EFFICIENT ON-CHIP POWER SUPPLY SAMPLING TO IMPROVE POST-SILICON DEBUG

A Thesis

by

LUKE MURRAY

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Sunil P. Khatri
Committee Members,	Jiang Hu
	Duncan M. Walker
Head of Department,	Chanan Singh

December 2014

Major Subject: Computer Engineering

ABSTRACT

In recent years, post-silicon debugging has become a significantly difficult exercise due to the increase in the size of the electrical state of the IC being debugged, coupled with the limited fraction of this state that is visible to the debug engineer. As the number of transistors increases, the number of possible electrical states increases exponentially, while the amount of information that can be accessed grows at a much slower rate. This difficulty is compounded by the outsourcing of IP blocks, which creates more black boxes that the debug engineer must work around. As a result, when an IC fails, tracking down the cause of the failure becomes a monumental task, and debugging becomes more art than science. One source of errors in a test circuit is the fluctuation of the power supplies during a single clock cycle. These supply variations can increase or decrease the speed of a circuit and lead to errors such as hold time violations and setup time violations. This thesis presents a circuit that precisely samples the power supply multiple times in a clock cycle, allowing the debug engineer to quantify the variations in the supply over a clock cycle. With this information, a better understanding of the electrical state of the test chip is made possible. The circuit presented in this thesis can sample the supply voltage with a quantization of 0.291mV, and the output is linear with an R^2 value of 0.9987.

ACKNOWLEDGMENTS

I'd like to thank Dr. Khatri for all of his support, guidance, and help in this process. I have learned a lot from him in these semesters and am indebted to him for all the work he has done. I'd also like to thank Dr. Hu and Dr. Walker for being on my committee and supporting me through this process. I also owe thanks to Monther Abusultan for helping me learn the ropes with HSPICE and resetting my password when I locked myself out.

I'd like to thank Dr. Narayanan for his advice and mentoring as I made the transition from an undergraduate to a graduate student. Also, I'd like to thank Ms. Carda for answering the many questions I had as I figured out how graduate school worked.

I'd like to thank my family for their support during this process. From when they helped me move in to when they help me move out (thanks in advance!) they have been beside me the whole time. I appreciate all that you do for me, Mom, Dad, Shannon, & Eva!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Post-silicon debug is complicated due to the presence of billions of on-chip nets and devices, resulting in a very large electrical state to reason about. The number of pins through which this state can be accessed is limited. Several electrical effects such as cross-talk, processing variation, temperature, and power supply variations make the task of post-silicon debug even more difficult. The task of debugging and error is naturally limited by the number of pins on the IC. Although techniques such as test pattern generation and boundary scans allow for the examination of the digital state of a chip, many of the failure modes of this IC are analog in nature. Power supply integrity is significant problem in modern VLSI designs, and power supply variations are minimized via CAD techniques during the design period. Knowing the fluctuations of the on-chip power supply over any clock cycle is important to understanding how the power supply is effecting the timing of circuits, and if these changes in timing are responsible for the error being debugged. Fore example, errors like setup time or hold time violations may be caused by power supply variations and may not be debuggable by digital techniques (such as scan testing) alone. This thesis presents a method to reconstruct the power supply waveform over a clock cycle.

In our scheme, an IC would have several Supply Testing Units (STUs), each connected to a set of test points on the chip, as shown in Figure I.1. In this figure three STUs (STU1, STU2 and STU3) are connected to eight test points (n1 through n8). The test points would be located at points in the power grid that are likely to have problems with droops or spikes; design-time CAD simulations and analysis could be used to identify these potentially problematic locations. Note that these test points may be located at sites that are not easily accessed during the post-silicon debug, such as the power pins of crit-

ical logic that are on lower metal layers. Each of the STUs is able to measure the supply voltage at one of the test points at a time and reconstruct the power supply waveform for that point. A multiplexer would be used to select the test point connected to a given STU. The STUs would sample the power supply of the appropriate test point multiple times during a clock cycle and output a binary number representing the voltage measured for each sample. This sequence of binary numbers would be used to reconstruct the power supply waveform during the clock cycle being tested. By doing this repeatedly for each clock cycle, the supply voltage waveform could be reconstructed for a given time interval of interest for debug purposes. This could then be used by the debug engineers to understand the effects that power supply variations were having on the timing of circuits, and if it led to the error being debugged.

In particular we sample the power supply fifteen times a clock period (assumed to be 1ns) and output a 10-bit number representing the voltage of each sample

The key contributions of this thesis are:

- The STU samples the power supply fifteen times a cycle, providing a high-accuracy measurement of the supply voltage at any test point.
- A complete reconstruction of the supply voltage waveform can be performed, enabling accurate debugging.
- Our scheme directly sample the power supply of any location on the die, including locations that are hard to probe using traditional techniques.
- Our scheme is integrated on-die, eliminating the need for expensive instrumentation, depackaging, or depassivation.

The rest of the thesis is organized as follows. Chapter II goes over the previous work in this area. In Chapter III the details of the circuit are explained. Then in Chap-

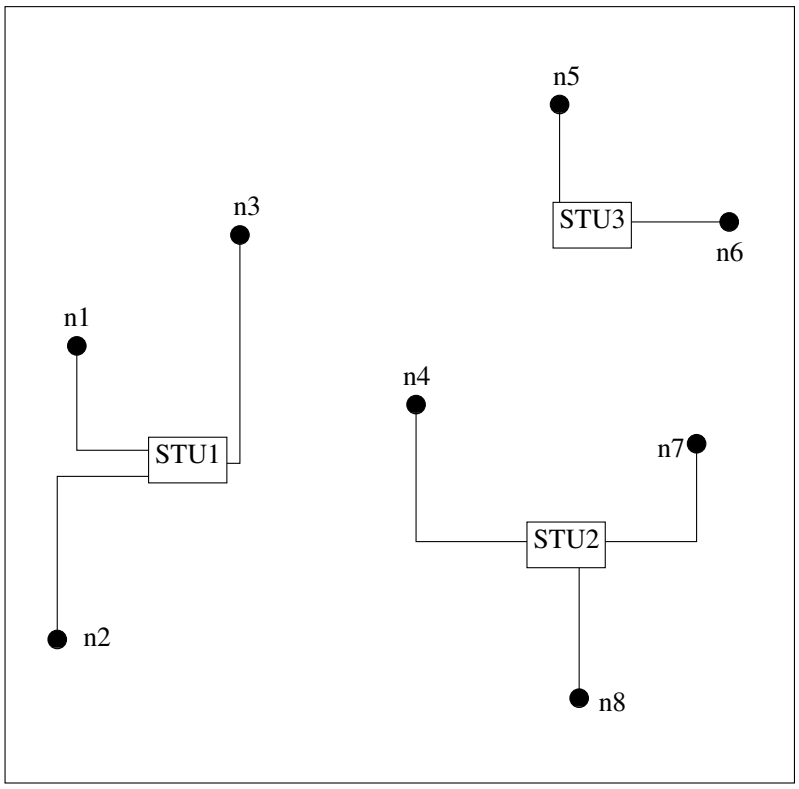

Fig. I.1. STU Based Architecture to Test Several Candidate Test Points

ter IV the results of simulations are reported, to validate the techniques and quantify its performance.

CHAPTER II

PREVIOUS WORKS

In this chapter, the previous approaches to address the problem of measuring power supply noise are examined. These methods include examining the effect that power supply noise has on the characteristics of a circuit, checking if the power supply has crossed a given threshold, or generating statistics about the power supply. Previous methods of sampling power supply voltages have usually only attempted to sample the power supply once per clock cycle, and have relied on indirect methods to measure variations in the power supply. Other approaches required special off-chip equipment, or were destructive in nature.

