Inter-Pulse Interval Based Mixed Signal Representations

Sanjay Ravi
B.S., Electrical and Computer Engineering, Carnegie Mellon University (1996)
M.S., Electrical and Computer Engineering, University of Texas at Austin (1999)

Presented to the Division of Biomedical Computer Science within
The Department of Science & Engineering
and the Oregon Health & Science University
School of Medicine
in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy
in
Electrical Engineering

December 2008

The dissertation “Inter-Pulse Interval Based Mixed Signal Representations” by Sanjay Ravi has been examined and approved by the following Examination Committee:

Dr. Dan Hammerstrom

Ph.D. Advisor

Professor of Electrical and Computer Engineering and Associate Dean for Research Maseeh College of Engineering & Computer Science at Portland State University

---

Dr. Siva Narendra

CTO, Tyfone, Inc., USA

John Hunt

Faculty, Department of Computer Science and Electrical Engineering

OGI School of Science & Engineering at Oregon Health & Science University

---

John Lynch

Faculty, Department of Computer Science and Electrical Engineering

OGI School of Science & Engineering at Oregon Health & Science University

Acknowledgments

I thank God who has made it possible for me to complete this Ph.D. thesis through His blessings.

I thank Dr. Dan Hammerstrom, my thesis advisor, for supervision of my Ph.D. research work and his comments and feedback which have benefited me greatly.

I also thank Dr. Siva Narendra, John Lynch, and John Hunt for their comments and feedback on this work.

I also thank my wife Santhoshini, and my parents for their support and patience during my Ph.D. years. They have always wanted me to be successful in my education and my professional career.

Contents

List of Tables

List of Figures

Abstract

Inter-Pulse Interval Based Mixed Signal Representations
Sanjay Ravi, B.S., M.S.

Ph.D., OGI School of Science & Engineering
at Oregon Health & Science University

December 2008
Thesis Advisor: Dr. Dan Hammerstrom

In this research we have developed and analyzed a set of techniques for converting analog voltages into pulse streams based upon the Inter-Pulse-Interval (IPI) as well as conversion back to analog voltage. A MOSIS 0.35 µm chip was designed and fabricated that contains both circuits. The basic chip design and test results are also presented. The IPI presented here is the asynchronous version, without a clock to drive the sampling process.

We have studied low power versions of this circuit. We show that the IPI conversion requires significantly less power than more traditional signaling techniques such as pulse-width modulation (PWM), sigma-delta ADC, etc. This low power is obtained without slowing the sampling rate of the input signal. We show that the IPI conversion process generates intrinsic distortion

between the voltage input and voltage output, and analytically prove that the way to compensate for this distortion is to increase the sampling rate. In asynchronous IPI, there is non-uniformity in sampling rate from one part of the input signal to another part of the input signal. For reducing distortion, the main thing that is relevant is the ratio of the worst-case or slowest sampling rate over the input signal frequency, based on Nyquist's theorem. The IPI output is suitable for transmission using broad-spectrum, pulse based ultra-wide band (UWB) techniques, with the potential application in the area of ultra-low power wireless sensors, especially for biomedical applications.

We also show that IPI representation has more immunity to high-frequency attenuation over an interconnect wire compared to more conventional signaling techniques, such as pulse-amplitude modulation (PAM), pulse-width modulation (PWM), etc. This concept can be useful to address high-speed signal integrity issues.

CHAPTER 1
INTRODUCTION

1.1 Motivation for IPI Research

Although digital signal representation has become almost universal, there are still many areas where an analog representation is required to interface with an analog sensor world or to meet various other objectives such as power dissipation, frequency, or cost [70]. One important group includes wireless data collection from implantable, portable and optoelectronic sensors. In these domains analog signal representation is essential for many input modalities such as instrumentation, sensor interfaces, and communications [70]. Likewise, there are related output applications, such as biomedical actuation and industrial control [70]. In addition, the needs of wireless and fiber-optical communication have reinvigorated analog design. However, there are problems concerning how to keep these analog components on a reasonable scaling curve as Moore's law continues unabated in the digital domain, and in integrating analog representations into large, complex digital systems ("system on a chip"). [70]

In this dissertation we present an approach to representing analog signals that we believe will integrate more cleanly in remote sensing applications. Today analog signals are almost exclusively represented by current or voltage quantities, which are susceptible to signal degradation. Our proposal is to borrow a page from neuroscience and use the time between

asynchronous pulses[1], the Inter-Pulse-Interval (IPI), to represent analog quantities. We have developed a mixed-mode analog/digital design methodology based on the IPI representation. [70] Mhaidat and Jabri have investigated the computation elements [1], and Ettienne-Cumings et al. have been doing work on image sensors [2-4], [6]. According to Ettienne-Cumings et al., the light intensity or the brightness of the pixels in the address-event imager is inversely proportional to the pulse inter-spike time interval, or inter-pulse time interval [3]. In fact, Address Event Representation (AER) [4-6] is a well developed and commonly used technique for communicating analog information. It has potential value in systems that are collecting data from arrays of distributed sensors.

In biological systems various signal representations are used, but pulses appear to be the dominant form for representing communication in neural circuits for reliable communication.

The IPI representation has been the subject of previous research [3], [27-29], [32], [41] but it has, as yet, found little applicability outside of the domain of neuromorphic hardware, optical fiber communications for analog and video applications, and light emitting diode communication systems. Some of this previous asynchronous IPI work was known as pulse interval modulation (PIM), and the synchronous IPI work was known as pulse position modulation (PPM) [32]. Much of the PIM work mentioned includes the case where PIM was combined with pulse width modulation (PWM) to generate the asynchronous pulse interval and width modulation (PIWM) scheme [28-29], [32]. This is where both the pulse time interval and the pulse width encode the data. However, PIWM consumes more power than PIM since the information is also encoded in the width of the pulses. The goal of this work is to characterize the use of the IPI representation as a general technique for

[1] In biology pulses are referred to spikes, and so much of what we present here is derived from work with and the modeling of the Inter-Spike-Interval (ISI).

analog signal representation, and bring it into a more general IC design flow to stimulate its use in a wider variety of applications.

There are various ways that pulses can be used to represent information. For example, one variation of pulsed information is rate-based encoding where the data are carried via the pulse frequency. This technique forms the basis of the 1-bit audio signal representation (pulse density modulation or pulse frequency modulation) that is being developed in Japan to power the next generation of high-fidelity sound. The original signal can be recovered from a pulse density stream by using a simple low pass filter.

However, the variation we are interested here uses the time between pulses, i.e., the inter-pulse interval, to convey information. This encoding is more efficient compared to conventional pulse signaling techniques such as pulse width modulation, pulse frequency modulation, etc., in terms of data representation and power utilization. Transmitting information by inter-pulse intervals raises interesting trade-offs in noise immunity, pulse density, information rates, etc., some of which are addressed in this paper. Another advantage of pulse streams is that the pulses can be converted to digital addresses and communicated over time-multiplexed buses with little loss of information, such as used by the Address Event Representation discussed above.

As we move to deep submicron and then on to nanometer/molecular devices, the problems that digital encounters with scaling, such as threshold inconsistency, subthreshold currents, hot-electron effects, doping variability, substrate coupling, and transmission line and complex cross-talk effects, are even more serious for analog circuitry. IPI representations will provide better immunity to these effects, as well as to the more traditional process, temperature, and reference voltage variations. For most applications, pulse based analog systems will require less power, which is mostly due to the fact that pulses are small bursts of energy rather than continuous current flow. We

do not believe that inter-pulse representations will replace all voltage/current representations in mixed-signal circuitry, but we do believe that the technique adds a valuable tool to the mixed-signal toolbox.

One area where IPI seems to be especially well suited is that of remote sensor data collection. As will be shown in this thesis, it is far less expensive in power to convert analog signals to IPI, which can be transmitted over a wire or wireless, than it is to convert to other signal representations. A potentially good application for IPI could be low power wireless sensor networks[2]. Wireless sensor networks are potentially useful for environmental monitoring, bio-medical applications such as patient monitoring, home security applications, and numerous other applications.

There are numerous advantages to using pulses to compute or to communicate. Most important, they are a way to represent analog quantities that can be multiplexed over a single wire and which are significantly more immune to noise. One of the problems with mixed signal circuits using traditional scalar current/voltage representations is that the switching noise of digital circuitry on the same die is disruptive, especially if they must traverse any distance - IPI representations would be immune to this noise. Since IPI-based signals are represented by full-scale pulses, they can be completely restored as they travel along a path, though it is important that path delay be consistent, since they are sensitive to jitter noise.

The potential advantages of an IPI based mixed signal representation then include:

- Significantly better immunity to noise and (digital spiking effects) over traditional mixed-mode analog-digital representations. IPI has better

[2] This has been studied by Rabaey et al. at the Berkeley Wireless Research Center [71-72].

immunity over traditional analog representation since IPI can have only two values, HIGH and LOW. The definition of IPI that is mentioned in both Chapter 1 and Chapter 5 explains why IPI can have potentially better noise immunity compared to some of the other signal representations.

- Lower power dissipation, since pulses occur only occasionally, instead of having a continuous stream of voltage or current. This point is addressed in Chapter 3 of this thesis.
- Better immunity to high-frequency attenuation in both on-chip and off-chip high-speed interconnects, since IPI pulses occur only occasionally. This will improve the signal integrity in high-speed digital applications. This issue is addressed in Chapter 6.

In the next section of this chapter, a formal definition of the Inter-Pulse Interval will be presented for the asynchronous form. After that, a table of definitions will be given that compare IPI with other signaling schemes, such as pulse-width modulation (PWM), etc.

1.2 Definition of IPI

1.2.1 Asynchronous IPI and Synchronous IPI

There are two basic forms of IPI, asynchronous and synchronous. In *asynchronous IPI*, AIPI, a continuous signal is represented by the time between any two adjacent data pulses. In AIPI the pulses are asynchronous. In *synchronous IPI*, SIPI, a continuous signal is represented by the time between a synchronous reference pulse and a data pulse.

IPI representation means that the analog information is stored in the interval between two pulses. In a plot of IPI representation, the x-axis is time, and the y-axis is the voltage of the pulse signal. That voltage can have only

two values, a high value and a low value. As the input analog voltage or current changes, the time between two pulses changes.

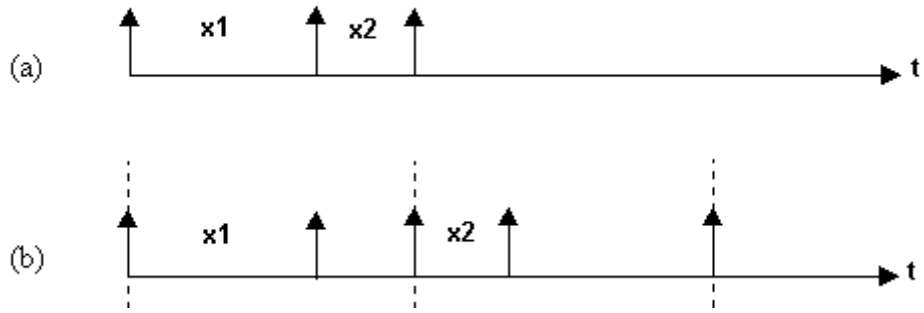

Figure 1.1: IPI signaling schemes: (a) Graphical definition of Asynchronous IPI, or AIPI, and (b) Graphical definition of Synchronous IPI, or SIPI. [1]

Figure 1.1 shown above demonstrates the definition of AIPI and SIPI in graphical format. As the top portion of the plot indicates, the data is represented by the time between two adjacent data pulses in AIPI. In the bottom portion of the plot, the pulses with the superimposed dashed lines represent the synchronous reference pulses in the SIPI representation, and the data is represented as the time between one of these clocked reference pulses and the data pulse that follows it in the SIPI representation.

In this section, Asynchronous Inter-Pulse-Interval (AIPI) representation will be defined more formally. Here are definitions of the terminology and the notations that will be used:

- t_i = inter-pulse interval (IPI) start time (or time at which first pulse occurs)
- t_f = inter-pulse interval (IPI) end time (or time at which second pulse occurs)
- t_d = *DeltaT* = t_f - t_i = IPI time difference between two asynchronous pulses that encodes the analog value x_d mentioned below

- x_d = analog value represented by the IPI time difference between the pulse starting at t_i and the second pulse ending at t_f
- $f(x)$ = analog-to-IPI conversion function
- $f^{-1}(x)$ = IPI-to-analog conversion function

Factors such as noise, accuracy, speed, and practical design limitations will affect how small the shortest interval can be [1].

In the linear AIPI representation, the time t_d is a real value that is linearly proportional to the input analog value x_d that is being encoded [1]. This representation is being defined here only for positive x_d. This linear conversion function can be mathematically expressed as:

$$t_d = f(x_d) = ax_d + b, \text{ where } x_d > 0 \qquad \textbf{Eq. 1.1}$$

In the above equation, a and b are real valued constants [1]. In the nonlinear representations such as the hyper-tangent or the logarithmic [1], the relationship between the encoded IPI time interval, t_d, and the analog input value, x_d, is not linear. In the logarithmic representation, for example, the general form of the conversion function is:

$$t_d = f(x_d) = a\log_b(x_d) + c \qquad \textbf{Eq. 1.2}$$

In the above equation, a, b, and c are real valued constants. A more general form of the logarithmic function that can be used to deal with the cases when x_d is zero or negative is

$$t_d = f(x_d) = a\log_b(x_d + d) + c \qquad \textbf{Eq. 1.3}$$

In the above equation, *a*, *b*, and *c* are real constant numbers, and *d* is a real positive constant that can be added to x_d so that the input to the log function is always positive. This constant *d* needs to be accounted for when converting back from the IPI domain to the analog domain [1]. The nonlinear IPI representation will be covered in Chapter 2 and the linear IPI representation will be covered in Chapter 3. Chapter 3 also explains the advantages of the linear IPI over the nonlinear IPI, and why linear IPI has been chosen over nonlinear IPI as a means of using AIPI for communication.

IPI signaling will be defined in this work as a pulse scheme where both pulse amplitude and pulse width are constant. An interesting question about AIPI concerns pulse shape. Although it can be defined as more or less a regular pulse with a well defined edge, as has been done in most of this thesis, the reality is that using a well defined pulse shape and edge is not necessary in AIPI. An AIPI pulse is really just a packet of energy that the circuit detects. This concept differentiates AIPI from the other pulse signal representations, where a rectangular pulse with a well defined edge is generally required. This energy packet model to describe AIPI will be discussed in more detail below.

This packet of energy is defined as an AIPI pulse if either **Eq. 1.4** or **Eq. 1.5** is satisfied as mentioned below:

$$(V_{AMP} = V_{MAX}) \text{ and } (F_{MAX} \geq F_{TH}) \qquad \textbf{Eq. 1.4}$$

$$(V_{AMP} > V_{LTH}) \text{ and } (F_{MAX} < F_{TH}) \qquad \textbf{Eq. 1.5}$$

In the above equations,

- V_{AMP} = voltage amplitude of a signal that represents a packet of energy
- V_{MAX} = maximum voltage swing for the AIPI signal

- V_{LTH} = threshold voltage level less than V_{MAX} where the packet of energy is considered a pulse if its amplitude is greater than this threshold. An AIPI pulse needs to have minimum amplitude for robustness, even if its amplitude is less than full swing due to attenuation or pulse shaping. This way, signals with amplitudes below this threshold can be filtered out as noise.
- F_{MAX} = maximum frequency that represents the frequency spectrum of this packet of energy
- F_{TH} = threshold frequency where the packet of energy is considered a pulse if the maximum frequency component in its frequency spectrum is less than this threshold

Here, V_{LTH} and F_{TH} are arbitrarily defined constants. These two conditions above (**Eq. 1.4** and **Eq. 1.5)** define an AIPI pulse as either 1) a pulse with high-frequency content at full voltage swing (to include the rectangular pulse with the full voltage swing with defined edges) or 2) a pulse that has voltage swing in the range between V_{LTH} and V_{MAX} but has the high-frequency content filtered out. An AIPI pulse needs to have minimum amplitude for robustness, even if its amplitude is less than full swing due to attenuation or pulse shaping. This way, signals with amplitudes below this threshold can be filtered out as noise. Most of this thesis uses the AIPI with well defined pulse edges to demonstrate the concepts in a simplified manner. However, this energy packet model will be useful for noise immunity, as will be discussed in Chapter 5. A signal that does not satisfy either **Eq. 1.4** or **Eq. 1.5** is considered to be noise.

AIPI is most useful in communication of analog information, since it can transmit a signal with more power efficiency than SIPI. SIPI, on the other hand, allows the definition of positive and negative values as well as a range of computational operations [1]. It is difficult to do arithmetic on AIPI

represented data, whereas a range of arithmetic operations can be performed directly on SIPI represented data. And in some cases, an AIPI conversion back to analog voltage or current can be done to perform signal processing such as basic filtering. This is a reasonable alternative since IPI conversion is inexpensive in terms of hardware and power.

However, the real strength of AIPI is its use as a means for representing and transmitting analog data over networks, whether between chips, via wireless connections, or even within a single chip, without converting to digital locally at the sensor, in an environment where power is at a premium, such as with portable, wireless sensors. Most of this thesis is devoted to the characterization of AIPI for such an application and comparing it to more traditional pulse representations. SIPI will not be covered here, since it has been convered very well by Khaldoon Mhaidat [1].

At its simplest, AIPI has traditionally been represented by the *integrate-and-fire* model [1], [41]. An input current I charges a capacitor C. When the capacitor voltage reaches a threshold, a pulse is generated at the output and the capacitor is discharged. Here, the time between pulses or the AIPI time interval is inversely proportional to the input current charging the capacitor. The nonlinear AIPI representation is based on this concept where the input current charging the capacitor varies, causing the AIPI time interval to vary. A more complex and more realistic model is the *leaky integrate-and-fire*. Some previous work has been done related to leaky integrate-and-fire neurons [42-43]. One of theses papers discusses the use of leaky integrate-and-fire neurons for sound feature detection [43]. In the leaky integrate-and-fire case, a resistance is used to leak charge off the capacitor, so that the inter-pulse interval then has precise control of the charge on the capacitance [70]:

$$C\frac{dV(t)}{dt}+\frac{V(t)}{R}=I(t) \qquad \textbf{Eq. 1.6}$$

Although these circuits tend to be low pass filters, there is an important and subtle difference. A low pass filter generates a continuous voltage or current representation by integrating over a series of pulses - the current or voltage is proportional to the pulse rate, in which case a single inter-pulse interval, IPI, does not carry significant information. [70]

However, the leaky integrate and fire element discharges the capacitor during pulse generation, thus “forgetting” all previous intervals. Consequently, the actual inter-pulse interval from one pulse to the next can have a significant effect on circuit output, whereas rate-based encoding requires a series of pulses to increase the rate accordingly. [70]

1.2.2 Precision and Resolution Issues

Perhaps the most serious criticism of the IPI representation is the fact that large dynamic range leads to long intervals intervals, which constraints the maximum frequency components that AIPI captures accurately. It also leads to potentially slower operation. One way to reduce this dynamic range problem is to use a logarithm-based representation. [70] Another approach is put the threshold mechanism directly into the sensor. The sensor then integrates charge representing the signal of interest to a fixed voltage threshold and then sends a pulse when the threshold is exceeded. For low amplitude input signals in the nonlinear IPI system, more integration time is used, leading to a more accurate representation of smaller signals.

Another issue with AIPI concerns the implementation of negative or zero values. This is one of those interesting situations where infinity (one continuous value) and zero (also one continuous value) have the same representation in the sense that there are no pulse edges. So, for example, pulses don’t necessarily have to be identified strictly by edges. As with many

analog circuits, negative values can always be implemented with complementary modes. [70] In the applications that have been studied, the issue of zero representation has not been a problem since by virtue of noise in the input, there are no intervals that cannot otherwise be represented by low value signals. And in many situations it is possible to add a small offset to more accurately control minimal value representation.

Though mostly insensitive to analog noise, AIPI is sensitive to jitter noise, i.e., non-uniform delays added during pulse communication. However, perhaps the biggest problem with the AIPI representation is that the sample rate changes depending on the magnitude of the voltage. This affects signal integrity in a number of ways. For example, a signal could be undersampled at the lower magnitudes, but sufficiently sampled at the larger magnitudes. The source, such as a sensor, can even control the sample rate by adding or subtracting a DC offset when necessary. There is also a slight non-linearity that can be manifested on a signal ramp from lower to higher amplitudes. This intrinsic distortion in AIPI systems due to this non-linearity is discussed in detail in Chapter 4.

1.3 Comparison of IPI with Previous Work

The concept of using pulses to encode and process information is not a new one. Pulse encoding is used in neurobiological systems, where information in the brain is encoded using pulses [1]. A number of people had done work on pulse modulation and communication for over 60 years [1]. Information on various pulse modulation techniques can be found in [7], [8], and [32].

1.3.1 Previously Studied Pulse Based Signal Representations

Table 1.1 below defines IPI along with some of the previous work on pulse based representations. In order to demonstrate the benefits of using IPI, IPI will be compared with a number of these other pulse.

Pulse Type	Definition	Mathematical Definition
Asynchronous Linear IPI	Inter-pulse interval, where the time between two pulses, *DeltaT* or T_d, is a linear function of the analog input voltage, V_{in}. In this case, *DeltaT* increases as V_{in} increases, causing the sampling rate to be lower at higher input voltages. Since this is asynchronous, there is no clock. Pulse amplitude and pulse width are constant.	$T_d = f(V_{in}) = aV_{in} + b$ $F_s = \frac{1}{T_d}$
PWM	Pulse width modulation, where the width or duty cycle of a synchronously pulsed signal is modulated with respect to the analog input. Pulse amplitude is constant, since PWM is encoded along the time axis and not the voltage axis. Pulse frequency or sampling rate is constant, based on the frequency of the clock that drives PWM. PWM is a special case of synchronous PFM where all the narrow pulses are adjacent to each other forming a wide pulse.	$P_w = f(V_{in}) = aV_{in} + b$
PAM	Pulse amplitude modulation, where the amplitude of a pulse is modulated with respect to the analog input voltage, V_{in}. The pulses are timed with respect to a synchronous clock. Therefore, pulse frequency or sampling rate is constant.	$P_{amp} = f(V_{in}) = aV_{in} + b$
PFM	Pulse frequency modulation, where the frequency of the pulse is modulated with respect to analog input. PFM can be either asynchronous or synchronous. Sigma-delta modulation (SDM) uses PFM in its analog-to-digital conversion algorithm. It makes more sense for PFM to be synchronous since then the input voltage signal can be sampled at fixed time intervals and then a constant frequency of pulses can be generated over a fixed time interval. Encoding/decoding is done based on average occurrence of multiple pulses over a synchronous time period. Pulse amplitude and pulse width are constant.	$P_f = f(V_{in}) = aV_{in} + b$

Table 1.1: This table defines several other popular pulse based data representations as well as IPI. More detailed information regarding the other pulse types is discussed below [1] [44].

In the equations for the table above, the parameters are defined as below:

- T_d = *DeltaT* = IPI time difference between two asynchronous pulses that encodes the analog value V_{in} mentioned below
- V_{in} = input analog voltage value to be encoded
- P_{amp} = pulse amplitude
- P_w = pulse width
- P_f = pulse frequency
- F_s = sampling frequency
- F_{clk} = clock frequency
- $f(V_{in})$ = analog-to-pulse conversion function, where the analog input V_{in} is the variable input
- a = arbitrary constant value that is real
- b = arbitrary constant value that is real

PWM, IPI, and PFM require only 2 voltage levels, HIGH and LOW. Regarding PWM, or pulse width modulation, P_w is the amount of time that the pulse signal is high in a given clock period. The duty cycle of the PWM signal is the proportion of time in the clock period that this signal is high. In PWM, the duty cycle is modulated with respect to the input. The number of possible pulse widths in a given clock period is equal to 2 raised to the power of n, where n is number of bits of resolution. For example, PWM-4 has 4 possible pulse widths where each clock period represents 2 bits.

Regarding PAM, or pulse amplitude modulation, P_{amp} is the voltage amplitude of the pulse signal. The number of voltage levels is equal to 2 raised to the power of n, where n is the resolution in bits. For example, PAM-4 has 4 levels of voltage amplitude where each pulse represents 2 bits, and PAM-8 has 8 levels of voltage amplitude where each pulse represents 3 bits.

In PFM, or pulse frequency modulation, P_f is the pulse frequency or number of pulses in a given time interval.

Figure 1.2 shows the pictorial diagram of each of the pulse modulation techniques: PAM, PWM, PFM, and IPI.

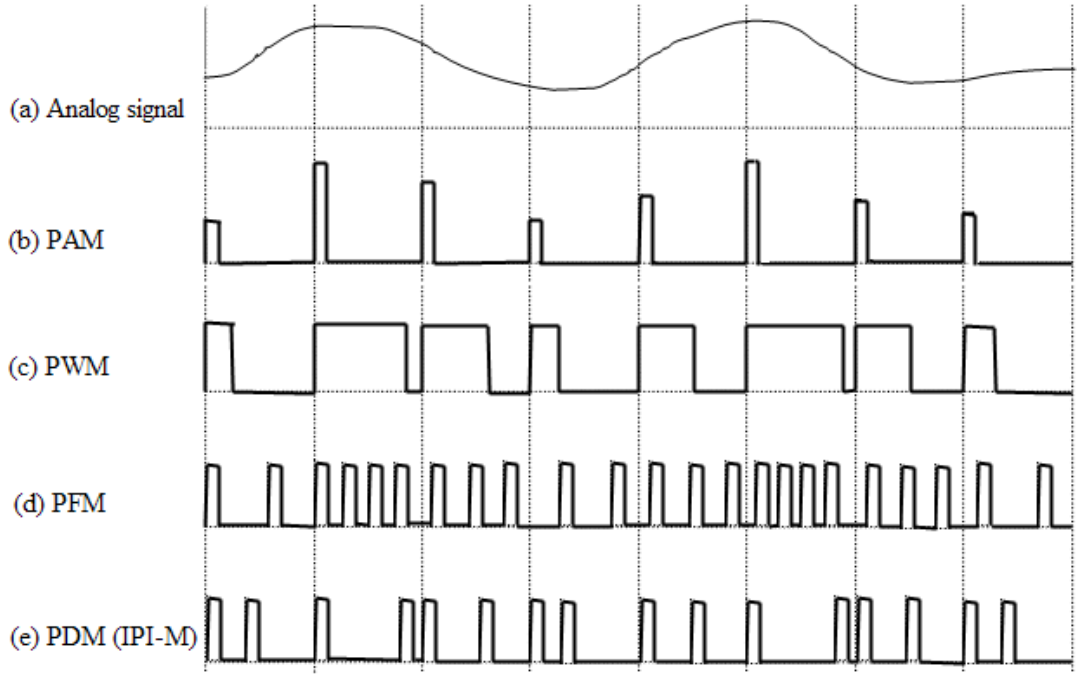

Figure 1.2: Pulse modulation techniques: (a) the analog signal, (b) pulse amplitude modulation (PAM), (c) pulse width modulation (PWM), (d) pulse frequency modulation (PFM), (e) pulse delay (or inter-pulse interval) modulation (PDM or IPI-M). As this pictorial diagram shows, PFM generally has a much higher density of pulses, or number of pulses per unit time compared to IPI. Figure 1.1 has shown the detailed comparison between asynchronous IPI (AIPI) and synchronous IPI (SIPI). [1]

1.3.2 Comparison of IPI with Synchronous PWM

Pulse width modulation (PWM) is where the width of the pulsed waveform, or amount of time that the pulse is high, varies with the input signal, and the pulse frequency remains the same [7-8], [32]. In contrast, for IPI, the pulse width remains constant while the pulse frequency, which is related to time between pulses, varies with the input signal. IPI is asynchronous in this research work, and PWM is synchronous [32].

Work on synchronous PWM has been done for applications involving optical fiber transmission of analog and video signals [1], [14], [17], [19], and [31]. PWM is also used in high-speed memory systems [36-37], [62]. PWM in the context of high-speed serial link applications will be discussed in detail in Chapter 6.

The similarities are that both PWM and IPI can be used for noise reduction since both can take on only two voltage levels, high and low. In PWM, the data is encoded along the time axis with no analog voltage present in the signal [7-8].

The fundamental advantage of IPI over PWM is IPI has lower power dissipation compared to PWM. IPI consumes less power than PWM since in IPI the encoding of the information is done in the time between fixed-width narrow pulses while in PWM the encoding is done in the variable width of the pulses [1], [32]. This paper mentions this in the context of synchronous IPI (SIPI), but this low power concept can be extended to asynchronous IPI (AIPI) as well. The reason AIPI will consume less power than SIPI is that there is no clock, a heavy consumer of power, to drive the AIPI system. Actual power figures comparing those two techniques can be found in Chapter 3 of this thesis.

The SIPI signal can be obtained from the PWM signal by differentiating the PWM signal to generate narrow pulses at the edge transitions [1]. Figure 1.3 shows the block diagram for PWM modulation. Sample and hold (S/H) is

performed on the analog input voltage in the PWM modulator and then that signal is compared with a sawtooth ramp signal. If the comparator detects that the input signal and the sawtooth ramp signal are equal, the comparator output goes from high to low. The PWM signal is taken from the output of the comparator. The SIPI signal is obtained by differentiating the PWM signal. In SIPI and PWM, a periodic clock is used to control the timing of the S/H operation and the ramp generation. The input signal has to be DC-shifted to accommodate the most negative value. Otherwise, it will be limited by the ramp minimum voltage and the comparator input range. [1], [32]

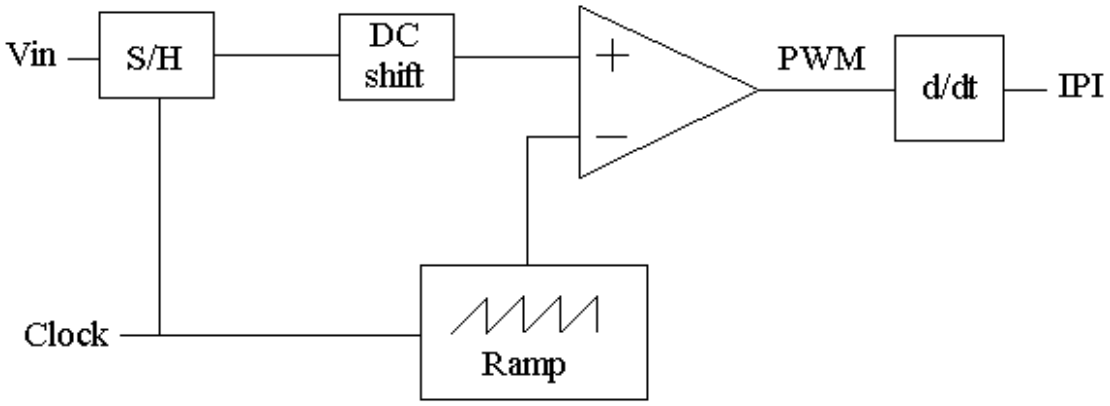

Figure 1.3: This diagram demonstrates synchronous IPI (SIPI) and PWM modulation. In asynchronous IPI (AIPI) and PWM, no clock is used and S/H and the ramp are restarted as soon as the comparator detects that the input sample and the ramp have equal values. [1], [32]

Demodulation of PWM is performed by converting the PWM signal to SIPI first via differentiation and then performing IPI demodulation. Figure 1.4 shows the block diagram for the demodulation of both PWM and IPI. For SIPI demodulation, a sawtooth ramp signal is generated using the first input pulse and then sample and hold is done on this ramp signal using the second input pulse. This produces the pulse amplitude modulation (PAM) equivalent to the PWM/IPI. A low pass filter (LPF) can then be applied to this PAM signal to recover the baseband signal component from the frequency spectrum. The ramp signal used in all the SIPI and PWM applications above is linear, which leads to a linear relationship between the voltage signal and the SIPI and the pulse width. [1], [32]

Since PWM is synchronously driven [32], there is a problem of clock jitter in scaled technologies that can degrade performance. A paper by Tang et al. discusses jitter in PWM when used in high-speed serial links [36]. In this paper, PWM jitter and clock jitter are equivalent since the clock is embedded in the PWM-encoded signal. Related work in phase-locked-loops (PLL) in a PWM high-speed I/O communication system is discussed in [37]. The PLL in this paper can be used in both the transmitter and receiver ends of the PWM I/O link. Here, both the transmitter PLL and the receiver PLL can generate the

8-phase clock by means of a voltage-controlled oscillator (VCO). PLL circuits can generate clock signals that have jitter due to phase noise, which can affect system performance [38]. One of the sources of this jitter or phase noise coming from clock generating circuits such as VCOs, PLLs, and ring oscillators is intrinsic noise generated by the circuits, such as 1/f flicker noise and thermal noise [39-40]. Another important advantage of AIPI is that it is asynchronous and does not use a clock that is subject to jitter noise.

