A Low Power Controller for a MEMS Based Energy Converter

by

Scott Meninger

Bachelor of Science in Electrical Engineering
Boston University, May 1996

Submitted to the Department of Electrical Engineering and Computer Science in Partial Fulfillment of the Requirements for the Degree of

Master of Science

at the

Massachusetts Institute of Technology

June 1999

© 1999 Massachusetts Institute of Technology. All rights reserved.

Signature of Author _____
Department of Electrical Engineering and Computer Science
May, 1999

Certified by _____
Anantha Chandrakasan, Ph.D.
Associate Professor of Electrical Engineering
Thesis Supervisor

Accepted by _____
Arthur Clarke Smith, Ph.D.
Professor of Electrical Engineering
Graduate Officer

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
JUL 1 5 1999
LIBRARIES

ARCHIVES

A Low Power Controller for a MEMS Based Energy Converter

by

Scott Meninger

Submitted to the Department of Electrical Engineering and Computer Science on May, 1999 in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering and Computer Science

Abstract

The trend in modern VLSI design towards low power DSP and sensing applications creates an opportunity for the development of self-powered systems based on harvesting ambient energy. Several different ambient sources have already been exploited. With advances in microelectromechanical (MEMS) technology, it is possible to implement a self powered system-on-a-chip with the MEMS device acting as the energy transducer in the form of a variable capacitor, with conversion controlled by employing low power digital control techniques. This thesis explores the design of such an energy converter. The theory behind the conversion process will be discussed, including the presentation of a mathematical model for the system. The design of a programmable delay line based digital controller and optimization of the accompanying complementary power switches is reviewed. Results from the fabricated controller are presented and discussed. The design of a self-locking controller, which is based on the present architecture, but uses a new energy feedback technique to phase lock to maximal energy transfer, is presented.

Thesis Supervisor: Anantha Chandrakasan
Title: Associate Professor of Electrical Engineering

Acknowledgments

First I would like to thank Professor Chandrakasan for giving me the opportunity to do this thesis. Anantha has been an excellent advisor to work for and with. The research has been stimulating and challenging, and I have learned a great deal throughout my association with him.

I owe numerous thanks to the motley crew of 38-107, VLSI commandos that they are. Raj Amirtharajah provided the inspiration for this thesis work, but I don't hold that against him. Raj has been a great source for both technical advice and casual conversation. Duke Xanthopoulos is probably more responsible for my layout functioning than I am. His advice and paranoia have been appreciated. I'm still trying to calculate the wavelength of my layout reference marker... Tom Simon, aside from being the circuit hacker supreme, provided me with enough Mahler to get me through writing this thesis. It's safe to say without his IO pads all would have been for naught. Jim Goodman has been an excellent lab mate and provided many laughs. His honest interest in discussing technical issues has been appreciated. Josh Bretz, SeongHwan Cho, Vadim Gutnik, Wendi Rabiner, Alice Wang, Amit Sinha, Gangs Konduri, and Rex Min have all been great to work with and I thank them for indulging me in afternoon football sessions, despite the fact that everyone was so busy. In general, I'd like to thank the group for collectively setting the bar so high, but being willing to help carry me over it.

Margaret Flaherty has provided great support throughout the past year. I especially appreciate her help in getting the paperwork through for the test board during winter break.

Professor Jeff Lang and Oscar Mur-Miranda have done the excellent work on the MEMS transducer which makes this thesis plausible, and they have been a pleasure to work with.

I would like to thank the members of D.O.A.I.H.P.S. (Rob Bean, Chris Brown, Rufus Chaffee, Ken Lambert) for their support and encouragement, but that would be hypocritical. I guess I can thank them most for not getting in my way too severely, and for not killing me off at York, ...*yet.* I would like to thank these jokers for helping me keep what little remnants of my sanity remain. Speaking of which, York VIII approacheth....

Of course the support of my family and friends throughout have been appreciated and I would like to express my gratitude to them all. In order to save a small forest, I will not name all of my family members, but want to let them know their encouragement has been appreciated.

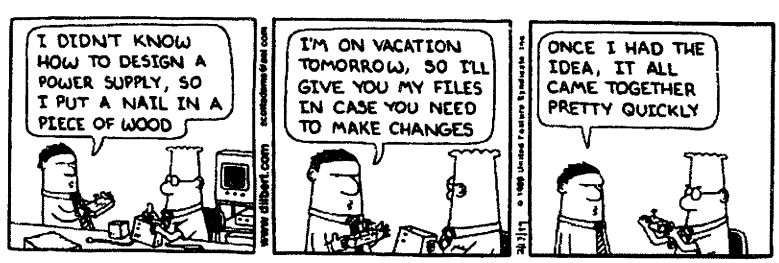
I would like to thank Barry and Joanne Rhoads, Linda Garland, and Sav Correale for their encouragement, support, and treatment as if I was one of the family. And, of course, I'd like to thank Anne (Mum) Garland for the love that showed through numerous whoopie pies, cookies, and conversations. She is missed.