One method used in the past to detect spikes or droops in the power supply has been to look at the *effects* the noise would have on a circuit. In [2, 3, 4], the effects of the noise in the power supply on the propagation delay of an inverter is used to detect if there is noise in the power supply. This detection is done by checking for the violation of a setup time or by measuring the change in the frequency of a ring oscillator. Similarly in [5] the effect of noise in the power supply on NAND gates is used to estimate the supply voltage. Another method is to use changes in the voltage threshold curve of an inverter due to power supply noise to estimate the power supply value, as in [6]. None of these methods measure the power supply voltage directly, but rather, they provide a coarse estimate of the average value of the power supply over an interval. Unlike our approach, none of these techniques can recreate the power supply waveform. Also, by using indirect measurements, they average the value of the power supply over time, and can only report when the power supply voltage deviated by a certain amount, but not report the actual voltage.

Another method that has been used in the past is to have the circuit check if the

supply voltage has crossed a given threshold, as described in [7, 8]. Similarly, in [9], circuits have been devised to either output the lowest or highest voltage seen on the power supply over a given period. These methods can detect if the supply deviates by a given amount, but do not allow the user to reconstruct the power supply waveform. Recreating the supply waveform may be essential because failure may occur due to many smaller deviations from the nominal value of the supply, rather than one large deviation from the nominal value.

One proposed method generated the Power Spectral Density (PSD) of the supply voltage, as in [10]. This can give more information than the previous methods, but doesn't allow the debug engineer to know *when* a spike or droop in the power supply occurred.

In [11], a circuit was proposed that would output the power supply noise signal to be read by an oscilloscope. This would require a specialized oscilloscope to be able to connect to specially designed small pads on the chip, and would be a destructive test because the scheme requires a depackaged and depassivated part. Also, depackaging and depassivation would change the electrical behavior of the IC.

A method to convert the analog supply voltage to a digital output was proposed in [12, 13], but these do not sample the voltage before performing the conversion. This results in the scheme measuring an *average* of the power supply voltage during the conversion time (usually longer than one clock cycle). Similarly, in [14] a circuit using a Voltage Controlled Oscillator (VCO) with a counter (to create an Analog to Digital Converter) was proposed. However, like [12, 13], the power supply voltage is not sampled before conversion, resulting in an average measurement of the supply over a single clock cycle, losing the changes in the supply that occur *during* a clock cycle.

To get around the need for an on-chip circuit to measure the supply voltage, [15] proposes the use of an electron microscope to measure the supply. The use of the electron microscope limits where testing can be performed, and is destructive in the sense that it

requires depassivated and depackaged parts. Further, the method cannot probe the power supply at lower layers of metal, due to occlusion by wires of higher metal layers.

Other methods have focused not on measuring the power supply on-chip, but on developing Automatic Test Pattern Generation (ATPG) methods that take power supply noise into account, as discussed in [16]. These methods are designed to come up with test patterns that are less likely to cause large amounts of power supply noise during testing. However, these methods only *model* the power supply noise, and don't directly measure it on-chip. Without this measurement, the debug engineer wouldn't know for certain if the failure is caused by a fault in the chip, or because the test patterns are creating too much power supply noise.

The previous methods for detecting power supply noise use indirect measurements to sample the power supply, at most once a clock cycle. Since the switching activity will change during a clock cycle, it is important to be able to sample the supply multiple times during one clock cycle. Unlike previous works, our scheme is non-destructive, and directly samples the power supply voltage multiple times per clock cycle. The next chapter will go over the circuit to perform the power supply sampling.

CHAPTER III

APPROACH

In the last chapter, the previous approaches for detecting power supply noise were presented, which didn't sample the power supply directly or didn't sample the supply multiple times during a single clock cycle. In this chapter a circuit that can both sample the supply directly and several times per clock period is presented. Each of the major components in the circuit will be presented in the following sections.

III-A Overview

Our approach works in two phases. First, during the *sampling phase*, the power supply is sampled at a sampling rate F_s (15GHz in our experiments) for S samples. Assuming a clock rate F (1GHz in our experiment) for the test circuit, this produces $S = \frac{F_s}{F}$ samples per clock. For our experiment, S is fifteen samples per clock. Then, in the *conversion phase*, each of the samples collected in the first phase is converted from the stored analog value to a digital count. The resulting sequence of C counts is the output of the STU and can be used to reconstruct the original supply waveform. Note that the number of samples per clock (fifteen in our design) is limited by the conversion speed of the STU.

In practice the conversion rate (F_c) is much slower than the sampling rate (F_s). In particular, our design yields an F_c of 15 MHz. Hence the conversion time for $S = 15$ samples is $\frac{S}{F_c} = 1\mu s$. Therefore, to reconstruct the power supply for K clock cycles, the corresponding experiment should be repeated for $min(K, \frac{F_s}{F_c})$ times. This is because every $\frac{F_s}{F_c}$ clock cycle can be sampled then converted during the next $\frac{F_s}{F_c} - 1$ clock cycles before another clock cycle is sampled. During the first run, the 1^{st}, $\frac{F_s}{F_c} + 1$, $2 \times \frac{F_s}{F_c} + 1$, and so forth clock cycles are sampled. Then during the second run, the 2^{nd}, $\frac{F_s}{F_c} + 2$, $2 \times \frac{F_s}{F_c} + 2$, and so forth clock cycles are sampled. After this, the complete waveform could be reconstructed

by interleaving the results from each of the runs. If $K < \frac{F_s}{F_c}$, then K experiments are run and the results are concatenated.

III-B Block Diagram

A brief overview of our approach is presented next, followed by a detailed discussion of each component of our design.

The datapath of each STU is made up of fifteen Sample to Pulse Converters (SPC) that are connected through a MUX to a counter unit, as shown in Figure III.2. At the start of the sampling phase, the control unit produces a 66.67ps pulse on each of the $Ctrl_{read}$ pins of the SPCs The falling edge of the pulse to SPC_1 coincides with the rising edge of SPC_2, whose falling edge coincides with the rising edge of SPC_3, and so on. The $Ctrl_{read}$ signals for SPC_1, SPC_2, SPC_3, and SPC_{15} are shown in Figure III.1. When the $Ctrl_{read}$ line goes high for any SPC, it reads the voltage from V_{test} (the voltage of the power supply test point being analyzed) into a capacitor inside the SPC.

During the conversion phase, the voltage stored in the capacitors of the SPC is compared (using a Difference Amplifier) to a sawtooth waveform, V_{ref}. This produces a pulse train, where each pulse is high when V_{ref} is greater than the voltage stored in the capacitor C_{SPC} in the SPC. The MUX selects the input to the counter, driving the counter input with a pulse from SPC_1, followed by SPC_2, and so on, up to SPC_{15}. The counter measures the width of each of the pulses and outputs it as V_{out} (a 10-bit number) in Figure III.2. Finally, at the end of the conversion phase, fifteen V_{out} samples have been computed.