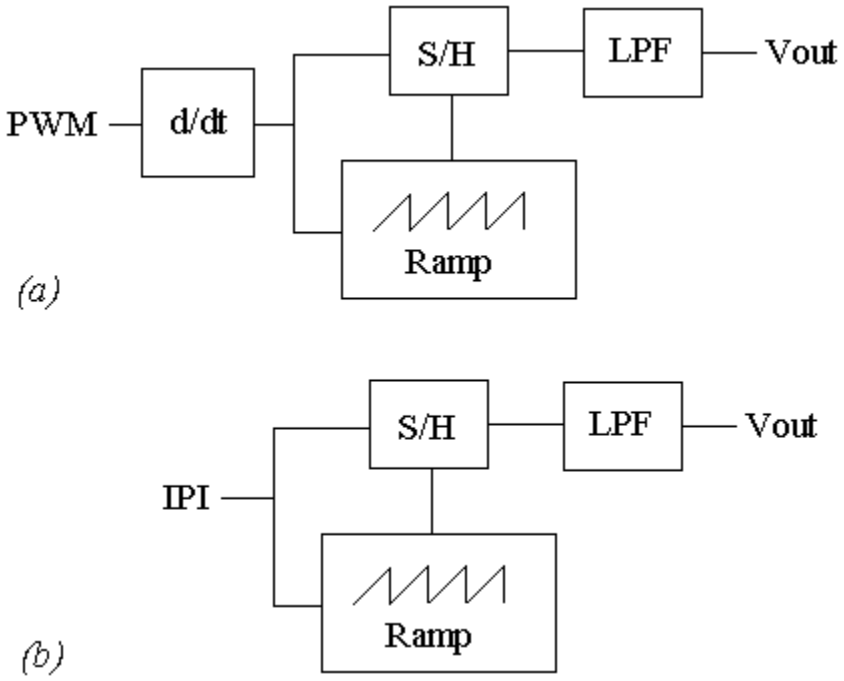

Figure 1.4: This diagram demonstrates *(a)* PWM demodulation, and *(b)* IPI demodulation. IPI demodulation here can be applied for both asynchronous IPI (AIPI) and synchronous IPI (SIPI) [1] [32].

1.3.3 Comparison of IPI with PAM

Pulse amplitude modulation (PAM) is where the amplitude of the pulse varies with the input signal.

PAM is commonly used in high speed serial link communication applications [56-61]. High speed communication applications will be discussed in more detail in Chapter 6. An advantage of IPI over PAM is that

IPI is more immune to noise since IPI has a fixed pulse amplitude and uses only time-dependent features to encode the information, while PAM uses the amplitude of the synchronous pulses to encode the information [1], [7-8].

1.3.4 Comparison of IPI with PFM

Pulse frequency modulation (PFM) is where the number of pulses within a given timeframe or time range varies with the input signal [7-8]. Work on PFM has been done for applications involving optical fiber transmission of analog and video signals [1], [23-26]. An analog signal can be converted to PFM by using a VCO followed by a monostable circuit to generate fixed-width narrow pulses [1], [32]. Decoding the PFM signal back to analog involves using a monostable circuit to reconstruct fixed-width narrow pulses from the input stream, followed by low-pass filtering to recover the base-band signal component from the frequency spectrum [1], [32]. The main drawback to using PFM is that many pulse signals in a given time frame are required, increasing the power requirements. IPI involves only two pulses bordering a time interval, causing the power dissipation to be reduced. Power dissipation in CMOS is related to transition frequency, where the power increases as transition frequency increases. For a given encoding time interval, IPI has only 2 rising transitions and 2 falling transitions, 2 pulses, while PFM has up to 2^N rising transitions and 2^N falling transitions, representing 2^N pulses, where N is the resolution in bits.

PFM can be either asynchronous or synchronous, and has been defined in some of the literature as asynchronous or anisochronous [28-29], [32]. However, it makes more sense for PFM to be synchronous since then the input voltage signal can be sampled at fixed time intervals and then a constant frequency of pulses can be generated in an interval. Sigma-delta modulation (SDM) uses PFM, which is also called pulse density modulation, in its analog-to-digital conversion algorithm [44].

Even though both IPI and PFM have varying pulse frequencies, asynchronous IPI can be encoded/decoded instantaneously, while in PFM the encoding/decoding is done based on average pulse occurrence over a synchronous time period.

1.3.5 Comparison of Impulse Radio with Other Wireless Technologies

One of the advantages of using IPI representation is for carrierless impulse based radios, such as UWB. The short duration of UWB impulse radio pulses causes an extremely wide bandwidth, on the order of several gigahertz. In spread-spectrum technologies, the signals are continuous-wave sinusoids that are modulated with a fixed carrier frequency. Spread-spectrum techniques have greater bandwidths, on the order of megahertz. Wideband communications such as UWB offers some advantages over narrowband communications such as covertness, frequency diversity for better performance in multipath channels, and resistance to jamming, or prevention of radio communication intentionally or unintentionally. UWB signaling techniques offer very low duty cycles, leading to very low transmission power and extra covertness compared to spread-spectrum techniques. [45]

Figure 1.5 shows the time and frequency domain plots for narrowband vs. spread-spectrum wideband vs. UWB, and compares all 3 signaling techniques with each other. This plot shows UWB having a narrow pulse in the time domain and a very wide bandwidth in the frequency domain while the other two techniques, narrowband and spread-spectrum, have continuous waveforms in the time domain and narrower bandwidths in the frequency domain.

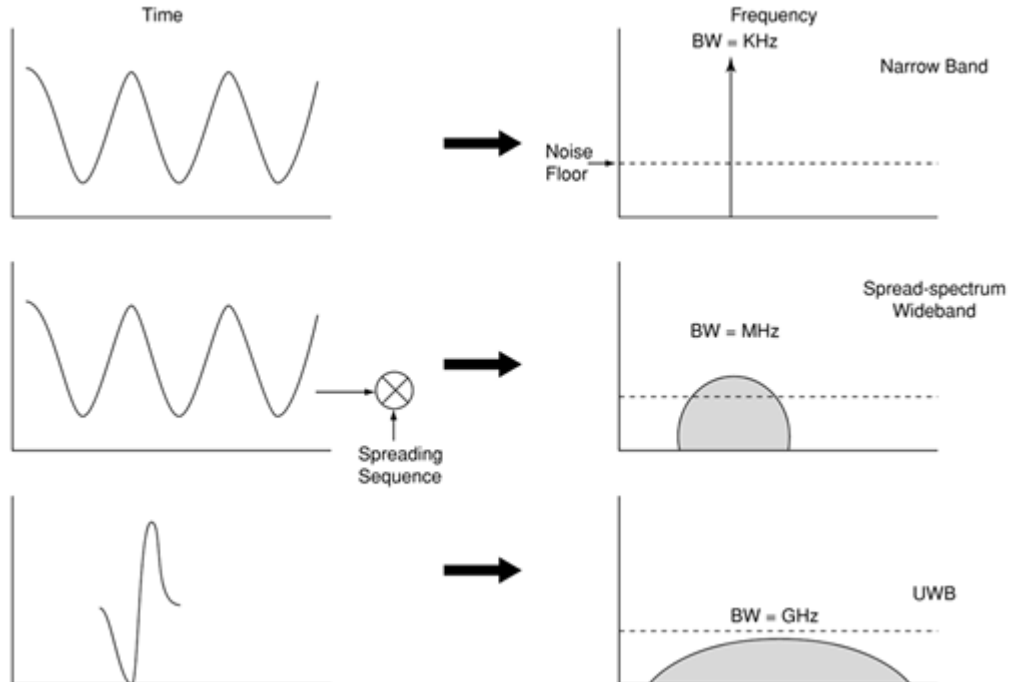

Figure 1.5: The transition from narrowband to wideband and ultra-wideband in the time and frequency domains [45].

1.4 Organization of Thesis

This thesis is divided into several chapters. In Chapter 2, designs for asynchronous IPI circuits using a non-linear IPI representation are shown. In Chapter 3, designs for asynchronous IPI circuits using the linear IPI representation are discussed, as well as the benefits of using linear IPI over non-linear IPI. In Chapter 4, MATLAB simulations and a more formal analysis of the IPI representation with respect to intrinsic distortion is presented. In Chapter 5, extrinsic distortion due to noise is covered. Chapter 6 deals with the application of high speed interconnect where IPI is compared with other pulse types. And chapter 7 summarizes the results of this work and discusses about some of the follow on research work that can be done in the future that is related to IPI.

CHAPTER 2

NONLINEAR IPI CONVERSION CIRCUITS

2.1 Introduction

This chapter covers preliminary designs for conversion circuits based on nonlinear IPI representation for voltage to IPI and IPI to voltage. Nonlinear IPI was used initially since this is based on previous work using the integrate-and-fire model mentioned in Chapter 1 [1], [41]. These designs were done to verify that conversion from analog voltage to IPI domain and back are possible in actual circuits to explore the various design issues. This work focuses on the asynchronous representation, since this is more suitable for ultra low power sensor based applications. There has been considerable work done on the Synchronous IPI representation (SIPI) by Khaldoon Mhaidat [1]. The SIPI representation is much more suitable for computation and allows direct arithmetic operation on pulse based forms.

The designs presented here were simulated in PSPICE. In addition to circuit representation, we also implemented the basic functionality in VHDL-AMS and as high level abstractions in MATLAB. VHDL-AMS results are not discussed in this thesis since the results were not conclusive.

The mathematical definition of AIPI was presented in Chapter 1 (Introduction).

2.2 Voltage to Asynchronous IPI Conversion

In this section, the simplest IPI representation is assumed, where the analog information is stored in the interval between two pulses. This is known as asynchronous IPI. The AIPI representation will be presented in the context of the voltage to AIPI conversion. In a plot of IPI representation, the x-axis is time, and the y-axis is the voltage of the pulse signal. That voltage has only two values. As the input analog voltage is changed, the time between two successive pulses is changed. Even within the constraints of the asynchronous IPI representation, there are a number of conversion options. An important option concerns the use of logarithmic or linear scale conversion. For most of the work reported here, unless indicated otherwise, the logarithmic representation is used. The logarithmic representation allows greater dynamic range without appreciably slowing the total system. On the other hand, a logarithmic representation is harder to use in many common signal processing applications.

A circuit for the voltage to IPI conversion has been designed. This converter is for the simplest asynchronous IPI representation without any reference or synchronization pulse. The main schematic is shown in Figure 2.1. The op-amp current mirror that converts the voltage signal into a current signal is shown in Figure 2.2. This current flows into a capacitor, charging it. Once that voltage on the capacitor reaches threshold, a pulse is generated and the capacitor is discharged. Ideally one would like to reuse the energy in the capacitor to generate the outgoing pulse to reduce power requirements even further. The current designs mentioned in this thesis do not do this, but it is a possible topic for future work.

The capacitor integrates the charge exponentially, i.e. $1-e^{-kt}$, which creates a logarithmic representation. An op-amp comparator generates a high output signal when the capacitor voltage reaches threshold. This comparator generates a high output signal when the IN+ pin, or the voltage on the

capacitor, becomes greater than the IN- pin, or the threshold voltage signal. This high output signal from the comparator generates the output pulse. The comparator schematic is shown in Figure 2.3.

The CMOS delay buffer, which adds some delay, is shown in Figure 2.4. This buffer is composed of two inverters, and is required to avoid timing glitches between the discharging done by the NMOS transistor (M4 of Figure 2.1) and the charging done by the op amp current mirror, since this current mirror is always on.

When the output of the CMOS delay buffer is high, the NMOSFET that takes the output of the buffer will turn on, causing the capacitor to discharge.

The plot of *DeltaT*, or the time interval between pulses, versus V_{in}, the input voltage, is shown in Figure 2.5. This plot appears to be exponential.

Initially the *DeltaT* versus V_{in} curves were too steep in some places and flat in other places. This problem was fixed by adding transistors and resistors to the op-amp current mirror to increase the gain, making it more dependent on the resistors than on the transistor characteristics, which, in turn, depend on temperature and supply voltage.

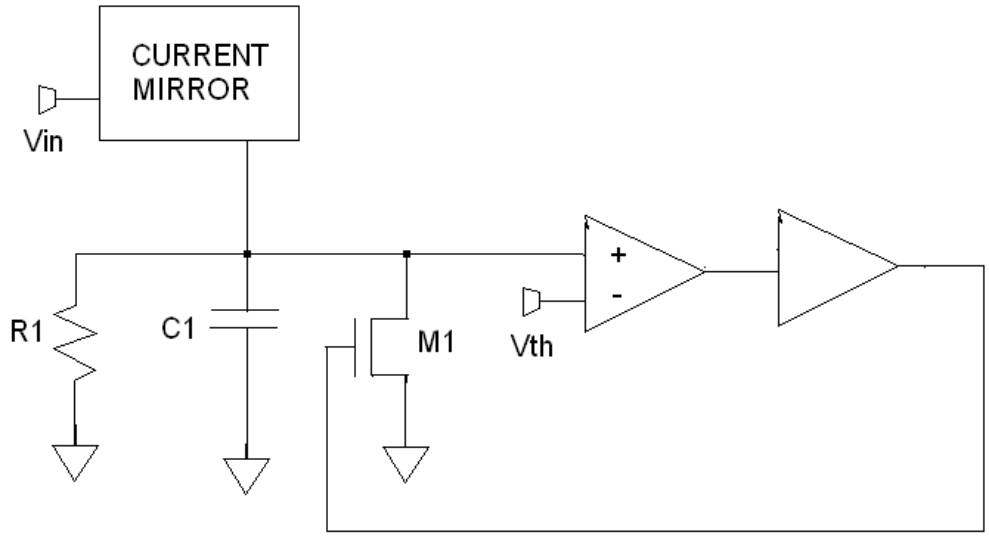

Figure 2.1: Voltage to IPI Converter - Main Schematic

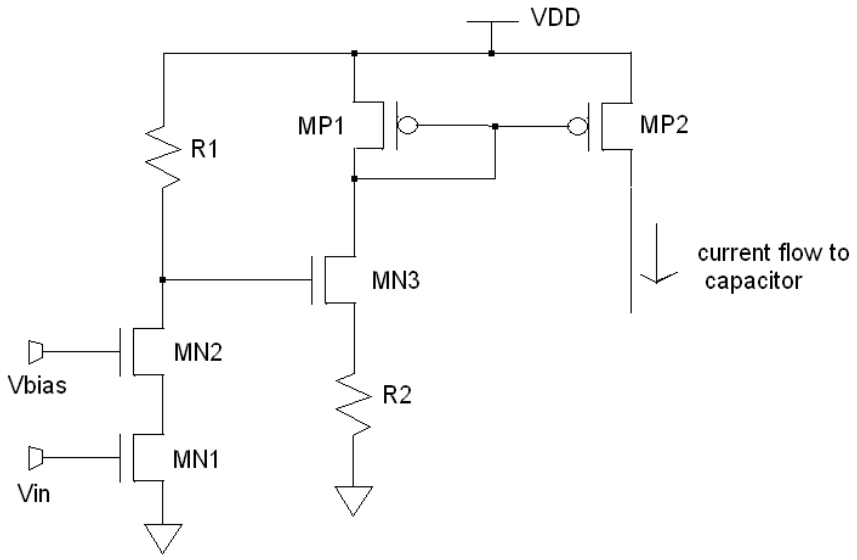

Figure 2.2: Voltage to IPI Converter - OpAmp Current Mirror

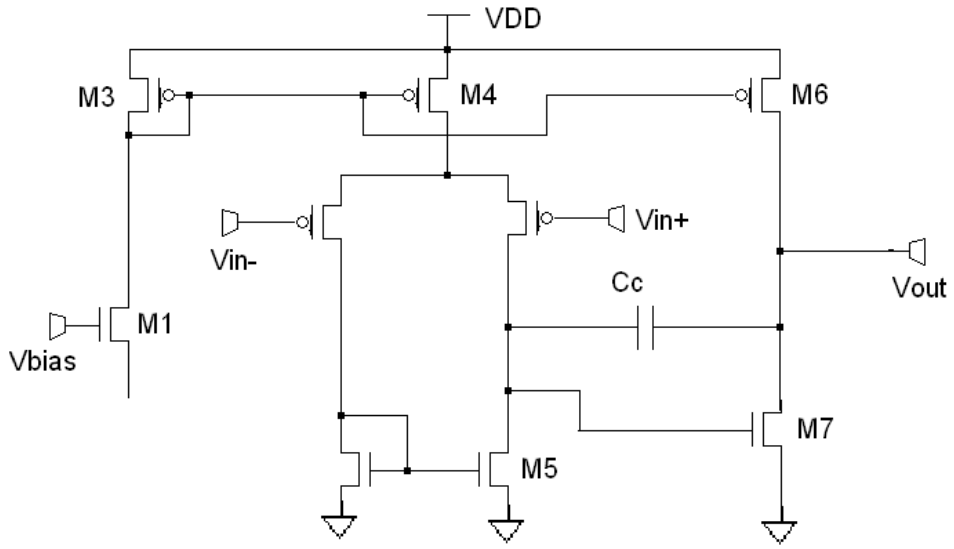

Figure 2.3: Voltage to IPI Converter - CMOS Comparator

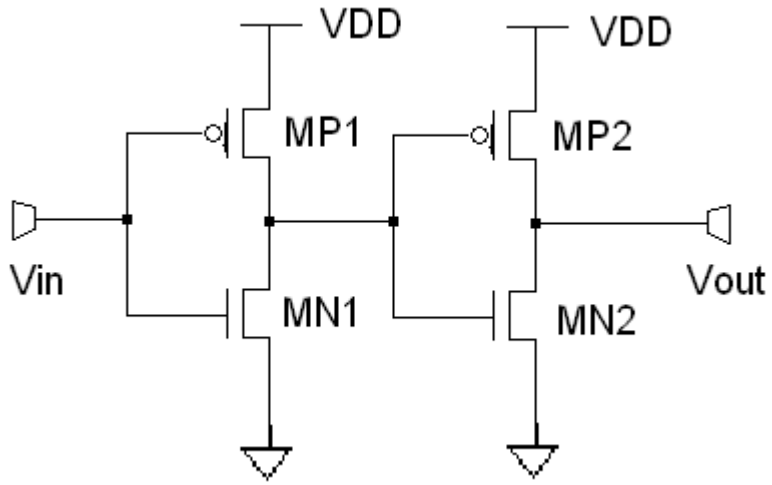

Figure 2.4: Voltage to IPI Converter - CMOS Delay Buffer

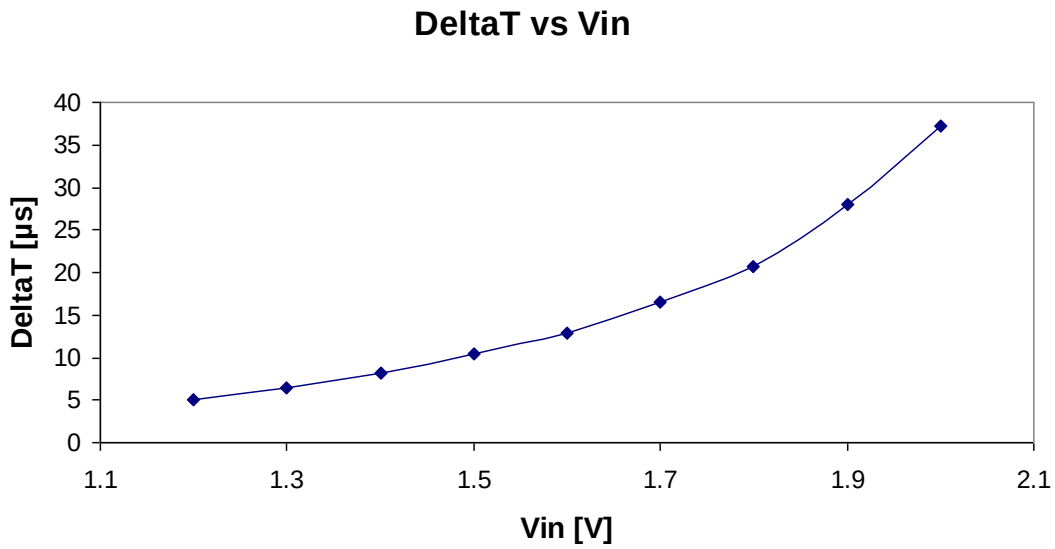

Figure 2.5: Plot of *DeltaT* versus $V_{in.}$ CMOS comparator has been used in this case.

2.3 Asynchronous IPI to Voltage Conversion

In addition to voltage to AIPI conversion, a circuit has also been designed that does IPI to voltage conversion. This circuit uses a current mirror and switched capacitors. Figure 2.6 shows the schematic for the IPI to Voltage Converter.

Figure 2.7 shows a plot of V_{out}, the output voltage, versus *DeltaT*, the time interval between two pulses. As expected, this plot is logarithmic in nature. Initially, ideal analog switches were used to insure sure that the basic circuit architecture was correct. Then those ideal switches were replaced by more realistic MOSFETs. Later, the capacitor values were tweaked in order to get the desired results.

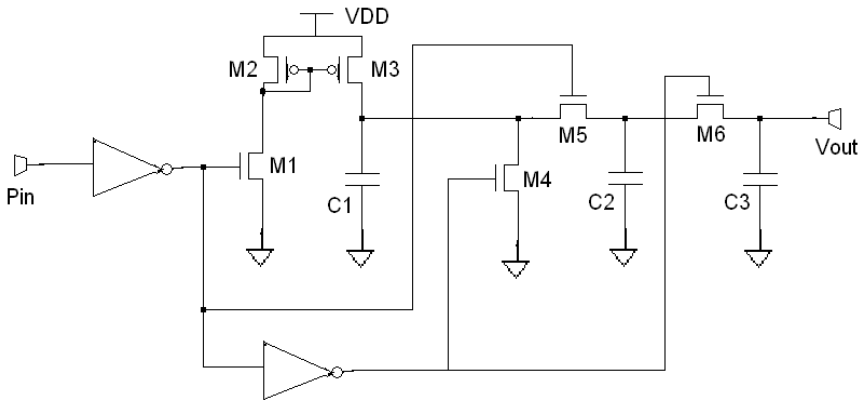

Figure 2.6: IPI to Voltage Converter Schematic

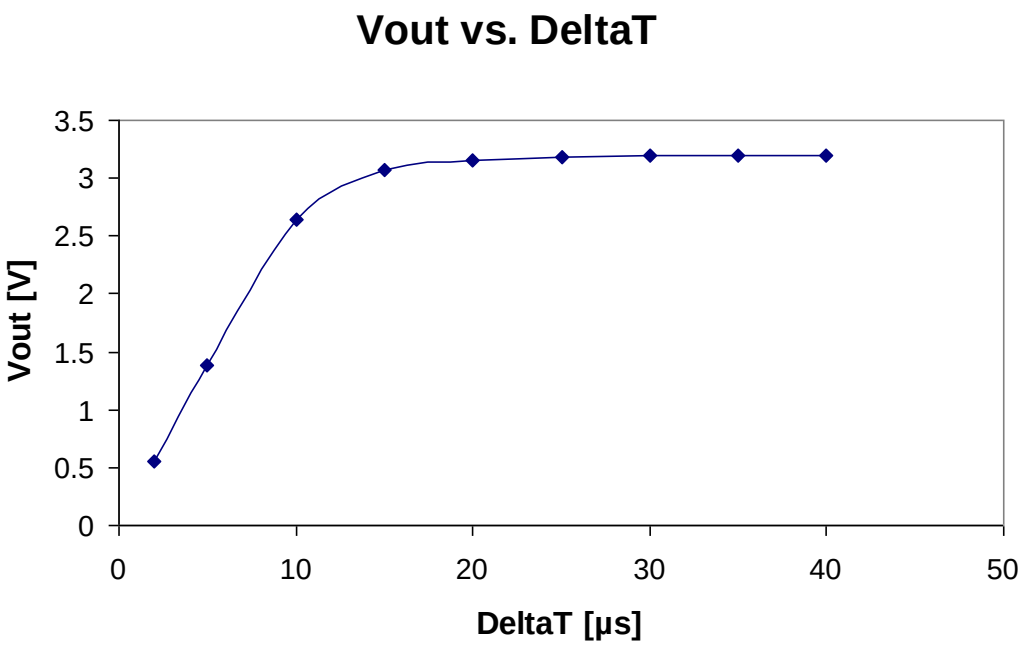

Figure 2.7: Plot of V_{out} versus *DeltaT*. MOS transitors have been used for the switches in the switched capacitor network.

2.4 Voltage to IPI to Voltage Test

As a final test of both converters, they were combined back-to-back into a dual converter that takes a voltage input, converts that into an IPI representation, and then converts that back into a voltage representation. Figure 2.8 shows the top level schematic for this circuit. The individual modules are identical to that presented with the exception that Figure 2.9 has the op-amp circuitry after the IPI to Voltage conversion that was needed to adjust the value of the output voltage, V_{out}. Since the AIPI representation does lose DC offset information, in these examples it is added back in for convenience sake. This is labeled as the subtractor block in the Voltage to IPI to Voltage Converter circuitry.

This Voltage to IPI to Voltage converter circuit behaves like a low-pass filter. According to the frequency domain plot (Figure 2.11), the corner frequency, f_c is around 100 Hz. This RC filter is where:

$$f_c = \frac{1}{2\pi RC} \qquad \textbf{Eq. 2.1}$$

In the above equation, f_c is the corner frequency, R is the value of resistance of the resistor, and C is the capacitor's capacitance.

A sinusoid input of various frequencies was applied to V_{in}, the input voltage signal. At low frequencies, where frequency is less than 50 Hz or f_c / 2, V_{out} matches V_{in} very closely, which is the desired result. In this case, V_{out} is a sinusoid that almost overlaps with the V_{in} input sinusoid with very little phase shift. Plots of the V_{out} and V_{in} signals as a function of time at the input frequency of 50 Hz are shown in Figure 2.10.

The frequency of the input voltage signal V_{in} was varied from 25 Hz up to 2 KHz. There is little distortion at the lower frequencies, where the frequency is less then f_c, but increasingly more distortion as the frequency

increases, where frequency is greater than f_c. As the input frequency increases, the output signal attenuates and has some phase shift. This means that the sinusoid peak of the V_{out} signal decreases as the input frequency increases. The deviation between V_{out} and V_{in} increases with increasing frequency, especially as frequency becomes much greater than f_c. Figure 2.11 contains the plot of the sinusoid peak voltage for V_{out} as a function of input frequency. Here, the voltage levels for input and output as well as the value for cut-off frequency are not of concern, since the main point here is demonstrating that the V→IPI→V converter behaves like a low-pass filter.

As indicated in Figure 2.11, the performance starts to degrade at several hundred Hertz. This frequency-induced performance degradation is due primarily to the capacitors in the IPI to Voltage Converter block. A solution to the degradation problem at higher frequencies would be to reduce the capacitance of the capacitor in the IPI to Voltage block that the current from the current mirror flows into. This way, the charge storing capacity of that capacitor decreases. By reducing that capacitance of the capacitor in the IPI to Voltage block, the corner frequency, $f_{c,}$ will increase based on reducing the capacitance of the RC low-pass filter based on Eq. 2.1.

Although some tweaking is possible, what is being seen here is a fundamental limitation in total frequency range that is due to the fundamental trade-offs between voltage and time. The logarithmic representation helps somewhat, but generally IPI representations will need to operate within well defined frequency ranges. This is not an unreasonable restriction since it applies to most mixed-signal circuitry. Though not done here, it is possible to increase the frequency range without significant circuit redesign. The f_c or cut-off frequency is different for one part of the wave from the other. This is due to the fact that the AIPI sampling rates are different for one part of the wave from the other. So even without the circuit constraints on input signal bandwidth or frequency range, there are fundamental issues that limit the frequency range

of the input voltage signal that gets converted to AIPI. According to Nyquist's Theorem, the input frequency must be less than half of the sampling frequency. For IPI, the sampling frequency means the worst case or slowest sampling frequency, since there are multiple sampling frequencies for a waveform that gets converted to IPI. For more complex waveforms which have sine wave components at different frequencies, the entire input waveform frequency range must be less than half of the worst case sampling rate. This concept will be discussed further in Chapter 4.

According to the frequency response plot in Figure 2.11, the poles are at approximately 50 Hz and 250 Hz, which are the points where the slope of the plot changes. The unity gain cutoff frequency is approximately 50 Hz, since that is the point where the peak V_{out} starts to drop below 2.5 V, the value of peak V_{in}. The phase angle at very low frequencies is 0 degrees, going to –45 degrees at 50 Hz, and then to –135 degrees at 250 Hz. It settles asymptotically at –180 degrees, since there are two poles. This system is stable since the phase at the unity gain cutoff frequency is around –45 degrees. We can tune the circuit so that the performance will be good for a wider frequency range by increasing the unity gain cutoff frequency, which in turn is done by increasing the frequency value of the poles. Stability should not be a problem since the first pole will always be the unity gain cutoff frequency, which will always have a phase angle of –45 degrees.

As shown in Figure 2.10, there is some distortion, or deviation of V_{out} from V_{in}, in certain places on the time axis. The maximum distortion is approximately 0.1 V.

Chapter 4 presents a more detailed analysis, which includes MATLAB simulations, to characterize the source of this distortion. A fundamental concept of IPI representation that leads to distortion is the difference in sampling rate of the upper portion of the input waveform and the sampling rate

of the lower portion of the input waveform. This difference in sampling rates increases as the input signal amplitude increases.

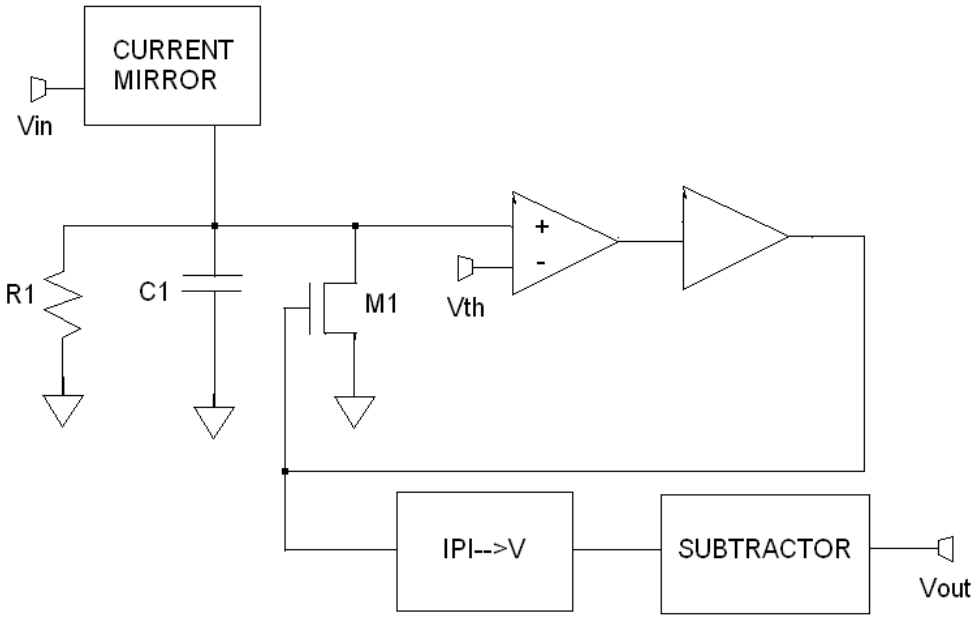

Figure 2.8: Voltage to IPI to Voltage Converter - Top Level Schematic

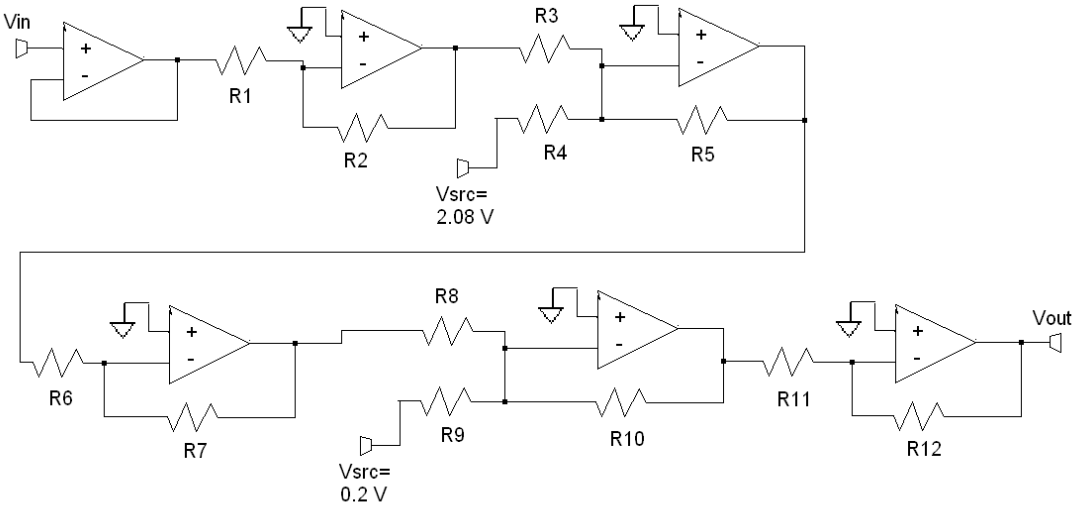

Figure 2.9: Subtractor Block of the Voltage to IPI to Voltage Converter

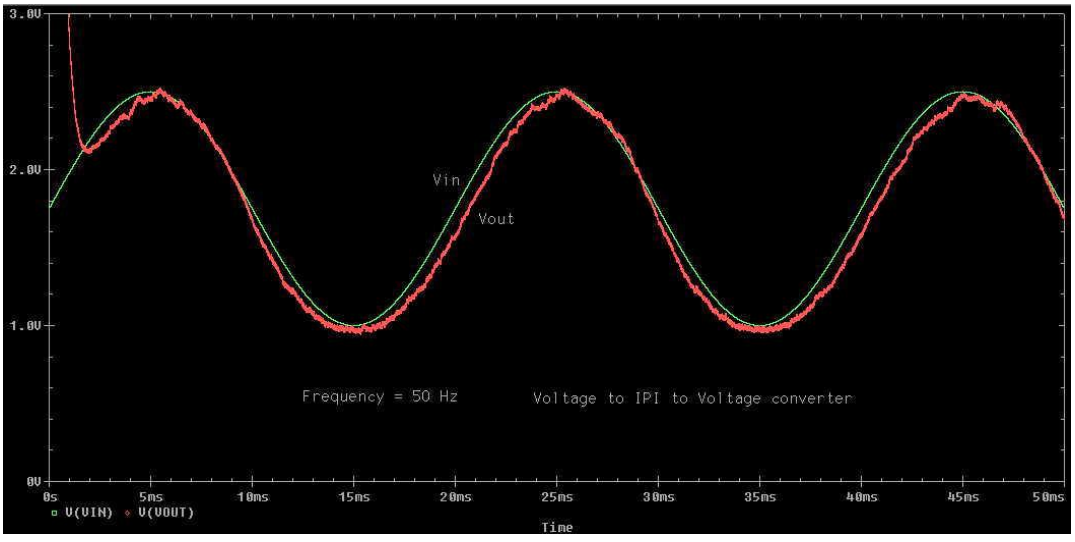

Figure 2.10: Plot of V_{out} and V_{in} as Function of Time. Input frequency is 50 Hz.