I thank my parents, Donald and Mary (Joan) Meninger for the support they give in so many ways. From Dad making egg sandwiches in the morning so I don't starve, to Mom worrying why it is so late and I am still at the lab, they have done everything possible to support and encourage my efforts. For this and for being an inspiration, I am eternally grateful.

The person who deserves the most credit/blame for this thesis is the love of my life, Stacy Rhoads. She has been a source of continued encouragement and support for many years now, and is most responsible for my return to school. I can not thank her enough for everything she has done for me and for giving me a reason to strive to succeed, but I intend to spend the rest of my life trying.

Finally, I would like to acknowledge the man, Bobby Rushhhh, for simply being who he is.

Figure 1 shows the basic design flow that was followed during the development of this thesis. It is amazing how appropriate it is, not only with respect to the carpenter analogy, but even down to the engineer's sloping forehead...



DILBERT Copyright © 1998 United Feature Syndicate, Inc
Redistribution in whole or in part prohibited

Figure 1: Art Imitating Life

Contents

1 Introduction **15**

1.1 Harvesting Energy 15

1.2 System Overview 16

1.3 Thesis Scope 16

2 Technique for Converting Energy **17**

2.1 Energy Conversion 18

2.1.1 Voltage Constrained Conversion 19

2.1.2 Charge Constrained Conversion 20

2.1.3 Modified Charge Constrained Conversion 21

3 Controller Implementation **23**

3.1 Mathematical Modeling 24

3.1.1 Derivation of Timing Values 25

3.2 Power Electronics 28

3.2.1 Q_o 28

3.2.2 Parallel Capacitor Optimization 29

3.2.3 Design of Switches 30

3.3 Other Considerations for the Power Electronics Circuitry 31

3.3.1 Reactive Ringing 32

3.3.2 Capacitor Resetting 32

3.3.3 Sizing C_{res} 33

3.4 Digital Controller 35

3.5 Synchronous Gated Oscillator 37
3.6 Fabricated IC Testing and Results 40
3.7 Testing Requirements 43
3.7.1 Serial Interface 44
3.8 Comparisons with Modeling 45
3.8.1 Power Electronics Functionality 45
3.8.2 Loss Estimation 46

4 Energy Feedback Controller 47
4.1 Delay Locked Loop: Overview 48
4.2 Energy Based Feedback 49
4.2.1 Energy Measurement 50
4.3 Feedback Approaches 53
4.4 Delay Line Length 55
4.4.1 Delay Line Length Calibration 56
4.5 Delay Line Architecture 57
4.5.1 Duty Cycle Requirements of the Delay Line 58
4.5.2 Decoding the Line Length 59
4.5.3 Single Buffer Delay Design 60
4.5.4 Reference Clock Polarity 63
4.6 Implementation of Lock Algorithm 64
4.6.1 Peak Detection 64
4.6.2 Delay Line Length Control 65
4.6.3 Lock Time 65
4.6.4 Lock Updating 66
4.6.5 Re-lock 68
4.7 Ancillary Issues 68
4.7.1 Conversion Rate and C_{res} 68
4.7.2 Proposed Control Algorithm 71
4.7.3 Possible Modifications to Proposed Algorithm 74

5 Conclusions and Future Work **77**

5.1 Conclusions 77

5.2 Future Work 77

5.2.1 Current Controller 77

5.2.2 Feedback Controller 77

5.3 Feedback 78

5.4 Digital Calibration 79

Bibliography **81**

A Bonding Diagram and Pin-out **83**

B Serial Interface Code **87**

List of Figures

1 Art Imitating Life 6

1.1 System Block Diagram 16

2.1 MEMS Transducer Plan View 17
2.2 MEMS Transducer Cross-Sectional View 18
2.3 Conversion Cycles 19
2.4 Modified Energy Conversion with C_{par} 22

3.1 System Architecture (Single Conversion Phase) 23
3.2 Block Diagram of MATLab Model 24
3.3 Circuit Topology in Various States During Energy Conversion . . . 25
3.4 Optimization Curves for Various L and C_{par} 29
3.5 Power FET Sizing Curves 31
3.6 Complete Power Electronics Circuit (Single Phase) 32
3.7 Layout of the Power Switches Section 33
3.8 Zoom in showing guard rings and isolation structures 34
3.9 SW3 Discharges the Capacitor Combination for Non-Ideal t4 35
3.10 Hybrid Delay Line 36
3.11 Gated Oscillator Circuit 38
3.12 Gated Oscillator Waveform 38
3.13 Chip Photograph 40
3.14 Test Circuit for Fabricated Controller 41
3.15 Serial Port Timing Waveforms 44
3.16 HSpice Simulation Results 45

