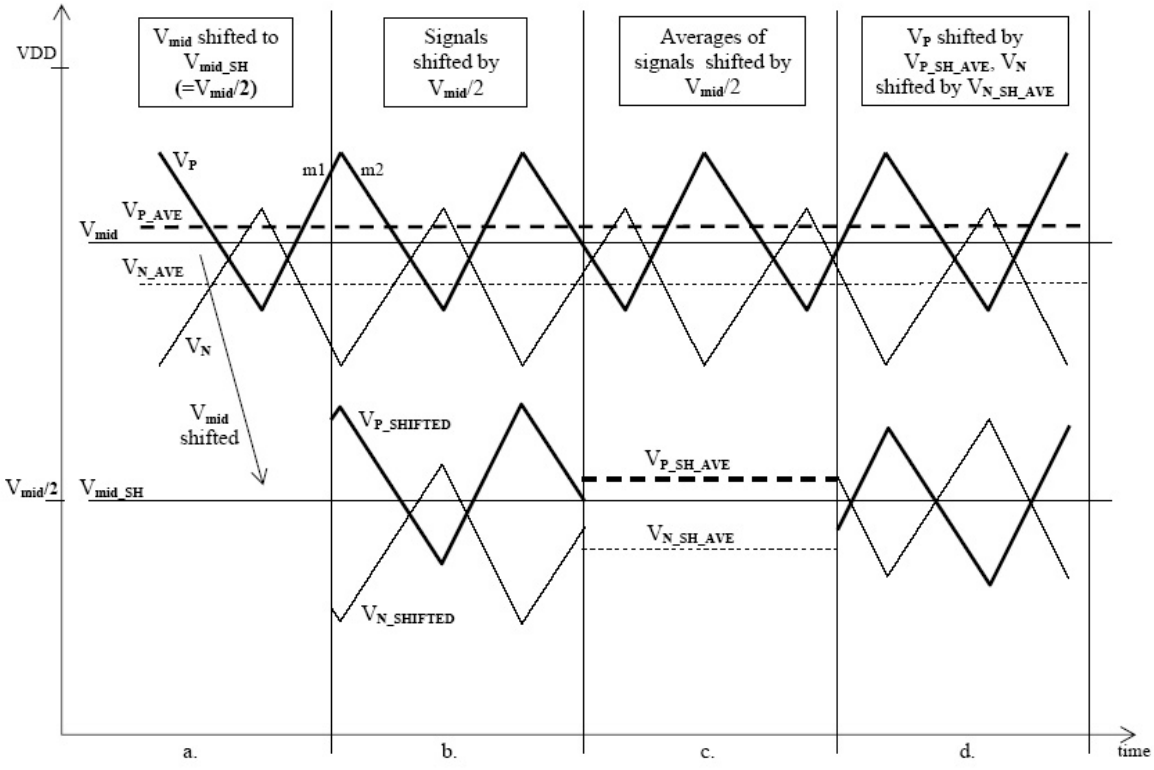


Figure 5.13: Signals at various places in post-integrator feedforward biasing block.

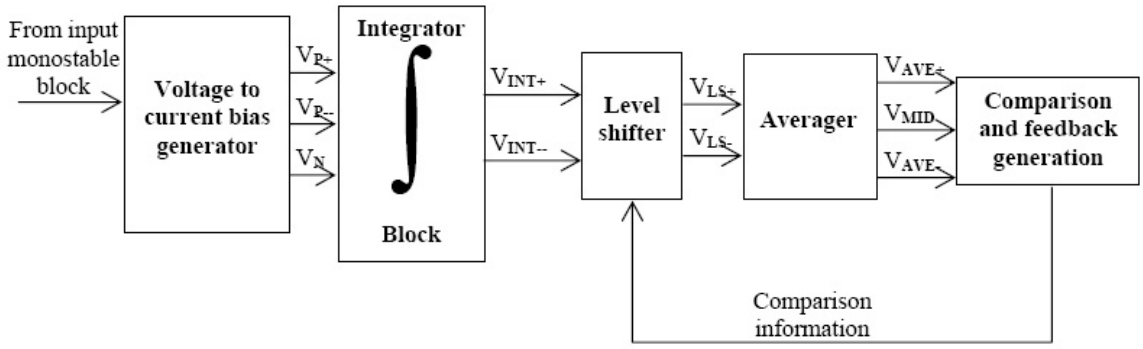


Figure 5.14: Block diagram for shifter feedback method.

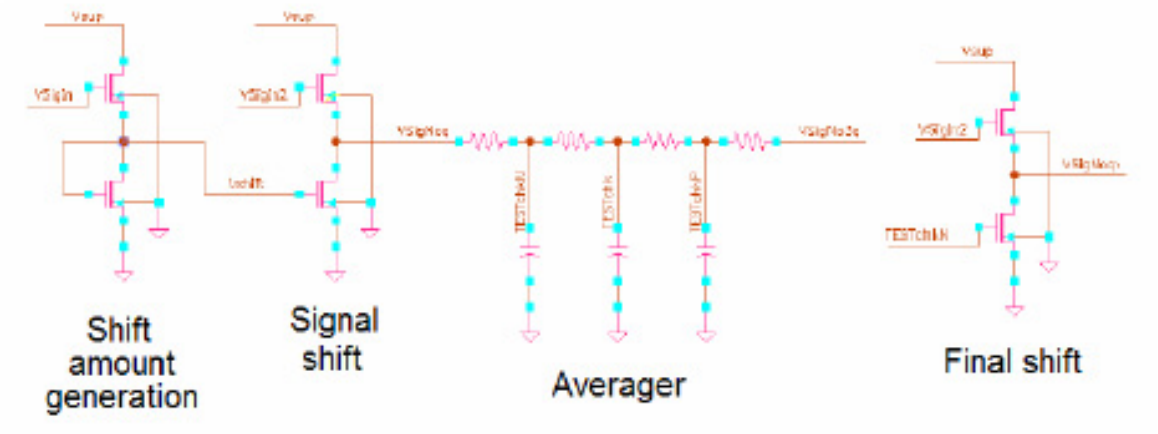


Figure 5.15: Circuitry of post-integrator feedforward biasing block.

Circuit Design

Figure 5.15 shows the implementation of the blocks in Figures 5.11 and 5.12.

One of the important functions of the shifters is to show the integrator a high resistive load and a small capacitive load. Using $CMOS$ provides the high input impedance, but, to provide the small capacitive load, the devices must be sized as small as possible. Using small devices make the devices more sensitive to velocity saturation and channel-length modulation and affect the amplitude of the shifted signal. The main design issue and a topic that, due to time constraints, will need to be pursued after this thesis is done, is how to better avoid loading of the integrator while reducing the effects of these two non-idealities. The design used here generates a smaller than ideal shifting value because the

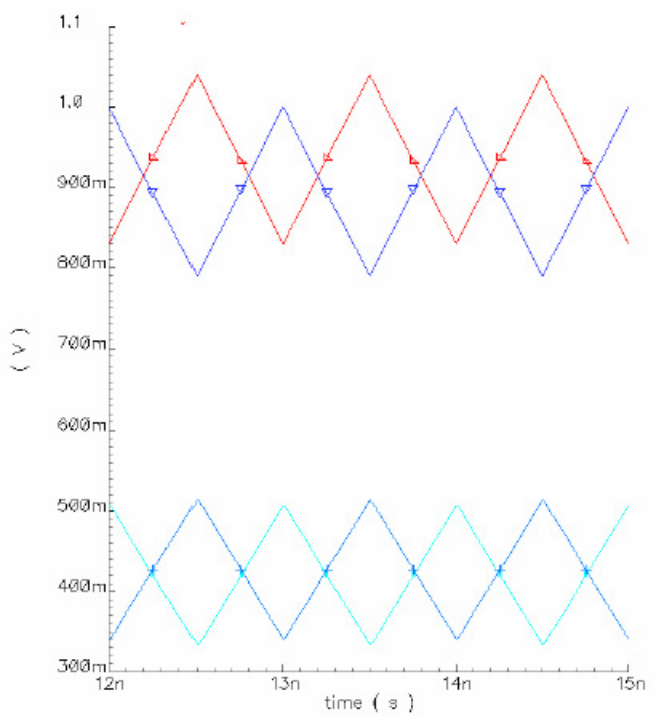


Figure 5.16: Simulation of output using post-integrator feedforward biasing block.