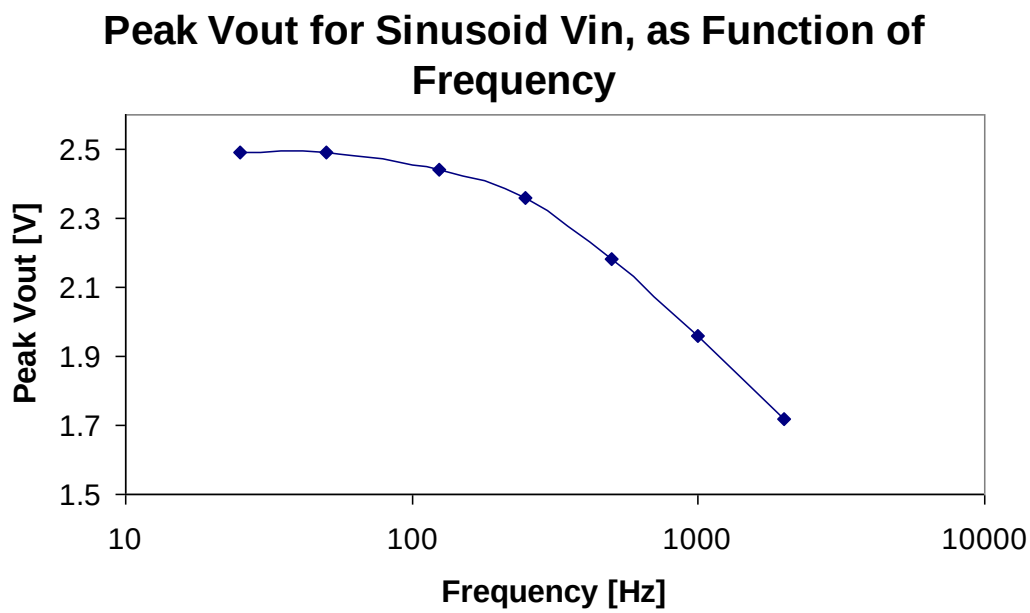

Figure 2.11: Plot of Peak V_{out} as Function of Input Frequency.

CHAPTER 3

LINEAR IPI CONVERSION CIRCUITS

3.1 Introduction

In the previous chapter, the basic design of the nonlinear IPI conversion circuits in PSPICE was discussed. Most of the work done here is based on the nonlinear IPI representation since this is based on previous work in using the integrate-and-fire model mentioned in Chapter 1 [1], [41]. The reasons why this thesis also covers linear IPI representation will be mentioned later in this chapter. Consequently, since we are proposing asynchronous IPI for data conversion and transmission, the remainder of this thesis discusses only AIPI. And when “IPI” is used, it will mean Asynchronous IPI.

In this chapter, a comparison between nonlinear and linear IPI is done to show the benefits of using linear IPI. IPI conversion circuits using a linear IPI representation were designed. Cost vs. performance analysis of the IPI converters in order to compare the IPI approach to existing methods is discussed in this chapter. Since transistors are almost free, the biggest cost in embedded remote sensor applications is going to be the power consumed by the circuit. Performance then deals with the amount of information transmitted in terms of sample rate and resolution, and any potential distortion of the signal that is introduced by the converters. The asynchronous IPI (AIPI) conversion circuits presented earlier were modified to reduce the power dissipation and to provide a better comparison with respect to commercial analog to digital to analog conversion in terms of power, speed, and accuracy. In order for the AIPI representation to be useful, it is important that the power

can be reduced without sacrificing speed or sampling rate. Also, low power utilization is critical for many of the potential applications of AIPI, such as wireless sensor networks.

ADC/DAC conversion was chosen as a reasonable benchmark for comparison since we view the most promising application of this work to be in remote sensing in environments which require sensor based analog to digital conversion prior to information transmission. However, there is a caveat, comparing a student developed chip to a highly optimized, mass manufactured commercial chip is not a completely fair comparison, but it does give us some ball-park figures. We only use commercial specifications here to give a rough idea of general ball-park of IPI to analog-digital conversion.

Our objective was to compare the power dissipated by the voltage (V) to AIPI converter (VIPC) to that required by an ADC, and then to compare the power dissipated by the AIPI to V converter (IPVC) and compare that with the power dissipated by the DAC. In this chapter, power dissipation for VIPC and IPVC was also compared to that of the other pulse modulation techniques such as pulse width modulation, pulse amplitude modulation, etc.

Finally, test chip fabrication was done for the AIPI conversion circuits using the TSMC 0.35-μm 4-metal layer process. This chip is also discussed in this chapter.

3.2 PSPICE Experimental Description

IPI conversion circuits were modified in PSPICE in order to reduce the power dissipation for comparison to an ADC of comparable accuracy in terms of power and sampling rate. The circuit simulations were performed using the TSMC 0.25μm CMOS technology for SPICE simulation. Some of the transistors in the voltage to IPI converter (VIPC) were operated in the subthreshold region in order to reduce the current and, therefore, the power

dissipation of the circuit. The threshold voltage of these transistors was reduced so that PSPICE can model subthreshold operation more accurately.

The capacitance that charges the voltage to the threshold was reduced to reduce power in the VIPC circuits. The smaller capacitance is better since less current is needed to charge the capacitor to get the same voltage ramp rate, which translates into lower power at the same speed.

Reducing the capacitance also reduces the area of the capacitor, and therefore reduces the overall area consumption of the circuit. When capacitance is lowered, the current mirror that feeds current into the capacitor is modified so that the current mirror can feed less current into the capacitor.

For the IPI to V converter (IPVC), the capacitance that is charged by the current mirror is reduced. As in the VIPC case, when the capacitance is reduced in the IPI to V case mentioned here, the current mirror that feeds current into the capacitor is modified to source less current.

3.3 Comparison of Logarithmic (Nonlinear) and Linear IPI Representations

The experiments mentioned in the previous chapter that were done in PSPICE using the logarithmic IPI representation. Experiments were performed in both MATLAB and PSPICE to compare the characteristics of both the logarithmic and linear representations of IPI. A linear representation is where *DeltaT* vs. V_{in} is linear for the VIPC and V_{out} vs. *DeltaT* is linear for the IPVC, which can be implemented by using a capacitor only integrator. Experiments were done in MATLAB to determine how the distortion of the voltage to linear IPI to voltage conversion compares with the distortion of the voltage to logarithmic (nonlinear) IPI to voltage conversion. In the MATLAB experiments, the amplitude (*e.ma*) was varied for two sets of offset bias (*e.bias*) values, *e.bias*=3 and *e.bias*=8. For the PSPICE experiments, Power vs. V_{in} simulations were done for both the logarithmic (nonlinear) and linear

cases. In PSPICE, an ideal op-amp comparator that dissipates no power was used. For the linear case, V_{thresh} (the threshold voltage needed to generate the pulse) is proportional to V_{in} and the input to the op-amp current mirror is a constant value of 0.3 V.

Figure 3.1 contains the plot of Distortion vs. Amplitude in MATLAB for the case where the offset bias is 3. This plot has results for both linear and logarithmic representations. Figure 3.2 shows the plot for Distortion vs. Amplitude in MATLAB when offset bias is 8. This plot also has results for both linear and logarithmic representations. As both plots show, the linear representation of voltage to IPI to voltage conversion leads to less distortion compared to the logarithmic representation.

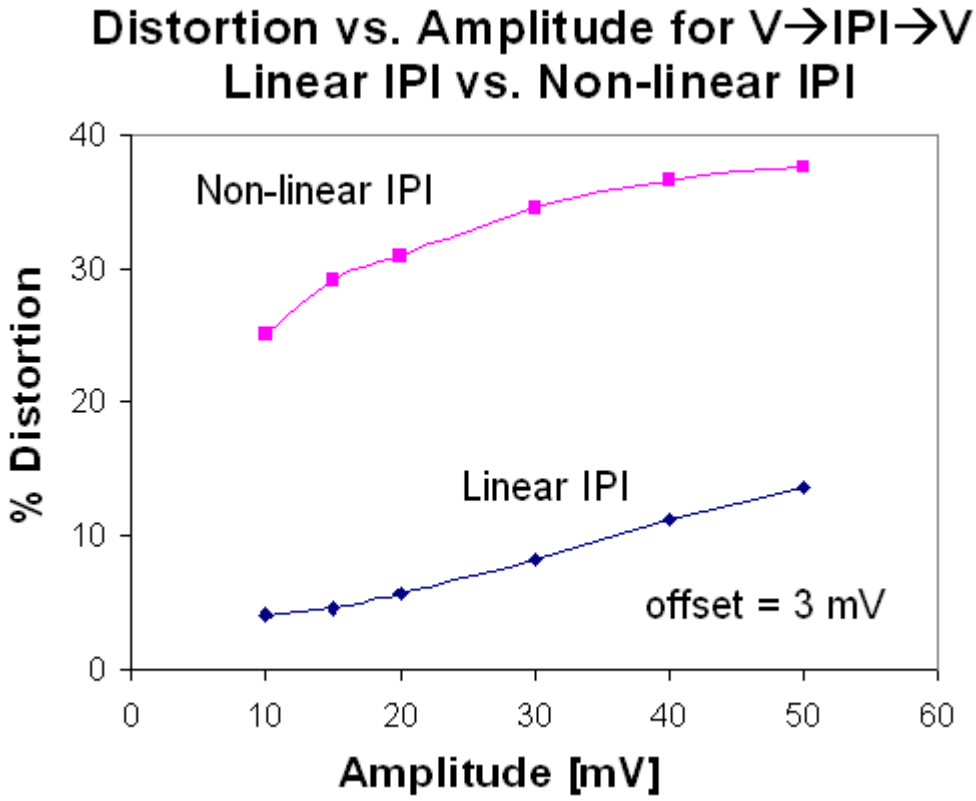

Figure 3.1: Plot of Distortion vs. Amplitude for both linear and non-linear (or logarithmic) representations of V→IPI→V conversion in MATLAB. In this case, offset bias (*e.bias*) is 3 mV.

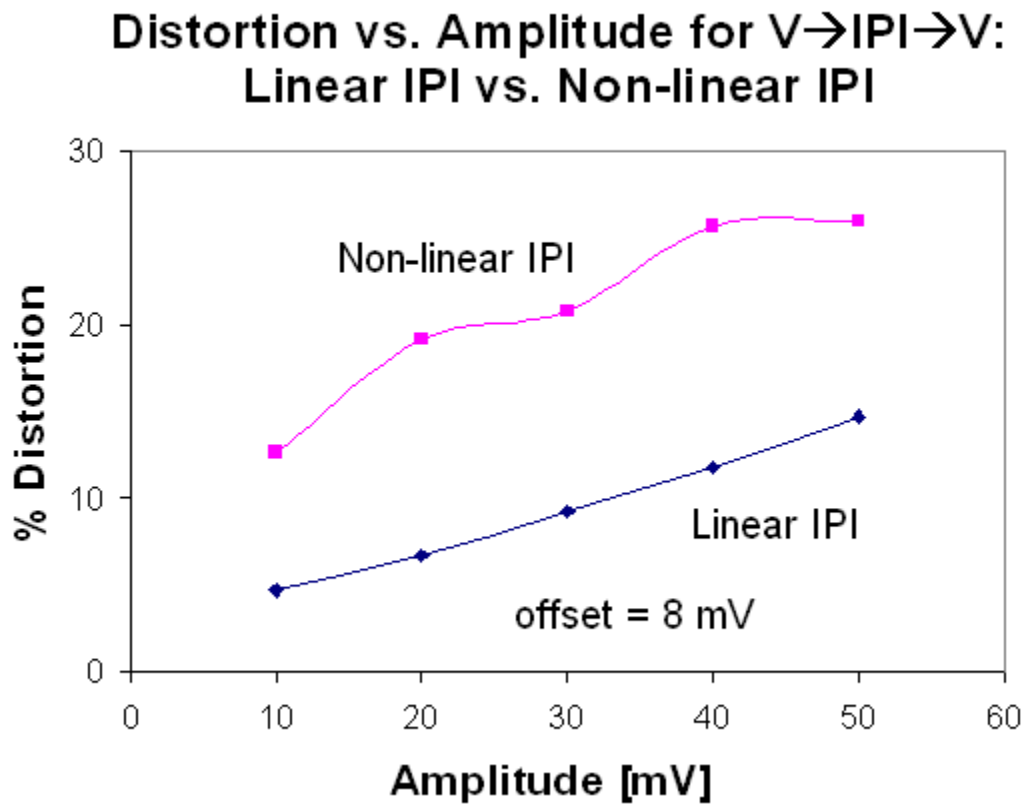

Figure 3.2: Plot of Distortion vs. Amplitude for both linear and non-linear (or logarithmic) representations of V→IPI→V conversion in MATLAB. In this case, offset bias (*e.bias*) is 8 mV.

The plot for Average Power vs. V_{in} for the voltage to IPI conversion (VIPC) experiments done in PSPICE that compare the linear representation with the logarithmic representation is shown in Figure 3.3. Here, C1, the capacitance being charged to V_{thresh}, is 50 pF, and V_{thresh} is 0.05 V for the logarithmic representation case. Figure 3.4 contains the plot for *DeltaT* vs. V_{in} for the linear representation. In this plot, the results are linear as expected. As Figure 3.3 demonstrates that using the linear representation for voltage to IPI conversion is a better choice since power remains almost constant as V_{in} increases, while in the logarithmic or nonlinear representation case, power increases significantly as V_{in} increases. A possible reason for this is that in the linear IPI case the current that the current mirror draws from the power

supply is constant, while in the nonlinear IPI case, the current mirror draws more current from the power supply as the input voltage, V_{in}, increases. Therefore, from both a distortion and power dissipation point of view, using the linear representation for voltage to IPI conversion is a better choice compared to the logarithmic or nonlinear representation. However, using linear IPI representation reduces dynamic range, therefore the decision on which type of IPI representation to use is application dependent.

These results also indicate that there is a potentially serious problem related to distortion, especially the intrinsic distortion, which is covered in Chapter 4.

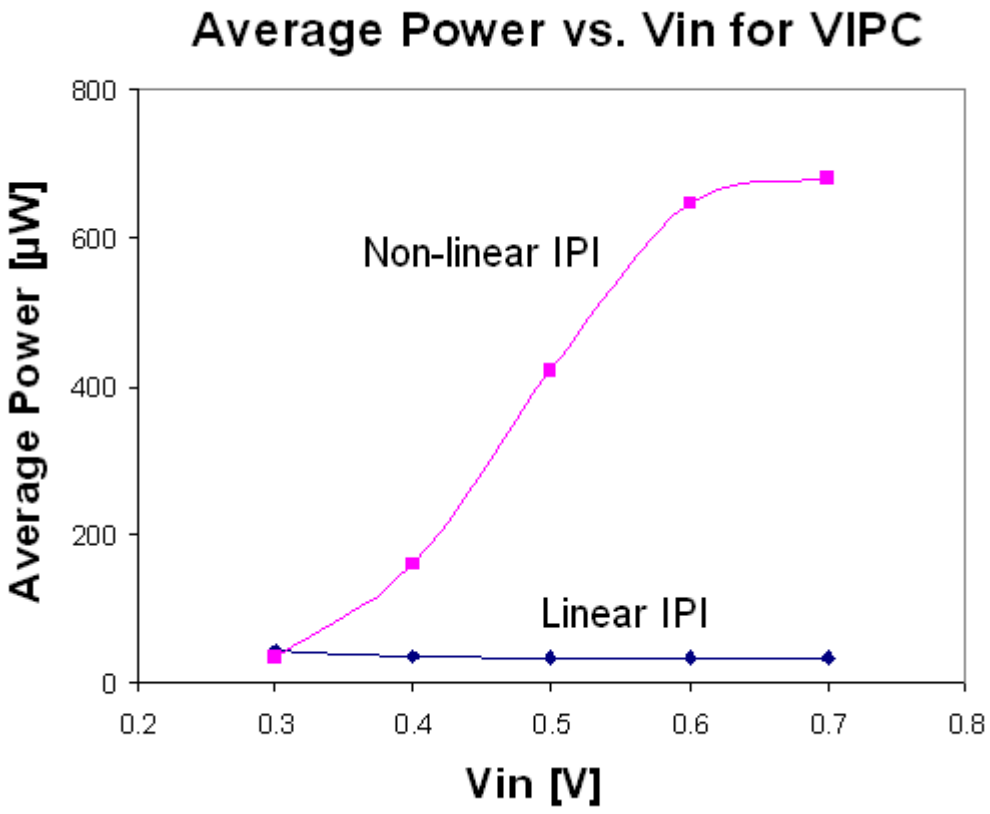

Figure 3.3: Plot of Average Power vs. V_{in} for VIPC. This work was done in PSPICE. Both the linear and logarithmic representations are shown. In this case, C1, the capacitance being charged to V_{thresh}, is 50 pF. V_{thresh} (threshold voltage needed to generate the pulse) is proportional to V_{in} for the linear case, and V_{thresh} is 0.05 V for the logarithmic representation case.

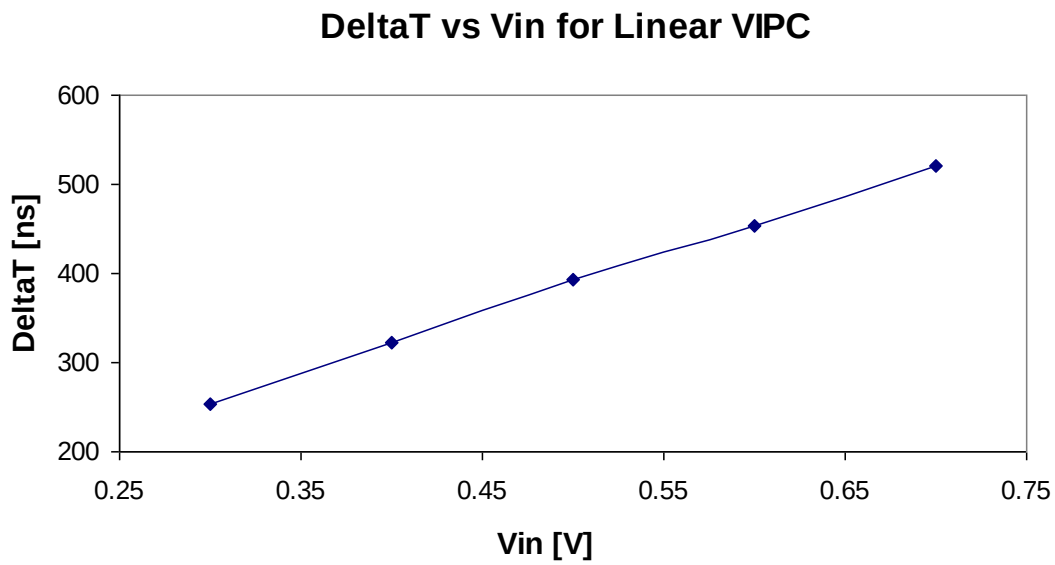

Figure 3.4: Plot of DeltaT vs. V_{in} for the VIPC using linear representation and ideal op-amp comparator. This plot shows a PSPICE simulation of a circuit that is actually producing linear results. Here, C1=50 pF, and Vthresh is proportional to V_{in}.

3.4 Voltage to IPI Conversion using CMOS Op-Amp Comparator

The CMOS op-amp comparator was added to the linear VIPC, replacing the ideal zero-power dissipating comparator. Initially, the ideal comparator was used so that the surrounding circuitry outside the comparator could be optimized first. However, for more realistic power estimates, it is important to have the CMOS op-amp comparator in the final schematic. Here, V_{DD} is 2 V, and C1 (capacitance being charged to V_{thresh}) is equal to 5 pF. The input to the op-amp current mirror is a constant of 0.3 V, and V_{thresh} is equal to V_{in}.

The plot for *DeltaT* vs. V_{in} using both the linear representation and the CMOS op-amp comparator in the VIPC circuit simulated in PSPICE can be

found in Figure 3.5. In this plot, the *DeltaT* curve as a function of V_{in} is linear as expected. Figure 3.6 shows the plot of Power vs. V_{in} for this IPI conversion circuit. Here, both quiescent and average powers are plotted. Figures 3.7-3.10 shows the schematics for the linear VIPC circuit. Figure 3.7 is the top level schematic. Figure 3.8 is the op-amp current mirror. Figure 3.9 is the CMOS op-amp comparator, and Figure 3.10 is the CMOS Delay block.

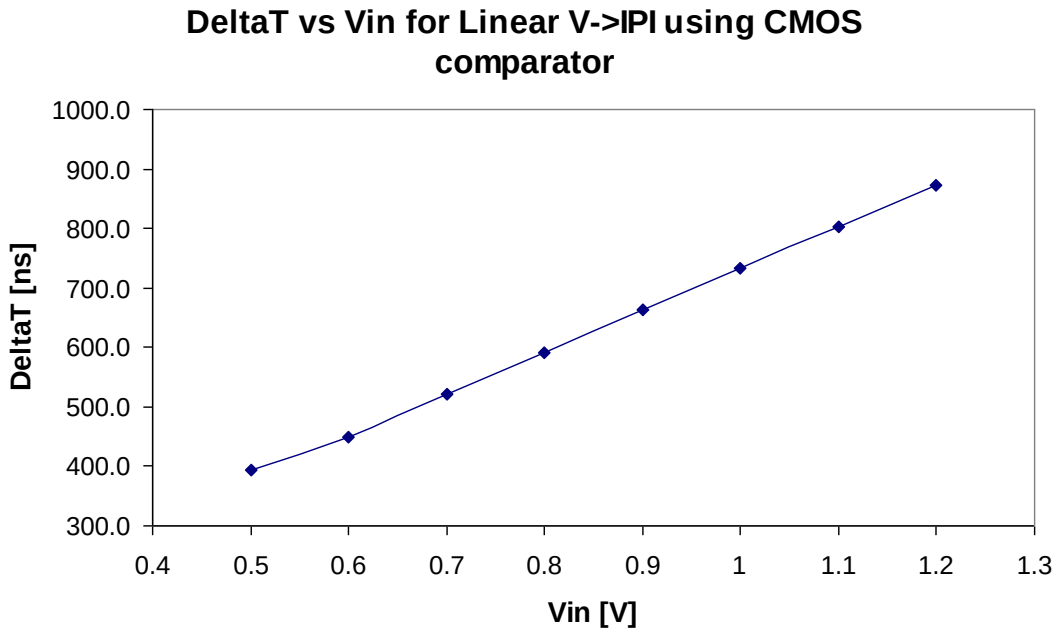

Figure 3.5: Plot of *DeltaT* vs. V_{in} for voltage to asynchronous IPI converter using linear representation and CMOS op-amp comparator. Here, *VDD* (power supply)=2 V, C1=5 pF, and Vthresh=V_{in}. The results are linear as expected.

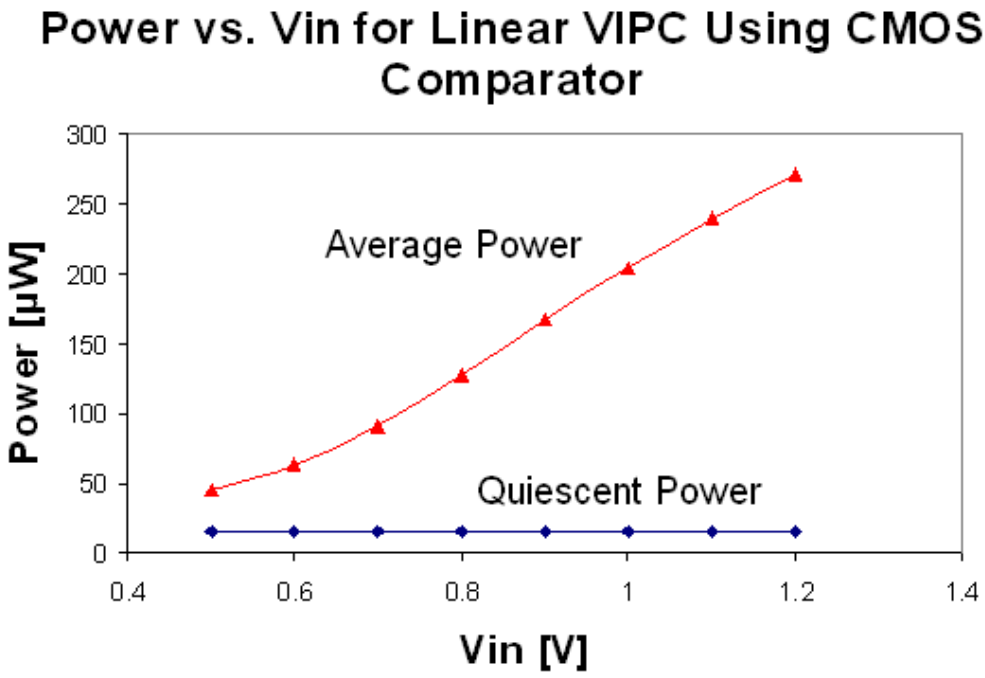

Figure 3.6: Plot of Power vs. V_{in} for voltage to asynchronous IPI converter using linear representation and CMOS op-amp comparator. Here, VDD=2 V, C1=5 pF, and $V_{thresh}=V_{in}$.

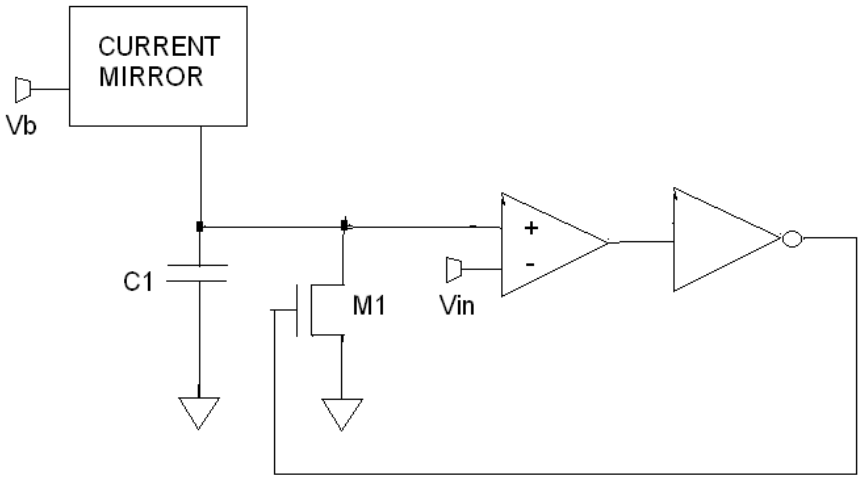

Figure 3.7: Top level schematic for the linear representation VIPC using the CMOS op-amp comparator.

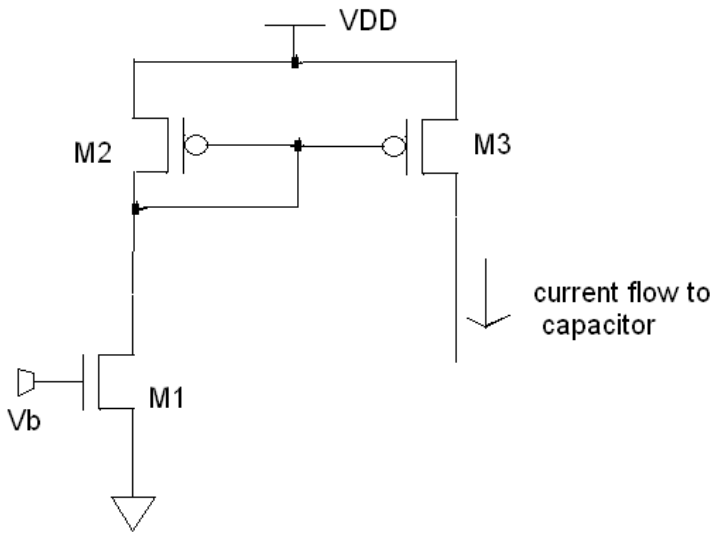

Figure 3.8: Schematic diagram of current mirror for linear VIPC.

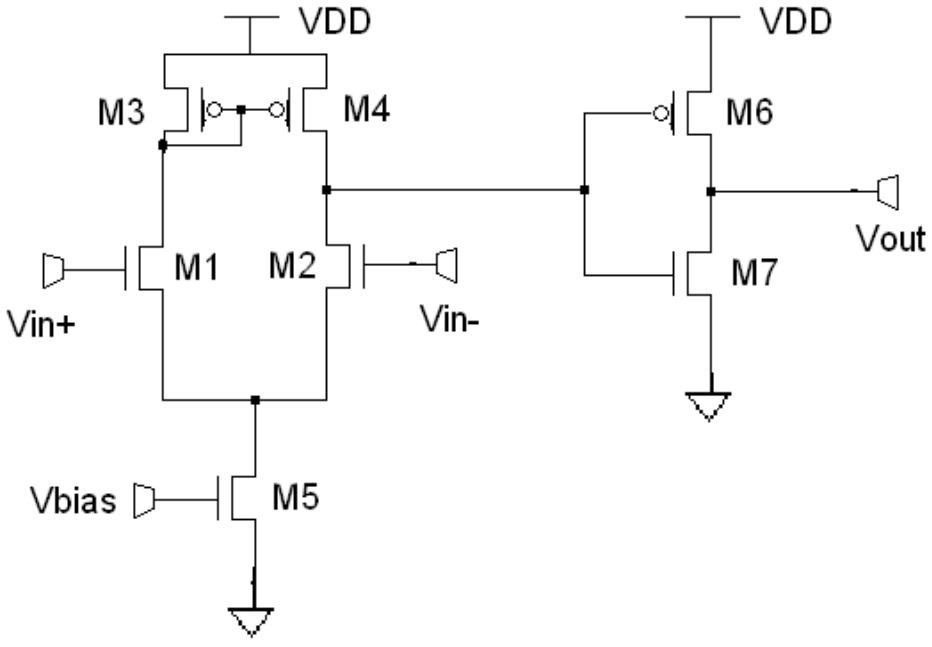

Figure 3.9: Schematic for CMOS comparator used in linear voltage to asynchronous IPI converter.

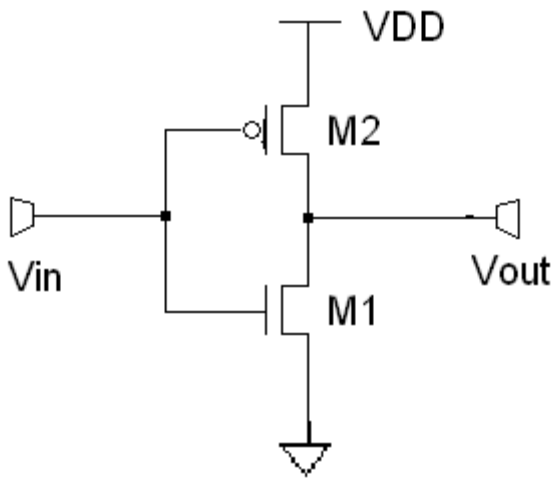

Figure 3.10: Schematic for CMOS Delay block for V→IPI converter. This component is necessary since the output of the CMOS op-amp comparator is inverting, and an extra inverter is needed to make the final IPI output non-inverting. This block is at the output of the op-amp comparator.