3.17 Data Comparison 46

4.1 Traditional Feedback 47
4.2 DLL Block Diagram 48
4.3 Energy Out vs. Delay 50
4.4 Energy Measurement Circuit 51
4.5 Energy to Count transfer Characteristic 51
4.6 Energy and Capacitor Voltage vs. Delay 52
4.7 Analog Multiplexor 53
4.8 Locking based on a count threshold 54
4.9 Locking based on examining a full period 54
4.10 Single Period Calibration Circuit 57
4.11 Pulse Width Growth Phenomena 58
4.12 Binary Weighted Delay Line 60
4.13 Buffer Delay Options 61
4.14 Overlap Current [1] 62
4.15 Reference Clock Inverter 63
4.16 Peak Energy Detector Circuitry 64
4.17 Delay Line Control Circuitry 65
4.18 Energy Converter Operating Modes 71
4.19 Open Loop Operation 72
4.20 Feedback Operation 73

A.1 Pin-out Schematic 83
A.2 Bonding Diagram 85

List of Tables

3.1 Controller Specifications (half circuit) 41
3.2 Power Outputs for Various V_{max} (Single Phase) 42

A.1 Chip Pin-out 84

Chapter 1

Introduction

As the field of VLSI expands further into the area of remote sensing and communications, the use of low power design methodologies to improve power source lifetime become increasingly important [2, 3]. Many techniques have been developed to create more effective DC-DC conversion technologies [4, 5]. While these approaches offer opportunities to increase battery or fuel cell life, the need for periodic system maintenance, in the form of power source replacement, still exists.

As the power requirement for the load circuit decreases, the opportunity to utilize ambient energy as an alternate power source becomes a viable alternative. Several ambient sources have already been exploited. These include RF[6], solar, fluid flow, and mechanical vibration [7, 8] sources.

Recent advances in MEMS (microelectromechanical systems) technology enable the creation of a self-powered system with the MEMS device acting as a electromechanical transducer in the form of a variable capacitor. By placing charge on the capacitor plates and then having a mechanical system (ambient vibration) move the plates in space, energy can be added to the stored charge and harvested for use by downstream load electronics. Power requirements of DSP applications with low duty cycles, such as periodic FFT monitoring of intermittent processes, have been reduced to the μW range [7, 9, 10], making this technique plausible.

1.1 Harvesting Energy

The concept of harvesting ambient energy to power electronic systems has been getting new attention recently as both process and design techniques for low power continue to improve. In [7], a system was presented capable of converting ambient mechanical vibration for use in low power systems into the 100μW range. Transduction was performed by employing a moving coil in a stationary magnetic field and using a rectifier and DC-DC converter to generate the required output voltage.

1.2 System Overview

The system this thesis proposes to harvest ambient mechanical vibration is shown in Figure 1.1. The mechanical system is modeled as a vibration source which couples into the electrical system through the MEMS transducer. A low power controller directs energy conversion and supplies power to the load. The controller consists of a power electronics subsystem, which is responsible for exciting the transducer through its energy conversion cycle and has been optimized to minimize losses, and a digital control core which generates the timing pulses that drive the gates of the power FETs in the power electronics subsystem.

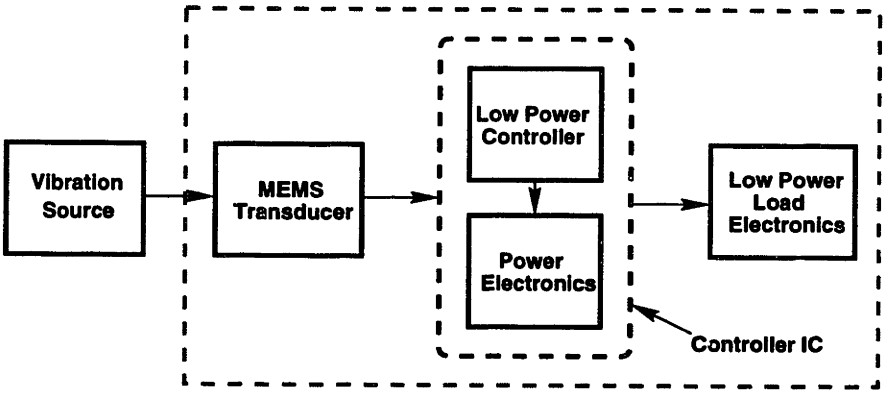


Figure 1.1: System Block Diagram

1.3 Thesis Scope

This thesis presents the design of an ultra low power digital controller with integrated power electronics for an ambient energy converter. Two different possible conversion processes will be explained, and the tradeoffs between them discussed. An optimal, modified conversion cycle has been derived and will be explained. The low power optimization approaches used to design the digital control core and power electronics will be discussed. Results from the fabricated controller will be compared to predictions made based on mathematical modeling of the system and Hspice simulations. Finally, the design of a second controller will be presented. This revised controller utilizes a new method of feedback based on measurement of converted energy, and the issues relevant to such an approach will be examined.