V_{DS} of the top $NMOS$ is smaller than the bottom $NMOS$ (See Figure 5.15, "Shift amount generator"). Because of this, the V_{GS} of the top $NMOS$ device is larger than the V_{GS} of the bottom device. The V_{mid_SH} value then is lower than the ideal $\frac{V_{mid\_SH}}{2}$ it should be and the "Final shift" is not as large as it should be. Even though, very good results were achieved using this method. A simulation through the entire system with no jitter is shown in Figure 5.16.

The post-integrator feedforward biasing block is not a stand-alone block and only has meaning when used in conjunction with the feedforward biasing block, integrator block, and input monostable block already introduced. Therefore no simulations beyond the one done in Figure 5.16 will be done here.

5.4.2 Post-Integrator Feedback Block

The idea with feedback is to make the integrator produce a correct output instead of, as the post-integrator feedforward block does, correcting an incorrect one. The function

of the feedback circuitry is to force the sawtooth signals to cross, preferably such that their averages are the same. Having this maximum overlap allows for the maximum input jitter if jitter is randomly distributed. The way to 'shift' the sawtooths to be on top of each other is to adjust the biasing to the integrator. By changing the bias values to the $PMOS+$ and $PMOS-$ inputs of the integrator each individual sawtooth signal can be moved up or down. Because the integrator's outputs are controlled by the values on the integrating capacitors, the mechanism to keep outputs centered on each other is different than a typical differential amplifier. Often the simplest way to adjust the common mode output value of a differential amplifier is to adjust the tail current source. Unfortunately, as shown in Chapter 2, adjustment to the tail current source alone is not enough to guarantee the ability to correct the basic equation values, $m1$ and $m2$, such that the equation would become true. Thus the need to control both $PMOS+$ and $PMOS-$. Though feedback was the original method of choice for adjusting the sawtooth voltages, a number of points eventually suggested that the purely feedforward solution needed to be seriously considered also:

1. The tail current sink of the integrator is always set to the sum of the $PMOS$ currents, and the ratio of $PMOS$ currents is the important issue when biasing the integrator. This means that changing the $PMOS$ current on one side effectively changes the behavior of both sides. This adds an additional challenge in that both sides must be adjusted at the same time and relative to each other in order to get the sawtooths to oscillate around the V_{mid} value.

2. The feedback loop is long, through multiple blocks, and has a small forward gain.

3. The $PMOS$ are sensitive enough that, until values settle, the output of the JAC will be worthless.

4. To make feedback work, its response has to be slow in order to avoid overshoot. This results in slow start-up and acquisition times.

The advantages the feedforward correction method has over feedback include the simplicity of its circuitry and theory, lack of instability issues, and minimal other non-ideality

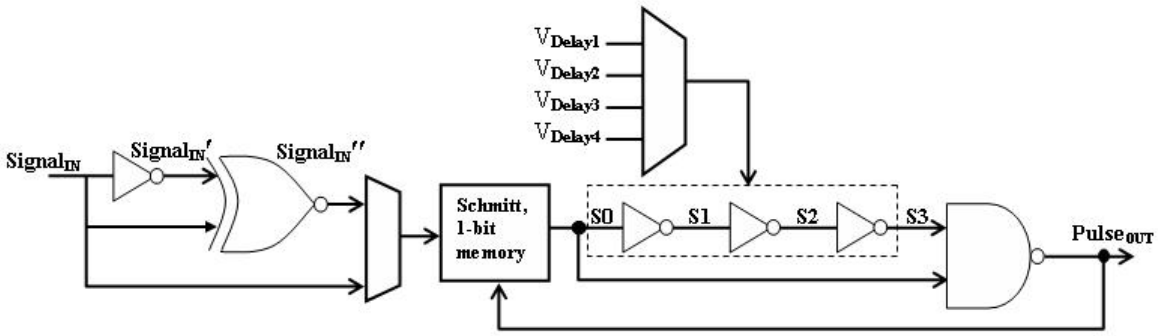


Figure 5.17: Options for four delays and two frequencies for eight possible duty cycle widths.

issues that make feedback challenging. The disadvantage to carrying out the adjustment using feedforwad is that, because the sawtooths will have a different range of movement at the output of the integrator, the effects of channel length modulation and maximum and minimum currents in the integrator itself may not be the same for the two waves, and this can lead to additional jitter. Feedback adds complexity to the circuit design, and introduces more possibilities for the $PMOS$ currents to not be constant. Because both methods use filters, both options have slow start-up and acquisition times. Because of the advantages of the purely feedforward circuit, the feedback circuit design will be introduced here but the final simulations will be done with only the pure feedforward design.

In the JAC system keeping t_p constant is very important. Any variation in t_p between clock cycles is propagated through the circuit and amplified at the output as jitter. For the system to work, t_p can be anything as long as it is unchanging between periods. The feedback path for adjustment to t_p would be from either the output of the integrator or the output of the level detector, both long paths which include multiple blocks. Because of the importance of keeping the value of t_p constant and the complexity of the feedback path, this option was investigated, but not polished. As a note, the input monostable could be set up using a DLL-like delay control, but adjusting the delay through the chain would still have the above named issues.

Other values that can be adjusted to make the basic equation hold true are $m1$ and $m2$, and those are controlled by the integrator block. Again there was more than one path

that could be used as a feedback control loop; Using the output of the integrator, or using the output of the level detector. It was found that with the non-idealities that could occur in the system, it was possible to have a situation where the outputs of the integrator block were not crossing and no output pulses were being generated by the level detector block. Without proper output of some sort, the path from the output of the level detector was not much use as a feedback path. The feedback path from the output of the sawtooth generator block was determined to be the most practical in a feedback system.

Block-Level Design

The feedback block design tested for this thesis uses much of the same circuitry as the post-integrator feedforward biasing block. It uses the averages of the output of the integrator and the ideal average to determine which signals need to be lowered and which need to be raised. That information is then used to adjust the current through the $PMOS$ current sources of the integrator.

Figure 5.18 shows the block diagram of the feedback circuitry. The feedback path of Figure 5.18 starts at the upper left-hand corner of the diagram. As done in the post-integrator feedback block, the outputs of the integrator are shifted and averaged to give information on whether the average of a sawtooth is too high or too low. If an average is too high, that means that the integrator's $PMOS$ is conducting too much and its biasing value needs to be raised to reduce current. It is useful that biasing needs to be moved higher if the average is too high and lower if the average is too low. The average can be used directly as the controlling signal back to the biasing circuitry. The current mirror on the left side of the feedforward biasing block in Figure 5.7 (which is the same as the second voltage to current converter in Figure 5.18), adjusts the current through its $PMOS$ through the $ViSRC$ input. If $ViSRC$ is raised, the current increases and VP drops. The increase in current is mirrored to the leftmost branch of the actual voltage to current converter, and, because of the current increase and $Vbias$ being a fixed value, VPL is dropped to increase the V_{GS} and accommodate the new current value. Because the $NMOS$ in the tail current sink for the voltage to current converter is in triode, changing its V_{DS} will affect the how