When comparing the voltage (V) to IPI converter (VIPC) mentioned above with the analog to digital converter (ADC), the power dissipation for the VIPC, is less than 300 µW if the average power is taken into consideration, even after adding the power from the CMOS comparator. The average power dissipation given for one of the typical ADCs from Silicon Laboratories is 1.350 mW [46]. The maximum DeltaT from the results in this work for the VIPC is less than 1000 ns, which causes the sampling rate to be greater than 1 Msps, while the sampling rate for this ADC is 100 ksps. The resolution for this ADC is 12 bits. Other ADCs were also studied. One ADC from Texas Instruments (TI) is a 16-bit resolution Delta Sigma ADC. This has a typical power consumption of 330 mW, and a maximum sampling rate of 1.25 Msps [47]. Another 16-bit resolution Delta Sigma ADC from TI has typical power consumption of 270 µW, and the maximum sampling rate is equal to 128 sps [48]. A 10-bit resolution SAR ADC from TI has a typical power consumption of about 3.8 mW, and the maximum sampling rate is equal to 1.25 Msps [49]. A 16-bit resolution Sigma Delta ADC from Maxim/Dallas Semiconductor has

typical power consumption equal to 960 µW, and conversion rate equal to 500 sps [50]. All these ADCs have higher power/sampling rate ratios compared to the VIPC circuit. The VIPC has power/sampling rate ratio that is less than 300 µW/1 Msps, which is 300 pJ. The ADCs discussed here were typical commercial chips for comparison purposes.

3.5 IPI to Voltage Conversion

The IPI to voltage converter (IPVC) using the linear representation, i.e., where the plot of V_{out} vs. *DeltaT* is linear, was also developed with PSPICE, since if the voltage to IPI conversion is linear, the reverse process, IPI to voltage conversion, also needs to be linear. The input *DeltaT*'s were reduced in order to make the final results linear and no other changes were necessary. Since for low *DeltaT*, the output voltage is proportional to the integration time of the capacitor, which is related to how long the input pulse is low, while for high *DeltaT*, the output voltage saturates towards a certain value that is close to *VDD*. The values for capacitor C1 were 10 pF and 5 pF. C1 was reduced along with modifying the current mirror in order to obtain lower power figures.

The plot for V_{out} vs. *DeltaT* for the IPVC using linear representation and with C1=10 pF is shown in Figure 3.11. The results are linear as expected. Figure 3.12 contains the plot for Power vs. *DeltaT* for this exact same circuit. Figure 3.13 is similar to the plot in Figure 3.11 except that C1=5 pF. Figure 3.14 is similar to the plot in Figure 3.12 except that C1=5 pF. Figure 3.15 shows the schematic for the IPVC circuit using the linear representation.

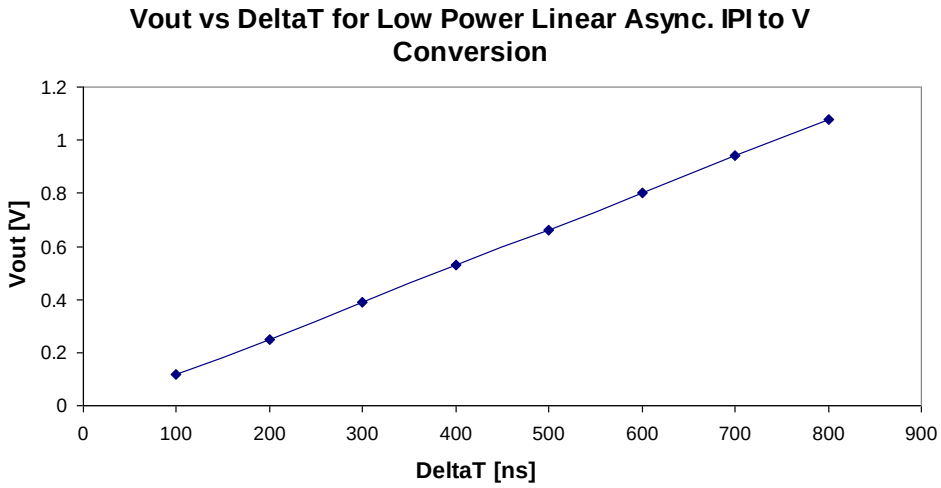

Figure 3.11: Plot of V_{out} vs. *DeltaT* for asynchronous IPI to voltage converter using linear representation, where C1=10 pF.

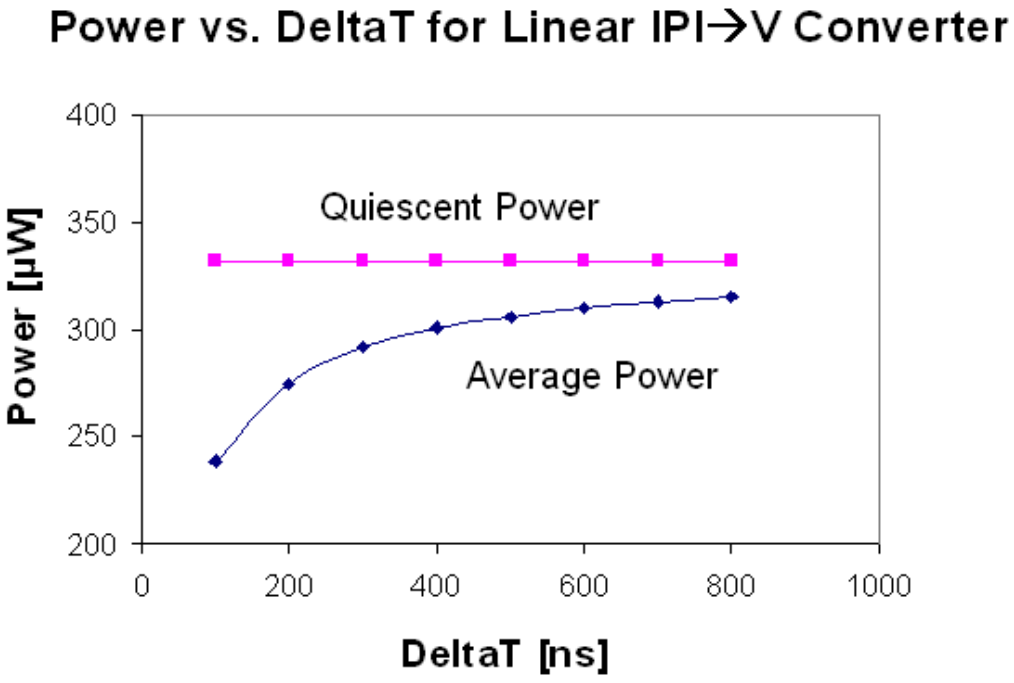

Figure 3.12: Plot of Power vs. DeltaT for asynchronous IPI to voltage converter using linear representation, PSPICE simulation. In this case, C1=10 pF. Here, both quiescent power and average power are plotted.

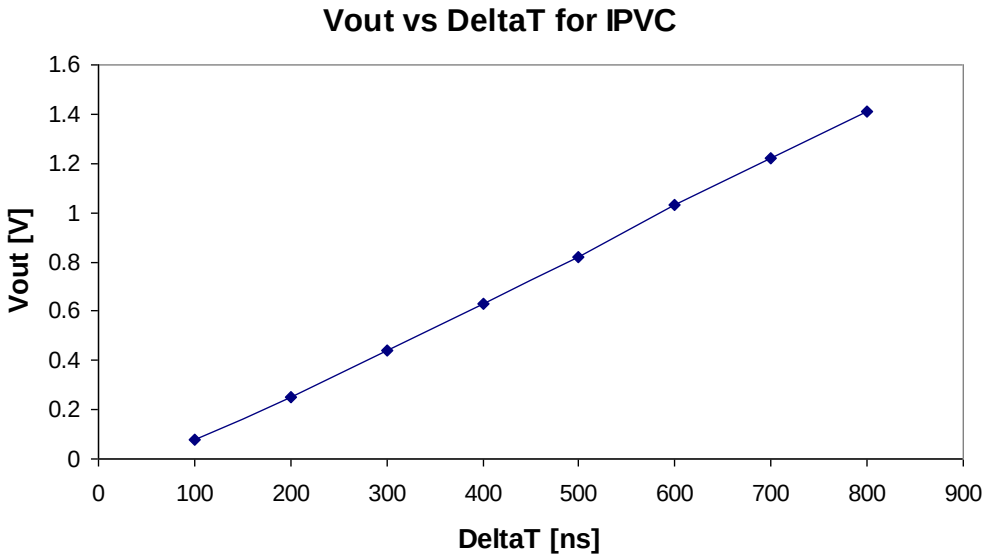

Figure 3.13: Plot of V_{out} vs. *DeltaT* for asynchronous IPI to voltage conversion using linear representation. In this case, C1=5 pF and *VDD*=2 V.

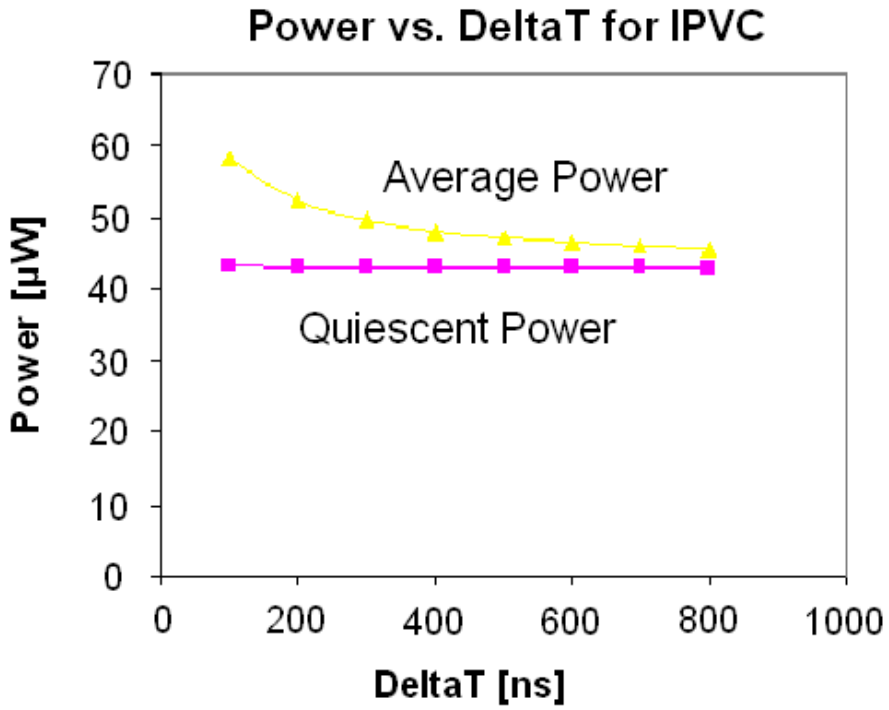

Figure 3.14: Plot of Power vs. DeltaT for asynchronous IPI to voltage converter using linear representation. In this case, C1=5 pF and *VDD*=2 V. Both average and quiescent power are plotted here.

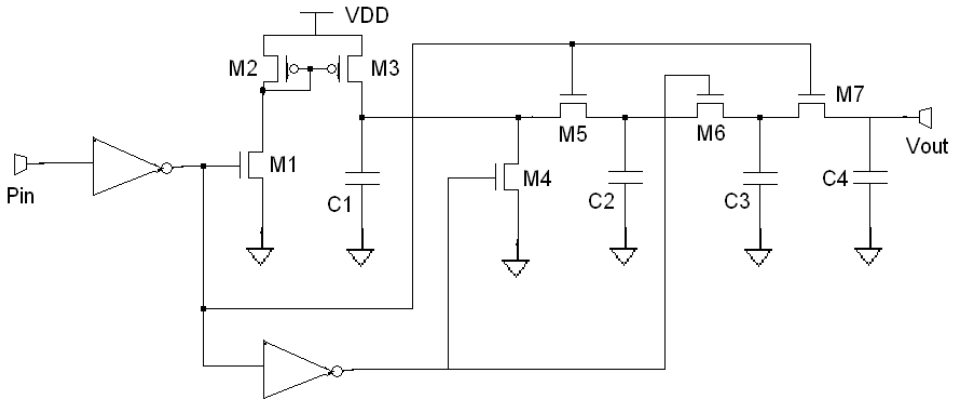

Figure 3.15: Schematic of IPI→V converter using linear representation.

As the plots for the IPI to V conversions (with a linear representation) demonstrate, reducing the capacitance charged by the current mirror along with modifying the current mirror can reduce power dissipation. The average power dissipation for the IPVC obtained from the results above is in the range of 40 to 60 μW, while the average power dissipation given for a typical 12-bit DAC from Silicon Laboratories is 330 μW [46]. These are only preliminary results since only power and sampling speed were taken into account here. The final figure of merit that takes into account power, speed, and resolution will be discussed later in this chapter.

3.6 Voltage to IPI to Voltage Conversion

3.6.1 V→IPI→V Simulations

The low power voltage to IPI to voltage converter was also simulated in PSPICE. Initially, experiments were conducted using a sinusoidal wave input in order to determine if this circuit contributes minimal distortion.

Figure 3.16 shows a plot of the V to IPI to V simulation. As the plot shows, the only significant distortion that exists is the time shift between the

input and output voltage sine waves. The delay of the output is approximately 10 us after the input. The average power dissipation is approximately 271 µW, which is less than the combined power dissipation of the ADC and DAC which equals 1.350 mW + 0.330 mW = 1.680 mW [46]. The ADC and DAC here are commercial chips from Silicon Laboratories.

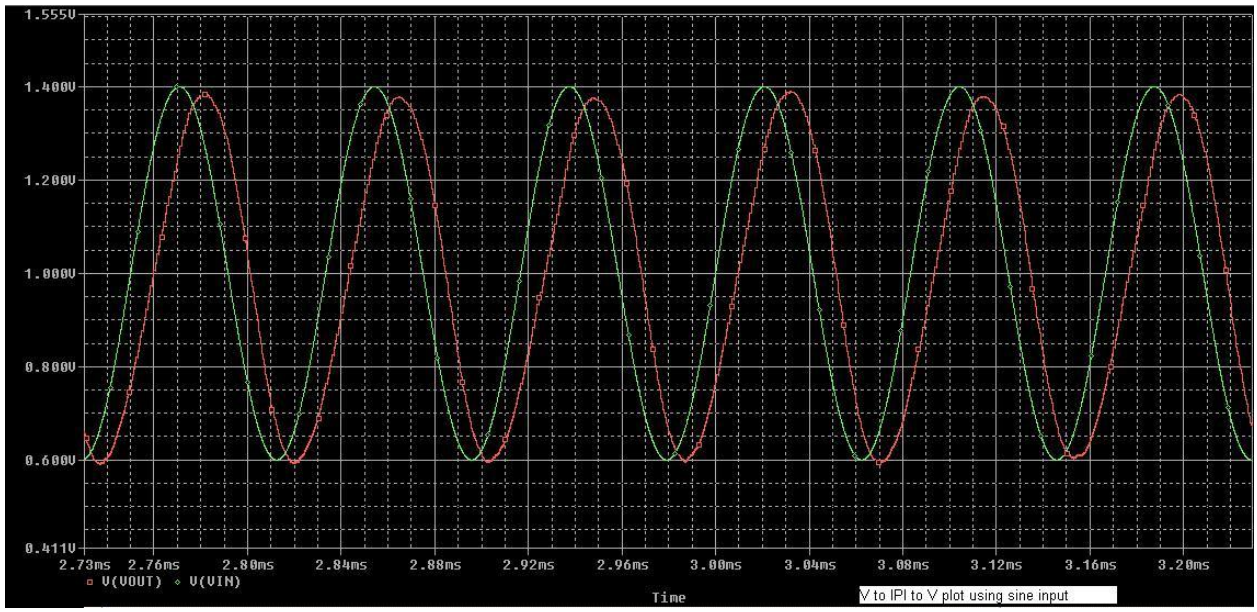

Figure 3.16: Plot of Voltage to IPI to Voltage simulation in PSPICE. Here, both the output voltage, Vout, and input voltage, Vin, are shown.

3.6.2 V→IPI→V Transfer Function

The transfer functions for the V→IPI and IPI→V converters were derived under DC conditions in the time domain, which means the charging and discharging of the capacitors by the current mirrors for a DC voltage input for the V→IPI→V conversion process. After that, the frequency domain transfer function for the V→IPI→V converter was derived by multiplying the time-domain transfer function derived in the first step by the transfer function of the switched-capacitor low-pass filter at the output of the IPI→V converter. The V→IPI transfer function was derived in the time domain. **Eq. 3.1A** describes the input/output relationship of the VIPC based on the charging of the capacitor *C1* based in Figure 3.7.

$$I_A = C_A \frac{dV_A}{dt} = C_A \frac{V_{IN}}{\Delta T} \qquad \textbf{Eq. 3.1A}$$

Where,

- I_A = current from current mirror in VIPC that charges the capacitor *C1*
- C_A = capacitance value of *C1* in the VIPC
- V_A = voltage across *C1*
- V_{IN} = input voltage for VIPC
- ΔT = *DeltaT*, or time between two IPI pulses. This is the VIPC's output.

Here, change in voltage is equal to V_{IN} since the capacitor charges from 0 to V_{IN} before the pulse is formed. The transfer function, H_{VIPC}, for the VIPC can be derived from **Eq. 3.1A** to obtain:

$$H_{VIPC} = \frac{\Delta T}{V_{IN}} = \frac{C_A}{I_A} \qquad \textbf{Eq. 3.1B}$$

In the IPI→V conversion, **Eq. 3.1C** describes the charging of capacitor *C1* shown in Figure 3.15.

$$I_B = C_B \frac{dV_B}{dt} = C_B \frac{V_2}{\Delta T} \qquad \textbf{Eq. 3.1C}$$

Where,

- I_B = current from current mirror in IPVC that charges the capacitor *C1*
- C_B = capacitance value of *C1* in the IPVC
- V_B = voltage across *C1*
- V_2 = maximum peak voltage of the triangular sawtooth wave V_B in the IPVC circuit
- ΔT = *DeltaT*, or time between two IPI pulses. This is the IPVC's input.

In the IPVC, the output voltage V_{OUT} is the average value of V_B since there is a low-pass filter at the output of the IPVC where V_B is its input and V_{OUT} is its output. Since V_B is a sawtooth wave with a period equal to ΔT, V_{OUT} is the area of triangle under this sawtooth wave divided by ΔT. **Eq. 3.1D** gives the formula for the area of the triangle (represented by A_{SAW}) under the sawtooth wave V_B and **Eq. 3.1E** gives the formula for computing V_{OUT} as a function of V_B.

$$A_{SAW} = \frac{(V_2)(\Delta T)}{2} \qquad \textbf{Eq. 3.1D}$$

$$V_{OUT} = \text{average}(V_B) = \frac{A_{SAW}}{\Delta T} = \frac{V_2}{2} \qquad \textbf{Eq. 3.1E}$$

From **Eq. 3.1C** and **Eq. 3.1E**, the DC transfer function for the IPVC, represented by H_{IPVC}, can be derived as a function of I_B and C_B.

$$H_{IPVC} = \frac{V_{OUT}}{\Delta T} = \frac{1}{2}\left(\frac{I_B}{C_B}\right) \qquad \textbf{Eq. 3.1F}$$

Therefore, the DC transfer function for the V→IPI→V circuit can be represented by K_{DC} which is the ratio of V_{OUT} over V_{IN} under DC or low-frequency conditions.

$$K_{DC} = (H_{VIPC})(H_{IPVC}) = \frac{V_{OUT}}{V_{IN}} = \frac{1}{2}\left(\frac{C_A}{I_A}\right)\left(\frac{I_B}{C_B}\right) \qquad \textbf{Eq. 3.1G}$$

The frequency domain component of the V→IPI→V transfer function is derived from the low-pass filter network in the IPI→V converter. This low-pass filter is a 3rd order filter portrayed in Figure 3.15 at the IPVC output. **Eq. 3.1H** portrays this low-pass filter's transfer function, represented as $K_{LPF}(s)$.

$$K_{LPF}(s) = \left(\frac{1}{1+sZ_1C_{F1}}\right)\left(\frac{1}{1+sZ_2C_{F2}}\right)\left(\frac{1}{1+sZ_3C_{F3}}\right) \qquad \textbf{Eq. 3.1H}$$

Where,

- Z_1 = impedance of MOSFET *M5* in IPVC circuit
- C_{F1} = capacitance value of *C2* in the IPVC
- Z_2 = impedance of MOSFET *M6* in IPVC circuit
- C_{F2} = capacitance value of *C3* in the IPVC
- Z_3 = impedance of MOSFET *M7* in IPVC circuit
- C_{F3} = capacitance value of *C4* in the IPVC

This above equation is based on the formula for the transfer function of the simple single-pole first-order RC low-pass filter, given in **Eq. 3.1I.** This transfer function is represented as $M_{LPF}(s)$. In **Eq. 3.1H**, Z_1, Z_2, and Z_3 represent impedances of the MOSFETs which behave like resistors.

$$M_{LPF}(s) = \left(\frac{1}{1+sRC}\right) \qquad \textbf{Eq. 3.1I}$$

Based on **Eq. 3.1H**, the 3 poles are at these following frequencies:

$$p_1 = \left(\frac{1}{2\pi Z_1 C_{F1}}\right) \qquad \textbf{Eq. 3.1J}$$

$$p_2 = \left(\frac{1}{2\pi Z_2 C_{F2}}\right) \qquad \textbf{Eq. 3.1K}$$

$$p_3 = \left(\frac{1}{2\pi Z_3 C_{F3}}\right) \qquad \textbf{Eq. 3.1L}$$

The overall transfer function of the V→IPI→V converter, $H(s)$, is the time-domain DC component of the transfer function multiplied by the frequency-domain component of the transfer function. It is derived from **Eq. 3.1G** and **Eq. 3.1H**.

$$H(s) = \frac{V_{OUT}(s)}{V_{IN}(s)} = (K_{DC})(K_{LPF}(s)) \qquad \textbf{Eq. 3.1M}$$

3.7 Figure of Merit

3.7.1 Definition of Figure of Merit for Pulse Comparison

To help the comparison process, we have developed a figure of merit (FOM) that takes into account: power, frequency, and resolution. Even though IPI has lower power than existing signaling schemes, IPI also has a slower sampling rate since it is asynchronously modulated along the time axis. The IPI results were simulated in SPICE using 0.25 µm models were compared to other circuits that use 0.25 µm process technologies, since the process technologies should be similar for a fair comparison. Frequency is the worst case or slowest sampling rate for AIPI representations, since there is no clock in AIPI and the sampling rate is non-uniform across the input voltage range. The frequency of the synchronous pulse representations (such as PWM, PAM) is just the clock frequency. Resolution for AIPI is defined in the next subsection, since AIPI involves a direct conversion from analog to IPI pulse representation. The resolution for the synchronous pulse representations is the number of bits per clock cycle. The FOM here will be:

$$FOM = E_{bit} = \frac{P / F_s}{N_b} = \frac{P}{F_s \times N_b} \qquad \textbf{Eq. 3.2}$$

Where,

- E_{bit} = energy used per bit, in multiples of joules/bit (units can be pJ/bit, nJ/bit, etc.)
- P = power consumption
- F_s = worst-case sampling frequency for AIPI, or clock frequency for synchronous pulse representation
- N_b = resolution in terms of number of bits

The energy consumed is power multiplied by time, where 1 joule is equal to 1 Watt multiplied by 1 second. In the FOM described in **Eq. 3.2**, the time is 1 / F_s, or the clock period for the synchronous signal representations. The FOM is in terms of energy per bit. So energy will decrease if power is same but frequency increases. Higher resolution will cause the FOM to decrease. A lower FOM is better.

3.7.2 Definition of Resolution for IPI

The variable A_{level} is the smallest increment in which the input voltage for a V→IPI→V converter can be changed without a corresponding change being reflected at the output. A smaller A_{level}, manes a higher resolution. The resolution is increased by increasing the gain of the comparator that compares the voltage on the capacitor with V_{in} in the linear IPI representation. However, there will be a tradeoff if the resolution is increased, since power increases when comparator gain is increased.

The variable V_{range} is the input voltage range for the V→IPI→V converter. This value is the difference between maximum V_{in} and minimum V_{in}.

$$n_{level} = 2^n = \frac{V_{range}}{A_{level}} \qquad \textbf{Eq. 3.3}$$

Here, n_{level} is the total number of analog input voltage levels that can be used in the V->IPI->V conversion. The resolution, N_b, or number of bits, is the largest integer that is less than the variable *n* mentioned in **Eq. 3.3**.

The resolution for IPI is therefore represented as the number of bits that can be transmitted in the timeframe between two IPI pulses.

Pulse Technique	Power	Sampling Frequency	Resolution	FOM: Energy/bit [pJ/bit]	Endnotes
IPI differential (linear)	262.5 µW	Min: 1.4 MHz Max: 2.1 MHz	8 bits	23.4	1, 2
PWM	66.5 mW	200 MHz	2 bits	166.25	3, 4
PAM	1 W	2.5 GHz	2 bits	200	5
SDM	65 mW	30 MHz	14 bits	154.8	6, 7, 8
SDM	65.8 mW	70.4 MHz	13.1 bits	71.4	9

Table 3.1: This table shows the comparison of Figure of Merit (FOM) of IPI vs. other signaling types for 0.25 µm process technology. Sampling frequency for IPI to compute the FOM is the worst case or slowest sampling frequency.

3.7.3 Analysis of FOM

Table 3.1 compares the FOM for IPI vs. other signaling techniques. IPI is compared against pulse width modulation (PWM), pulse amplitude modulation (PAM), and pulse density modulation (PDM). PDM is the same as pulse frequency modulation (PFM) as defined in Chapter 1. The PFM circuits mentioned in the table above are sigma-delta modulation (SDM) ADCs since SDM uses PFM in the analog-to-digital conversion algorithm [44].

As this table shows, linear IPI has lower energy/bit compared to the other techniques. A possible reason is that IPI is asynchronous while the other signaling techniques are synchronous. Since IPI is asynchronous, extra power won't be consumed to generate the sampling clock that is required for synchronous pulse techniques. Since IPI is defined as the time between two narrow width pulses, the duty cycle is lower compared to the other pulse techniques, leading to lower power. Energy per bit has been calculated for differential IPI, which is given in the table. A more detailed explanation of

differential IPI, including the benefits of using differential IPI instead of single-ended IPI, is covered in Chapter 5 of this thesis.

3.8 AIPI Conversion Test Chip

3.8.1 Chip Design

To see how our simulated designs would work on real physical circuits, an IPI (AIPI) test chip was designed and manufactured for both V to IPI and IPI to V conversion. The test chip was fabricated using the TSMC 0.35 µm 4-metal layer process. The circuit design is similar to that mentioned earlier in this thesis, so the details are not repeated here. The only modification to the circuit design was to proportionally increase the size of the transistors from a 0.25 µm technology to a 0.35 µm technology. The Mentor Graphics IC Station was used for layout. Due to area constraints, the design used external capacitors for both V to IPI and IPI to V circuits. Both of these circuits were fabricated on the same chip as the IPI computation circuits of another student, Khaldoon Mhaidat [1].

The V to IPI and IPI to V conversion circuits were tested using linear IPI representation mode, because of the greater advantages of using the linear IPI representation.

3.8.2 Chip Test Results

The V to IPI converter was tested first. An external capacitor was used to collect charge for pulse generation. The capacitance for this external capacitor was 1 nF. In order to maintain linearity, as discussed earlier in this chapter, the input, V_{in}, was applied at the op-amp comparator threshold node. Figure 3.17 shows the plot for the real V to IPI circuit. In this case, the supply voltage, or *VDD*, is equal to 2.62 V, and the input to the op-amp current mirror, which is constant for the linear case, is equal to 1.0 V. This plot shows that

the fabricated V to IPI converter circuit works correctly, and that the results are linear as expected.

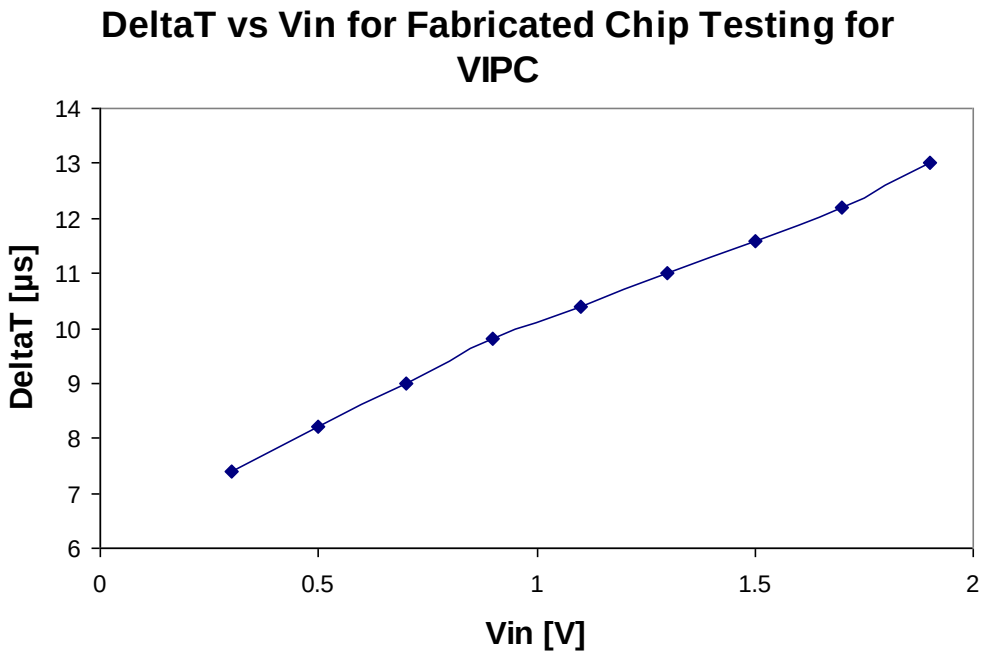

Figure 3.17: Plot of *DeltaT* vs. V_{in} for the V to IPI circuit on the fabricated chip. Here, *VDD*=2.62 V, and the op-amp current mirror input = 1.0 V.

This series of plots shown here represent the data taken from a single chip.

Testing of the V to IPI conversion was done for several chips, and there was some variation in the *DeltaT* values for a given V_{in} as a result of process variations. There are ways to compensate for process variation such as by varying the power supply voltage, by varying the bias current controlling the voltage for the op-amp current mirror that feeds current into the V to IPI converter capacitor, and by varying the bias current controlling voltage for the op-amp comparator. These externally applied voltage values need to be varied from chip to chip, so that the *DeltaT* values can be the same for a given V_{in}. A power supply variation analysis was also performed for both *VDD*=2.3 V and *VDD*=2.6 V. Input to the op-amp current mirror in this case is equal to

0.80 V. Figure 3.18 contains the plot for *DeltaT* vs. V_{in} for the on-chip V to IPI converter for the power supply variation analysis. As this plot shows, the basic operation is still linear, but the *DeltaT* for a given V_{in} varies as *VDD* varies.

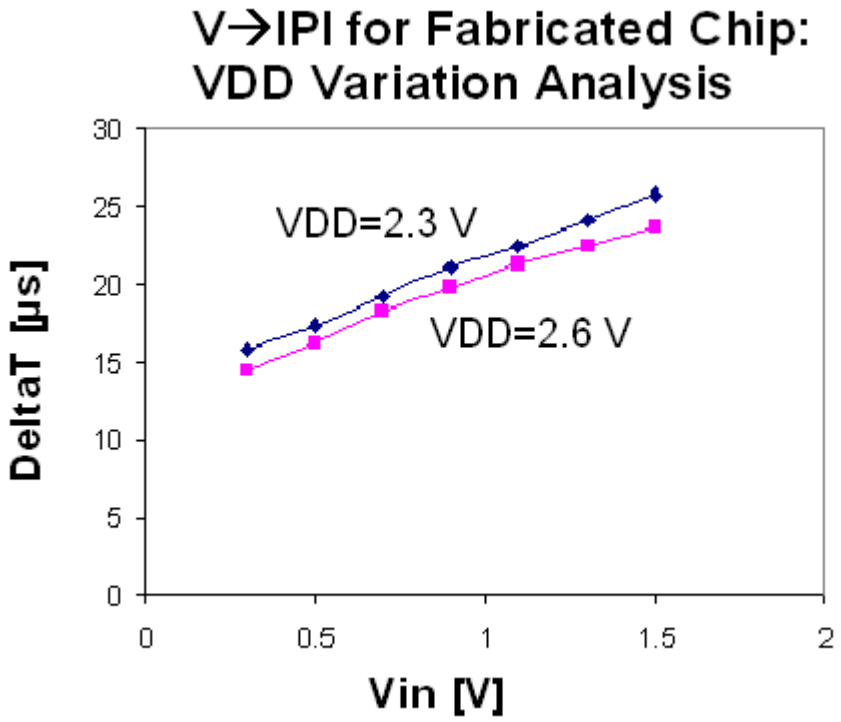

Figure 3.18: Plot of *DeltaT* vs. V_{in} for VIPC on the fabricated chip, after performing the power supply (*VDD*) variation analysis. In this case, the op-amp current mirror input = 0.80 V.