Along with a control circuit to synchronize the datapath, there is a ring oscillator clock circuit that produces a 15GHz clock. Additionally, there are two charge pumps to produce -VDD and $2\times$VDD, which are used by the SPCs

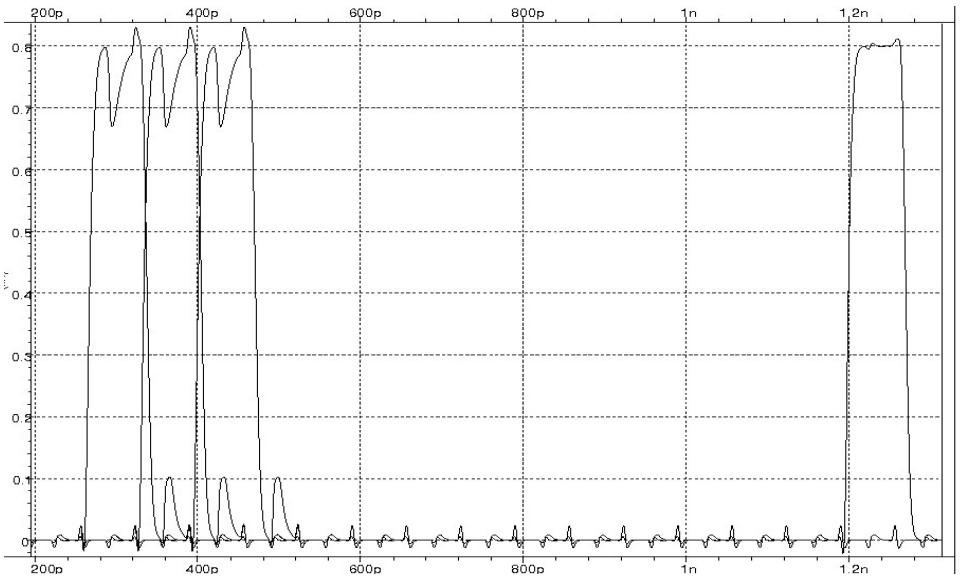

Fig. III.1. $Ctrl_{read}$ Waveforms for SPC_1, SPC_2, SPC_3, and SPC_{15}, With the Rest Suppressed for Clarity.

III-C Sample to Pulse Converter

The two main parts of the Sample to Pulse Converter are the Sample & Hold (SH) circuit and the Difference Amplifier, which will be covered in Section III-D & Section III-E, respectively. The voltage samples (V_{SH}) stored in the SH circuits (in C_{SH} in Figure III.4) can range between $VDD + V_R$ and $VDD - V_R$, where V_R is the maximum on-chip ripple (assumed to be 100mV). The rest of the SPC converts the voltage stored in the SH circuit (in C_{SH}), to a lower voltage that is less than VDD. This is done by charge sharing between the capacitor inside the SH circuit, C_{SH}, and the capacitor between the Sample & Hold and the Difference Amplifier, C_{SPC}, shown in Figure III.4 & Figure III.3, respectively. When the STU is in the sampling phase, $Ctrl_{out}$ is low, which means that M_1 in Figure III.3 is conductive and C_{SPC} is grounded. Then when $Ctrl_{out}$ goes high, C_{SH} and C_{SPC} are effectively shorted, making the voltage of the output of the SH block equal to

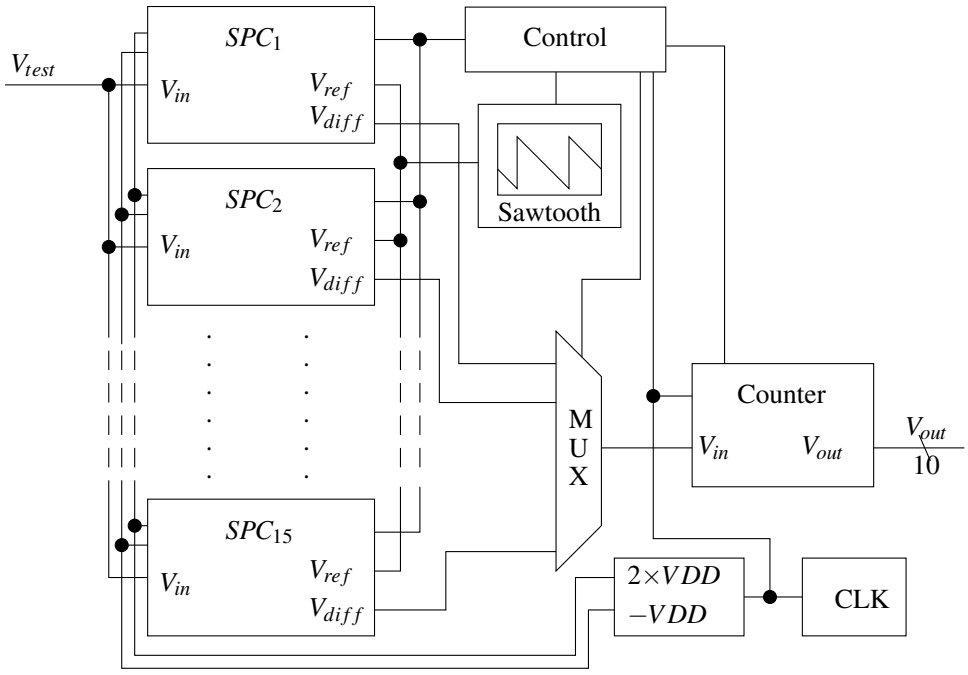

Fig. III.2. Block Diagram of Each Supply Testing Unit (STU)

$\frac{C_{SH}}{C_{SPC}+C_{SH}} \times V_{SH}$, where V_{SH} was the sampled voltage of the SH block.

III-D Sample & Hold

The Sample & Hold circuit is used to quickly sample the voltage of the power supply at the given test point and store it as a voltage (V_{SH}) in the capacitor C_{SH}. This is done using two complimentary passgates and a capacitor, C_{SH}, as shown in Figure III.4. For SPC_i, the $Ctrl_{read}$ pin is connected to the output $Ctrl_i$ from the Sample & Hold part of the control circuit, see Figure III.9. This signal will go high for a 66.67ns pulse during the sampling phase, allowing C_{SH} to sample the voltage on V_{in}. The C_{out} signal will stay low during the sampling phase, then go high during the conversion phase when the voltage in C_{SH} is shared with the voltage in C_{SPC}, as described above.

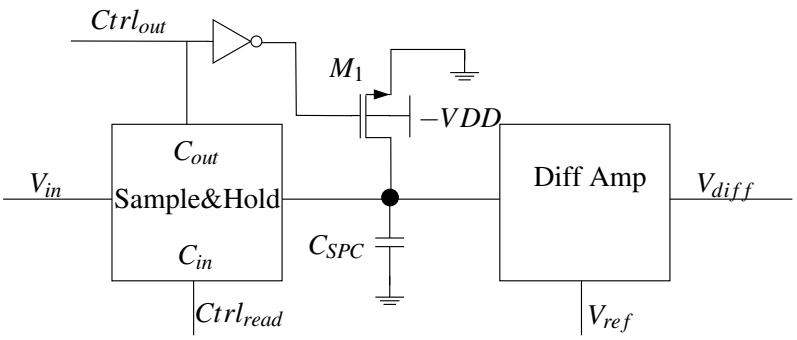

Fig. III.3. Sample to Pulse Converter (SPC)

There are two specifications that the Sample & Hold circuits must meet. First, the voltage on V_{in} should be faithfully stored into C_{SH} during the sampling phase. Second, the voltage in C_{SH} should be preserved during the conversion phase. To meet the first requirement the sizing of the first passgate (on the left of Figure III.4) and the capacitor (C_{SH}) must be such that their RC value is small enough to allow charging during a $\frac{1}{15\text{GHz}}$ period. The second requirement can be met by limiting the leakage during the conversion phase. Part of how this is meet is by heavily reverse biasing all of the MOSFETs in the passgates (set Figure III.4) to increase the magnitude of their threshold voltages, hence reducing their leakage. The MOSFET M_1 in the SPC (Figure III.3) is reverse biased for the same reasons. The leakage is also controlled by reducing the size of the passgates.