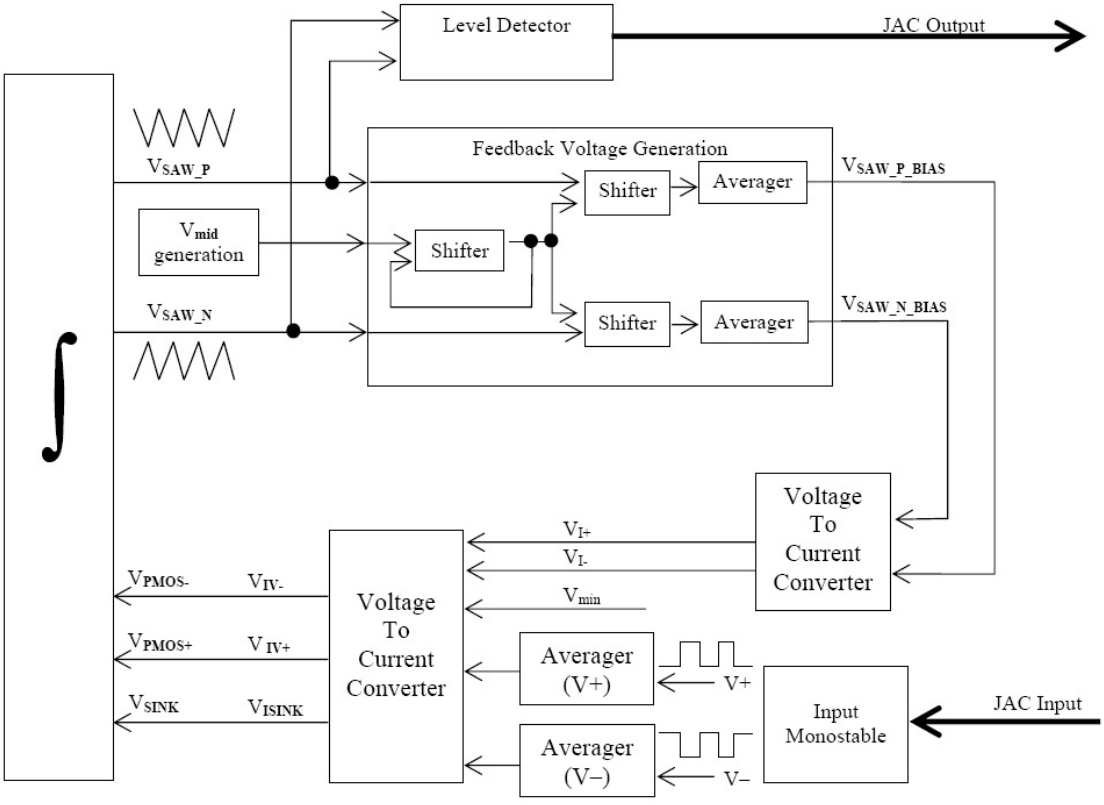


Figure 5.18: Feedback block level diagram.

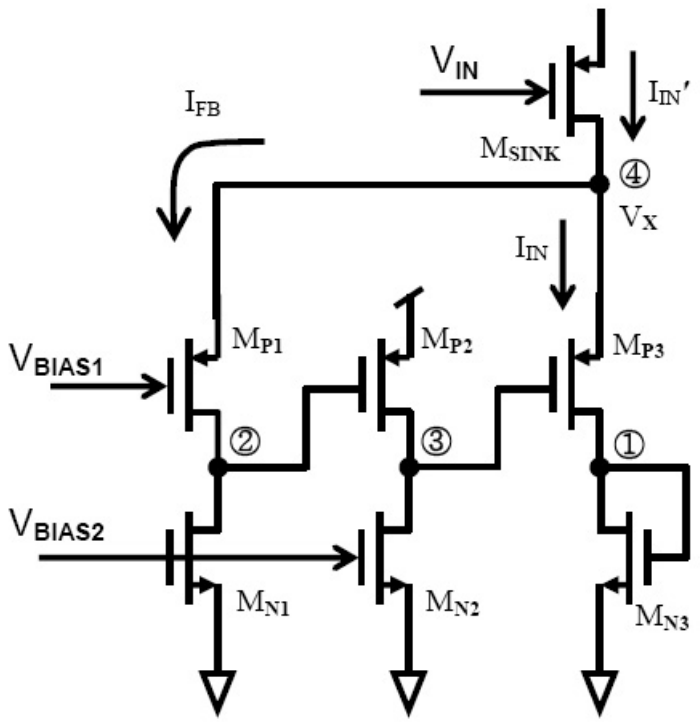


Figure 5.19: PMOS version of voltage to current generator.

much current it sinks. With an increase in $ViSRC$, more current is conducted through the leftmost branch, and because VPL drops, less current is conducted through the tail. This results in a reduction in current through the transistor which sets the biasing value for the integrator's $PMOS$ and a lowering of the output average voltage of the sawtooth. Note that $ViSRC$ must rise for a lowering of the average. That is where the first voltage to current converter becomes necessary. The first voltage to current converter is uses $PMOS$ as its active devices and therefore produces and output that rises when the input falls, and falls when the input rises - exactly what the second voltage to current converter needs. Because of the effective gain produced in the first voltage to current converter by the increase in current in the leftmost branch, and the decrease in voltage at VPL, there is a chance for instability in this loop. But, because of the low-pass filter in the feedback path, transitions occur so slowly, this is not a problem. What is a problem is how long it takes to settle.

Circuit Design

The feedback circuitry has already been introduced in the description of the block diagram and in earlier sections. The final topics of the design which need to be discussed are some of the design decisions, the stability of the feedback loop, and how the integrator $PMOS$ are actually correctly set.

The post-integrator feedforward and feedback blocks both use information on whether an integrator output average is above, below or near another value. The value the output average is compared to could be the average of the other integrator output or a fixed value which has been determined as a good central value for the signals to be oscillating around. The method used in this work is comparison with a fixed value. The first step in comparing values is to get their averages. The method used here to generate the averages of both input signals $VSigN$ and $VSigP$ is the low-pass filter shown in Figure 5.20. The input to this RC network is the output of a level shifter so as not to load the integrator. By adjusting the relative sizes of the inner and outer resistors, the difference between the averages can be adjusted. A large inner resistor size and smaller outer resistance size, will divide the input signal in such a way that $VNave$ and $VPave$ will be farther from $Vmid$ and each other. Having smaller outer resistors also increases the frequencies allowed through to $VNave$ and $VPave$ and makes them less flat. The center value, $Vmid$ is a virtual GND in that the inputs, $VSigN$ and $VSigP$, are moving in opposite directions at the same rate at all times ideally. In reality, process variation and other non-idealities make $VSigN$ and $VSigP$ not perfect mirrors of the other, and $Vmid$ is not a perfect virtual GND. But with filtering well below the frequency of $VSigN$ and $VSigP$, $Vmid$ can be made to be kept, to a high degree, stable. Making the inner resistor sizes smaller and the outer resistance sizes larger, will reduce the noise on $VNave$ and $VPave$, but will also divide the input signal in such a way that $VNave$ and $VPave$ will be closer to $Vmid$ and each other, therefore making the relative relationship less clear. The final ratio selected gave approximately $100mV$ between $VNave$ and $VPave$, and a steady $Vmid$, $VNave$ and $VPave$.

As an aside, an argument for using V_{MID} as the comparison value can also be made.

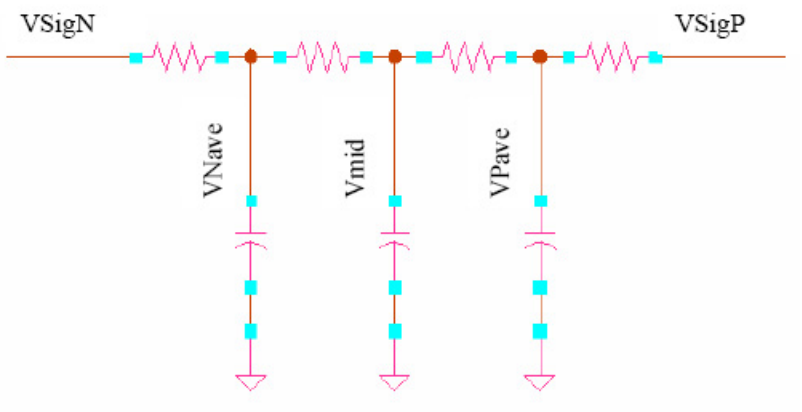


Figure 5.20: Circuit for generation of average and center values of sawtooth signal.

The justification for using V_{MID} is based on the assumptions that, if the $NMOS$ integrator tail biasing circuit (which sums the two $PMOS$ currents) is near the integrator, that the behavior of the $PMOS$ and $NMOS$ in the summer and integrator will have similar process variation and behavior. And, as another requirement, the size of the integrator switches must be large enough to guarantee that they can conduct the amount of current that goes through the tail current source without forcing the tail current source into triode. If these conditions are met, then the amount the the two sawtooths will be displaced off of each other will be dependent solely on the duty cycle of the two waveforms. For example, take the case where, for some reason, more current is supplied than ideally would be expected on a rising slope, and the same amount of extra current is sunk than ideally expected on a falling slope. Also assume that the extra supplied and the extra sunk is a constant amount per unit time. This is clearly an oversimplification, but makes it possible to illustrate a point. In this case, a 50% duty cycle will return to its original height or voltage at the end of a period because an equal amount of current was added in excess on the rising slope as was sunk on the falling slope. But, as in most cases in usual operation of a JAC, the duty cycle is not 50% and because of this, over the course of a period, the sum of the extra current from the rising slope will not equal the sum of the extra current sunk on the falling slope and the sawtooth will be displaced. The reason the sawtooths do not diverge to the rails is that, first of all, the pre-integrator feedforward circuitry works within approximately 0.5% to 1.5% error, so the currents are close to the ideal currents, but that, also, as a sawtooth

rises, the V_{DS} value across the $PMOS$ is reduced and the current pulling the sawtooth up is reduced. The same phenomena occurs when the sawtooth drops. This "load-line" effect is a useful self-correction mechanism. When the initial biasing is close to ideal as is the pre-integrator feedforward biasing is, and the difference between the averages of the sawtooths is largely caused by causes other than the biasing, one sawtooth will be below the ideal average and one will be above. When the displacement is extreme and the sawtooths are completely out of useful range, they will still be on either side of a correct center value and, as they are adjusted in, their average will also be adjusted such that, by the time the final adjustment has occured, the center value and their biasing will be correct. An example of a sawtooth with only initial biasing was shown in Figure 5.9. If the initial biasing is not well done, comparing the two sawtooths to their mid-value can be dangerous. In early designs, before proper biasing was available for the integrator, both sawtooths sometimes were above the ideal value and sometimes both were below. In either of these cases, if the mid-value was used, the sawtooths would actually be moved away from their ideal average. For safety sake, the post-integrator feedforward and feedback blocks in this work compare the integrator outputs to a generated value and do not use V_{MID}.