The IPI to V circuits on the various test chips were non-functional. Therefore, there are no results to be presented here for the IPI to V test chip circuit. The layout used for the V to IPI test chip circuit is shown in Figure 3.19, and for the layout used for the IPI to V test chip circuit in Figure 3.20. No capacitors are shown in the layouts since external capacitors off-chip were used. Power consumption was not measured on the fabricated chip. Only the basic functionality in terms of DeltaT was tested here.

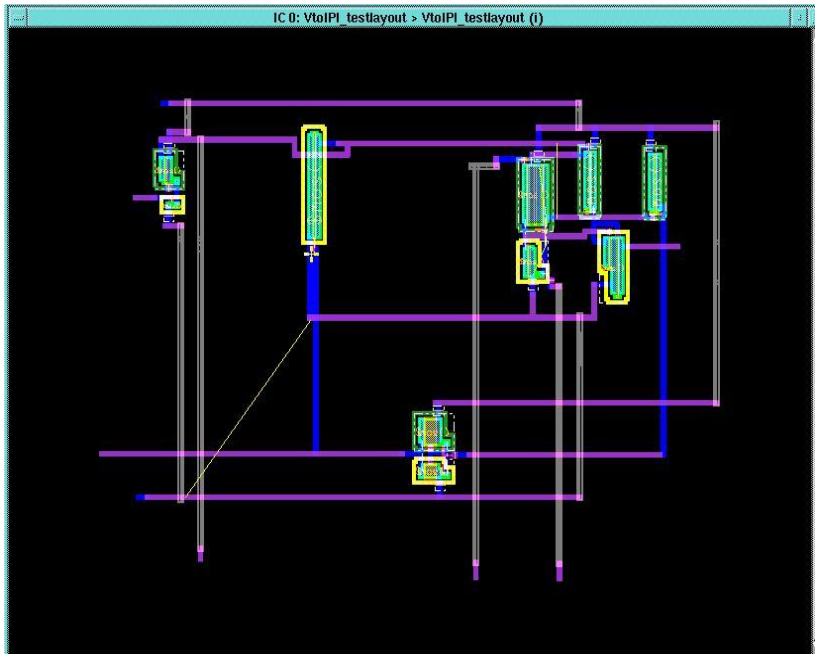

Figure 3.19: Test chip layout for the V to IPI circuit.

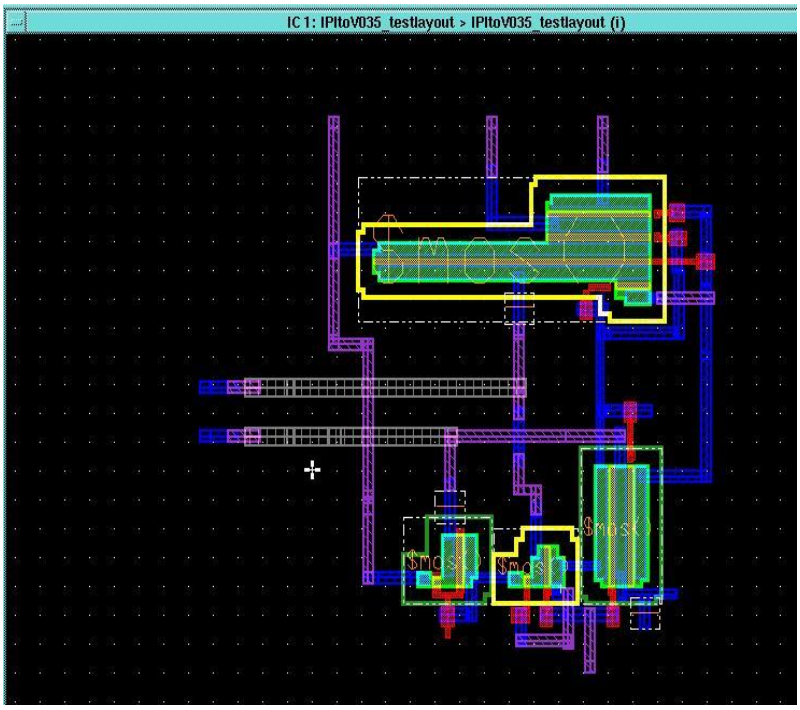

Figure 3.20: Test chip layout for IPI to V circuit.

3.9 Conclusions and Next Step

In this chapter we have demonstrated that it is possible to build power efficient IPI conversion circuits using linear IPI representation. Since one potential application for AIPI is low power wireless sensors, especially for biomedical applications where battery life is critical, and since power consumption is an extremely important factor in the use of such sensors, this makes IPI particularly attractive to that application space. The results in this chapter show that IPI is more appropriate for applications that require very low power for long battery life and where speed is not crucial. Currently in most remote sensor applications a signal conversion from analog to digital is done, and then the digital representation is transmitted via an RF wireless connection, adding significant overhead. The IPI representation provides a much cheaper alternative conversion, both in terms of circuit complexity and power, of an analog signal into a representation that can be easily transmitted. In fact, pico-radio being developed by Jan Rabaey and his group at the Berkeley Wireless Research Center, is pulse based and would be an appropriate means for transmitting IPI based signals [51], [52], [53].

The next step is to investigate the source of intrinsic distortion in the V->IPI->V conversion in MATLAB, since distortion is a very important issue in IPI conversion, which is investigated in the next chapter.

Chapter 3 Endnotes (from FOM table):

[1] IPI sampling frequency is worst-case or slowest frequency.

[2] IPI simulations are V->IPI->V simulations done in 5Spice/WinSpice.

[3] fabricated chip

[4] W. Chen, G. Dehng, J. Chen, S. Liu, “A CMOS 400 Mb/s Serial Link for AS-Memory Systems Using a PWM Scheme,” *IEEE Journal of Solid-State Circuits*, vol. 36, no. 10, October 2001, pp. 1498-1505.

[5] J. T. Stonick, Gu-Yeon Wei, J. L. Sonntag, D. K. Weinlader, “An Adaptive PAM-4 5-Gb/s Backplane Transceiver in 0.25-µm CMOS,” *IEEE Journal of Solid-State Circuits*, vol. 38, no. 3, March 2003, pp. 436-443.

[6] post-layout simulation

[7] SDM = Sigma-Delta Modulation

[8] J. Chiang, H. Chen, P. Chou, “A 2.5-V, 14-bit MASH Sigma-Delta Modulator for ADSL,” *2004 IEEE Asia-Pacific Conference on Advanced System Integrated Circuits*, August 4-5, 2004, pp. 24-27.

[9] R. del Rio et. al., “A 2.5-V CMOS Wideband Sigma-Delta Modulator,” *Instrumentation and Measurement Technology Conference, 2003*, May 20-22, 2003, pp. 224-228.

CHAPTER 4

INTRINSIC DISTORTION FOR V→IPI→V CONVERSION

4.1 Introduction

In the previous two chapters designs for a number of circuits for conversion to and from asynchronous IPI were demonstrated. These designs were done for both nonlinear and linear IPI representations, and were accompanied by the results from SPICE simulations.

To complement the experimental circuit designs that were simulated in SPICE and implemented in the chip, an analysis was done to obtain a better understanding of the intrinsic distortion from the analog voltage to/from IPI conversions. Characteristics that are important to us include bandwidth, distortion, discrete Fourier analysis, and both spatial and jitter (temporal) noise. Intrinsic distortion, which is primarily spatial, is discussed here. Extrinsic distortion (spatial and temporal) is discussed in the next chapter. As a part of this analysis the asynchronous voltage to IPI and the IPI to voltage conversions were implemented in MATLAB, which eliminated circuit specific distortion.

First, experiments were performed using MATLAB to simulate the distortion under various conditions. Then, an analysis was done to understand the source and the nature of this distortion. The theoretical analysis will be presented first, then the simulated MATLAB results that confirm the theory are presented.

The MATLAB experiments were done using the linear IPI representation. The benefits of the linear IPI representation over the nonlinear IPI representation were described in detail in Chapter 3.

4.2 Analytical Work for Source of Distortion

4.2.1 45-degree Ramp

The analysis assumes a full $V_{IN} \rightarrow IPI \rightarrow V_{OUT}$ double conversion. A simple ramp for V_{IN} was used as input. Based on a plot, as shown in Figure 4.1, the area per unit time that is above the V_{OUT} curve and below the V_{IN} curve, represents the distortion, and can be written as,

$$A_{DIST} = \frac{\left|\left(\int V_{IN}\, dt - \int V_{OUT}\, dt\right)\right|}{T_{TOT}} \qquad \textbf{Eq. 4.1}$$

In this equation, A_{DIST} is the distortion area per unit time, T_{TOT} is the time interval over which the distortion is measured, and the integrals of the V_{IN} and V_{OUT} curves mentioned above in **Eq. 4.1** represent the areas under each of those two curves respectively. Area per unit time is considered here since we need to take into account the difference in shape. Two areas can have different shapes but still be equal in area.

To illustrate the intrinsic distortion in a simplified manner, the input voltage is assumed to be a 45 degree linear ramp. This means that the area that represents the distortion will be based on the sum of multiple right-angle triangles where the horizontal and vertical legs have equal length. Figure 4.2 shows the 45-degree input ramp, the staircase output, and the triangles that represent intrinsic distortion between the input voltage and the output voltage.

The intrinsic distortion is caused by the sample-and-hold process that occurs in the V→IPI→V conversion process and the fact that the quantization intervals are changing size as the voltage ramps. Real circuits have a low-pass filtering mechanism that smoothes the output voltage curve. However, to simplify things, the low-pass filtering at the output will be ignored in this analysis, so that we deal directly with the staircase output. **Eqs. 4.2-4.4** describe the phenomena that occur based on the ramp input in Figure 4.2, where the ramp input is passed through the V→IPI converter, and then through the IPI→V converter to generate the output voltage shown in that figure.

V_{MAX} falls between two values that represent V_{MIN} multiplied by powers of 2. **Eq. 4.2** is based on this assumption. Then a 45-degree sub-triangle at the upper-right most portion of the ramp diagram (Figure 4.2) is considered. The horizontal and vertical legs of that triangle are length $V_{MAX} - 2^N V_{MIN}$. The area of this triangle is added to the areas of the other triangles each of which have horizontal and vertical legs of some power of 2. This power of 2 gets larger as we move from left to right along the time axis in Figure 4.2, therefore the area of each triangle except the upper-right most one keeps increases by a power of 2 as we move right along the time axis. The sum of areas of all these triangles is expressed in **Eq. 4.3**. This total area is then divided by the time interval, which is $V_{MAX} - V_{MIN}$ since it is a 45-degree ramp, to obtain the average area per unit time expressed in **Eq. 4.4**.

$$2^N V_{MIN} < V_{MAX} < 2^{N+1} V_{MIN} \qquad \textbf{Eq. 4.2}$$

$$A = \left(\sum_{i=0}^{N-1} \tfrac{1}{2}(2^i V_{MIN})^2\right) + \tfrac{1}{2}(V_{MAX} - 2^N V_{MIN})^2 \qquad \textbf{Eq. 4.3}$$

$$A_{AV} = \frac{(\sum_{i=0}^{N-1} \frac{1}{2}(2^i V_{MIN})^2) + \frac{1}{2}(V_{MAX} - 2^N V_{MIN})^2}{V_{MAX} - V_{MIN}} \qquad \textbf{Eq. 4.4}$$

- V_{MIN}: minimum voltage level of the input analog signal
- V_{MAX}: maximum voltage level of the input analog signal
- N: number of triangles minus 1; the triangles that represent the area between the input voltage and the output voltage. There are total of N + 1 triangles.
- A: total area or sum of the area of triangles that are between the input voltage and the output voltage signals
- A_{AV}: average per unit time of the total area of the triangles that are between the input voltage and the output voltage signals. This takes into account the differences in shape between objects that have the same area. For a 45-degree sawtooth input ramp, the time interval (x-axis) is assumed to be equal to the voltage interval (y-axis), therefore the time interval is set equal to V_{MAX} - V_{MIN}.

In order to simplify this analysis, **Eq. 4.4** is approximated as:

$$A_{AV} = \frac{\sum_{i=0}^{N-1} \frac{1}{2}(2^i V_{MIN})^2}{(2^N - 1)V_{MIN}} \qquad \textbf{Eq. 4.5}$$

Here, V_{MAX} is set to equal $2^N V_{MIN}$. After some manipulation we get:

$$A_{AV} = \frac{(\frac{1}{2}V_{MIN})\sum_{j=0}^{N-1}(4^i)}{2^N - 1} \qquad \textbf{Eq. 4.6}$$

This shows that intrinsic distortion, represented by the average total area of triangles per unit time, is caused by both the input offset and amplitude. Higher offset causes the fastest sampling rate (at bottom of waveform) to be at

a lower frequency. If the amplitude increases, there will be more triangles, causing the total area to increase non-linearly. As the input voltage increases by a factor of *M*, the sampling time interval (or time between two IPI pulses) multiplies by *M*, causing the area of the individual triangle corresponding to the sampling error between output and input voltages to multiply by M^2 . This means that there is a non-linear relationship between the difference in sampling rates between different parts of the input wave and the difference in error represented by the triangles. Therefore, there is structural non-linearity in the intrinsic distortion of the V→IPI→V conversion process. This non-linearity is due to the difference in sampling rate between the lowest and highest values of the input wave form. There is then, a trade-off between distortion and dynamic range. This non-linearity in the distortion of the intrinsic V→IPI→V conversion process means it is not so easy to compensate for this distortion. Compensation for this type of distortion is beyond the scope of this dissertation and is left as suggested future work.

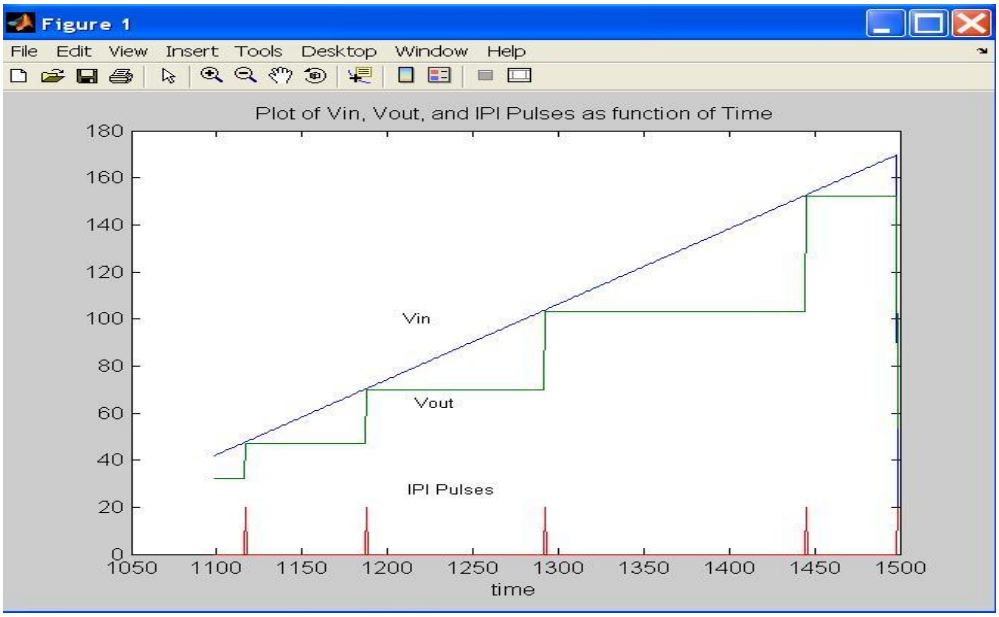

Figure 4.1: MATLAB plot of V_{IN}, V_{OUT}, and IPI Pulses as a function of Time. V_{IN} is a simple ramp or sawtooth input wave in MATLAB. V_{OUT} is a staircase where its value changes only at time points where the IPI pulses occur. This plot shows that the distortion area is the sum of the triangles that are between the V_{IN} and V_{OUT} waveforms. This simulation was done using a linear IPI representation. Simulation conditions: time per wave period (*e.tp*) = 500, amplitude (*e.ma*) = 80, offset adding variable (*e.bias*) = 10, and sampling rate increase factor (K_{SAMP}) = 1.

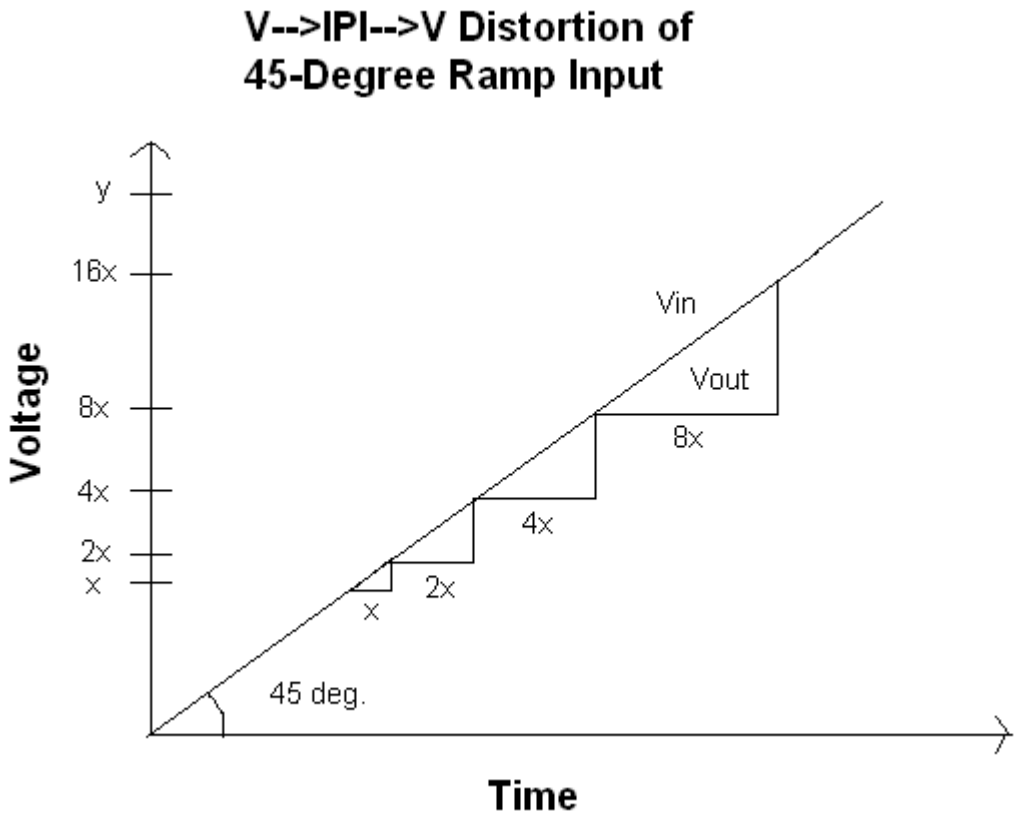

Figure 4.2: This plot shows the 45-degree input ramp voltage and the output staircase voltage for the V→IPI→V conversion. This is not from MATLAB, but just an example to illustrate how the equations were derived to demonstrate the source of intrinsic distortion. The distortion is based on total area between input and output voltage signals per unit time, which is the sum of the triangles in this figure, divided by the total time interval which equals the difference between the maximum input voltage (*y*) and the minimum input voltage (*x*). From this figure, we can infer that increasing the sampling frequency can reduce the total area of triangles, reducing distortion, or more precisely decreasing the ratio between the highest and lowest sample rates reduces the distortion.

4.2.2 Generic Ramp and Sine Wave

Here, equations for the generic ramp with arbitrary angle as well as the approximate equations for the sine wave are derived. Figure 4.3 shows the diagram for the generic ramp input. The methodology for deriving the equations is similar to that of the 45-degree ramp, but the slope will be some arbitrary value *m* instead of being set to 1 as we did for the 45-degree case. By using triangles we are using a piece-wise linear approximation which

simplifies the analysis. As a consequence, the predicted results will be slightly off from the simulated results.

Assuming the horizontal leg of the right triangle is equal to the minimum value for input voltage V_{MIN}, the length of the vertical leg will equal the horizontal leg multiplied by the slope, or $m * V_{MIN}$. Therefore, **Eq. 4.7** describes the equation for the area of that triangle, represented by A_{T1}.

$$A_{T1} = \frac{1}{2} m V_{MIN}^2 \qquad \textbf{Eq. 4.7}$$

If we consider the triangle to the immediate upper-right, the length of the horizontal leg is equal to $V_{MIN} + m * V_{MIN}$, while the vertical leg is equal to $m * (V_{MIN} + m * V_{MIN})$, or $m * V_{MIN} * (1 + m)$. Therefore, the area of this larger triangle, represented by A_{T2}, is:

$$A_{T2} = \frac{1}{2}\left(V_{MIN}(1+m)\right)\left(mV_{MIN}(1+m)\right) \qquad \textbf{Eq. 4.8}$$

Using **Eq. 4.7**, we derive an equation for A_{T2}:

$$A_{T2} = \frac{1}{2} m V_{MIN}^2 (1+m)^2 = A_{T1}(1+m)^2 \qquad \textbf{Eq. 4.9}$$

The horizontal leg which represents the time interval increases by a factor of $(1 + m)$ for each succeeding triangle to the upper-right, while the area of the succeeding triangle to the upper-right increases by a factor of $(1 + m)^2$. Therefore, the distortion can be represented by total area per unit time, which is the sum of the areas of the triangles divided by the sum of the all the horizontal legs which represent the total time interval. This average area per unit time is represented as B_T in **Eqs. 4.10-4.11**:

$$B_T = \frac{\sum_{i=0}^{N} \frac{1}{2} m V_{MIN}^2 (1+m)^{2i}}{\sum_{i=0}^{N} V_{MIN} (1+m)^i} \qquad \textbf{Eq. 4.10}$$

$$B_T = \frac{1}{2} m V_{MIN} \frac{\sum_{i=0}^{N} (1+m)^{2i}}{\sum_{i=0}^{N} (1+m)^i} \qquad \textbf{Eq. 4.11}$$

Based on the above equations, the distortion represented as total area per unit time is based on offset represented by V_{MIN} and the difference in sampling rate since the numerator increases at a much higher rate than the denominator if the number of triangles which is $N + 1$ increases.

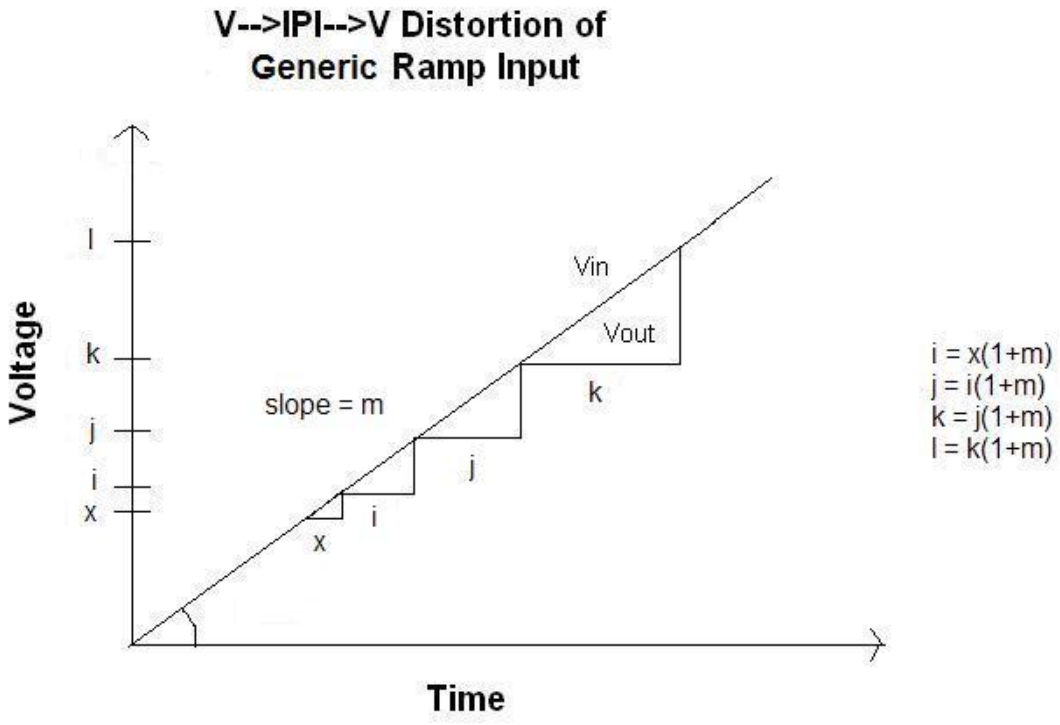

Figure 4.3: This plot shows the generic, linear input ramp voltage with arbitrary angle and the output staircase voltage for the V→IPI→V conversion. This is not from MATLAB, but just an example to illustrate how the analytical equations were derived. The distortion is based on the total area between input and output voltage signals per unit time, which is the sum of the triangles in this figure, divided by the total time interval which equals the sum of the horizontal legs of all the triangles. Here, *x* is the minimum value of input voltage, and the vertical leg of the smallest area triangle on the lower-left is equal to slope *m* multiplied by *x*. From this figure, we can infer that increasing the sampling frequency can reduce the total area of triangles, reducing distortion.

The sine wave can be approximated as a piecewise linear waveform that has several straight lines with different slopes connected to each other. Also, to simplify the analysis, only a quarter-phase of the sine wave, from 0 to π/2, is being considered here. The area per unit time for this linear approximation to the sine wave can be described by **Eq. 4.12** below:

$$B_{T,SIN} = \frac{\sum_{i=1}^{N} \left(\frac{1}{2} M_i X_i^2 \right)}{\sum_{i=1}^{N} X_i} \qquad \textbf{Eq. 4.12}$$

- $B_{T,SIN}$: average area per unit time, which is sum of areas of triangles used here divided by the total time interval. Total time interval is sum of horizontal legs of all the triangles, which is sum of all the IPI intervals that quantize or sample the input waveform.
- X_i: sampling or quantization interval between two adjacent IPI pulses; this represents the horizontal leg length of each triangle
- M_i: slope of each triangle; for this linear approximation of the sine wave, the slope varies from triangle to triangle.
- N: total number of triangles; the triangles that represent the area between the input voltage and the output voltage.

Based on the above equation, **Eq. 4.12**, the distortion represented as total area per unit time is based on the difference in sampling rate since the numerator increases at a much higher rate than the denominator if the number of triangles increases. A larger quantization interval results in a larger area per unit time, therefore a longer sampling interval between two IPI pulses results in more distortion. Figure 4.4 shows the diagram of the piecewise linear approximation of the sine wave.

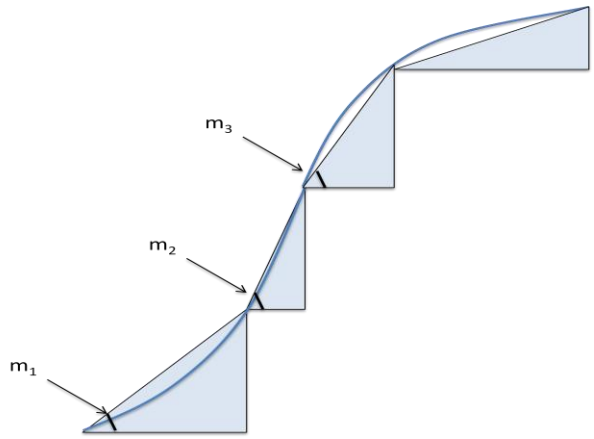

Figure 4.4: This diagram shows the piecewise linear approximation of a sine wave. The distortion is represented by the triangles between the input which is the linear approximation and the output which is the staircase. The input is approximated as multiple ramp lines each with different slopes. In this diagram, m_1, m_2, and m_3 represent the slopes of each respective ramp line.

4.3 Linear IPI Representation Simulations in MATLAB

In the previous section, an analysis of the actual source of the intrinsic distortion of the V→IPI→V conversion was presented. This section discusses the results of simulations done in MATLAB, which confirm the theory mentioned in the previous section.

4.3.1 Experiment Description for MATLAB Simulations

MATLAB simulations were done on the sine wave, using the asynchronous linear representation of IPI. The benefits of a linear representation are lower power and distortion, compared to that of the non-linear representation. Results showing this comparison are in Chapter 3 of this thesis. These MATLAB simulations were for the basic algorithmic operation of the V→IPI→V conversion process.

The analog signal input was converted into a stream of pulses. This was done by using a dummy capacitance variable, which is an ideal capacitor that behaves like the integrating capacitance in the conversion circuits. This

capacitance is initialized to zero, and through time, voltage is “added” to the capacitor. This is similar to the increase in voltage on a real capacitor as the current from the current mirror charges up the capacitor. The value of the capacitance variable in MATLAB increases until it reaches the threshold voltage, which is the input voltage value for that linear IPI representation. When the threshold voltage is reached, the pulse is formed and the capacitance variable is set to zero. This is an idealistic version of the actual circuit. This capacitor discharges instantly to zero in MATLAB, which is different from the IPI conversion circuits where the capacitor slowly discharges. Also, unlike real circuits there is no charge leakage off the capacitor.

Next, the pulses were converted back to an analog waveform using a simple sample and hold algorithm. The analog input and output waves were normalized to the range 0 to 1, which eliminated scaling concerns.

Finally, the distortion between the input and output waves was computed. The distortion in this case was the RMS difference multiplied by 10. The RMS difference is the square root of the average of the squares of the differences between the values of the normalized output wave and the normalized input wave.

Therefore, the input waveform parameters such as amplitude, etc., are described in the non-normalized form, while the distortion is relative to the normalized input and output signals.

In this case, the time per wave period, *e.tp*, is 500, and the offset, *e.bias*, is 10. The input variable *e.bias* is the minimum voltage level of the input sine wave, which needs to be positive and generally small, but non-zero. The amplitude, *e.ma*, was varied from 10 to 100. Simulations were run for both the case without increased sampling rate (the default case) and for the case where sampling rate was increased by a factor of 50. Sampling rate was increased to see how increasing sampling rate reduces distortion.

The maximum interval, i.e., the maximum time between two adjacent pulses of the IPI signal, and the minimum interval, i.e., minimum time between two adjacent pulses of the IPI signal were calculated during each simulation. This is due to the fact that the time interval between two pulses is inversely proportional to the sampling rate, and in the linear IPI representation, the upper portion of the input wave will have a lower sampling rate compared to the lower portion of the input wave.

Figure 4.5 shows a plot of the distortion as a function of amplitude and sampling rate, based on simulation results.

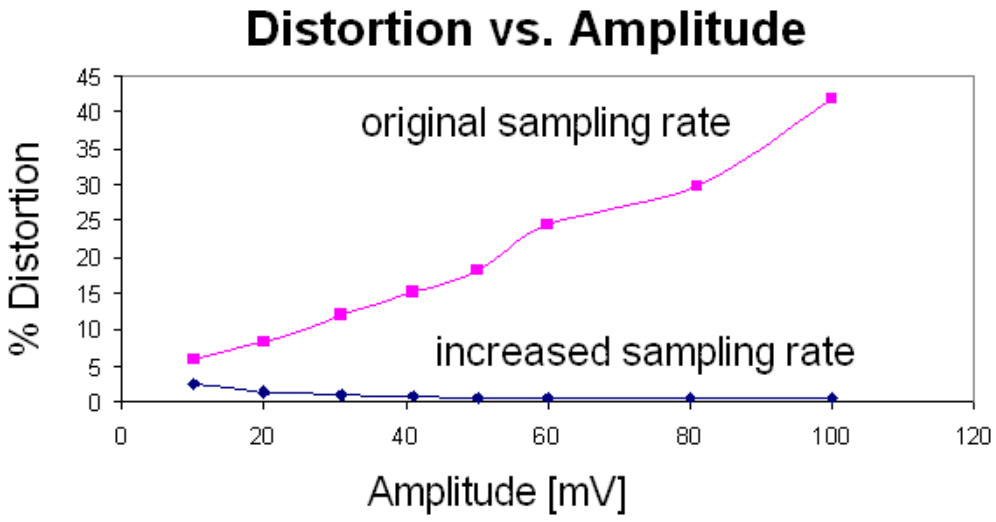

Figure 4.5: This plot shows distortion as a function of input signal amplitude, both for the case where the sampling rate was at default and for the case where the sampling rate was increased by a factor of 50. As this plot indicates, if sampling rate is increased, the distortion is much lower and does not increase with amplitude. Without increased sampling rate, the distortion increases with amplitude. Simulation conditions are: *e.tp* = 500; *e.bias* = 10. Description of these parameters can be found in main text above. Input was sine wave.