III-E Difference Amplifier

The Difference Amplifier takes as inputs the sawtooth V_{ref} signal and the voltage stored in C_{SPC}, which are labeled as V_{ref} and V_{SPC} in Figure III.5. When $V_{ref} > V_{SPC}$, the output V_{diff} is pulled high and when $V_{ref} < V_{SPC}$ it is pulled low. This creates a train of pulses on the output (V_{diff}) of the Difference Amplifier that have a frequency equal to the frequency of V_{ref} (15MHz). Note that this choice frequency results in fifteen sample being converted

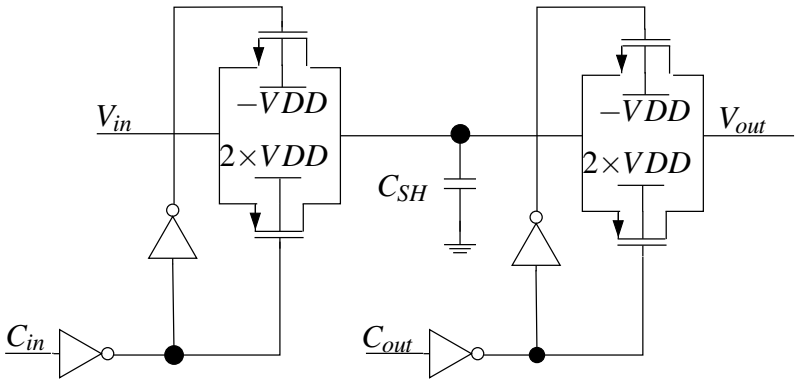

Fig. III.4. Sample & Hold Circuit

in $\frac{15}{15\text{MHz}} = 1\mu s$ The width of each pulse in the pulse train changes with the voltage stored in C_{SPC}. Lower voltages create wider pulses and higher voltages shorter ones. This train of pulses will continue until a new sample is read into C_{SPC}

III-F Sawtooth Waveform Generator

The Sawtooth circuit works in two stages, shown in Figure III.6. When V_{start} is low the gates of M_{vref} and M_7 are charged to VDD and ground respectively, then when V_{start} goes high the voltage at V_{ref} is linearly discharged, creating the sawtooth waveform.

The charging part of the circuit consist of M_{vref}, M_5, M_6, M_7 and M_8, shown in Figure III.6. During the charging phase the gate capacitance of M_{vref} is pulled high by M_5 while M_7's gate is pulled low through M_8. To prevent a short circuit from VDD to ground during the charging M_6 is used to separate the gates of M_{vref} and M_7. Similar to the SPC, the two gate capacitors M_7 and M_{vref} are used to adjust the starting voltage of the sawtooth circuit through charge sharing.

The discharging stage of the sawtooth circuit is composed of M_1, M_2, M_3 and M_4 in Figure III.6. The discharging is performed by using a simple current source, M_1, and

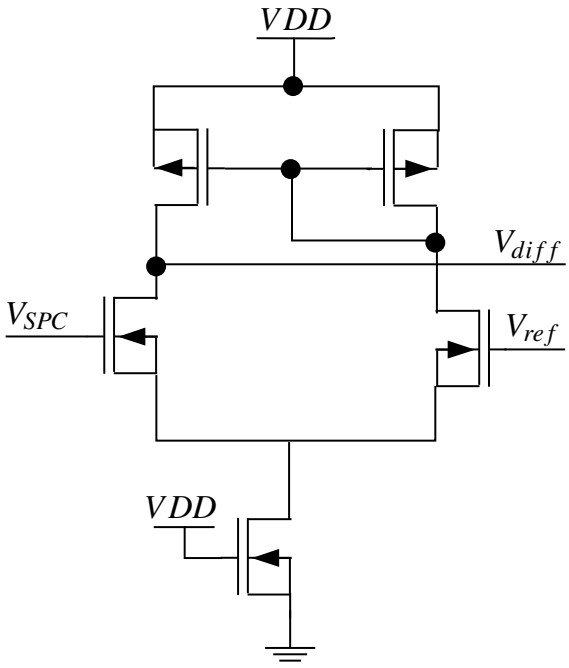

Fig. III.5. Difference Amplifier Circuit

mirroring it to V_{ref} using M_2 and M_3. The discharging is controlled by M_4, which acts as a foot device, turning off the discharging when V_{start} is low and the circuit is in the charging phase. During the discharge phase, a constant current is discharged from the capacitors M_7 and M_{vref}, resulting in a linear decreasing voltage (a sawtooth) on the output V_{ref} of the circuit, shown in Figure III.7.

III-G MUX

To connect the outputs of the fifteen SPCs to the single counter, a 16:1 MUX is used. The MUX is composed of sixteen complementary passgates and control logic for each of the passgates. By cycling through the fifteen SPCs at a 15MHz rate, the MUX output is a series of fifteen pulses. The first pulse is from V_{diff} from SPC_1, the second pulse is from

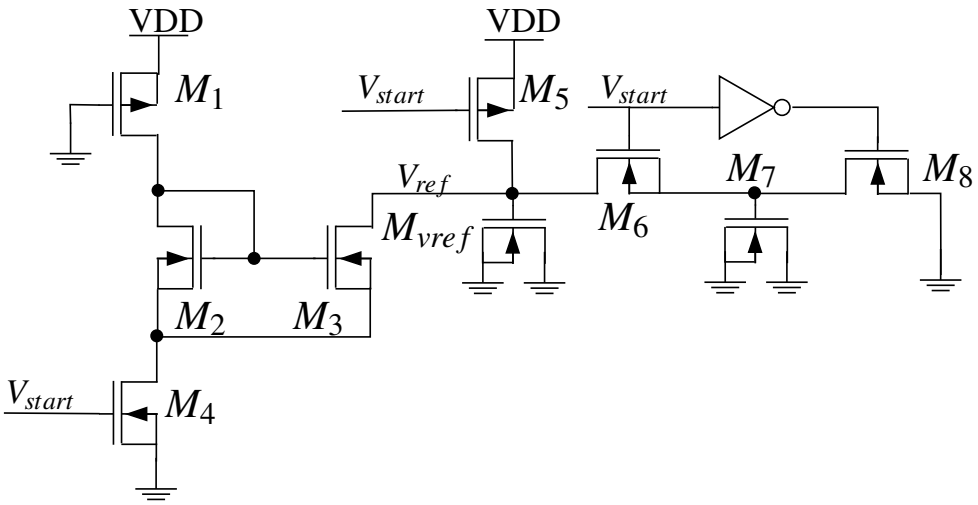

Fig. III.6. Sawtooth Circuit

V_{diff} from SPC_2, and so on. The sixteenth input to the MUX is grounded.

III-H Counter

The counter circuit, see Figure III.2, measures the width of the pulse on V_{in}, and outputs a 10-bit number, V_{out}. The value on V_{out} is proportional to the width of the pulse on V_{in} A chain of ten toggle flip-flops, with the output of the i^{th} flip flop connected to the clock of the next flip-flop, is used to create a standard counter circuit. The clock to the chain of flip-flops is gated with V_{in}, so that the counter counts the number of clock pulses while V_{in} is high. When the STU control switches the MUX, to select the V_{diff} from an new SPC, a reset signal is also sent to the counter circuit. Since the counter circuit is standard, a figure for this circuit is omitted.