Another design decision was whether to adjust the bias values to the $PMOS$ before summing them to generate the tail current source current, or after. If the $PMOS$ bias values are adjusted before they are summed, then the adjustment is, in effect, just changing the ratios of the two $PMOS$ currents. Adjusting the $PMOS$ values after the integrator tail current has been determined is equivalent to individual transistor adjustment. In retrospect, the later may have more potential for success than initially thought. The later option allows adjustment to be done without attention to how it would affect the other $PMOS$ and the rest of the circuit. It also has a shorter feedback path and a much easier circuit to evaluate. An early design decision led to the investigation of the first option. Further research into the second option would be worthwhile.

The stability of the feedback loop is guaranteed by having the averager circuit in the feedback signal path. The slow response of the low-pass filter introduces a low-frequency pole which starts to drop the magnitude of the amplitude from a small frequency. The slow

response to change in the output of the integrator is also what makes this feedback system work. Because the filter output changes much slower than the output of the integrator, the integrator's new values have time to propogate back around before the feedback value can change more than a very small amount. Once the average of the integrator's output matches the value to which it is being compared, adjustments stop and the value stays constant. The drawback of this method of feedback is that quick adaption to new frequencies is impossible.

As with all the other blocks, this block is also sensitive to supply and GND noise as well as process variations, intrinsic noise, coupling and other non-idealities. In this case again, adding capacitors from the gates to a supply can keep V_{GS} a bit more stable and improve noise performance. Other design issues that should be pursued is replacing the simple shifters and current mirrors with circuits more impervious to noise.

Chapter 6

Results and Conclusions

6.1 Introduction

The basic function of the JAC is to generate a low-jitter square wave from a jittery sinusoidal or square-wave input signal. This chapter tests the jitter reduction performance of the purely feedforward JAC design for typical, slow, and fast corners, and supply and intrinsic noise. The results are analyzed and discussed, and a list of future topics is presented.

6.2 Simulation Results

6.2.1 Noise and Jitter Simulation Results

Figure 6.1 shows output jitter in *ps rms* for a $1GHz$ input signal. For all simulations, approximately 400 consecutive input edges that caused output crossings were watched. With no jitter on the input and no supply noise, simulation gave a perfect output with zero jitter. Adding supply noise increased output jitter noticably. This jitter came largely from t_p variation and the effects of the noise on the integrator. Note that, at approximately $200mV$ of supply noise and $320ps$ of peak-to-peak input jitter, the slope of the graph starts to increase quickly. As noted in Chapter 2, input jitter greater that $T - T_p$ can cause the system to fail, and, because the duty cycle is 65%, or $650ps$ for the $1GHz$ input signal, the

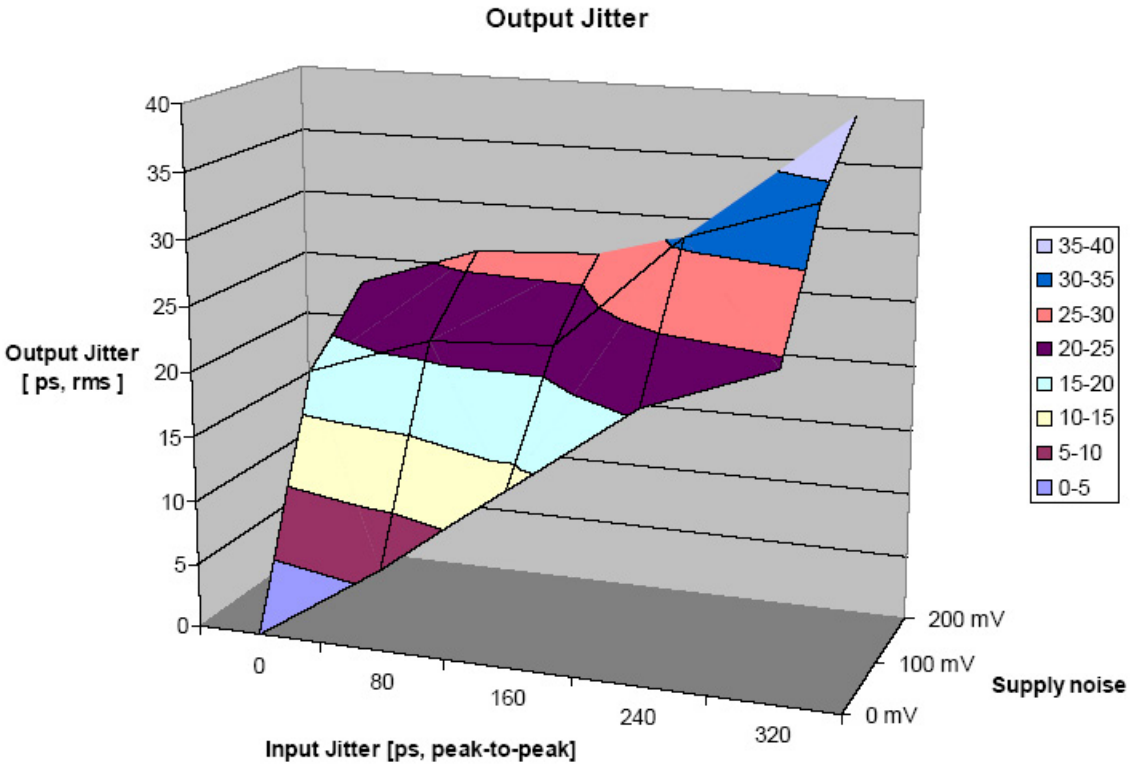


Figure 6.1: Output jitter for various input jitter and supply noise values.

allowable jitter is approximately $350ps$. With jitter added due to supply noise and taking rise and fall times into account, the limit is reduced even more. In Figure 6.1 the limit is around mid to high $200ps$ range. When this limit has been exceeded, errors on the output increase because of non-crossing outputs. Removing the output failures, the non-crossing outputs, from the data still shows jitter reduction, but there is some corruption due to the system trying to correct itself. The data for Figure 6.1 is in Table 6.1.

	Input Jitter				
Supply noise	0	80	160	240	320
$0mV$	0	6.31	13.56	20.68	24.33
$100mV$	18.05	21.32	21.80	30.77	33.9
$200mV$	22.88	26.25	26.76	20.72	38.60

Table 6.1: Figure 6.1 data.

The results here were not as good as the original goals of this project. Much of the

jitter comes from t_p variation. Reducing the length of the input monostable improved t_p accuracy almost proportionally. In other words, removing one of the three "chain links" (a delay block, pulse generator block, and flip-flop), improved t_p variation by approximately one-third. If the GND for the circuit is relatively quiet, reducing t_p to 20% or 30% is a viable solution. But, if GND is noisy, the integrator will see that noise directly on one side of its output and the circuit will not perform well. For a noisy GND, a longer t_p provides a steep slope for the crossing signals and better results. Another way to reduce GND noise problems is to keep the gate-source voltage to the integrator current sink constant.

Some of the circuitry in this design, such as the current mirror in the voltage to current converter and the shifters are actually single input and output circuits. To get differential circuit type common mode rejection, they were used in pairs and a common GND and supply was used for each pair. Using these blocks in pairs did successfully nullify the noise errors.