4.4 Nyquist Limit Simulations

MATLAB simulations were done where the input sine wave frequency was varied. These simulations show how the distortion of input signals above the Nyquist frequency limit compare to the distortion of input signals below the Nyquist frequency limit. Nyquist Theorem states that in order to reconstruct the original input signal from a sampled signal, this condition must be satisfied [54]:

$$f_i < \frac{f_s}{2} \qquad \textbf{Eq. 4.17}$$

Here, f_i is the input frequency of the sine wave and f_s is the sampling frequency. The issue with V→IPI→V conversion is that the sampling rate varies within the input signal based on amplitude. With higher amplitudes, the sampling rate becomes less uniform. Therefore, the sampling frequency being considered here is the worst-case, or lowest sampling frequency. This occurs at the upper portion of the input signal for the linear IPI representation. Here, only the ratio of the input wave frequency to the minimum sampling rate is relevant since this is the worst-case scenario. Ratio of the input wave frequency to the maximum sampling rate is irrelevant here. The data are normalized to the period of the signals simulated. In order to satisfy the Nyquist rule for reconstruction of the sampled signal:

$$T_i > 2 * T_s \qquad \textbf{Eq. 4.18}$$

Here, T_i is the input period of the sine wave and T_s is the sampling period, or the maximum sampling period for AIPI representation. Figure 4.6 contains the plot of distortion vs. input signal period for the case where the maximum sampling period was 5 μs. Figure 4.7 contains the plot of distortion vs. input

signal period for the case where the maximum sampling period was 1 µs. Distortion in these plots is for differential IPI, which will be explained in detail in Chapter 5 of this thesis. As both of these plots show, the distortion gets much worse at input periods less than the Nyquist period limit based on **Eq. 4.18**, or at input frequencies greater than the Nyquist frequency limit based on **Eq. 4.17**.

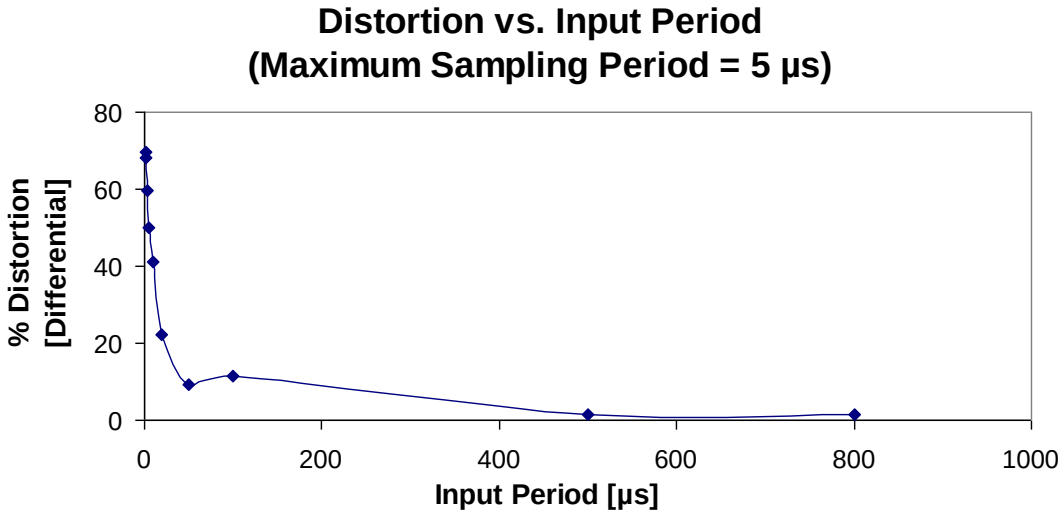

Figure 4.6: Plot of Distortion vs. Input Period, where maximum sampling period is 5 µs. V→Differential IPI→V simulations were done in MATLAB. Here, e.ma = 20 and e.bias=10. Analog input voltage is sine wave. Maximum sampling period in this plot is 5 µs. Therefore, the Nyquist limit or minimum for input period is half of the maximum sampling period, or e.tp=10 µs. As this plot shows, distortion gets significantly worse for input periods below the Nyquist limit of 10 µs.

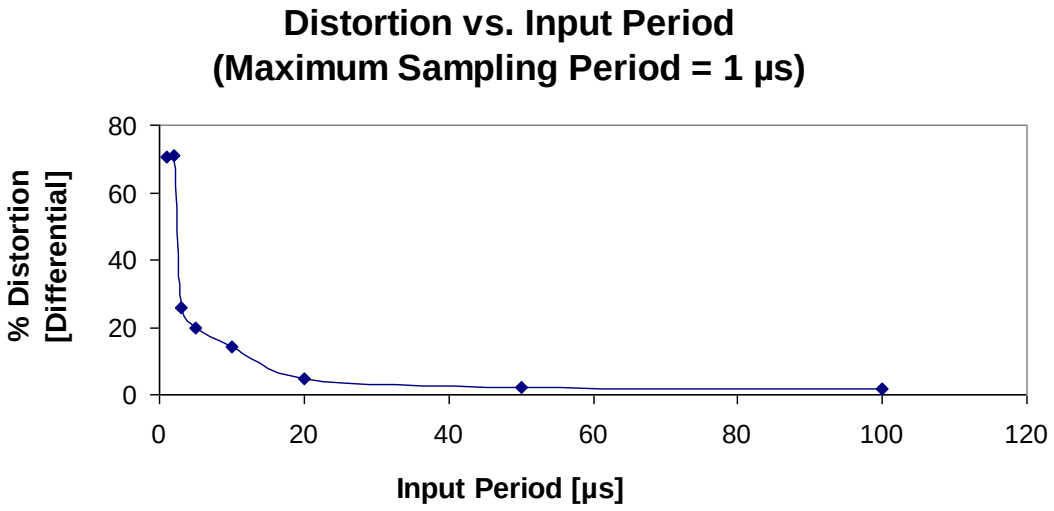

Figure 4.7: Plot of Distortion vs. Input Period, where maximum sampling period is 1 µs. V→Differential IPI→V simulations were done in MATLAB. Here, e.ma = 20 and e.bias=10. Analog input voltage is sine wave. Maximum sampling period in this plot is 1 µs. Therefore, the Nyquist limit or minimum for input period is half of the maximum sampling period, or e.tp=2 µs. In other words, the slowest sampling rate should be at least twice the input frequency to satisfy the Nyquist rule. As this plot shows, distortion gets significantly worse for input periods below the Nyquist limit of 2 µs.

4.5 Conclusions and Next Step

It was shown analytically that the linear AIPI representation has a source of intrinsic distortion. This was analyzed and simulated. It was shown that increasing the sampling rate by either reducing the input wave amplitude or adjusting the input wave offset reduces the intrinsic distortion of the V→IPI→V conversion process. In AIPI the sampling rate is proportional to the input voltage level. In the linear IPI representation, the sampling rate decreases as the input voltage level increases. This difference in sampling rates between upper and lower portion of the input wave causes the distortion to increase at higher amplitudes. The worst-case or slowest sampling frequency must be greater than twice the input frequency, in order to satisfy Nyquist's criteria.

The next chapter of this thesis focuses on extrinsic distortion, since noise added to the IPI pulse stream causes distortion in the V→IPI→V conversion process. Noise being added to the input signal won't be given importance, since the V→IPI→V converter behaves as a low-pass filter, so the high frequency noise components at the input will get filtered out in the V→IPI→V conversion process.

CHAPTER 5

EXTRINSIC DISTORTION FOR V→IPI→V CONVERSION

5.1 Introduction

In the previous chapters, the basic design of the IPI conversion circuits in PSPICE for both nonlinear and linear IPI representations, the MOSIS 0.35 µm chip fabrication and measurement, and the work on intrinsic distortion for V→IPI→V conversion in MATLAB were discussed.

In this chapter, the analysis of the IPI converters is extended to extrinsic distortion. The source of extrinsic distortion is assumed to be noise that is added to the IPI pulse stream after the signal has been converted from analog to IPI. Noise will definitely cause distortion in the asynchronous V→IPI→V converter circuit output. It was stated in the Introduction (Chapter 1) that IPI is being investigated since it has the advantage of better noise immunity. Therefore, the IPI circuits have been modified to improve the noise immunity. First, the distortion of the V→IPI→V conversion with extrinsic noise added to the IPI pulse stream was compared to the distortion of the unconverted analog signal with that same noise in order to examine the robustness of the IPI representation in the presence of noise. Next, the IPI circuits have been modified to improve the noise immunity. Differential IPI converters in MATLAB were simulated to investigate how the noise immunity of differential IPI compares with the noise immunity of single-ended IPI. In this case, the noise is added to the IPI pulse stream. Later, differential IPI conversion circuits are then designed and simulated in SPICE in order to determine the cost in terms

of extra power consumed. Finally, MATLAB simulations were done to analyze the impact of jitter noise on V→IPI→V distortion.

Since we are proposing asynchronous IPI for data conversion and transmission, this chapter discusses only AIPI. And when "IPI" is used, it will mean Asynchronous IPI.

5.2 Definition of Noise in IPI Context

IPI pulses are defined as a packet of energy which may or may not have defined edges. This has been mentioned in Chapter 1. This packet of energy is defined as an AIPI pulse if either **Eq. 5.1** or **Eq. 5.2** is satisfied as mentioned below:

$$(V_{AMP} = V_{MAX}) \text{ and } (F_{MAX} \geq F_{TH}) \qquad \textbf{Eq. 5.1}$$

$$(V_{AMP} > V_{LTH}) \text{ and } (F_{MAX} < F_{TH}) \qquad \textbf{Eq. 5.2}$$

In the above equations,

- V_{AMP} = voltage amplitude of a signal that represents a packet of energy
- V_{MAX} = maximum voltage swing for the AIPI signal
- V_{LTH} = threshold voltage level less than V_{MAX} where the packet of energy is considered a pulse if its amplitude is greater than this threshold. An AIPI pulse needs to have minimum amplitude for robustness, even if its amplitude is less than full swing due to attenuation or pulse shaping. This way, signals with amplitudes below this threshold can be filtered out as noise.
- F_{MAX} = maximum frequency that represents the frequency spectrum of this packet of energy

- F_{TH} = threshold frequency where the packet of energy is considered a pulse if the maximum frequency component in its frequency spectrum is less than this threshold

Here, V_{LTH} and F_{TH} are arbitrarily defined constants. These two conditions above (**Eq. 5.1** and **Eq. 5.2)** define an AIPI pulse as either 1) a pulse with high-frequency content at full voltage swing (to include the rectangular pulse with full voltage swing with defined edges) or 2) a pulse that has voltage swing in the range between V_{LTH} and V_{MAX} but has the high-frequency content filtered out. An AIPI pulse needs to have minimum amplitude for robustness, even if its amplitude is less than full swing due to attenuation or pulse shaping. This way, signals with amplitudes below this threshold can be filtered out as noise.

A packet of energy is defined as noise if neither of the above two equations or conditions are satisfied. Since noise usually has edges and has been defined here as having higher frequency components compared to AIPI pulses less than full voltage swing, low-pass filters can be used to filter out much of the noise. A low-pass filter causes the edges of a signal to disappear since the high-frequency components are attenuated. The energy of the pulse representing noise that travels through this filter spreads out and at some point, after its amplitude gets attenuated to below the threshold, ceases to be a pulse anymore. Such a low-pass filter is inherent to metal interconnect.

An advantage of using AIPI is that, based on the energy-packet model, AIPI pulses can be defined, i.e.. shaped, to have only low frequency components making it more immune to high-frequency attenuation when low-pass filters are used to attenuate the noise in the pulse stream. Other signaling techniques such as PWM, etc., have attenuation due to the presence of high-frequency components, since these techniques use well defined edges. AIPI can utilize this energy-packet model to improve immunity from

high-frequency attenuation in high-speed interconnects that have parasitic low-pass filters. There is more discussion of this issue in Chapter 6, where AIPI is compared with other signaling techniques.

However, the definition of pulse given here, both verbal and mathematical, is not a complete “energy packet” based pulse definition. Here, noise was defined as too steep of an edge that can be filtered using low-pass filters. Noise can also be defined as a pulse with too shallow of an edge, and what we really want is a band-pass filter. This band-pass filter can filter out both edges that are too steep and edges that too shallow, minimizing both types of noise. A complete investigation of an “energy packet” based pulse definition that takes this band-pass filter into account is beyond the scope of this work and is left for future research.

There are cases where the noise can qualify as a pulse based on the above definition, that is, where the noise has high amplitude that is equal to or close to V_{MAX}. The noise will be detected as an AIPI data pulse. Differential IPI signaling is then used to filter out that type of noise. Differential IPI representation is discussed in detail later on in this chapter, since there is a cost involved in terms of extra power consumed.

5.3 Noise Simulations of Single-Ended IPI vs. Analog Signals

Simulations were done in MATLAB that compare the extrinsic distortion of the V→IPI→V converter (between the post-converted voltage signal and the original voltage signal) in the presence of noise added to the IPI pulse stream to the distortion of the unconverted analog signal (V→V) in the presence of this same noise. The V→V distortion is the between the original clean signal and that same signal with the same noise added but without the IPI conversion. In this case, the IPI representation is single-ended.

The results for these simulations can be found in Figure 5.1. For low noise strength, IPI has less distortion compared to the original analog signal.

However, as noise strength increases or the noise level increases, IPI has more distortion than the original analog signal. This is because when the noise amplitude is below the pulse detection threshold, the noise gets filtered out at the IPI→V converter, and when the noise amplitude is above the pulse detection threshold, the IPI→V converter detects the noise as spurious pulses causing errors in the decoding. This causes the V→IPI→V distortion to get worse at higher noise levels. This is an important characteristic of IPI signal representation, it is very fault tolerant up to a particular noise threshold and then it more or less fails catastrophically when pulses and noise can no longer be distinguished. To improve this failure threshold, we have investigated differential IPI encoding, which is discussed below.

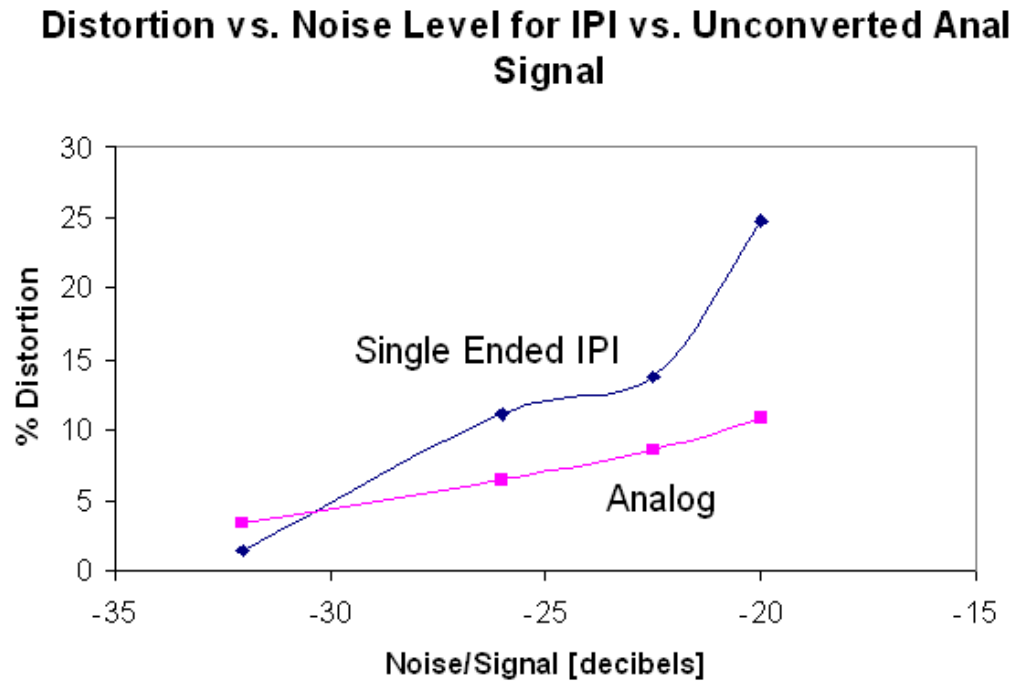

Figure 5.1: Plot of Distortion vs. Noise-to-Signal Ratio for Single-ended IPI (V→IPI→V) vs. Unconverted Analog Signal (V→V). These simulations were done in MATLAB. Noise-to-Signal ratio is ratio of noise amplitude over IPI pulse amplitude expressed in decibels. These results show that single-ended IPI representation is robust only to low noise levels but not to high noise levels where the noise-to-signal ratio is greater than approximately -30 dB.

5.4 Noise Simulations of Single-Ended and Differential IPI

5.4.1 Experiment Description

Noise was added to the IPI pulse stream to show the robustness of this IPI representation, since we have claimed that one of the advantages of IPI representation is improved noise immunity. Distortion of the voltage to IPI to voltage conversion with noise added was measured in MATLAB. This was done for both the single-ended and the differential IPI representations. In the differential IPI representation, two pulses signals are generated from the original IPI pulse signal. The first pulse is the same as the original, and the second pulse is the inverse of the first pulse, low voltage level when the first pulse is high and high voltage level when the first pulse is low. A comparator is then used to compare the two pulse signals and convert the IPI signal from differential mode back to single-ended mode. The purpose of using differential IPI signaling technique is to eliminate the common mode noise that gets added to each of the two pulse signals. Therefore, the differential IPI has better noise immunity compared to the single-ended IPI representation, if the noise is spatial and not jitter. The effectiveness of differential IPI in the presence of jitter noise will be discussed at the end of this chapter. Using a differential complement of the original signal to reduce spatial noise is one characteristic of IPI representation that causes it to have some advantage over conventional analog signaling.

The single-ended pulse signal after noise is added can be described by **Eq. 5.3**:

$$p_n = p + e_n \qquad \textbf{Eq. 5.3}$$

Here, p is the single-ended pulse signal before noise is added, or the output of the V to IPI converter, e_n is the noise that gets added to the pulse stream, and p_n is the pulse signal after noise is added, which becomes the input to the IPI to V converter.

The differential IPI representation can be described by the first pulse signal in **Eq. 5.4** and the second pulse signal (inverse of first pulse signal) in **Eq. 5.5**:

$$p_{n1} = p_1 + e_n \qquad \textbf{Eq. 5.4}$$

$$p_{n2} = p_2 + e_n \qquad \textbf{Eq. 5.5}$$

In **Eq. 5.4**, p_1 is the first pulse signal before noise is added, $\mathbf{e_n}$ is the noise that gets added to the pulse stream, and p_{n1} is the pulse signal after noise is added to p_1. In **Eq. 5.5**, p_2 is the second pulse signal before noise is added, $\mathbf{e_n}$ is the noise that gets added to the pulse stream, and p_{n2} is the pulse signal after noise is added to p_2.

The comparator that converts back from differential IPI to single-ended IPI can be described by these equations below, which are derived from **Eq. 5.4** and **Eq. 5.5**:

$$p_{n1} - p_{n2} = (p_1 + e_n) - (p_2 + e_n) \qquad \textbf{Eq. 5.6}$$

$$p_{n1} - p_{n2} = p_1 + e_n - p_2 - e_n \qquad \textbf{Eq. 5.7}$$

$$p_{n1} - p_{n2} = p_1 - p_2 \qquad \textbf{Eq. 5.8}$$

The inputs to the comparator are p_{n1} and p_{n2}. If $p_{n1} - p_{n2} > 0$, then the comparator output, which is the single-ended IPI signal, will be high. If p_{n1} –

$p_{n2} < 0$, then the comparator output will be low. **Eq. 5.8** indicates that the differential input to the comparator, $p_{n1} - p_{n2}$, is totally independent of the noise that gets added to the pulse stream, and is dependent only on the difference between the pre-noise pulse signals. Therefore, we have proved here analytically that using differential IPI representation causes the noise to cancel each other out, preventing extra distortion from being added due to noise. Using differential IPI representation, the only types of distortion that exist are due to non-linearity in the V→IPI→V converter and due to jitter noise, which will be covered at the end of this chapter. However, using the single-ended IPI representation, **Eq. 5.3** indicates that there is noise in the pulse signal that is used as input to the IPI to V converter, causing noise to make the overall distortion worse.

The noise that gets added to the pulse stream is represented by an array of normally distributed random numbers that gets multiplied by the noise-to-signal ratio. This noise-to-signal ratio is expressed in decibels. The linear representation of AIPI was used here.

5.4.2 Analysis of Results

The plot for the MATLAB noise simulations that compare single-ended IPI, differential IPI, and the unconverted analog signal with each other can be found in Figure 5.2. This is the case where noise has been added to the pulse stream. In this plot, the distortion is plotted against noise-to-signal ratio expressed in decibels. Based on this plot, the MATLAB simulated results agree with the analytical analysis done in the previous section. There is lot less distortion for the differential IPI representation than for the single-ended IPI representation. For the differential IPI case, distortion remains the same even as noise strength is increased, which means more noise being added to the IPI pulse stream. The distortion of the differential IPI signal conversion for all values of noise added is equal to the distortion of the single-ended IPI

signal conversion without any noise added to it, which means that the only distortion for the differential IPI case is due to intrinsic non-linearity, since there is no jitter added here. Distortion increases somewhat logarithmically as noise strength increases for the single-ended IPI case. Differential IPI has less distortion compared to the original analog signal in the presence of the same noise, if the noise is only spatial without any jitter.

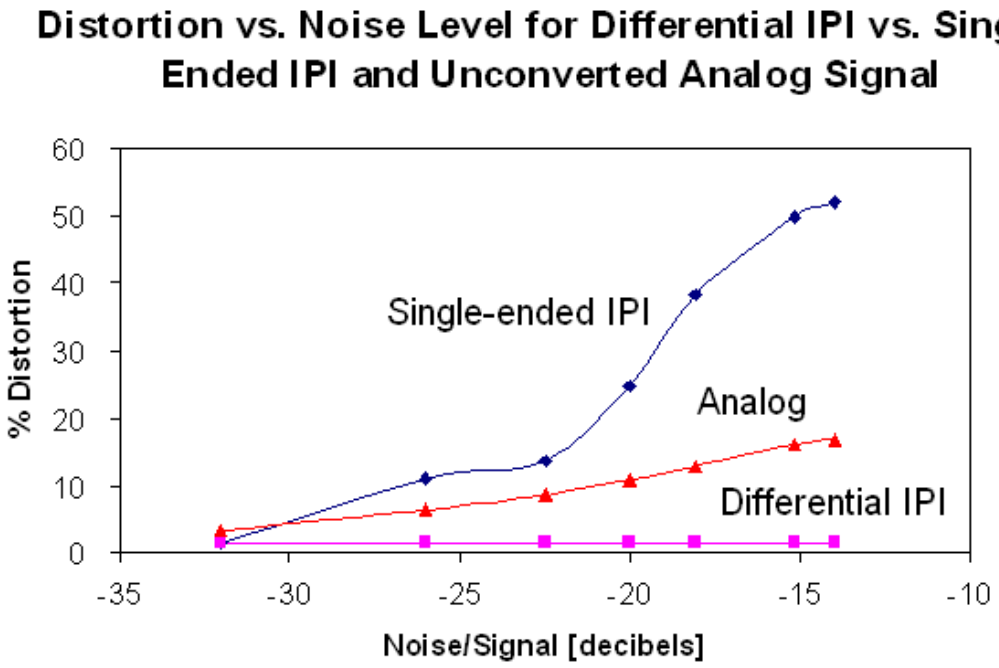

Figure 5.2: Plot of Distortion vs. Noise-to-signal Ratio for V→IPI→V simulations in MATLAB. Noise-to-Signal ratio is ratio of noise amplitude over IPI pulse amplitude expressed in decibels. Single-ended IPI, differential IPI, and the unconverted analog signal in the presence of the same noise are all shown in this plot. Simulation conditions: Sine wave, time per wave period (*e.tp*) = 500, offset adding variable (*e.bias*) = 10, amplitude (*e.ma*) = 20, sampling rate increase factor = 50. The distortion mentioned here includes both intrinsic and extrinsic distortion, therefore the differential distortion is the floor that represents intrinsic distortion. The results shown here do not take into account jitter noise.

Spatial noise added to the input signal is not much of a problem if the noise is a high frequency component, since the V→IPI→V converter behaves as a low-pass filter, filtering out the high frequency noise components. Therefore, noise that gets added to the voltage input wasn't given much attention. Noise that gets added to the IPI pulse stream was given more focus in this research work.

5.5 Differential IPI Representation: Implementation in SPICE

Both the single-ended and differential IPI converter circuits were simulated in WinSpice. The TSMC 0.25-µm models were used. Both the single-ended and differential IPI converter circuits were V→IPI→V circuits. The V→Differential IPI converter, or differential VIPC, was very similar to the VIPCs that were designed in the previous chapters. The circuit architecture is similar to that of the previously designed linear representation single-ended VIPC designed in Chapter 3, with the only difference being that an extra inverter has been added at the output of the VIPC comparator. The purpose of this extra inverter is to convert the IPI output from that comparator to its differential complement. Both of these differential complementary signals of IPI are passed on to the Differential IPI->V converter, or differential IPVC, as inputs to that circuit. The schematic for the top-level differential V→IPI→V is shown in Figure 5.3.

The differential IPVC is very similar to the linear representation single-ended IPVC that was designed in Chapter 3, with the exception that there was a comparator at the input which converts the differential IPI signal back to the single-ended version. This comparator is necessary to filter out the common mode noise that gets added to both of the pulse signals that represent the differential IPI. This single-ended IPI signal is then converted to voltage. The schematic for the differential IPVC is shown in Figure 5.4.

Differential IPI results in less distortion compared to single-ended IPI, when noise is added to the pulse stream. This is because the common-mode noise added to both signals in the differential IPI gets eliminated by the comparator. The distortion results in MATLAB have already been shown earlier in this chapter.

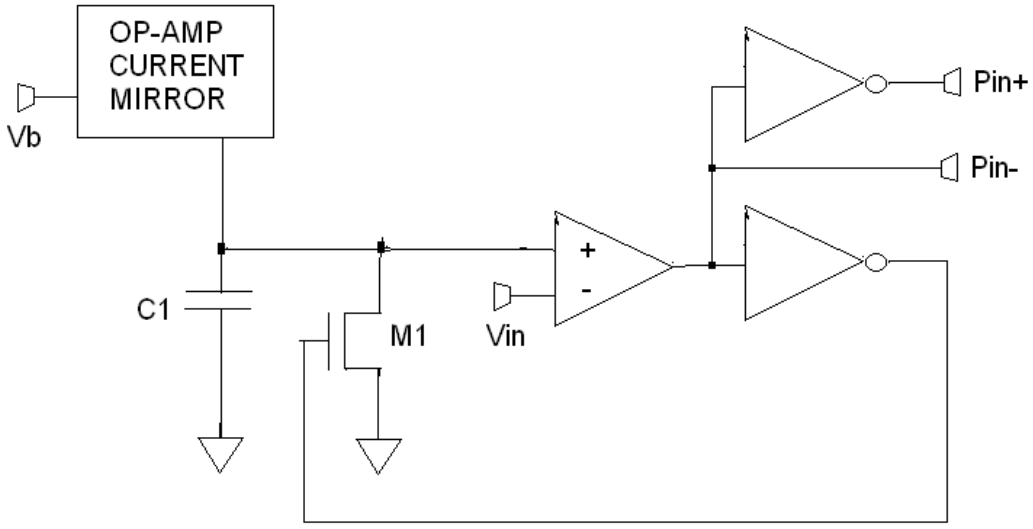

Figure 5.3: Top-level schematic for differential V→IPI→V converter circuit. This circuit converts voltage to differential IPI, and then the IPITOV_D_BLK2 sub circuit converts the differential IPI back to voltage.

Spice simulations were done for both single-ended and differential IPI to examine how much more power dissipation increases due to using differential IPI. Power dissipation increases only a little when differential IPI is used. The increase in power consumption by using differential IPI, when compared to single-ended IPI, is only 10.8%. Power dissipation results for both single-ended and differential IPI are shown in Table 5.1. This table also shows the RMS distortion between the input and output signals for both single-ended and differential IPI, when the maximum noise is applied to the pulse stream. This data shown here in this table indicates that the distortion reduces by 97% if differential IPI is used. These results demonstrate that a little extra cost in terms of power for using differential IPI results in significant distortion reduction.

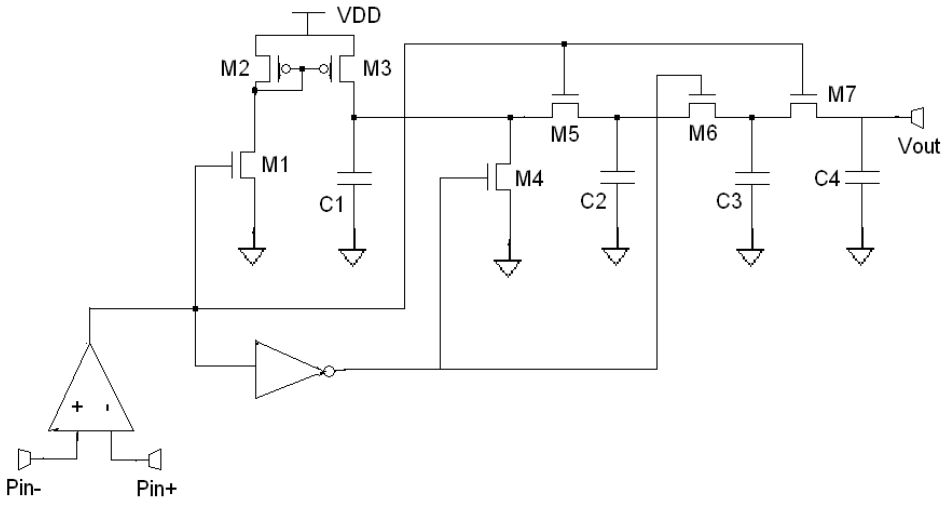

Figure 5.4: Schematic for differential IPVC, or circuit that converts from differential IPI back to voltage. This circuit has a comparator that converts from differential IPI to single-ended IPI first, before the conversion to voltage. The sub circuit box on the extreme left before the main IPVC circuitry is the comparator that converts from differential IPI to single-ended IPI.

IPI Representation Type	Power Supply Current [µA]	Power Supply Voltage (VDD) [volts]	Average Power Consumption [µW]	RMS Distortion
Single-ended	158	1.5	237	0.52
Differential	175	1.5	262.5	0.015

Table 5.1: Table showing power supply current, power supply voltage, average power consumption, and RMS distortion for both single-ended and differential IPI representations, for the V->IPI->V simulations. The power results are based on WinSpice simulations. The distortion results are based on MATLAB simulations. Here, average power consumption is equal to power supply current multiplied by power supply voltage.

RMS distortion measures the difference between the normalized input and normalized output waveforms. This distortion is calculated based on the maximum noise strength, or noise-to-signal ratio of -14 dB, being multiplied by the array of random numbers. The distortion mentioned here includes both intrinsic and extrinsic distortion, therefore the differential distortion is the floor that represents intrinsic distortion.

5.6 IPI Jitter Simulations

Simulations were done in MATLAB to measure how the V→IPI→V distortion gets affected based on IPI jitter. IPI jitter is defined as the displacement or deviation of an IPI pulse within a pulse stream along the time axis from the target value along the time axis. No additional noise was added to the IPI pulse stream. Both single-ended and differential IPI representations were used here.

The results for these jitter simulations can be found in Figure 5.5. This is the plot of Distortion vs. Pulse Displacement. This pulse displacement is for one pulse per input wave period within the IPI pulse stream. This analysis was done for both sides of the pulse displacement from the original position along the time axis, up to a maximum of 20 µs on either side of the original pulse position. For single-ended IPI, this pulse displacement was just the absolute time deviation of the IPI pulse from the original time point. For differential IPI, this pulse displacement was the relative time deviation between the two complementary IPI pulses in the differential pair. Jitter can cause a

differential pair of signals to be misaligned, therefore differential IPI simulations were done where the two complementary pulses were misaligned from each other. Jitter in cases where the two complementary differential IPI pulses are aligned together will be the same as for the single-ended IPI representation. The distortion is based on relative pulse displacement on the time axis with respect to the two time intervals both between the preceding pulse and the displaced pulse and between the displaced pulse and the succeeding pulse. These results show that distortion is much worse for differential IPI compared to single-ended IPI when jitter is present in the IPI pulse stream. In differential IPI, the complementary pulses get misaligned during jitter, causing the pulses to get eliminated when differential IPI is converted back into single-ended IPI, which leads to catastrophic errors in the IPI→V decoding process. In differential IPI, even a slight jitter can cause catastrophic errors, while in single-ended IPI, slight jitter is not a serious problem.

These noise simulations show that differential IPI has advantages over single-ended IPI only when the noise is spatial Gaussian noise, but has disadvantages over single-ended IPI when jitter noise is added.