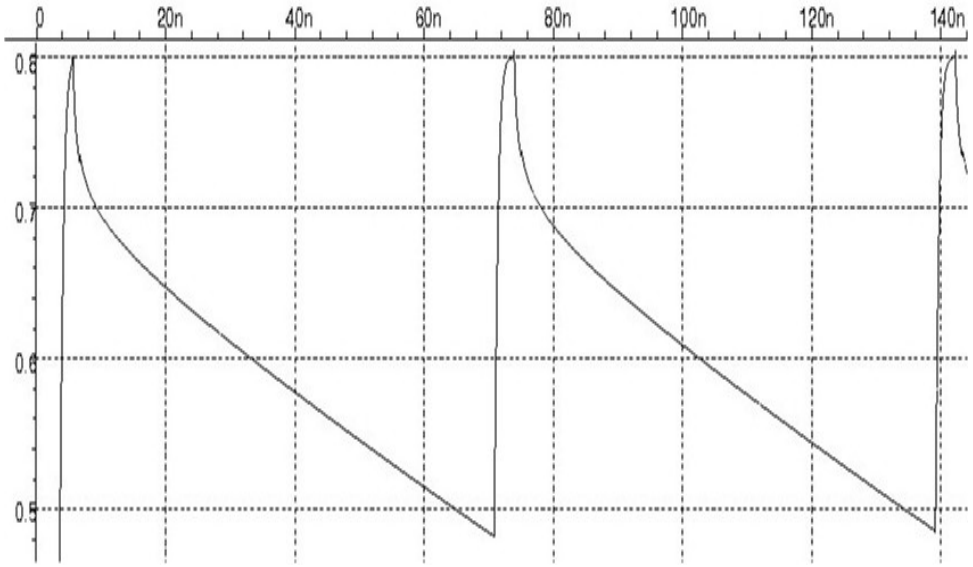

Fig. III.7. Sawtooth Circuit Output at V_{ref}

III-I Control Circuit

The control circuit is composed of a circuit controlling the $Ctrl_{read}$ pins of the SPC during the sampling phase, and a separate circuit controlling the STU during the conversion phase. The single input to the control circuit is a 2ns wide active high reset pulse, V_{reset}, which dictates when the sampling will start.

When the control circuit sees the reset pulse, it produces the fifteen $Ctrl_{read}$ pulses, each 66.67ps long, that are sent to the SPCs This is done by an edge detection circuit that outputs a 66.67ps pulse when a rising edge is seen on V_{reset}, shown in Figure III.8. The circuit computes a one cycle delay, V_{delay}, of V_{reset} and a two cycle delay, V_{delay2}, of V_{reset} using two flip-flops in series. The clock to both of the flip-flops is the 15GHz resonant clock. The logic in the circuit detects when V_{delay2} is low and V_{delay} is high, and generates a high pulse between the rising edge of V_{delay1} and V_{delay2}. The output of

the edge detection circuit, V_{pulse} (in Figure III.8) is connected to the $Ctrl_{in}$ input of the Sample & Hold control circuit (see Figure III.9). The Sample & Hold control is a series of fifteen flip-flops that produce the delayed versions of the 66.67ps wide V_{pulse} signal of Figure III.8. The signals $Ctrl_i$ in the Sample & Hold control circuit are connected to the $Ctrl_{read}$ pin of SPC_i, respectively (for $1 \leq i \leq 15$)

The second part of the control circuit generates the signals for the STU when it is in the conversion phase. These signals are based off of the clock divider circuit that creates clock frequencies at $(\frac{1}{2})$ to $(\frac{1}{2})^{14}$ of the 15GHz clock. These signals are called Div_1 through Div_{14} respectively, and are shown in Figure III.10. The signals Div_{11} to Div_{14} are used to control the MUX. They can effectively be thought of as a 4 bit counter operating at 15MHz and are used to change the SPC connected to the counter during the conversion phase.

The signal V_{start}, that controls the sawtooth circuit, is generated by a standard CMOS circuit implementation of the logic function $V_{start} = \overline{(Div_5 + Div_6) \cdot Div_7 \cdot Div_8 \cdot Div_9 \cdot Div_{10}}$. This creates a 15MHz signal (since the slowest signal, Div_{10}, is a 15MHz signal) that is low for three clock cycles of Div_4 (because of the $Div_5 + Div_6$ part) which equals $\frac{3}{937.5MHz} = 3.2$ns, during which time the sawtooth circuit recharges as described previously. Similarly the reset to the counter ($V_{controlReset}$) is generated from the logic function $V_{controlReset} = Div_5 \cdot \overline{Div_6} \cdot Div_7 \cdot Div_8 \cdot Div_9 \cdot Div_{10}$. This creates an active high signal that goes high during the last of the three clock cycles of Div_4 that V_{start} is low, insuring that the counter is reset right before the sawtooth starts discharging again. The last control signal is a readout signal ($V_{readout}$) that specifies when the current values of count are ready and should be read out. It is described by the logic function $V_{readout} = Div_5 \cdot Div_6 \cdot Div_7 \cdot Div_8 \cdot Div_9 \cdot Div_{10}$. This creates an active high signal that goes high during the first of the three clock cycle of Div_4 that V_{start} is low. Since this is when the sawtooth has just finished discharging, it insures that V_{out}, in Figure III.2, is read after the

conversion is completed, but before the counter is reset for the next conversion.

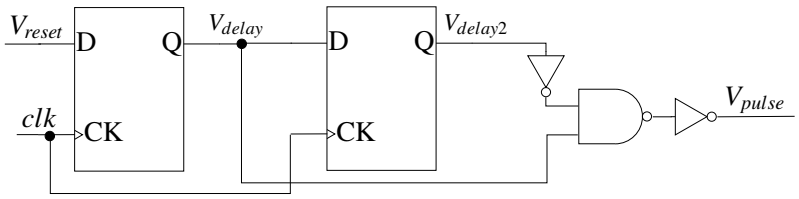

Fig. III.8. Edge Detection Circuit

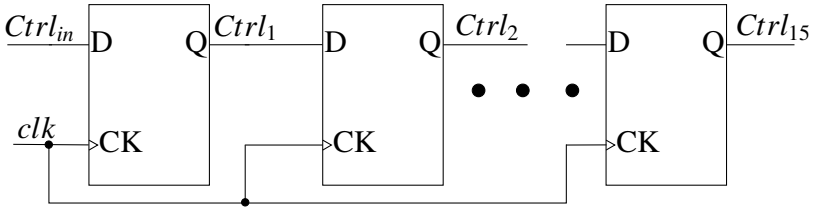

Fig. III.9. Sample & Hold Control Circuit

III-J Resonant Clock

The sampling performed by the STU is at a 15GHz frequency, and hence the 15GHz clock must be generated by the STU. This is done using a resonant clock, as described in [1]. This clock works by creating a sinusoidal standing wave oscillation in a Mobius shaped transmission line (Figure III.11). The Mobius ring is created by taking two concentric rings of metal and crossing them at one point, shown at the top of Figure III.11. The pair of inverters at the Mobius crossing force the two parallel sections of the band to have an

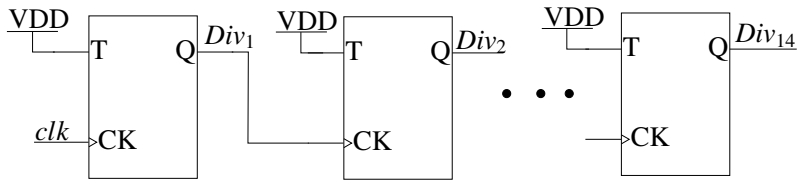

Fig. III.10. The Control Clock Divider Circuit

amplitude that is negative of each other. These two voltages will then propagate around the ring until (because of the Mobius crossing) they will each end up back at the inverter pair but on the opposite wire than which it started out on, creating a sinusoid oscillation in the Mobius band. The frequency of the standing wave is related to the inductance and capacitance of the Mobius band ($f = \frac{1}{\sqrt{LC}}$), which in turn is related to the perimeter of the band. The frequency can thus be tuned by selecting the correct size of the Mobius band. The inverter pair provides the negative resistance to sustain the oscillation. The standing wave can be connected to one or more recovery circuits (as shown in the figure) to convert the sinusoid standing wave to a square wave clock. Note that our design only uses one recovery circuit. However, for an IC with multiple STUs, neighboring STUs could share one Mobius ring and each have their own recovery circuit, cutting down on the active area for each STU. The recovery circuit is similar to the difference amplifier presented in Figure III.5.