As shown in Figure 6.1 and Table 6.1, output jitter is less than the input jitter. Looking at the output jitter verses the input jitter as a straight ratio and in dB ($20Log\left(V_{OUT}/V_{IN}\right)$) gives some more information. Figure 6.2 and Figure 6.3 show the straight ratio of output jitter to input jitter and dB of that value respectively. The sharp spike at $200mV$ supply noise and $400ps\ peak-peak$ shows a failure. Also note that it looks like the ratio improves as input jitter increases. This is because, when there is very little input jitter, there can be very little improvement. Of these graphs, the most informative point is, therefore, the lowest point on the graph. That point shows the actual ability to reduce jitter. Table 6.2 shows the best performances for each amount of supply noise.

Supply noise	Best Performance (dB)	Best Performance (ratio)
$0mV$	-16.20	6.46
$100mV$	-11.60	3.80
$200mV$	-13.31	4.63

Table 6.2: Best performances for various levels of supply noise.

The above results were found running the simulations using typical device values. Running the simulations using the fast and slow corners without supply noise gives the results

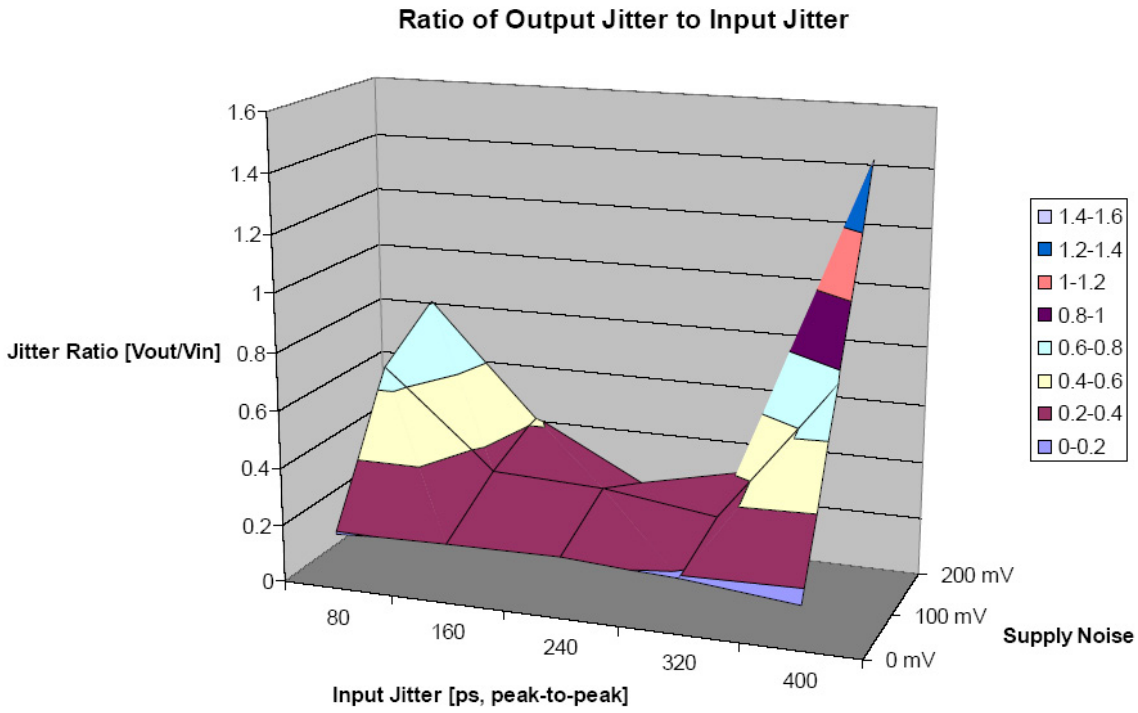


Figure 6.2: Ratio of output jitter (*ps rms*) to input jitter (*ps rms*).

shown in Table 6.3. The slow process corner failed consistently. As shown in Figure 6.4, the higher V_T reduces current and slows slopes so much that the width of the input squarewave's duty cycle is not enough time to allow the wave to rise and fall. Increasing the current to make the slow process corner succeed will affect other corners' performance. The solution to this problem is to add control which automatically adjusts current so the sawtooth will always swing the maximum it can swing. This topic is listed below under future topics. The fast corner succeeds where the typical corner fails because t_p is reduced, and, therefore, $T - t_p$, the amount of jitter that can be corrected, increases.

6.2.2 Process Variation

Monte Carlo simulations were run with $0ps$, $40ps$, $80ps$, $120ps$, and $160ps$ of input jitter. All widths and lengths of devices, resistors and capacitors were set to vary $\pm 10\%$, and ten runs at each input jitter were made. Figure 6.5 shows the lower limit of the sawtooth Monte

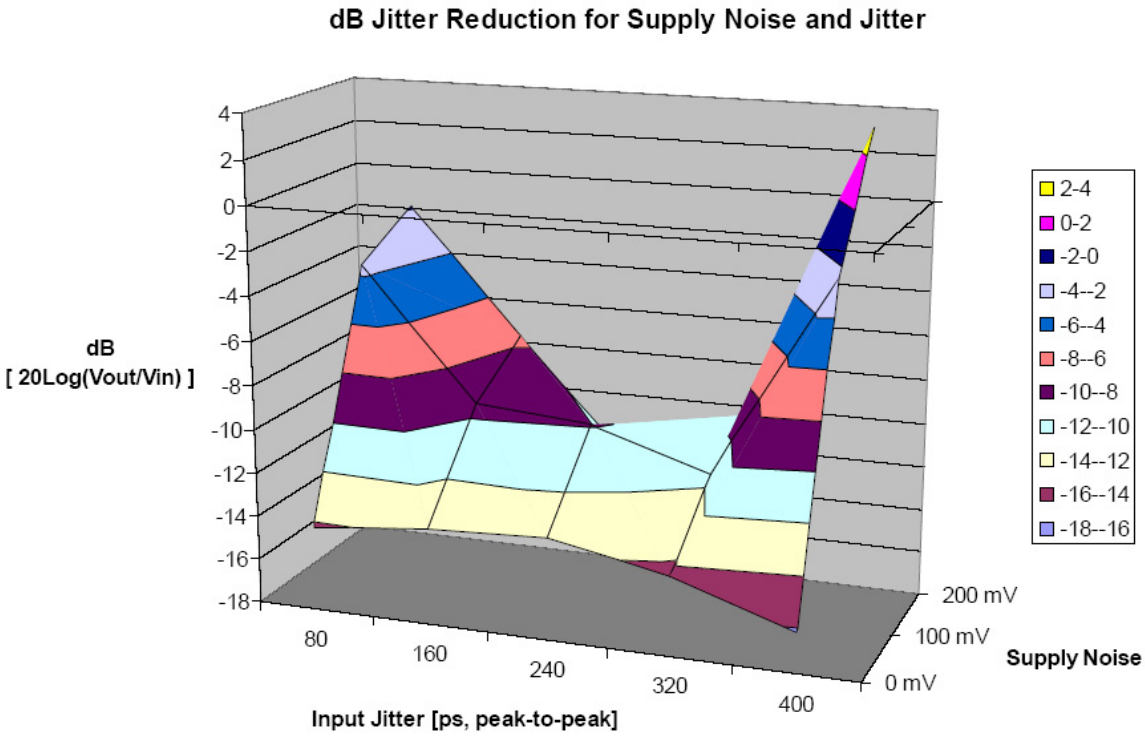


Figure 6.3: dB of output jitter to input jitter ratio.

Input Jitter (pk-pk)	Input Jitter (rms)	Output Jitter Fast Corner (rms)	Output Jitter Typical Corner (rms)	Output Jitter Slow Corner (rms)
$0ps$	$0ps$	$1.01ps$	$\approx 0ps$	Failed
$80ps$	$32.16ps$	$16.96ps$	$6.31ps$	Failed
$160ps$	$63.95ps$	$31.30ps$	$13.56ps$	Failed
$240ps$	$95.12ps$	$46.25ps$	$20.68ps$	Failed
$320ps$	$126.70ps$	61.35	$24.33ps$	Failed
$400ps$	$159.08ps$	75.72	Failed	Failed

Table 6.3: Effects of corners on output jitter.