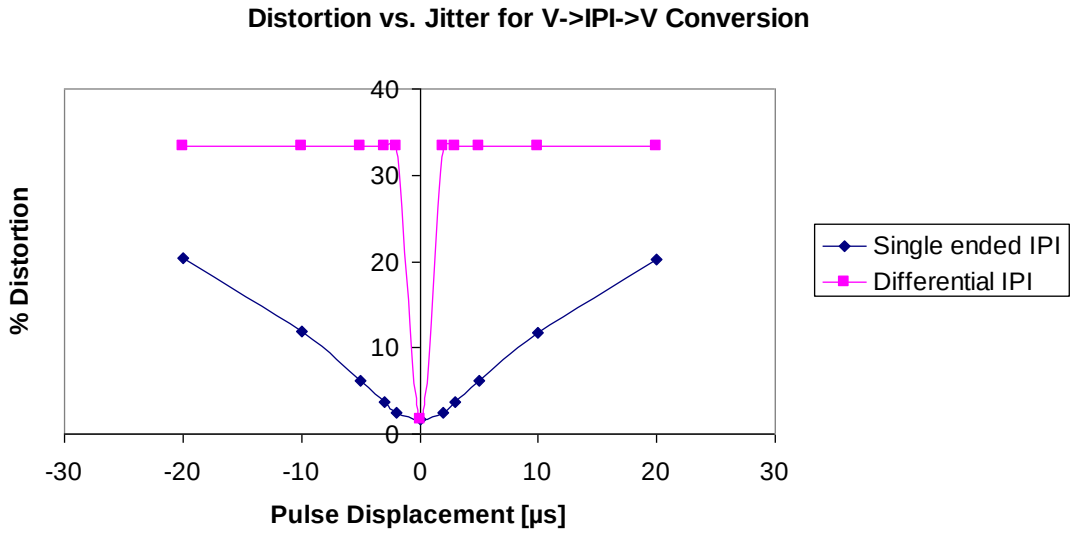

Figure 5.5: Plot of Distortion vs. IPI Pulse Jitter. This plot shows the V→IPI→V distortion in the presence of IPI pulse jitter and in the absence of other noise sources. Here, simulations were done in MATLAB. Pulse displacement is the amount of time that the IPI pulse occurrence deviates from the target time point that the IPI pulse should be occurring at. In differential IPI, this displacement is the time misalignment between two complementary pulses, where one pulse is in its correct time position and the complementary pulse is displaced. Distortion is worse for differential IPI compared to single-ended IPI. Input is a sine wave where the offset is 10 mV, the amplitude is 20 mV, and the wave period is 5 ms.

CHAPTER 6
SYSTEM-LEVEL APPLICATION OF IPI IN HIGH SPEED INTERCONNECT

6.1 Introduction

In the previous chapters, the basic design of the asynchronous IPI conversion circuits in SPICE for both nonlinear and linear IPI representations, the MOSIS 0.35 μm chip fabrication and measurement, the theoretical work in MATLAB to find the source of the intrinsic distortion for the V→IPI→V conversion, and the work on differential IPI to solve the problem of extrinsic distortion for the V→IPI→V conversion were discussed. In these chapters, we have assumed that the most likely application for using IPI was for low power analog sensor communication.

In this chapter, IPI will be taken a step further so that it is not limited to only analog sensor communication. Here, IPI will be investigated to address signal integrity issues in high-speed electronic hardware applications which is an problem in the mainstream hardware industry.

In this chapter, IPI distortion over a low-pass filter interconnect line will be compared with that of other signal representations. The system-level application is for the case where two chips communicate with each other via the I/O and off-chip PCB interconnect in a high-speed digital scenario.

6.2 RLC Low Pass Filter Distortion Simulations: Problem Definition

Off-chip interconnect is generally modeled as a transmission line for speeds in the gigahertz range. The transmission line has signal loss due to reflections, crosstalk, and high-frequency attenuation. The transmission line needs to be terminated at the receiver end to match the characteristic impedance, in order to minimize reflections. There are two sources of high-frequency attenuation in off-chip interconnect. The first is copper trace loss. The amount of this loss depends on the effective cross-sectional area and length of the copper trace. As frequency increases, the current in the metal trace gets pushed towards the surface of the conductor. This is known as the skin effect and it causes the effective cross-sectional area of the conductor to decrease, causing the effective resistance of the conductor to increase. This, in turn, leads to high-frequency attenuation of the signal going through this metal interconnect conductor. Second is dielectric loss. Attenuation of high frequency components is a serious problem in high speed digital communications over a transmission line when one chip communicates with another chip in a system. [64], [66]

Because of this attenuation of high frequency components, the off-chip interconnect or transmission line behaves as a low-pass filter when one chip communicates with another chip in a system. The transmission line can be modeled as an RLGC network. This means that resistance (*R*), capacitance (*C*), inductance (*L*), and conductance (*G*), are distributed over the interconnect wire. In this study, a second-order low-pass RLGC filter will be used as a simple of the high-frequency attenuation. Here, conductance (*G*) represents the resistor that is connected to ground and in parallel to the capacitor [73]. The reason for this is because the objective here is to compare the high-frequency attenuation for IPI with the high-frequency attenuation for other pulsed techniques while focusing only on the properties of the pulse signaling

techniques and ignoring the second-order effects of the transmission line. So it was decided that IPI distortion over a low-pass filter interconnect line would be compared with the distortion of other signal representations in comparable situations. The application is for the case where two chips communicate with each other via the I/O and off-chip PCB interconnect in a high speed digital signal integrity context.

6.3 Experimental Procedure

It is important to look at pulse signal characteristics at the algorithm level without worrying about circuit details, and it is important to analyze the extrinsic distortion due to the low-pass filter without worrying about the intrinsic distortion of the analog/digital to pulse conversion, so ideal sources for each of the pulse signal inputs were used. Here, both input and output are pulsed signals. There is no conversion to or from analog or digital. Hence, the IPI distortion initially will be for IPI→IPI only. The V→IPI→V distortion where the IPI signal gets attenuated over the interconnect low-pass filter will be covered briefly at the end of this chapter.

One paper shows that the bandwidth for an off-chip PCB trace is 4.0 GHz. This PCB trace is 5" minimum via stub. This paper is indicated as reference [55]. This value, which is 4.0 GHz, is used as the cut-off frequency for the RLC low-pass filter circuit simulations that compare high frequency IPI attenuation with other pulse high frequency attenuation. These low-pass filter circuits were simulated in SPICE.

The schematic for the low-pass filter interconnect that is modeling the off-chip transmission line is given below in Figure 6.1.

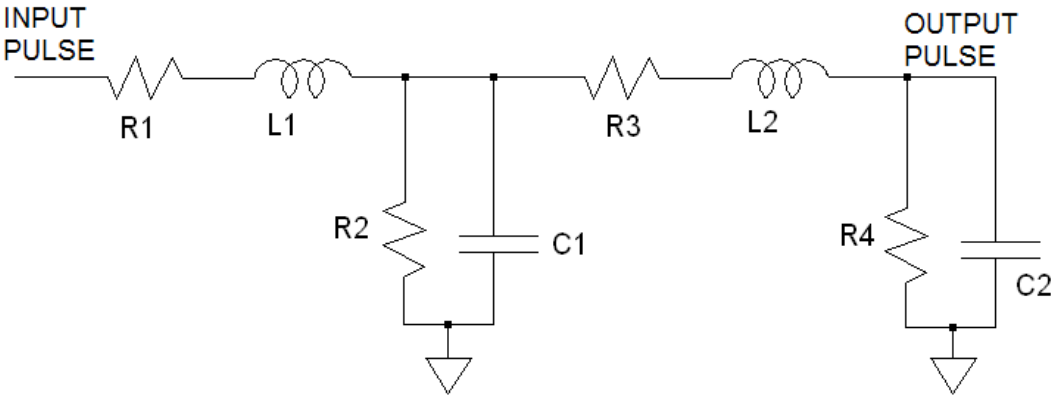

Figure 6.1: Schematic for low pass filter interconnect transmission line. This circuit was simulated in SPICE to measure distortion of IPI and other signal representations. The cut-off frequency here is 4.0 GHz.

Distortion is defined as the RMS difference between output voltage and input voltage of the V→IPI→V conversion, as described in Chapter 4. **Eq. 6.1** gives the mathematical equation for % distortion. Distortion simulations will be described briefly at the end of this chapter.

$$D = 100 * \sqrt{\text{mean}(V_{OUT} - V_{IN})^2} \qquad \textbf{Eq. 6.1}$$

- D: V→IPI→V distortion after the IPI pulse gets attenuated over the low-pass filter in the interconnect. This is expressed in the form of percent distortion.
- V_{OUT}: output of V→IPI→V after this wave gets normalized to the range 0 to 1
- V_{IN}: input of V→IPI→V after this wave gets normalized to the range 0 to 1

The equations for pulse attenuation below do not describe how attenuation was defined in the V→IPI→V MATLAB simulations at the end of this chapter. These equations define the pulse attenuation for the Pulse→Pulse simulations done in SPICE that cover most of this chapter.

Pulse attenuation is defined as mean of the absolute value of differences between the output of the low pass filter and the input of the low pass filter. The formula to compute pulse attenuation is given below in **Eq. 6.2**:

$$A = \text{mean}(|P_{OUT} - P_{IN}|) \qquad \textbf{Eq. 6.2}$$

- A: pulse attenuation
- P_{OUT}: output of low-pass filter
- P_{IN}: input of low-pass filter, which is pulsed signal from ideal source

The final figure of merit used here is normalized attenuation or attenuation over voltage swing ratio instead of just attenuation. Lower voltage swing makes the pulse more voltage-level sensitive, which means reduced noise margin. High-speed interconnects have externally applied noise added due to crosstalk. PAM-4 and LVDS have lower voltage swings compared to IPI. PAM-4 is 2-bit resolution pulse amplitude modulation. This has already been defined in Chapter 1 of this thesis. LVDS, or low-voltage differential signaling, is a form of PAM-2, or 1-bit resolution PAM with only 2 possible levels of voltage amplitude in a given pulse. One benefit of using IPI is that IPI uses the full rail-to-rail voltage swing, improving the noise margin. The formula to compute normalized attenuation is given in **Eq. 6.3**:

$$A_{NORM} = \frac{A}{V_{AMP}} \qquad \textbf{Eq. 6.3}$$

- A_{NORM}: normalized pulse attenuation

- A: pulse attenuation
- V_{AMP}: voltage amplitude or voltage swing of pulse signal

Here, IPI is being compared to 2-bit resolution pulse amplitude modulation (PAM-4), to low-voltage differential signaling (LVDS), and to 2-bit resolution pulse width modulation (PWM-4).

PAM-4, or the 2-bit resolution PAM, is commonly used in high speed serial link communication applications [56-61]. PAM-4 has 4 possible levels of voltage amplitude in a given pulse.

LVDS is a form of PAM-2, or 1-bit resolution PAM with only 2 possible levels of voltage amplitude in a given pulse. LVDS has very low voltage swing, usually at 400 mV. This low voltage swing enables low power and high speed simultaneously for off-chip data communication. LVDS is covered extensively in some of the literature [65], [67], [68].

PWM-4, or the 2-bit resolution pulse width modulation, is chosen here, since that is also used for high speed serial link communications [62]. PWM-4 has 4 possible different duty cycles in a given pulse.

Since off-chip interconnects do not have a process technology node, the following were simulated for different I/O speeds for PWM and PAM:

1) 2 Gbps:

    1 GHz I/O sampling rate for PAM-4 and PWM-4

2) 10 Gbps:

    5 GHz I/O sampling rate for PAM-4 and PWM-4

3) 20 Gbps:

    10 GHz I/O sampling rate for PAM-4 and PWM-4

For PAM-4 and PWM-4, the I/O sampling rate, or clock frequency, is based on the sampling cycle time for a set of 2 bits. Therefore for PAM the pulse width was set to 1 ns for a sampling rate of 1 GHz, since pulse width represents the time the PAM-4 voltage output level will remain constant before

the next sampling. The period is double the pulse width for the PAM-4 simulations. For PWM-4, since the duty cycle of the periodic pulse waveform is modulated, the period was set to 1/sampling rate. For example, for a 1 GHz sample rate (2 Gbps) the period is 1 ns and for a 5 GHz (10 Gbps) sample rate, the period was set to 200 ps.

For PAM-4, the voltage levels used are 0 V, 0.5 V, 1.0 V, and 1.5 V. In LVDS, which is equivalent to PAM-2, the voltage levels used are 0.55 V and 0.95 V. Full voltage swing, which is from 0 V to 1.5 V, is used for IPI and PWM. PWM-4 was simulated at 25%, 50%, and 75% duty cycles.

The IPI representation does not correlate to any clock frequency, since it is asynchronous. Therefore, the IPI resolution was set to 3, 5, 7, and 10 bits. The IPI symbol maximum period was set to be the values given in the table below. The goal is to make the comparison between IPI and PAM/PWM to be as fair as possible by keeping the data rates constant.

Data Rate	**Period @ 3-bit resolution**	**Period @ 5-bit resolution**	**Period @ 7-bit resolution**	**Period @ 10-bit resolution**
2 Gbps	3 bits/2 Gbps = 1.5 ns	5 bits/2 Gbps = 2.5 ns	7 bits/2 Gbps = 3.5 ns	10 bits/2 Gbps = 5 ns
10 Gbps	3 bits/10 Gbps = 300 ps	5 bits/10 Gbps = 500 ps	7 bits/10 Gbps = 700 ps	10 bits/10 Gbps = 1 ns
20 Gbps	3 bits/20 Gbps = 150 ps	5 bits/20 Gbps = 250 ps	7 bits/20 Gbps = 350 ps	10 bits/20 Gbps = 500 ps

Table 6.1: IPI symbol maximum period calculation, based on data rate and resolution.

The low-pass filter interconnect attenuation simulations were also done for pulse frequency modulation (PFM), where PFM is the pulse repetition rate, which is varied according to the analog input. Since PFM is similar to frequency modulation of a carrier wave, PFM is considered to be synchronously sampled using a clock. The resolution for the PFM was set to 6

bits, since dynamic range of 40 dB is approximately 6 bits [63]. **Eq. 6.4** demonstrates the relationship between dynamic range and bit resolution.

$$D_{RANGE} = 20\log_{10} 2^{N} \qquad \textbf{Eq. 6.4}$$

- N: bit resolution
- D_{RANGE}: dynamic range

The sampling clock cycle period is the bit resolution divided by the data rate. This sampling clock period is then divided by 64 (number of combinations in 6-bit resolution system), and then rounded up to the nearest multiple of 5 ps. This calculated value becomes the pulse period for the densest pulse case. Table 6.2 shows the values of the pulse periods. Simulations were done for the case where the pulse rate is 1/10 of the rate for the densest pulse case, or where the pulse period is 10 times higher, as well as for an arbitrary pulse period that is shorter than this. Table 6.2 shows the calculation of densest pulse period of PFM for each data rate.

Data Rate	**Sampling Clock Period**	**Densest Pulse Period**	**Densest Pulse Period (rounded up)**
2 Gbps	6 bits/2 Gbps = 3 ns	3 ns/64 = 46.9 ps	50 ps
10 Gbps	6 bits/10 Gbps = 600 ps	600 ps/64 = 9.4 ps	10 ps
20 Gbps	6 bits/20 Gbps = 300 ps	300 ps/64 = 4.7 ps	5 ps

Table 6.2: This table shows the calculation of densest pulse period of PFM for each data rate.

6.4 Mathematical Model for Frequency

For all the signal representations discussed in this chapter, the sampling frequency is given in **Eq. 6.5** below:

$$f_s = \frac{1}{T_s} = \frac{D_{RATE}}{R_{BIT}} \qquad \textbf{Eq. 6.5}$$

- f_s: sampling frequency
- T_s: sampling period
- D_{RATE}: data rate in terms of number of bits per second
- R_{BIT}: resolution in terms of number of bits

In the above equation, R_{BIT} is equal to 2 bits for PAM-4, 1 bit for LVDS, and 2-bits for PWM-4.

In this chapter, the transition frequency is discussed, since the high-frequency attenuation is based on pulse transition frequency. The transition period is the time from one rising edge to the next, the transition frequency then is the frequency of rising edge occurrence.

The Maximum period for IPI is mentioned here since IPI has 2^{RBIT} discrete periods or specific values for *DeltaT*, which is the time between two successive AIPI pulses. In AIPI, the period is not fixed but varies according to the modulation. In AIPI, as the period is varied, i.e. as period decreases (or sampling frequency increases) the data rate increases for a given resolution since the pulses will be closer to each other. When two IPI pulses are close to each other, the demodulation back to analog or digital happens more quickly.

For example, for 10-bit resolution and 1 ns sampling period, the data rate is 10 bits divided by 1 ns, which equals 10 Gbps. When we reduce this sampling period to 500 ps, the data rate will increase to 10 bits/500 ps = 20 Gbps. So in AIPI the data rate is also varying just as sampling frequency is.

The attenuation for maximum sampling period of AIPI was compared with that of other signal representations so that the data rates will be equal for all of them.

In PAM-4, the transition rate is half of the sampling rate or clock frequency, since the voltage level is modulated and not the timing of the edge, causing the PAM signal to remain constant for the entire clock cycle. In PAM, sampling rate equals the clock frequency since this pulse technique is synchronous.

LVDS is a form of PAM where the bit resolution is half the resolution of PAM-4. Therefore, the transition frequency is half of the sampling clock frequency. The sampling clock frequency of LVDS will be double that of PAM-4, if the data rates are the same for both LVDS and PAM-4.

PWM-4 is synchronous; therefore, the sampling rate is constant, which is equal to the clock sampling frequency. For PWM-4, sampling frequency and rising edge transition rate are the same since the modulation is along the time axis just like IPI.

The data rate and sampling frequency for PFM are constant since PFM is synchronous. In PFM, the transition rate varies within a fixed sampling interval based on changes in the input analog or digital data. A higher transition frequency means that there are more pulses within a given sampling clock period. The minimum transition period is defined as the smallest time interval between two PFM pulses, which is the case where there are the maximum number of pulses in a sampling period. The maximum transition period is defined as the largest time interval between two PFM pulses, which is the same as the sampling clock period.

Table 6.3 shows the mathematical equations for sampling frequency and transition frequency for the signal representations mentioned in this chapter.

Signal Representation	Sampling Frequency	Transition Frequency
AIPI	$f_{s,MIN} = \frac{D_{RATE}}{R_{BIT}}$	$f_{TR,MIN} = \frac{D_{RATE}}{R_{BIT}}$
PAM-4	$f_{CLK} = \frac{D_{RATE}}{2}$	$f_{TR} = \frac{f_{CLK}}{2} = \frac{D_{RATE}}{4}$
LVDS	$f_{CLK} = D_{RATE}$	$f_{TR} = \frac{f_{CLK}}{2} = \frac{D_{RATE}}{2}$
PWM-4	$f_{CLK} = \frac{D_{RATE}}{2}$	$f_{TR} = f_{CLK} = \frac{D_{RATE}}{2}$
PFM	$f_{CLK} = \frac{D_{RATE}}{R_{BIT}}$	$f_{TR,MIN} = f_{CLK}$ $f_{TR,MAX} = \left(2^{R_{BIT}}\right)\left(f_{CLK}\right)$

Table 6.3: This table shows the mathematical equations for both sampling frequency and transition frequency for each of the signal representations discussed in this chapter. The minimum frequencies are mentioned here for the IPI representation. In AIPI, the transition frequency is equal to the sampling frequency.

Where:

- $f_{s,MIN}$: minimum sampling frequency used in the AIPI representation
- f_{CLK}: sampling clock frequency for PAM-4, LVDS, PWM-4, and PFM, since they all are synchronous
- f_{TR}: transition frequency, which is defined as the frequency of rising edge occurrence of the pulses
- $f_{TR,MAX}$: maximum transition frequency
- $f_{TR,MIN}$: minimum transition frequency
- D_{RATE}: data rate in bits per second
- R_{BIT}: resolution in bits

Based on the equations in Table 6.3, PFM will have much lower transition period (or higher transition frequency) in general compared to IPI if resolution and data rate are kept constant for both of them. Even though both IPI and PFM have varying pulse or transition frequencies, the data rate will increase for IPI when the pulse frequency increases since IPI is asynchronous and can be encoded/decoded more or less instantaneously, while the data rate for PFM is constant since the encoding/decoding is based on average pulse occurrence over a synchronous time period.

6.5 Results

6.5.1 IPI vs. PAM-4 Results

Table 6.4 shows the attenuation results for IPI and PAM-4. The purpose of this table is to do a comparison of IPI attenuation vs. PAM attenuation.

For a given data rate, IPI has a lower attenuation, when compared to PAM-4, in the cases where the IPI resolution is 5 bits or greater, but for 2 Gbps, PAM-4 has a lower absolute attenuation, so IPI may not be the best candidate in this situation. The IPI representation then has more attenuation than PAM-4 when the IPI resolution is 3 bits.

Pulse Type	Attenuation	Normalized Attenuation [1/V]	Input Source Pulse Width	Input Source Period	Voltage Levels	Data Rate, Resolution
IPI	5.21e-2	3.47e-2	20 ps	1.5 ns	0 V, 1.5 V	2 Gbps, 3 bits
IPI	3.15e-2	0.021	20 ps	2.5 ns	0 V, 1.5 V	2 Gbps, 5 bits
IPI	2.25e-2	0.015	20 ps	3.5 ns	0 V, 1.5 V	2 Gbps, 7 bits
IPI	1.58e-2	1.05e-2	20 ps	5 ns	0 V, 1.5 V	2 Gbps, 10 bits
PAM-4	0.12	0.080	1 ns	2 ns	0 V, 1.5 V	2 Gbps, 2 bits
PAM-4	8.08e-2	8.08e-2	1 ns	2 ns	0 V, 1.0 V	2 Gbps, 2 bits
PAM-4	4.01e-2	8.02e-2	1 ns	2 ns	0 V, 0.5 V	2 Gbps, 2 bits
PAM-4	4.10e-2	8.20e-2	1 ns	2 ns	0.5 V, 1.0 V	2 Gbps, 2 bits
IPI	0.24	0.16	20 ps	300 ps	0 V, 1.5 V	10 Gbps, 3 bits
IPI	0.15	0.10	20 ps	500 ps	0 V, 1.5 V	10 Gbps, 5 bits
IPI	0.11	7.33e-2	20 ps	700 ps	0 V, 1.5 V	10 Gbps, 7 bits
IPI	7.76e-2	5.17e-2	20 ps	1 ns	0 V, 1.5 V	10 Gbps, 10 bits
PAM-4	0.57	0.38	200 ps	400 ps	0 V, 1.5 V	10 Gbps, 2 bits
PAM-4	0.38	0.38	200 ps	400 ps	0 V, 1.0 V	10 Gbps, 2 bits
PAM-4	0.19	0.38	200 ps	400 ps	0 V, 0.5 V	10 Gbps, 2 bits
PAM-4	0.19	0.38	200 ps	400 ps	0.5 V, 1.0 V	10 Gbps, 2 bits
IPI	0.43	0.29	20 ps	150 ps	0 V, 1.5 V	20 Gbps, 3 bits
IPI	0.28	0.19	20 ps	250 ps	0 V, 1.5 V	20 Gbps, 5 bits
IPI	0.21	0.14	20 ps	350 ps	0 V, 1.5 V	20 Gbps, 7 bits
IPI	0.15	0.10	20 ps	500 ps	0 V, 1.5 V	20 Gbps, 10 bits
PAM-4	0.94	0.63	100 ps	200 ps	0 V, 1.5 V	20 Gbps, 2 bits
PAM-4	0.62	0.62	100 ps	200 ps	0 V, 1.0 V	20 Gbps, 2 bits
PAM-4	0.31	0.62	100 ps	200 ps	0 V, 0.5 V	20 Gbps, 2 bits
PAM-4	0.31	0.62	100 ps	200 ps	0.5 V, 1.0 V	20 Gbps, 2 bits

Table 6.4: This table shows the off-chip interconnect low pass filter attenuation results for IPI and PAM-4. For the IPI case, the input source period is the maximum period.

When the IPI has a higher bit resolution, there are more bits per sampling symbol, which means that for a given data rate, IPI has lower

transition frequency. PAM-4 has a resolution of only 2 bits. This means that for a given data rate and when IPI resolution is 5 bits or greater, IPI has a lower frequency of transitions compared to PAM. This translates into lower attenuation for IPI for an IPI resolution of 5 bits or greater.

There is also some attenuation in the IPI pulse itself. But exactly how that pulse is shaped is not of concern, as long as that pulse is detectable, since IPI is time sensitive, and not voltage level sensitive like PAM. As we discussed in Chapter it is possible to define IPI pulses as energy packets where the pulse does not have a definite shape or edge. This IPI pulse shape attenuation affects only the pulse height, but not the time between two pulses.

6.5.2 IPI vs. LVDS Results

Table 6.5 shows the attenuation results for LVDS. The IPI results from the previous table are repeated here so that it can be compared more easily with LVDS.

6.5.3 IPI vs. PWM Results

Table 6.6 shows the attenuation results for IPI and PWM-4. The IPI results from the previous table are repeated here for comparison purposes.

IPI attenuation is less than PWM attenuation for all the cases in the table above, because the sampling rate for IPI is less than that for PWM, for all the IPI bit resolutions considered above.

Pulse Type	Attenuation	Normalized Attenuation [1/V]	Input Source Pulse Width	Input Source Period	Voltage Levels	Data Rate, Resolution
IPI	5.21e-2	3.47e-2	20 ps	1.5 ns	0 V, 1.5 V	2 Gbps, 3 bits
IPI	3.15e-2	0.021	20 ps	2.5 ns	0 V, 1.5 V	2 Gbps, 5 bits
IPI	2.25e-2	0.015	20 ps	3.5 ns	0 V, 1.5 V	2 Gbps, 7 bits
IPI	1.58e-2	1.05e-2	20 ps	5 ns	0 V, 1.5 V	2 Gbps, 10 bits
LVDS	6.34e-2	0.16	500 ps	1 ns	0.55 V, 0.95 V	2 Gbps, 1 bit
IPI	0.24	0.16	20 ps	300 ps	0 V, 1.5 V	10 Gbps, 3 bits
IPI	0.15	0.10	20 ps	500 ps	0 V, 1.5 V	10 Gbps, 5 bits
IPI	0.11	7.33e-2	20 ps	700 ps	0 V, 1.5 V	10 Gbps, 7 bits
IPI	7.76e-2	5.17e-2	20 ps	1 ns	0 V, 1.5 V	10 Gbps, 10 bits
LVDS	0.25	0.62	100 ps	200 ps	0.55 V, 0.95 V	10 Gbps, 1 bit
IPI	0.43	0.29	20 ps	150 ps	0 V, 1.5 V	20 Gbps, 3 bits
IPI	0.28	0.19	20 ps	250 ps	0 V, 1.5 V	20 Gbps, 5 bits
IPI	0.21	0.14	20 ps	350 ps	0 V, 1.5 V	20 Gbps, 7 bits
IPI	0.15	0.10	20 ps	500 ps	0 V, 1.5 V	20 Gbps, 10 bits
LVDS	0.22	0.56	50 ps	100 ps	0.55 V, 0.95 V	20 Gbps, 1 bit

Table 6.5: This table shows the off-chip interconnect low pass filter attenuation results for IPI and LVDS. For the IPI case, the input source period is the maximum period.

PWM has more attenuation when compared to PAM. This is due to the fact that PAM is voltage level modulated, the PAM voltage level can remain constant for the entire sampling clock cycle, while PWM is time modulated like IPI, causing PWM to have double the transition frequency when compared to PAM with a comparable data rate.

Even though PWM and IPI are both time modulated, PWM is driven by a synchronous clock that can be very fast while IPI is asynchronous, and in PWM, the duty cycle or pulse width varies with respect to the input, while in IPI the pulse period varies with respect to the input. Therefore, PWM has higher sampling frequency compared to IPI, causing more high frequency attenuation in PWM.

Pulse Type	Attenuation	Normalized Attenuation [1/V]	Input Source Pulse Width	Input Source Period	Voltage Levels	Data Rate, Resolution
IPI	5.21e-2	3.47e-2	20 ps	1.5 ns	0 V, 1.5 V	2 Gbps, 3 bits
IPI	3.15e-2	0.021	20 ps	2.5 ns	0 V, 1.5 V	2 Gbps, 5 bits
IPI	2.25e-2	0.015	20 ps	3.5 ns	0 V, 1.5 V	2 Gbps, 7 bits
IPI	1.58e-2	1.05e-2	20 ps	5 ns	0 V, 1.5 V	2 Gbps, 10 bits
PWM4	0.24	0.16	250 ps	1 ns	0 V, 1.5 V	2 Gbps, 2 bits
PWM4	0.24	0.16	500 ps	1 ns	0 V, 1.5 V	2 Gbps, 2 bits
PWM4	0.23	0.15	750 ps	1 ns	0 V, 1.5 V	2 Gbps, 2 bits
IPI	0.24	0.16	20 ps	300 ps	0 V, 1.5 V	10 Gbps, 3 bits
IPI	0.15	0.10	20 ps	500 ps	0 V, 1.5 V	10 Gbps, 5 bits
IPI	0.11	7.33e-2	20 ps	700 ps	0 V, 1.5 V	10 Gbps, 7 bits
IPI	7.76e-2	5.17e-2	20 ps	1 ns	0 V, 1.5 V	10 Gbps, 10 bits
PWM4	0.71	0.47	50 ps	200 ps	0 V, 1.5 V	10 Gbps, 2 bits
PWM4	0.94	0.63	100 ps	200 ps	0 V, 1.5 V	10 Gbps, 2 bits
PWM4	0.69	0.46	150 ps	200 ps	0 V, 1.5 V	10 Gbps, 2 bits
IPI	0.43	0.29	20 ps	150 ps	0 V, 1.5 V	20 Gbps, 3 bits
IPI	0.28	0.19	20 ps	250 ps	0 V, 1.5 V	20 Gbps, 5 bits
IPI	0.21	0.14	20 ps	350 ps	0 V, 1.5 V	20 Gbps, 7 bits
IPI	0.15	0.10	20 ps	500 ps	0 V, 1.5 V	20 Gbps, 10 bits
PWM4	0.64	0.43	25 ps	100 ps	0 V, 1.5 V	20 Gbps, 2 bits
PWM4	0.84	0.56	50 ps	100 ps	0 V, 1.5 V	20 Gbps, 2 bits
PWM4	0.61	0.41	75 ps	100 ps	0 V, 1.5 V	20 Gbps, 2 bits

Table 6.6: This table shows the off-chip interconnect low pass filter attenuation results for IPI and PWM-4. For the IPI case, the input source period is the maximum period.

6.5.4 IPI vs. PFM Results

Table 6.7 shows the attenuation results for IPI and PFM. The IPI results from the previous table are repeated here so that it can be more easily compared with PFM. The results show that the attenuation is less for IPI when

compared to PFM. PFM has more attenuation since PFM has many more pulses occurring within a given synchronous sampling period, causing it to have a higher transition frequency compared to IPI. IPI involves measuring instantaneous time between two asynchronously sampled pulses, while PFM involves measuring the average rate of pulse occurrence within a fixed synchronous sampling period, resulting in a very high transition rate for high bit rates.

Figure 6.2 shows the normalized attenuation for all the signal representations mentioned in this chapter. To summarize, IPI has less attenuation (without normalization) compared to PAM-4 if the IPI resolution is 5 bits or greater, for a given data rate or bit rate. IPI has less attenuation than PWM and PFM for a given data rate.

IPI has a lower normalized attenuation compared to PAM-4, LVDS, PWM, and PFM, if the data rate is kept constant, because IPI has a lower transition frequency compared to the other four signal representations. The mathematical formulas given earlier in this chapter demonstrate this for a given data rate. Lower transition frequency in IPI causes the high frequency attenuation to be less.

In this work, power is not being considered since the pulse signal characteristics are being investigated at the algorithm/system level and not at the circuit level. Ideal sources are being used for the pulse generation instead of conversion circuits.

AIPI has been defined based on the energy-packet model in both Chapters 1 and 5, where the pulse shape does not have to be rectangular with defined edges. An area for follow on research can be finding a way to shape these AIPI pulses so that their maximum frequency component can be less than the interconnect low-pass filter cut-off frequency value to reduce attenuation along the interconnect wire. This energy-packet model makes

AIPI unique compared to other signal representations, where the pulses always have a definite edge and shape.