III-K Charge Pump

Since several MOSFETs in the SPC are reverse biased, a -VDD source and a 2×VDD source are needed. Two Dickson charge pumps are used to do this, in Figure III.12 and Figure III.13. These are based off of the charge pumps in [17]. The basics of the 2×VDD charge pump (Figure III.12) is that the MOSFETs act as diodes allowing current to only

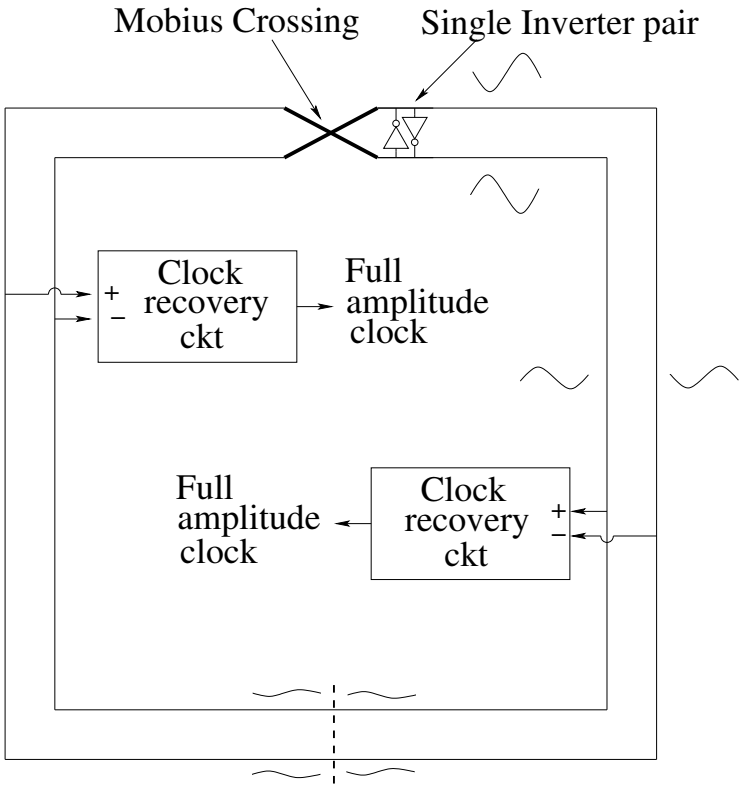

Fig. III.11. Resonant Clock Topology [1]

flow from the left to the right in the figure. By attaching the terminals of the capacitors (C_1-C_4) in each stage to *clk* and $\overline{clk}$, the voltage is increased from one stage to the next. When *clk* is low the gate of M_1 is at 2×VDD, since $\overline{clk}$ is at VDD and C_3 has a charge of VDD stored across it from the previous half clock cycle. This means M_1 is conducting and C_1 is being charged to VDD. While this is happening, M_2 is not conducting. Then when *clk* goes high, the top plate of C_1 is pumped up to 2×VDD. Now the gate of M_2 is at 3×VDD, since *clk* is at VDD and C_4 has 2×VDD stored across it from the previous half clock cycle, so M_2 is conducting and M_1 is not conducting. This allows C_2 to be charged to 2×VDD. The charging of C_3 and C_4 work in a similar manner.

Using *clk* to pump up the voltage from one stage to the next allows the circuit to create a square wave that oscillates from 2×VDD to 3×VDD with the same frequency as *clk* at V_1 and V_2. The two voltages (V_1 and V_2) are out of phase from each other since C_2 and C_4 are connected to $\overline{clk}$ and *clk* respectively. The MOSFETs M_3 and M_6 work like diodes only allowing current to flow when the voltage at their drain is greater than the voltage at their source. This allows C_5 to be charged to the highest voltage seen on V_1 and V_2. The capacitor C_5 is also used to smooth out the naturally induced ripples. The two stages are needed because in reality the highest voltage seen at V_1 and V_2 are lower than 3×VDD because of losses across M_1, M_2, M_4 and M_5, plus M_3 and M_6 drop the voltage seen at V_1 and V_2 by V_{th}, like a diode would. Resulting in an output at C_5 that is lower than the theoretical 3×VDD. This turned out to be 2×VDD in our design.

The -VDD charge pump works in a similar manner.

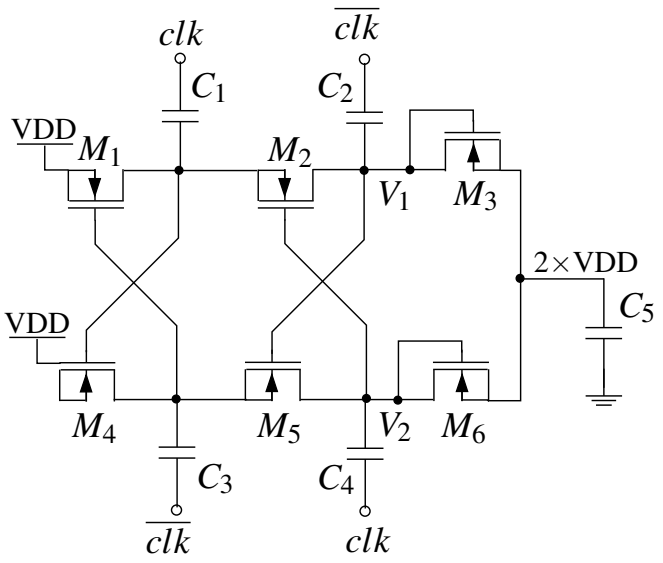

Fig. III.12. Dickson Charge Pump to Create 2×VDD.

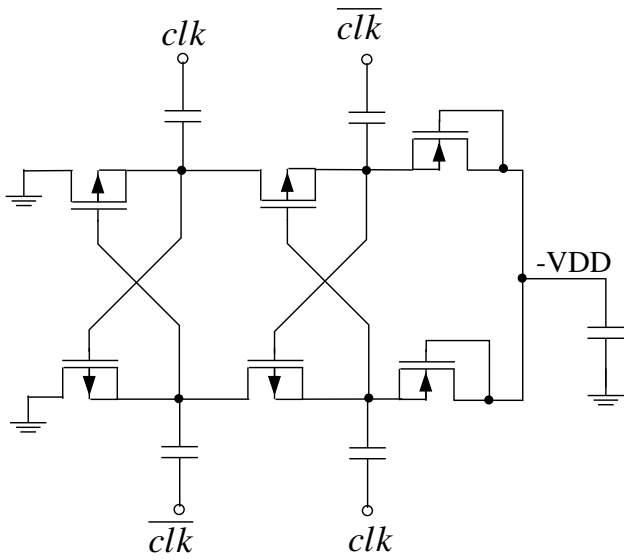

Fig. III.13. Dickson Charge Pump to Create -VDD.