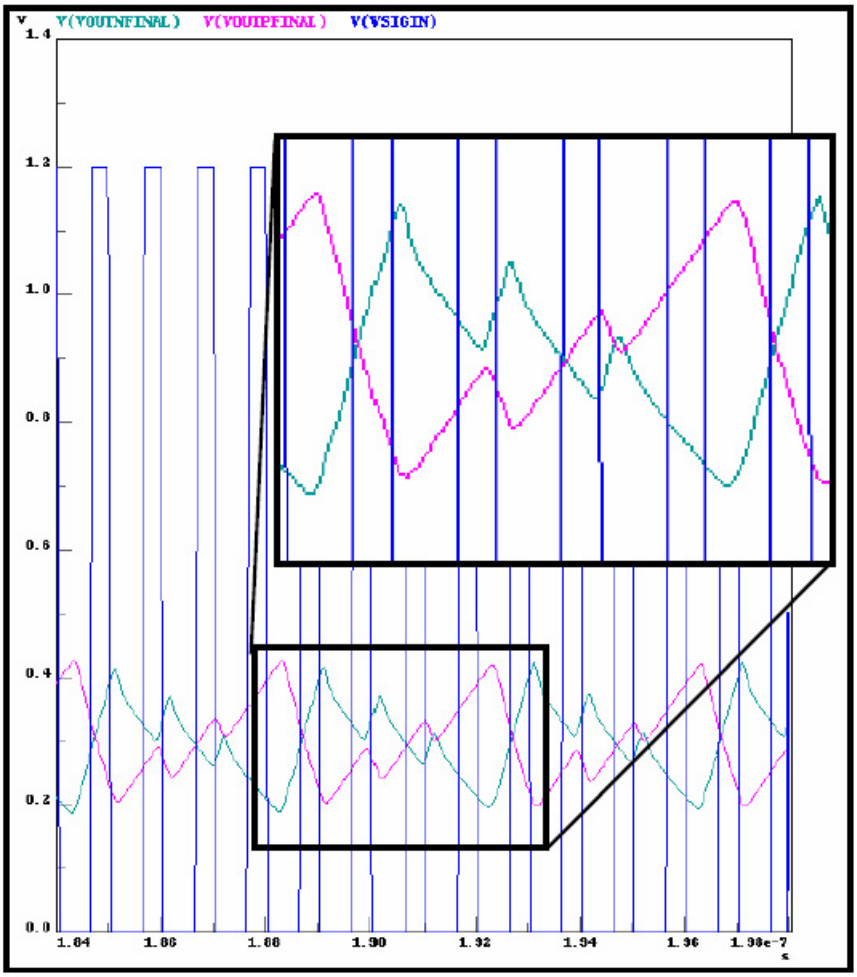


Figure 6.4: Slow corner failure.

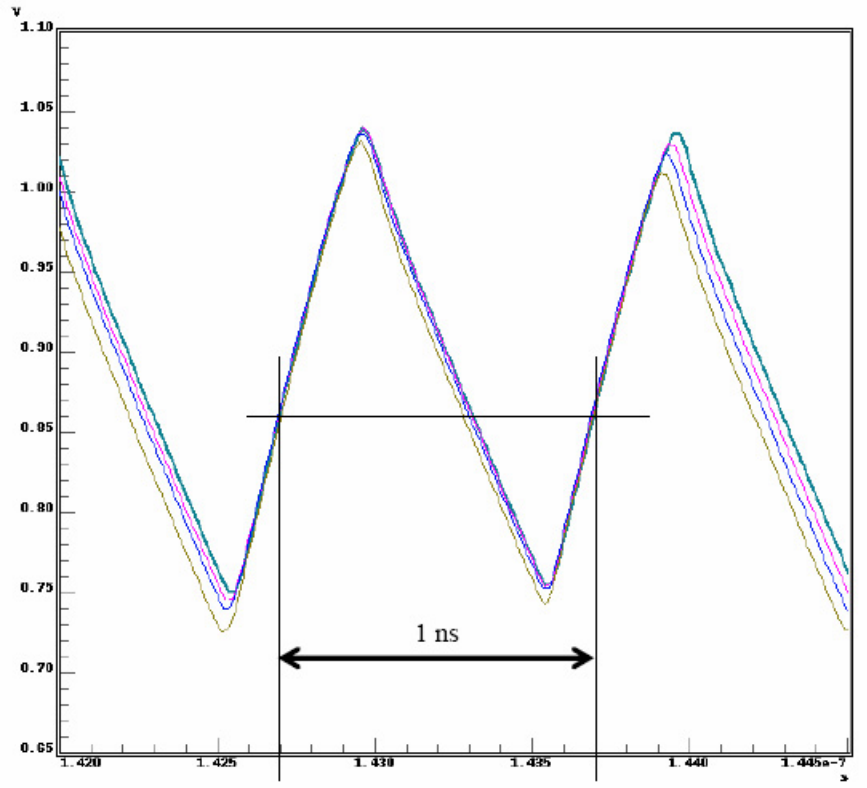


Figure 6.5: Lower limit of Monte Carlo simulation of sawtooth with 0 to 160ps of jitter.

Carlo simulation with 0ps, 40ps, 80ps, 120ps, and 160ps of jitter on input. Note that the crossing slope (the rising slope in Figure 6.5) is approximately equal among all amounts of jitter, but the slope that moves with jitter clearly is moving. Behavior of the important slope is minimally effected by process variation.

6.2.3 Intrinsic Noise

Intrinsic noise for fast, typical, and slow corners was analyzed. Calculations were then done using Equation 6.1, which was derived in Chapter 2, to find error on the crossing of the input to the level detector. That means that the signal has been shifted by the post-integrator correction block. The system that was evaluated had a $1GHz$ input signal and the duty cycle generated by the input monostable is 65%. The slopes of the shifted sawtooths are approximately $830 \times 10^6 \frac{V}{s}$ on the crossing slope. Results are summarized in Table 6.4.

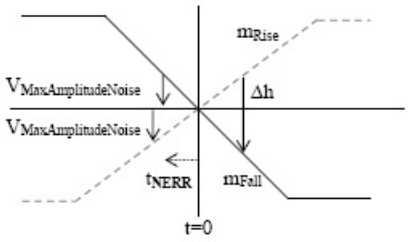


Figure 6.6: Worst case crossing timing error.

$$t_{NERR} = \frac{\Delta h}{m_{Rise} + m_{Fall}} = \frac{2 \times V_{MaxNoise}}{m_{Rise} + m_{Fall}} \tag{6.1}$$

Corner	N Ave Noise (rms)	P Ave Noise (rms)	Average Jitter (pk-pk)	N Max Noise (rms)	P Max Noise (rms)	Max Jitter (pk-pk)
Fast	$1.6mV$	$1.7mV$	$1.41ps$	$3.1mV$	$3.7mV$	$2.90ps$
Typical	$1.35mV$	$1.5mV$	$1.21ps$	$2.4mV$	$2.9mV$	$2.26ps$
Slow	$1.2mV$	$1.4mV$	$1.11ps$	$2.3mV$	$2.7mV$	$2.13ps$

Table 6.4: Jitter caused in JAC from average and maximum intrinsic noise.

A shorter input monostable was used in this design which reduced the number of devices in the signal path. The input monostable used in Chapter 3 used three delay segments whereas this design only uses two. This reduced the number of stages by ten, and reduced the t_p variation by approximately $\frac{1}{3}$ since the number of devices have been reduced by $\frac{1}{3}$.

6.2.4 Power

Power is reported here but is not a design aspect that was paid much attention to. In almost all blocks, a higher current improved performance, but, when deciding currents, a balance was taken between the performance and power and a center value was taken. Designing for lowest power or highest performance would have led to different sizing.

6.3 Future Topics

This section discusses topics that will be researched in the near future.

6.3.1 Start-up Time

Figure 6.7 shows two delays that need to be addressed in future research. Figure 6.7 shows the output of the integrator (top set of sawtooths), and the output of the post-integrator feedforward block (lower set of sawtooths). At start-up, it takes approximately $40ns$ for signals on the integrator to arrive near their final values. This is the time that it takes for the integrating capacitors to be charged up and find equilibrium. Another slow adjustment can be seen on the lower set of sawtooths from $40ns$ to approximately $170ns$. This is due to the large sizes of resistors and capacitors in the low-pass filter used in the post-integrator feedforward block. The value generated to be used as a comparison values in the post-integrator feedforward block could be used to initialize the sawtooths and averaging blocks when the sawtooth and average values are too far from

A value for use in the post-integrator feedforward block generates a value to compare the sawtooths against. The value is the "'ideal"' value and could be used upon start-up to set averages and initial sawtooth values. This circuitry would greatly reduce start-up time and would not be dependent on frequency or duty-cycle.