Pulse Type	Attenuation	Normalized Attenuation [1/V]	Input Source Pulse Width	Input Source Period	Voltage Levels	Data Rate, Resolution
IPI	5.21e-2	3.47e-2	20 ps	1.5 ns	0 V, 1.5 V	2 Gbps, 3 bits
IPI	3.15e-2	0.021	20 ps	2.5 ns	0 V, 1.5 V	2 Gbps, 5 bits
IPI	2.25e-2	0.015	20 ps	3.5 ns	0 V, 1.5 V	2 Gbps, 7 bits
IPI	1.58e-2	1.05e-2	20 ps	5 ns	0 V, 1.5 V	2 Gbps, 10 bits
PFM	0.28	0.19	20 ps	250 ps	0 V, 1.5 V	2 Gbps, 6 bits
PFM	0.15	0.10	20 ps	500 ps	0 V, 1.5 V	2 Gbps, 6 bits
IPI	0.24	0.16	20 ps	300 ps	0 V, 1.5 V	10 Gbps, 3 bits
IPI	0.15	0.10	20 ps	500 ps	0 V, 1.5 V	10 Gbps, 5 bits
IPI	0.11	7.33e-2	20 ps	700 ps	0 V, 1.5 V	10 Gbps, 7 bits
IPI	7.76e-2	5.17e-2	20 ps	1 ns	0 V, 1.5 V	10 Gbps, 10 bits
PFM	0.69	0.46	20 ps	50 ps	0 V, 1.5 V	10 Gbps, 6 bits
PFM	0.56	0.37	20 ps	100 ps	0 V, 1.5 V	10 Gbps, 6 bits
IPI	0.43	0.29	20 ps	150 ps	0 V, 1.5 V	20 Gbps, 3 bits
IPI	0.28	0.19	20 ps	250 ps	0 V, 1.5 V	20 Gbps, 5 bits
IPI	0.21	0.14	20 ps	350 ps	0 V, 1.5 V	20 Gbps, 7 bits
IPI	0.15	0.10	20 ps	500 ps	0 V, 1.5 V	20 Gbps, 10 bits
PFM	0.71	0.47	20 ps	40 ps	0 V, 1.5 V	20 Gbps, 6 bits
PFM	0.69	0.46	20 ps	50 ps	0 V, 1.5 V	20 Gbps, 6 bits

Table 6.7: This table shows the off-chip interconnect low pass filter attenuation results for IPI and PFM. For the IPI case, the input source period is the maximum period.

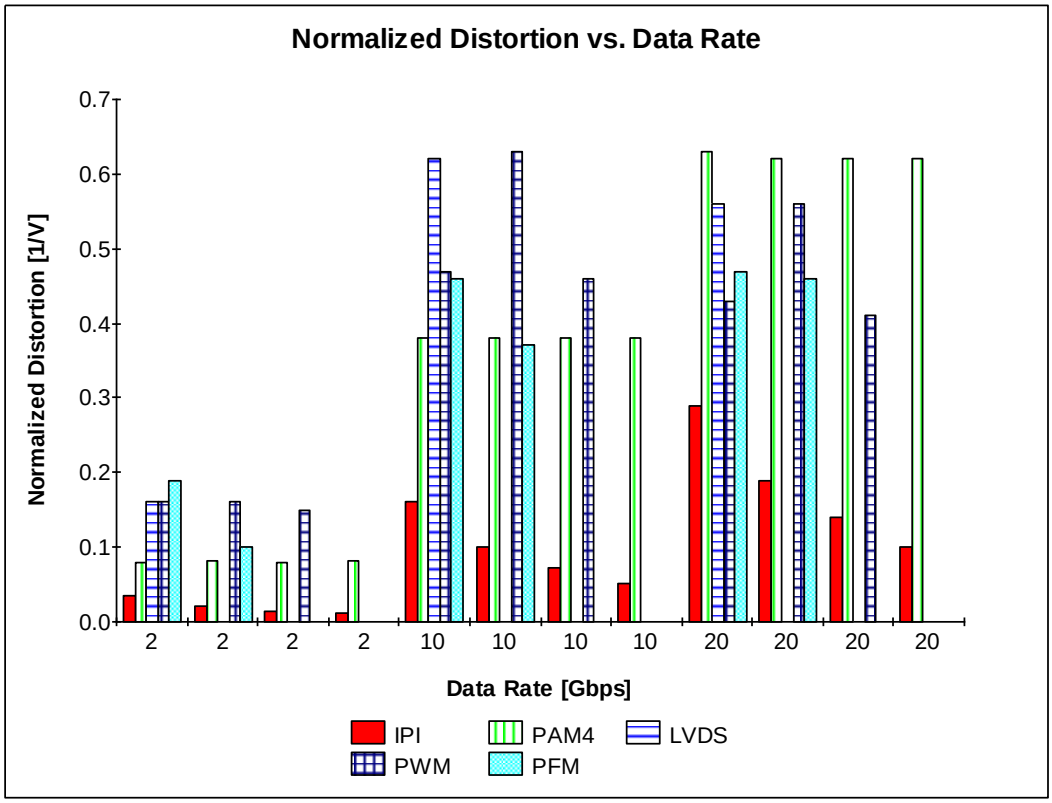

Figure 6.2: This bar graph shows the normalized attenuation for all the signal representations mentioned in this chapter. The data is grouped according to data rate, so IPI attenuation can be compared to that of the other signal representations for a given data rate. There are 4 groups of bars since IPI attenuation has been measured at 4 different resolutions. In this bar graph, normalized distortion actually represents the normalized pulse→pulse attenuation.

6.6 V→IPI→V Simulations over Low-Pass Filter Interconnect

MATLAB simulations were done for the V→IPI→V conversion where the IPI pulse stream gets attenuated over the low-pass filter. In order to simplify things here, jitter was not taken into account here. These simulations were done to analyze the impact of the IPI pulse attenuation on the decoding error, which contributes to the distortion. Only the reduction in IPI pulse amplitude was taken into account, and the reduction in rise/fall slope was ignored here since in-depth research involving the rise/fall slope is beyond the scope of this thesis. The IPI→V circuit designed in this thesis is level-sensitive, where it can detect IPI input pulses that have amplitudes above the threshold of the input inverter, but is not edge-sensitive.

The plot of results for the V→IPI→V distortion vs. % attenuation for the IPI pulse amplitude is shown in Figure 6.3. For the attenuation cases where the pulse amplitude is still above the detection threshold, the distortion is not at all affected, remaining at the floor that represents intrinsic distortion (discussed in Chapter 4). However, if the attenuation gets worse to the point that the IPI pulse amplitude is below the pulse detection threshold, the distortion gets significantly worse where the original analog signal is lost, since the IPI→V decoder won't be able to detect the pulses to do its decoding function. This work proves that the V→IPI→V conversion process is robust in environments where the IPI signal gets attenuated over the low-pass filter interconnect, as long as the attenuated pulse amplitude is high enough to be detected by the IPI→V converter.

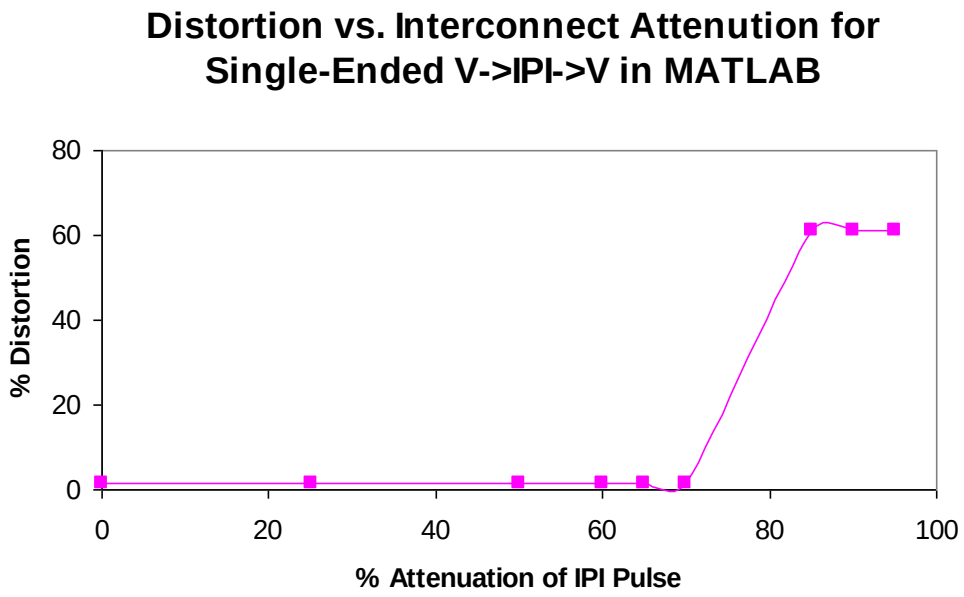

Figure 6.3: This plot shows % Distortion vs. % Attenuation of IPI Pulse. % Attenuation is the ratio of IPI pulse amplitude at low-pass filter output to IPI pulse amplitude at low-pass filter input. Distortion is measured between the output voltage and the input voltage of the V→IPI→V converter in MATLAB. Distortion in cases where the attenuated IPI pulse is above the detection threshold is at the floor that represents intrinsic distortion. Low-pass filter interconnect induced distortion occurs only when the IPI pulse is attenuated below the pulse detection level.

CHAPTER 7
CONCLUSIONS AND FUTURE WORK

7.1 Summary of Results

This section summarizes the contributions of this thesis.

7.1.1 IPI for Low Power Analog Applications

In Chapters 2 through 5 the basic designs of asynchronous IPI conversion circuits for both nonlinear and linear IPI representations were presented and circuits designed. SPICE simulation was performed and a MOSIS 0.35 µm chip was fabricated and tested. Likewise, higher level, more abstract simulations using MATLAB for both intrinsic and extrinsic distortion for V->IPI->V conversion were also presented. Most of the analysis assumed that the IPI was being used to in applications involving low power analog sensor communication.

Power and distortion simulations showed that linear IPI is generally a better choice compared to nonlinear IPI. IPI conversion circuits were optimized for low power. These circuits were simulated assuming a 0.25 µm process technology. The figure of merit, energy per bit, was the lowest for asynchronous linear IPI, when compared with existing signaling techniques. From the perspective of an energy per unit of information transferred, the IPI representation does very well, making it well suited for low power analog communication in a remote sensing environment.

It was also shown experimentally and analytically that there is intrinsic distortion in the V→IPI→V conversion process. This distortion is due to difference between the high and low sampling rates of the maximum and minimum voltages of the signals being converted. In the linear IPI representation, this sampling rate decreases as the input voltage level increases, and can become a severe problem for high amplitude or high offset analog input signals. One way to reduce this intrinsic distortion is to increase the overall sampling rate which reduces the ratio between the high and low voltages. However, this comes at a cost in greater power dissipation. MATLAB simulation results indicated that the intrinsic distortion is aggravated at input periods less than the Nyquist period limit based on **Eq. 4.18**, or at input frequencies greater than the Nyquist frequency limit based on **Eq. 4.17**. In AIPI representation, the sampling rate varies depending on the amplitude and offset of the input signal. The Nyquist rule is violated if the minimum sampling rate is below the Nyquist frequency limit.

There is also extrinsic distortion due to external noise in the V→IPI→V conversion. Both MATLAB and SPICE simulations were done to compare differential IPI with single-ended IPI. The MATLAB results show that differential IPI results in less extrinsic distortion compared to single-ended IPI, when noise is added to the pulse stream. This is because the common-mode noise added to both signals in the differential IPI gets eliminated by the comparator. As the noise-to-signal ratio gets worse, the single-ended IPI conversion has worse distortion compared to that of the unconverted V→V signal, in the presence of the same noise. The differential IPI conversion results in reduced distortion compared to the unconvered V→V signal in the presence of the same noise.

SPICE simulations were done for both single-ended and differential IPI to examine how power dissipation increases due to the use of differential IPI. The increase in power by using differential IPI, compared to single-ended IPI,

is only 10.8%, while the extrinsic distortion can be reduced up to 97%. These results demonstrate that a little extra cost in terms of power for using differential IPI results in significant extrinsic distortion reduction if the noise is from non-jitter sources. However, distortion due to jitter noise gets worse if differential IPI is used.

7.1.2 IPI for High Speed Applications

In Chapter 6, IPI was taken a step further so that it was not limited to only analog sensor communication, a very specialized niche area. IPI was investigated to address signal integrity issues in high-speed electronic hardware applications, a problem that the mainstream hardware industry is facing today.

In Chapter 6, the attenuation of IPI signals over a low-pass filter interconnect line was compared with that of other pulse based signal representations. This is important in cases where IPI is used to communicate analog information between two chips using off-chip PCB interconnect in a high-speed digital scenario.

It was important to look at pulse signal characteristics at the algorithm level without worrying about circuit details, and also it was important to analyze the extrinsic distortion due to the low-pass filter without worrying about the intrinsic distortion of the analog/digital to pulse conversion, so ideal sources for each of the pulse signal inputs were used. Here, both the input and output are pulsed signals. There is no conversion to or from analog or digital. Hence, the IPI attenuation was for IPI->IPI only.

IPI has a lower normalized attenuation compared to PAM-4, LVDS, PWM, and PFM, if the data rate is kept constant, because IPI has a lower transition frequency compared to the other four signal representations. The mathematical formulas given in Chapter 6 prove this for a given data rate.

Lower transition frequency in IPI causes the high frequency attenuation to be less.

Some V→IPI→V simulations in MATLAB were done where the IPI signal was passed through a low-pass filter interconnect. Distortion is not at all affected if the IPI pulse attenuation causes the amplitude to be above the pulse detection threshold. However, distortion gets worse where the input analog signal is lost if the attenuated IPI pulse amplitude falls below the pulse detection threshold.

7.2 Future Work

7.2.1 Information per Pulse

We believe that one of the main advantages of the asynchronous IPI representation is that each pulse carries more information than more traditional pulse based representations. However, creating a methodology for measuring such information is beyond the scope of this current work and remains an important topic for future research in this area. Interestingly there is quite a bit of work in the neuroscience community in measuring the asynchronous spike trains that neurons use to communicate with each other that may be applicable to analyzing the information characteristics of AIPI.

7.2.2 Electrocardiogram Studies

Electrocardiogram (ECG) is a signal that shows the electrical activity of the heart. ECG was chosen as an example application for asynchronous low power IPI, since when the sensor senses the ECG of a patient to be transmitted somewhere, long battery life is critical.

V->IPI->V conversions can be done on the ECG analog input signal to look at both power and distortion.

7.2.3 Broadband Pulse Based RF

Broadband Pulse Based RF such as Ultra-Wide Band (UWB) communication is useful for a wide variety of wireless sensor applications, especially for low power short distance communication such as digital health applications. Such representations have good immunity from noise and it is difficult to intercept. An UWB signal has a frequency-domain spectrum that has very low power spectral density and very wide bandwidth. [69]

IPI can be considered for UWB applications since IPI is composed of narrow pulses that occur over large time intervals, which can have wide bandwidth. In this case, the frequency spectrum (or FFT) of IPI will need to be compared to that of other pulse representations such as PWM, PAM, etc, and then those bandwidths will need to be compared with each other.

There are two versions of UWB. One is an UWB standard that uses more spread spectrum signaling, but it is more complex than just being pulse based. Then there is the generic UWB, which is the pulse based very broad spectrum communication. Pico-radio, for example, falls into that category. [refs]

7.2.4 Circuit Simulations

V->IPI->V converter circuit simulations can be done for more advanced process technologies, such as 90 nm and 45 nm technology nodes, in order to determine how competitive the IPI converter circuits are at those nodes. So far, work has been done mainly for 0.25 µm process technology node, since for ultra-low power low speed analog sensors, process scaling is not as critical compared to high speed digital circuits.

Another fruitful area for follow on research is to develop techniques where some of the dissipated power in the VIPC will be re-used to generate the pulse to be transmitted. Currently, power in the VIPC is dissipated when the discharging current flows from the capacitor through the MOS transistor to

ground. This could be used to further reduce power consumption without slowing down the performance or making the resolution worse.

Regarding the low-pass filter circuit simulations for high speed interconnect applications, V->IPI conversion can be done using the VIPC, then the IPI signal gets applied to the low-pass filter, then the filtered IPI signal goes through the IPVC, and then voltage distortion between the output and input can be measured.

It is possible to define in terms of an energy-packet model, where the pulse shape does not have to be rectangular with defined edges. VIPC circuits that generate these pulses and IPVC circuits that detect these pulses are certainly possible and could have significant value. These AIPI pulses can be shaped so that their maximum frequency component can be less than some frequency value to reduce attenuation along a wire and to improve noise immunity.

Another topic for follow on research is to design circuits where the AIPI signal generated by the VIPC is converted into a synchronous IPI (SIPI) signal. SIPI is more suitable compared to AIPI for computation. SIPI work has been done extensively by Khaldoon Mhaidat [1]. An application for this can be a case where the AIPI signal gets transmitted from one chip to another and then gets converted to SIPI for computation at the receiver chip.

REFERENCES:

[1] Khaldoon Mhaidat, Representations and Circuits for Time Based Computation, Ph.D. Dissertation, Oregon Health and Science University, March 2006.

[2] R. M. Philipp, D. Orr, V. Gruev, J. Van der Spiegel, and R. Etienne-Cummings, “Linear Current-Mode Active Pixel Sensor,” *IEEE Journal of Solid-State Circuits*, Vol. 42, No. 11, November 2007, pp. 2482-2491.

[3] E. Culurciello, R. Etienne-Cummings, and K. Boahen, “A Biomorphic Digital Image Sensor,” *IEEE Journal of Solid-State Circuits*, Vol. 38, No. 2, February 2003, pp. 281-294.

[4] E. Culurciello, R. Etienne-Cummings, and K. Boahen, “Arbitrated Address Event Representation Digital Image Sensor,” *IEEE International Solid-State Circuits Conference 2001*, February 5, 2001.

[5] Kwabena A. Boahen, “Point-to-Point Connectivity Between Neuromorphic Chips using Address-Events,” *IEEE Transactions on Circuits and Systems*, Vol. XX, No. Y, 1999, pp. 100-117.

[6] E. Culurciello, R. Etienne-Cummings, and K. Boahen, “Arbitrated address-event representation digital image sensor,” *Electronics Letters*, Vol. 37, No. 24, 22nd November 2001, pp. 1443-1445.

[7] A. F. Murray, D. Del Corso, and L. Tarassenko, “Pulse-Stream VLSI Neural Networks Mixing Analog and Digital Techniques,” IEEE Trans. Neural Networks*,* vol. *2,* no. 2, March 1991, pp. 193-204.

[8] Hamilton, A.; Murray, A.F.; Baxter, D.J.; Churcher, S.; Reekie, H.M.; Tarassenko, L.; "Integrated pulse stream neural networks: results, issues, and pointers", Neural Networks, IEEE Transactions on, Volume 3, Issue 3, May 1992 Page(s):385 - 393

[9] D. Hammerstrom, M. Jabri, and R. Etienne-Cummings, "Inter-Pulse-Interval Based Mixed Signal Representations," Research proposal submitted to the Semiconductor Research Corporation, 2001.

[10] COOKE, D., JELONEK, Z., OXFORD, A.J., and FITCH, E., ‘Pulse communication’, J. IEE, 94, Part IIIA, 1947, pp. 83-105

[11] FITCH, E., 'The spectrum of modulated pulses', J. IEE, 94, Part IIIA, 1947, pp. 556-564

[12] JELONEK, Z., 'Noise problems in pulse communication', J. IEE, 94, Part IIIA, 1947, pp. 533-545

[13] LEVY, M.M., 'Some theoretical and practical considerations of pulse modulation', J. IEE, 94, Part IIIA, 1947, pp. 565-572

[14] SCHROCKS, C.B., 'Proposal for a hub controlled cable television system using optical fiber', IEEE Trans. on cable television, vol. CATV-4, no. 2, 1979, pp. 70-77

[15] BERRY, M.C., 'Pulse width modulation for optical fibre transmission,' PhD Thesis, Nottingham University, England, 1983.

[16] BERRY, M.C., and ARNOLD, J.M., 'Pulse width modulation for optical fibre transmission of video'. IEE Int. Conf. on the Impact of VLSI Technology on Communication Systems, London, 1983

[17] Suh, S.Y., "Pulse Width Modulation for Analog Fiber-Optic Communications," *Journal of Lightwave Technology*, Vol. LT-5, No. 1, Jan. 1987, pp. 102-112.

[18] WILSON, B., and GHASSEMLOOY, Z., 'Optical pulse width modulation for electrically isolated analogue transmission', J. Phys. (E), 1985, 18, pp. 954-958

[19] Wilson, B., and Ghassemlooy, Z., "Optical PWM data link for high quality video and audio signals," *IEEE Transactions on Consumer Electronics*, Vol. 40, No. 1, Feb. 1994, pp. 55 – 63.

[20] WILSON, B., and GHASSEMLOOY, Z., 'Optical fibre transmission of multiplexed video signals using PWM, Int. J. Optoelectron., 1989, 4, pp. 3-17

[21] HEATLEY, D.J.T., 'Video transmission in optical fibre local networks using pulse time modulation'. ECOC 83 - 9th European Conference on Optical Communication, Geneva, September 1983, pp. 343-346

[22] OKAZAKI, A., 'Still picture transmission by pulse interval modulation', IEEE Trans., 1979, CATV4 pp. 17-22

[23] HEATLEY, D.J.T., 'Unrepeatered Video Transmission Using Pulse Frequency Modulation Over 100 km of Monomode Optical Fibre', *Electronics Letters*, Vol. 18, No. 9, April 29, 1982, pp. 369-371.

[24] HEATLEY, D.J.T., and HODGKINSON, T.G., 'Video Transmission Over Cabled Monomode Fibre at 1.523 μm Using PFM with 2-PSK Heterodyne Detection', *Electronics Letters*, Vol. 20, No. 3, February 2, 1984, pp. 110-112.

[25] HEKER, S.F., HERSKOWITZ, G.J., GREBEL, H., and WICHANSKY, H., 'Video Transmission in Optical Fiber Communication Systems Using Pulse Frequency Modulation', *IEEE Transactions on Communications*, Vol. 36, No. 2, Feb. 1988, pp. 191-194.

[26] KANADA, T., HAKODA, K., and YONEDA, E., 'SNR Fluctuation and Non-linear Distortion in PFM Optical NTSC Video Transmission Systems', *IEEE Transactions on Communications*, Vol. COM-30, No. 8, August 1982, pp. 1868-1875.

[27] OKAZAKI, A., "Pulse-Interval Modulation Applicable to Narrowband Transmission," *IEEE Transactions on Cable Television*, Vol. CATV-3, No. 4, October 1978, pp. 155-164.

[28] SATO, M., MURATA, M., and NAMEKAWA, T., "Pulse Interval and Width Modulation for Video Transmission," *IEEE Transactions on Cable Television*, Vol. CATV-3, No. 4, October 1978, pp. 165-173.

[29] Wilson, B., Ghassemlooy, Z., and Cheung, J.C.S., "Optical pulse interval and width modulation for analogue fibre communications," IEE Proceedings Journal, Vol. 139, No. 6, Dec. 1992, pp. 376-382.

[30] WILSON, B., GHASSEMLOOY, Z., and CHEUNG, J.C.S., 'Spectral predictions for pulse interval and width modulation', Electron.Lett., 1991, 27, (7), pp. 580-581

[31] Wilson, B. and Ghassemlooy, Z., "Pulse time modulation techniques for analogue optical fibre transmission," IEE Colloquium on Analogue Optical Communications, 18 Dec 1989, pp. 7/1-7/4

[32] Wilson, B. and Ghassemlooy, Z., "Pulse time modulation techniques for optical communications: a review," IEE Proceedings Journal, Vol. 140, no. 6, Dec. 1993, pp. 346-357

[33] Lu, C., Wilson, B., and Ghassemlooy, Z., "Pulse time modulation techniques for low cost analog signal transmission systems," Proceedings of IEEE Singapore International Conference on Networks, vol. 2, 6-11 Sept. 1993, pp. 635-638

[34] Cowen, S., "Fiber Optic Video Transmission System Employing Pulse Frequency Modulation," OCEANS, vol. 11, Sep 1979, pp. 253-259

[35] DAS, J., and SHARMA, P.D.: 'Pulse interval modulation', Electron. Lett., 1967, 3, pp. 288-289.

[36] Tang, R.; Kim, Y.; Choi, M.; Lombardi, F.; "Jitter Analysis of PWM Scheme in High Speed Serial Link," *IMTC 2006-Instrumentation and Measurement Technology Conference*, 24-27 April 2006, pp. 494-497.

[37] Tang, R.; Kim, Y.; "A Novel 8-Phase PLL Design for PWM Scheme in High Speed I/O Circuits," [date, vol, journal title ???] pp. 119-122.

[38] Hanumolu, P.K.; Casper, B.; Mooney, R.; Wei, G.; Moon, U.; "Analysis of PLL Clock Jitter in High-Speed Serial Links," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 50, No. 11, Nov. 2003, pp. 879 – 886.

[39] Lee, D.C.; "Analysis of Jitter in Phase-Locked Loops," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 49, No. 11, Nov. 2002, pp. 704 – 711.

[40] Gierkink, S.L.J.; Klumperink, E.A.M.; van der Wel, A.P.; Hoogzaad, G.; van Tuijl, E.A.J.M.; Nauta, B.; "Intrinsic 1/f Device Noise Reduction and Its Effect on Phase Noise in CMOS Ring Oscillators," *IEEE Journal of Solid-State Circuits*, Vol. 34, No. 7, July 1999, pp. 1022 – 1025.

[41] Abrahamsen, J.P.; Hafliger, P.; Lande, T.S.; "A Time Domain Winner-take-all Network of Integrate-and-fire Neurons," *Proceedings of the 2004 International Symposium on Circuits and Systems, 2004, ISCAS '04*, Volume 5, 23-26 May 2004, pp. V-361 - V-364.

[42] Standage, D.I.; Trappenberg, T.P.; "Differences in the subthreshold dynamics of leaky integrate-and-fire and Hodgkin-Huxley neuron models," *IEEE International Joint Conference on Neural Networks, 2005*, Vol. 1, July 31 - August 4, 2005, pp. 396 – 399.

[43] Smith, L.S.; Fraser, D.S.; **"**Sound Feature Detection Using Leaky Integrate-and-fire Neurons," *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004*, *ICASSP 2004,* Vol. 1, 17-21 May 2004, pp. I - 617-620.

[44] http://en.wikipedia.org/wiki/Pulse-density_modulation

[45] http://www.phptr.com/articles/article.asp?p=433381&seqNum=7&rl=1

[46] Silicon Laboratories datasheet webpage: http://www.silabs.com/public/documents/tpub_doc/dsheet/Microcontrollers/Precision_Mixed-Signal/en/C8051F12x-13x.pdf

[47] TI ADC webpage: http://focus.ti.com/docs/prod/folders/print/ads1601.html

[48] TI ADC webpage: http://focus.ti.com/docs/prod/folders/print/ads1100.html

[49] TI ADC webpage: http://focus.ti.com/docs/prod/folders/print/ads7887.html

[50] Maxim/Dallas Semiconductor ADC webpage: http://para.maxim-ic.com/cache/en/results/4942.html

[51] J. Rabaey, J. Ammer, T. Karalar, S. Li, B. Otis, M. Sheets, T. Tuan, "PicoRadios for Wireless Sensor Networks: The Next Challenge in Ultra-Low-Power Design," *Proceedings of the International Solid-State Circuits Conference*, San Francisco, CA, February 3-7, 2002.

[52] J.L. da Silva Jr., J. Shamberger, M.J. Ammer, C. Guo, S. Li, R. Shah, T. Tuan, M. Sheets, J.M. Rabaey, B. Nikolić, A. Sangiovanni-Vincentelli, P. Wright, "Design Methodology for PicoRadio Networks," *DATE 2001*.

[53] Jan M. Rabaey, "PicoRadio: Ultra-Low Energy Wireless Sensor and Monitor Networks," Invited Presentation, CBS-ETAPS Workshop, Berlin - April, 2000.

[54] Walt Kester and James Bryant, “Basics of ADCs and DACs, Part 1,” *DSP DesignLine*, July 19, 2007, www.dspdesignline.com.

[55] A. Emami-Neyestanak, et. al., "A Low Power Receiver with Switched-Capacitor Summation DFE," 2006 Symposium on VLSI Circuits, Digest of Technical Papers.

[56] R. Farjad-Rad, C. K. Yang, M. A. Horowitz, and T. H. Lee, “A 0.4-µm CMOS 10-Gb/s 4-PAM Pre-Emphasis Serial Link Transmitter,” *IEEE Journal of Solid-State Circuits*, vol. 34, no. 5, May 1999, pp. 580-585.

[57] R. Farjad-Rad, C. K. Yang, M. A. Horowitz, and T. H. Lee, "A 0.3-µm CMOS 8-Gb/s 4-PAM Serial Link Transceiver," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 5, May 2000, pp.757-764.

[58] J. T. Stonick, Gu-Yeon Wei, J. L. Sonntag, D. K. Weinlader, “An Adaptive PAM-4 5-Gb/s Backplane Transceiver in 0.25-µm CMOS,” *IEEE Journal of Solid-State Circuits*, vol. 38, no. 3, March 2003, pp. 436-443.

[59] J. L. Zerbe, C. W. Werner, V. Stojanovic, F. Chen, J. Wei, G. Tsang, D. Kim, W. F. Stonecypher, A. Ho, T. P. Thrush, R. T. Kollipara, M. A. Horowitz, K. S. Donnelly, “Equalization and Clock Recovery for a 2.5-10-Gb/s 2-PAM/4-PAM Backplane Transceiver Cell,” *IEEE Journal of Solid-State Circuits*, vol. 38, no. 12, December 2003, pp. 2121-2130.

[60] C. Menolfi et. al., “A 25 Gb/s PAM4 Transmitter in 90nm CMOS SOI,” *2005 IEEE International Solid-State Circuits Conference*, February 7, 2005, pp. 72-73.

[61] T. Toifl et. al., “A 22 Gb/s PAM-4 Receiver in 90-nm CMOS SOI Technology,” *IEEE Journal of Solid-State Circuits*, vol. 41, no. 4, April 2006, pp. 954-965.

[62] W. Chen, G. Dehng, J. Chen, S. Liu, “A CMOS 400 Mb/s Serial Link for AS-Memory Systems Using a PWM Scheme,” IEEE Journal of Solid-State Circuits, vol. 36, no. 10, October 2001, pp. 1498-1505.

[63] K. Kagawa, N. Yoshida, T. Furumiya, J. Ohta, M. Nunoshita, “Application of pulse frequency modulation photosensors to subretinal artificial retina implantation,” Proc. SPIE, vol. 4596, October 2001, pp. 314-319.

[64] D. Miller, *Designing High-Speed Interconnect Circuits: Advanced Signal Integrity Methods for Engineers*, Intel Press, 2004.

[65] A. Boni, A. Pierazzi, and D. Vecchi, “LVDS I/O Interface for Gb/s-per-Pin Operation in 0.35-µm CMOS,” *IEEE Journal of Solid-State Circuits*, Vol. 36, No. 4, April 2001, pp. 706-711.

[66] H. Blennemann and R. Fabian W. Pease, “High Aspect Ratio Lines as Low Distortion, High Frequency Off-Chip Interconnects,” *IEEE Transactions on Components, Hybrids, and Manufacturing Technology,* Vol. 16, No. 7, November 1993, pp. 692-698.

[67] Sua Kim, Bae-Sun Kong, Chil-gi Lee, Jin-Hyun Kim, Young-Hyun Jun, Changhyun Kim, **“**A 6-Gbps/pin 4.2mW/pin Half-Deuplex Pseudo-LVDS Transceiver,” *Solid-State Circuits Conference, 2006. ESSCIRC 2006. Proceedings of the 32nd European*, Sept. 2006, pp. 484–487.

[68] M. Chen, J. Silva-Martinez, M. Nix, and M. Robinson, “Low-Voltage Low-Power LVDS Drivers,” *IEEE Journal of Solid-State Circuits*, Vol. 40, No. 2, February 2005, pp. 472-479.

[69] Nathaniel August, “UWB Applications and Technologies,” Presentation for PersonalTelco Project, VTVT Lab, Virginia Tech, March 31, 2004.

[70] D. Hammerstrom, M. Jabri, and R. Etienne-Cummings, "Inter-Pulse-Interval Based Mixed Signal Representations," Research proposal submitted to the Semiconductor Research Corporation, 2001.

[71] Emeritaatsviering Prof. H. De Man, Jan M. Rabaey, “Ubiquitous Sensor Networks and Ambient Intelligence – Technology Focusing on Society’s Woes,” Invited Presentation, Academic Colloquium, Leuven, September 2005.

[72] Y.H. Chee, A.M. Niknejad, J. Rabaey, “A 46% Efficient 0.8 dBm Transmitter for Wireless Sensor Networks,” *2006 Symposium on VLSI Circuits Digest of Technical Papers*, June 15-17, 2006, pp. 43-44.

[73] http://www.altera.com/technology/signal/fundamentals/basics/sgl-basics.html

BIOGRAPHICAL SKETCH

I did my undergraduate education at Carnegie Mellon University, and graduated with a Bachelor of Science in Electrical and Computer Engineering in May 1996. I did my Master of Science in Electrical and Computer Engineering at University of Texas at Austin. I graduated with that degree in December 1999. My area of specialization during my master's degree program was in semiconductor device physics and processing. I did some part-time jobs as an intern at both SEMATECH and AMD during that time.

I started my PhD program of study in September 2001. My area of interest is analog and mixed-signal design, which includes low power design and signal integrity. Some of my internship experiences were at Intel which involved analog design and validation. These internships were from April to October 2005 and from July to December 2006.