This chapter presented the circuit for sampling the power supply, showing how the supply is first sampled very quickly by several STU circuits, after which each sample is serially converted into a digital word. This setup allows the power supply to be sampled multiple times during a single clock cycle and for the supply to be directly sampled. This process can be repeated with the same input test vectors to the chip to find the power supply voltage during the next clock cycle. By doing this repeatedly, it is possible to completely reconstruct the supply voltage over an arbitrary amount of time. The next chapter will go over the experiments used to test the functionality of our design.

CHAPTER IV

EXPERIMENTS

The previous chapter detailed the design of our circuit for sampling the power supply voltage over a single clock cycle. This chapter will go over the experiment setup to test the linearity of the output value of the circuit compared with the input supply voltage. This is a critical figure of merit that will show how accurate the circuit is. The second experiment will look at the effect the order in which the samples are converted has on the output data. Again, a linear relationship is needed for the circuit to work well. The area overhead associated with our scheme will be quantified as well.

IV-A Testing Linearity

All of the circuits described in this thesis were modeled in HSPICE [18] using a 22nm PTM technology [19]. The nominal VDD for this process is 0.8V. The length of the wire in the Mobius band for the resonant clock was 1026 μm with a width of 20 μm, and a spacing of 20 μm. This achieved a square wave clock signal at 15GHz.

The main experiment was to apply a test voltage to V_{test} and measure the change in V_{out} (see Figure III.2) as V_{test} was varied. At the start of this test, V_{test} was initially set to 0.8V, so that the test voltage would not be accidentally sampled during the start up. After the circuit had booted up, V_{test} was changed to the test voltage for that run, and the STU received a 2ns pulse on V_{reset} that would cause the STU to start sampling V_{test}. During this test, V_{test} was swept from 0.7V to 0.9V in 10mV increments. Figure IV.1 shows the outputted counts on V_{out} for the given voltages on V_{test}. This plot shows that the output count varies linearly, with an R^2 value of 0.9987, showing that it is possible to reconstruct the original waveform from the output count. Using the output count seen on V_{out}, and the corresponding voltage on V_{test}, we performed a linear curve fit to obtain

the equation $count = 3431.6 \times V_{test} + 2165.5$. From this equation we find that the voltage resolution of our design is $\frac{\Delta V}{count} = 0.291mV$. During this test the 2×VDD and -VDD circuits were not simulated since the HSPICE runtimes were prohibitive. However they were both simulated with fifteen SPCs connected as loads. The measured outputs from the charge pumps from this simulation were used in the overall test of the circuit. The high amount of switching in the charge pump circuits makes it infeasible to run HSPICE when the charge pumps are simulated along with the rest of the STU. By measuring the ripple under load in the separate charge pump simulation, and "replaying" the outputs of the charge pump while simulating the rest of the STU, we achieve reasonable HSPICE runtimes without compromising accuracy.

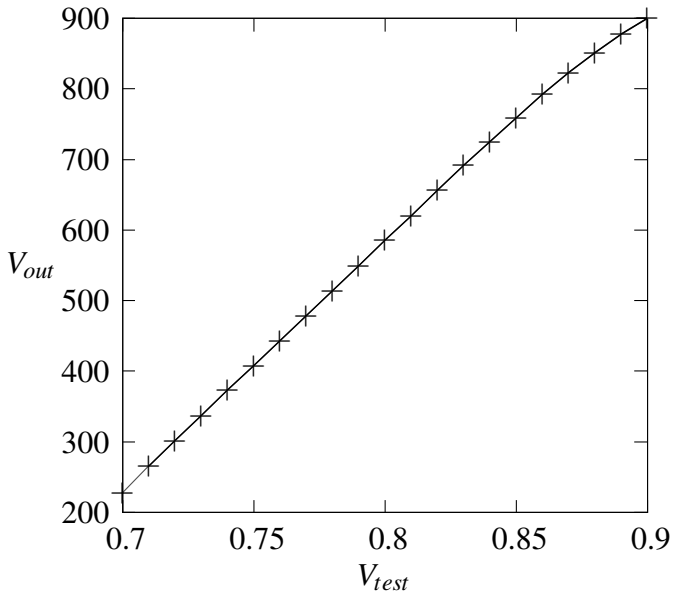

Fig. IV.1. Plot of V_{test} Versus the Digital Count on V_{out}

IV-B Sampling Order Effects

One potential concern with our approach is that after all fifteen SPCs sample the supply value, only one SPC converts its stored value at a time. This leaves open the possibility that even if they all sample the same voltage, they could produce different results, since the stored value of SPC_i could droop (due to leakage currents) before it is converted. To test what happened to the voltage stored in C_{SH} and C_{SPC} over time a voltage of 780mV was sampled by an SPC and the width of V_{diff} (the train of pulses produced by the difference amplifier) was measured over a one microsecond (the conversion period for our setup) timeframe. The result was that there was a decrease of 0.2ns for each pulse, which would decrease the output count by three for each conversion. This means that if, for example, all the SPCs sampled the same V_{test} and SPC_1 outputted 467, then SPC_2 would output 464, and SPC_3 would output 461, and so on. However, this change in the output versus the order of conversion is linear, with R^2 equal to 0.998. Hence, this effect can easily be compensated for by adding $3 \times (i-1)$ to the results of SPC_i.

IV-C Process Variation

To test the effects of **processes variation** on the design, several corner cases were run on the Sample & Hold Circuit, since the effect of variations on other blocks can be canceled during calibration. For the *fast corner* the widths of all the transistors were increased by 5%, the lengths were decreased by 5%, and the temperature was lowered to 0 degrees Celsius. For the *slow corner* the widths of all the transistors were decreased by 5%, the lengths were increased by 5%, and the temperature was raised to 50 degrees Celsius. The nominal temperature was 25 degrees Celsius. For the fast corner, the voltage change results in an average change of 31.87 counts in the digital value of the 10-bit output V_{out}. For the slow corner, the average change in the count of V_{out} was 26.07. In other words,

the change in the digital count is at most $\pm$ 3.12% and $\pm$ 2.55% respectively. The R^2 for the fast and slow corners are 0.9918 and 0.9976, respectively. Since the transistors for all the SPC blocks will vary in width, length and temperature together, the near unity R^2 value shows most of the error will be eliminated by calibration. Tolerance to temperature fluctuations after calibration was also tested, with the result that a 1% fluctuation in temperature gave a 0.04% fluctuation in V_{out} and a 5% fluctuation in temperature gave a 2% fluctuation in V_{out}. This shows that a small change in temperature will have an even smaller effect on the results.

IV-D Sawtooth Linearity & Total Area

The linearity of the sawtooth circuit was also tested, because of the effect of this circuit on the overall circuit's linearity. The result was that over the range of [0.7V, 0.9V] for V_{test}, the sawtooth had an R^2 equal to 0.9977. The range [0.7V, 0.9V] is the specified operating range of the STU. The percent of the clock cycle spent in this range was also calculated, since it equals the percent out of the total number of values of the 10 bit output that are actually used. The result was that the sawtooth spends 87.75% of the clock cycle in the [0.7V, 0.9V] range.

The total active area for the STU is tabulated in Table IV.1. The overall area of the STU is 10.77 μm^2. Of the total area, the largest part comes from the control circuit. The resonant clock and the sawtooth are the second and third largest sub-circuits of the STU. The control circuit, as shown in Figure III.10 and Figure III.9, has twenty-nine flip-flops which adds to its size. The resonant clock requires a pair of large inverters to operate and the sawtooth circuit has two large gate capacitors. In addition to the active area, the circuit also requires 27fF worth of MIM (metal-insulator-metal) capacitors that are used for C_{SH}, C_{SPC} and the capacitors in the charge pumps.