6.3.2 Filters

One of the original goals of this project was to build the circuit without using filters. The inspiration for this goal was the limit on performance introduced by the filters in this design.

Corner	Power
Fast	$7.1e-03$
Typical	$5.7e-03$
Slow	$4.5e-03W$

Table 6.5: Power.

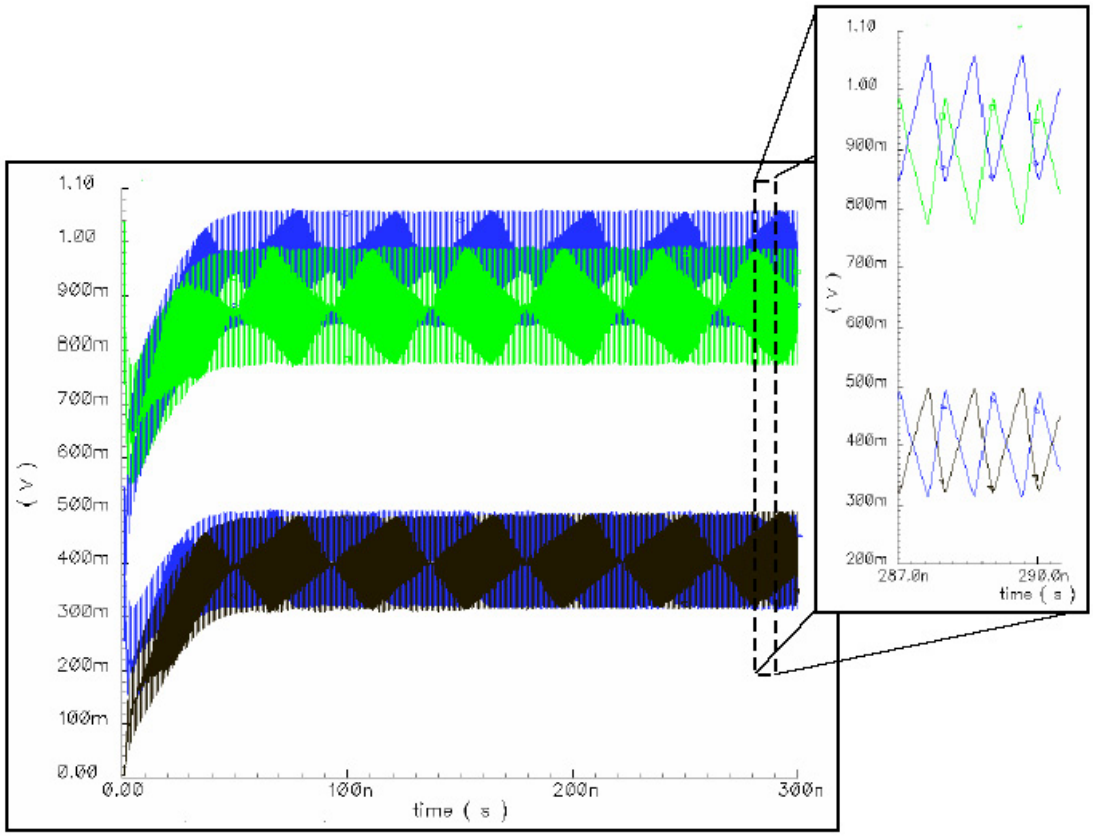


Figure 6.7: Start-up delays.

Just as with a PLL, the two low-pass filters used in the JAC determines both performance and start-up or adjustment time. When larger resistors and capacitors are used in the filters, the less the circuit was sensitive to long-term jitter, but the slower the start-up time and adjustment to new frequencies. The larger resistors and capacitors also take area. Filters are used in the JAC in the voltage to current converter and the post-integrator feedforward circuit, both of which are effected by the delay and passed frequencies.

A number of ideas were explored, but none came to fruition by the time of this writing.

6.3.3 Automatic Delay Adjustment

For a wideband JAC design, both t_p and the amplitude of the output of the integrator need to be adjustable. To avoid needing an extreme amount of current through the $PMOS$ or tail current source of the integrator, t_p should be somewhere between 25% and 75% of the period. Also, as the frequency increases, the sawtooth out of the integrator will be charging and discharging for less time and therefore produce a sawtooth with a smaller amplitude. A smaller amplitude makes it more difficult to get the most accurate reading on the crossing point of the sawtooths. To keep the amplitude of the sawtooths high, and therefore the accuracy of the level detector high, an adaptive current control mechanism is needed.

A mechanism for setting the width of t_p was introduced in Chapter 3, and the filter used in the post-integrator feedforward block can be used to determine amplitude of the integrator output sawtooths. Figure 6.8 is repeated here for convenience. Note that the volatge divider between the two sawtooths ($VSigNoq$ and $VSigNo3q$) divides according to the ratio of the outer resistor and inner resistor, the center value being a virtual GND when not in start-up or a frequency transition. Using a combination of either the averages found by the filter ($TESTchkN$ and $TESTchkP$), the center value, and/or the divider ratios, the amplitude of integrator sawtooths can be estimated. The estimation can then be used to either increase currents in the integrator or amplify the signal as it is put through the post-integrator feedforward circuitry.

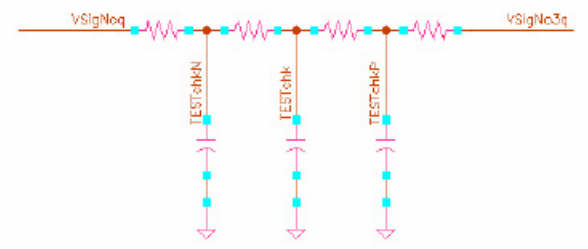


Figure 6.8: Filter used in post-integrator feedforward block.

6.3.4 Applications

There is a wide range of applications appropriate for the JAC circuit. One that the author is especially interested in is clock distribution. Because the JAC can multiply and clean a signal, clock signals could be distributed using a lower frequency sine wave or other The challenge here is the synchronization of signals across a chip or block. The phase correction block, as well as the JAC itself, are larger than envisioned and one of the goals from here on out is to further simplify and reduce the size of both circuits.

Other applications that could be targeted are any that need a clean, jitter free signal. Clock recovery from data, analog to digital and digital to analog conversion clocks, and demodulation local oscillators are some examples.

6.3.5 Design Topics

The sizing of the digital part of this design was based on the method described in Chapter 3. Though convenient, this method of selecting device sizes may not be the optimal size in terms of noise, delay, or gain. The analog parts of this design followed theory of good performance, but are not, in most cases, optimized for noise, channel-length modulation effects, velocity saturation effects, drive, or for minimized size. The thought behind the sizing of analog parts was to pick currents which were enough to perform the function required of the circuit, but not to use excessive power. Most cases a mid-value currents were selected. The exception was the voltage to current converter. It was consciously oversized to increase drive and reduce the corner frequency of the block so it would reject

any residual noise from the original signal. This means it uses more power than the actual minimized converter necessary. Research will continue to describe designing for optimal power, or area, or jitter suppression.

6.4 Conclusion

This work shows that a purely feedforward jitter removal circuit is possible for frequencies in the $800MHz$ to $5GHz$ range. The design is relatively simple and modular, which allows a designer to customize the JAC to the type of system where the circuit will be used. MOSFET devices were used and no special processing is required. In some cases it is recommended that certain analog blocks are placed near each other to minimize process variation effects but, other than that, no special layout considerations are required or recommended. The transistors used in the switching $NMOS$ in the integrator have a channel length of $110nm$. All other devices in the design are sized above $130nm$. No effort was made to minimize the number of transistors used. The final number of transistors used was 403 from the input single-to-differential converter through the output level detector. Exactly 300 of these were used in the input monostable block.

Simulations were done using ST Microelectronic $90nm$ technology. Simulations looked at the performance of the circuit in the presence of supply noise, GND noise, intrinsic noise, and input jitter. All simulations were done at typical, fast, and slow corners. Attenuation of jitter was tested with input jitter from $0pspk-pk$ to the limit the system could process. For a system running at $1GHz$ and driven with a pulse of width $650ps$, the maximum jitter that can be processed is slightly less than $350pspk-pk$. The results varied from $-13.81dB$ (a 4.9x reduction in rms jitter) for $200mV$ of random noise on the supply, intrinsic noise, and worst case evaluations for 10% process variation, to $-14.68dB$ (a 5.4x reduction in rms jitter) for no supply or GND noise. Using an ideal source instead of the pre-processing input monostable block used in these results, gives a maximum jitter reduction of $-35.5dB$. The loss in performance can be attributed to the large amount of circuitry in the input

processing block. These results show that this purely feedforward system is, in fact, effective in reducing jitter.