Name	Active Area (μm^2)
STU	10.77
SPC	0.04
Sawtooth	1.58
MUX	0.46
Counter	0.80
Control	3.88
Resonance Clock	3.17
Charge Pumps	0.28

Table IV.1. Sizes of STU and Its Sub-Circuits

IV-E Results

The experiments in this section showed that the circuit presented in the previous chapter can accurately measure the power supply voltage. Importantly, the output results have been shown to be linear with an R^2 of 0.9987, which shows that the input supply voltage can faithfully be reconstructed from the outputted words. The circuit was also shown to have a resolution of 0.291mV allowing for small changes in voltage to be detected. Furthermore the second experiment showed that though there is a change in the output word based on the order in which it was converted, this dependency is linear with an R^2 equal to 0.998. Taken together, the two experiments show that the circuit presented in this thesis can be used to accurately measure the supply voltage multiple times during a clock cycle, and output the results.

CHAPTER V

CONCLUSION

Previous works, as discussed in Chapter II, have focused on designs that don't measure the supply voltage directly, and don't sample the supply multiple times a cycle. For a debug engineer, there is a need to be able to reconstruct the value of the power supply, not just know whether it crossed a given threshold. A knowledge of the changes in the value of the power supply during a single clock cycle is important for the debug engineer. This thesis presented a circuit in Chapter III that accomplishes that goal, while the experiments in Chapter IV show that the circuit is very accurate while performing these conversions.

This circuit can sample the power supply voltage at a fast (15GHz) rate, taking several (15 in our experiments) samples per clock period. These samples are then each converted to a 10 bit output. The circuit is substantially linear with a resolution of ΔV = 0.291mV and an R^2 of 0.9987. This output, along with calibration data, can be used to completely reconstruct the supply voltage waveform over any interval of time. Along with the linearity of the conversion from input supply voltage to the output word, it was shown that the variation of the output word based on the conversion order is also linear, with an R^2 of 0.998. This means that any effect from the conversion order can be corrected. This allows a debug engineer to be able to accurately measure the power supply voltage at a given point in the design, for a clock cycle of interest. The whole process can be repeated to reconstruct the power supply waveform over an arbitrary time frame.

Furthermore, by sampling the power supply at a rate faster than the circuit under test clock rate, the debug engineer will be able to see how the supply varies over a clock period. Since the amount of circuit switching activity will vary over a single clock cycle, the power supply will also change, causing the circuit to slow down and speed up during the clock period. Thus is is important to measure the power supply at a faster rate than the

clock rate of the circuit under test to understand the effect that the power supply is having on the timing of the circuit under test.

Having detailed waveforms of the power supply with high resolution in both the voltage axis and the time axis can prove to be a crucial aid in the post-silicon debugging of the test chip. The sampling technique presented in this thesis provides this information to the debug engineer.

REFERENCES

[1] Victor Cordero and Sunil Khatri, "Clock distribution scheme using coplanar transmission lines," in *Design Automation, and Test in Europe (DATE) Conference*. IEEE, 2008, pp. 985–990.

[2] Josep Vázquez and José de Gyvez, "Power supply noise monitor for signal integrity faults," in *Design, Automation and Test in Europe Conference and Exhibition, 2004. Proceedings*. IEEE, 2004, vol. 2, pp. 1406–1407.

[3] Yasuhiro Ogasahara, Masanori Hashimoto, and Takao Onoye, "All-digital ring-oscillator-based macro for sensing dynamic supply noise waveform," *Solid-State Circuits, IEEE Journal of*, vol. 44, no. 6, pp. 1745–1755, 2009.

[4] Mariagrazia Graziano and Marco Vittori, "A fully digital power supply noise thermometer," in *SOC Conference, 2009. SOCC 2009. IEEE International*. IEEE, 2009, pp. 173–176.

[5] Cecilia Metra, Luca Schiano, M Favalli, and B Ricco, "Self-checking scheme for the on-line testing of power supply noise," in *Design, Automation and Test in Europe Conference and Exhibition, 2002. Proceedings*. IEEE, 2002, pp. 832–836.

[6] Arunkumar Vijayakumar, Raghavan Kumar, and Sandip Kundu, "On design of low cost power supply noise detection sensor for microprocessors," in *VLSI (ISVLSI), 2012 IEEE Computer Society Annual Symposium on*. IEEE, 2012, pp. 120–125.

[7] Ali Muhtaroglu, Greg Taylor, and Tawfik Rahal-Arabi, "On-die droop detector for analog sensing of power supply noise," *Solid-State Circuits, IEEE Journal of*, vol. 39, no. 4, pp. 651–660, 2004.

[8] Anuja Sehgal, Peilin Song, and Keith Jenkins, "On-chip real-time power supply noise detector," in *Solid-State Circuits Conference, 2006. ESSCIRC 2006. Proceedings of the 32nd European*. IEEE, 2006, pp. 380–383.

[9] Yuki Tamaki, Toru Nakura, Makoto Ikeda, and Kunihiro Asada, "A toggle-type peak hold circuit for local power supply noise detection," in *Quality Electronic Design (ASQED), 2010 2nd Asia Symposium on*. IEEE, 2010, pp. 29–32.

[10] Elad Alon, Vladimir Stojanovic, and Mark Horowitz, "Circuits and techniques for high-resolution measurement of on-chip power supply noise," *Solid-State Circuits, IEEE Journal of*, vol. 40, no. 4, pp. 820–828, 2005.

[11] Takeshi Okumoto, Makoto Nagata, and Kazuo Taki, "A built-in technique for probing power-supply noise distribution within large-scale digital integrated circuits," in *VLSI Circuits, 2004. Digest of Technical Papers. 2004 Symposium on*. IEEE, 2004, pp. 98–101.

[12] Kenji Shimazaki, Makoto Nagata, Takeshi Okumoto, Shouzou Hirano, and Hiroyuki Tsujikawa, "Dynamic power-supply and well noise measurement and analysis for high frequency body-biased circuits," in *VLSI Circuits, 2004. Digest of Technical Papers. 2004 Symposium on*. IEEE, 2004, pp. 94–97.

[13] Makoto Nagata, Jin Nagai, Takashi Morie, and Atsushi Iwata, "Measurements and analyses of substrate noise waveform in mixed-signal IC environment," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 19, no. 6, pp. 671–678, 2000.

[14] Rex Petersen, Pankaj Pant, Pablo Lopez, Aaron Barton, Jim Ignowski, and Doug Josephson, "Voltage transient detection and induction for debug and test," Test Conference, 2009. ITC 2009. International, 2009.

[15] Keith Jenkins and Robert Franch, "Measurement of VLSI power supply current by electron-beam probing," *Solid-State Circuits, IEEE Journal of*, vol. 27, no. 6, pp. 948–950, 1992.

[16] Mohammad Tehranipoor and Kenneth M Butler, "Power supply noise: A survey on effects and research," *IEEE Design & Test of Computers*, vol. 27, no. 2, pp. 51–67, 2010.

[17] Gaetano Palumbo and Domenico Pappalardo, "Charge pump circuits: An overview on design strategies and topologies," *Circuits and Systems Magazine, IEEE*, vol. 10, no. 1, pp. 31–45, 2010.

[18] Synopsys, "HSPICE," `http://www.synopsys.com/tools/Verification/AMSVerification/CircuitSimulation/HSPICE/Pages/default.aspx`, Accessed: 10/07/2013.

[19] Arizon State University, "Predictive Technology Model," `http://ptm.asu.edu`, Accessed: 10/21/2013.