In implementing this system, a number of new blocks were developed including a differential Schmitt trigger, feedforward correction block to align signals, high-speed pulse-mode flip-flop, and a monostable that can produce a duty cycle close to 100%. The feedforwarding biasing circuitry was also unique as was the effort to create a completely feedforward design.

Bibliography

Al-Sarawi, S., Low power Schmitt trigger circuit, *Electronics Letters*, *38*(18), 1009 – 1010, 2002.

Allstot, D., S. Kiaei, and R. Zele, Analog logic techniques steer around the noise, *Circuits and Devices Magazine, IEEE*, *9*(5), 18–21, 1993.

Anand, S., A 2.75Gb/s CMOS Clock Recovery Circuit with Braod Capture Range, in *ISSCC*, 2001.

Ashok, S., T. Reilly, and P. Lappalainen, Comments on 'A high-performance monostable multivibrator', in *Solid-State Circuits, IEEE Journal of*, vol. 10, pp. 80 – 80, 1975.

Banerjee, D., The Impact of Various PLL Parameters on System Performance, 2006.

Bigongiari, A., S. Brigati, G. Caiulo, G. Franchi, and F. Maloberti, An analog high-speed wide-range programmable monostable multivibrator, in *Circuits and Systems, 1995. ISCAS '95., 1995 IEEE International Symposium on*, vol. 2, pp. 845 – 849, 1995.

Bluetooth.org, Bluetooth Basics, 2006.

Chen, S., and M. Ker, A new Schmitt trigger circuit in a 0.13-/spl mu/m 1/2.5-V CMOS process to receive 3.3-V input signals, *Circuits and Systems II: Express Briefs, IEEE Transactions on [see also Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on]*, *52*(7), 361 – 365, 2005.

Chung, W.-S., H.-W. Cha, and H.-J. Kim, Current-controllable monostable multivibrator using OTAs, in *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on [see also Circuits and Systems I: Regular Papers, IEEE Transactions on]*, vol. 49, pp. 703 – 705, 2002.

Dejhan, K., P. Tooprakai, T. Rerkmaneewan, and C. Soonyeekan, A high-speed direct bootstrapped CMOS Schmitt trigger circuit, in *Semiconductor Electronics, 2004. ICSE 2004. IEEE International Conference on*, pp. 69 – 71, 2004.

Heydari, P., Design and Analysis of Low-Voltage Current-Mode Logic Buffers, in *Proceedings of the Fourth International Symposium on Quality Electronic Design (ISQED03)*, vol. 53, pp. 1274 – 1278, 2006.

Howe, R. T., and C. G. Sodini, *Microelectronics, An Integrated Approach*, Prentice Hall, 1997.

Lappalainen, P., A high-performance monostable multivibrator, in *Solid-State Circuits, IEEE Journal of*, vol. 9, pp. 190 – 192, 1974.

Lo, A., Transistor Trigger Circuits, in *Proceedings of the IRE*, vol. 40, pp. 1531 – 1541, 1952.

Lo, Y.-K., and H.-C. Chien, Current-Mode Monostable Multivibrators Using OTRAs, in *Circuits and Systems II: Express Briefs, IEEE Transactions on [see also Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on]*, vol. 53, pp. 1274 – 1278, 2006.

Lotfi, R., M. Taherzadeh-Sani, M. Azizi, and O. Shoaei, A 1-V MOSFET-only fully-differential dynamic comparator for use in low-voltage pipelined A/D converters, *Signals, Circuits and Systems, 2003. SCS 2003. International Symposium on*, *2*(2), 377 – 380, 2003.

Lu, X., Z. Li, W. Qiu, D. Walker, and W. Shi, PARADE: parametric delay evaluation under process variation [IC modeling], in *5th International Symposium on Quality Electronic Design, 2004*, pp. 276–280, 2004.

Maneatis, J., Selecting PLLs for ASIC Applications requires tradeoffs, 2003.

Meyer, R., P. Gray, P. Hurst, and S. Lewis, *Analysis and Design of Analog Integrated Circuits*, fourth ed., John Wiley & Sons, Inc., 2001.

Microelectronics, S., Low Power Single CMOS Timer, 2003.

Mizuno, M., 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, *Solid-State Circuits, IEEE Journal of*, *31*(6), 784 – 791, 1996.

Musicer, J., An Analysis of MOS Current Mode Logic for Low Power and High Performance Digital Logic, Master's thesis, University of California at Berkeley, 2000.

Oppenheim, A., and A. Willsky, *Signals and Systems*, second ed., Prentice Hall Signal Processing Series, 1997.

Pedroni, V., Low-voltage high-speed Schmitt trigger and compact window comparator, *Electronics Letters*, *41*(22), 1213 – 1214, 2005.

Pfister, A., Novel CMOS Schmitt trigger with controllable hysteresis, *Electronics Letters*, *28*(7), 639 – 641, 1992.

Pialis, T., and K. Phang, Analysis of Timing Jitter in Ring Oscillators Due to Power Supply Noise, in *Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on*, vol. 1, pp. I–685 – I–688, 2003.

Rabaey, J., *Digital Integrated Circuits*, Prentice-Hall, Inc., 1996.

Razavi, B., *Monolithic Phase-Locked Loops and Clock Recovery Circuits: Theory and Design*, first ed., IEEE, 1996.

Razavi, B., *Design of Analog CMOS Integrated Circuits*, McGraw Hill, 2001.

Savoj, J., A 10Gb/s CMOS Clock and Data Recovery Circuit with a Half-Rate Linear Phase Detector, 2001.

Schultheiss, P., , and H. Reich, Some Transistor Trigger Circuits, in *Proceedings of the IRE*, vol. 39, pp. 627 – 632, 1951.

Toda, A., A. Pandit, and K. A. Tran, Ultra Low Power Clock Generation using Sub-Threshold MCML, 2006.

Torralba, A., J. Martinez-Heredia, R. Carvajal, and Ramirez-Angulo, Low-voltage transconductor with high linearity and a large bandwidth, *Electronic Letters*, *38*(25), 1616–7, 2002.

Tuwanut, P., J. Koseeyaporn, and P. Wardkein, A Novel Monostable Multivibrator Circuit, in *TENCON 2005 2005 IEEE Region 10*, pp. 1 – 4, 2005.

Underhill, M., The Anti Jitter Circuit for the Suppression of Wideband Phase-Noise, in *IEEE Colloquium on Microwave and Millimetre-Wave Oscillators and Mixers*, pp. 1–14, 1998.

Underhill, M., The adiabatic anti-jitter circuit, in *Proceedings of the 1999 Joint Meeting of the European Frequency and Time Forum and IEEE International Frequency Control Symposium*, vol. 2, pp. 611–614, 1999.

Underhill, M., The adiabatic anti-jitter circuit, *IEEE Transaction on Ultrasonic, Ferroelectrics and Frequency Control*, *48*(3), 666–674, 2001.

Underhill, M., The noise and suppression transfer functions of the Anti-Jitter circuit, in *Proceedings of the 2003 IEEE International Frequency Control Sympposium and PDA Exhibition Jointly with the 17th European Frequency and Time Forum*, pp. 490–498, 2003.

Underhill, M., and J. Brodrick, The performance of the anti-jitter circuit with enhanced feedback ('EF-AJC'), in *Proceedings of the 2004 IEEE International Frequency Control Symposium and Exposition*, pp. 158–164, 2004.

van Roon, T., Phase-locked loop, 2006.

Van Zeghbroeck, B., Pinciples of Semiconductor Devices, 2004.

Wang, Z., CMOS adjustable Schmitt triggers, *Instrumentation and Measurement, IEEE Transactions on*, *40*(3), 601 – 605, 1991.

Zhang, C., A. Srivastava, and P. Ajmera, Low voltage CMOS Schmitt trigger circuits, *Electronics Letters*, *39*(24), 1696 – 1698, 2003.

Zilic, Z., Phase- and Delay-Locked Loop Clock Control in Digital Systems, 2001.