Chapter 2

Technique for Converting Energy

The method proposed to convert ambient mechanical vibration into electrical energy is to use a MEMS variable capacitor. By placing charge on the capacitor plates and then moving the plates apart, mechanical energy can be converted into electrical energy which can then be stored and utilized by a load.

The MEMS transducer is shown in Figure 2.1 in plan view and in Figure 2.2 in cross-sectional view. [11] The structure is built using deep reactive ion etching and wafer bonding technology. The lower wafer, the *handle* wafer, serves to support the identical upper *device* wafer. The two wafers are bonded with a thin layer of silicon dioxide. The parasitic capacitance created by the bonding oxide has a beneficial effect which will be discussed in section 2.1.3. The mechanism by which the MEMS

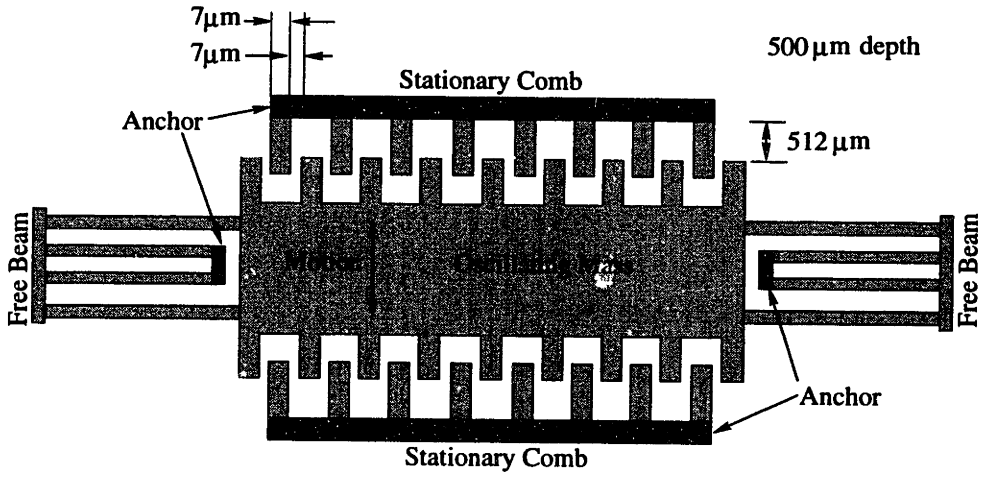


Figure 2.1: MEMS Transducer Plan View

transducer creates a variable capacitance is as follows. The central mass is suspended

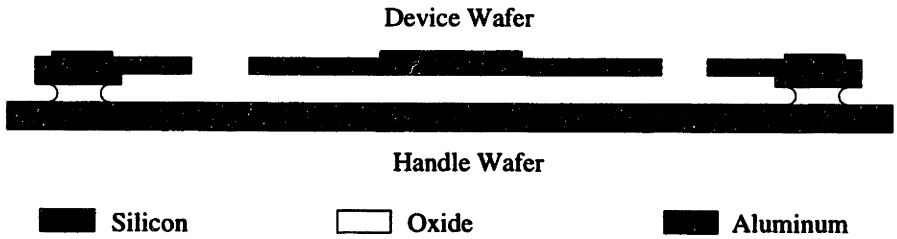


Figure 2.2: MEMS Transducer Cross-Sectional View

by the spring system depicted on both sides, which are attached to the handle wafer at the anchor points denoted in Figure 2.1. As the system is shaken by the vibration, the inertia of the mass causes the interdigitated fingers of the stationary combs and oscillating mass to move back and forth with respect to each other. The spring system is designed to restrict motion of the mass in only the lateral direction. The fingers on the moving mass and stationary comb fingers make up two terminals of a variable capacitance. The values of maximum and minimum capacitance achieved depend on the number of fingers, surface area overlap of the interdigitated fingers, and the maximum and minimum spacing in the gap at the extremes of the proof mass deflection. The fingers are covered with a teflon-like coating as part of the etching process which also serves to prevent shorts from occurring should the plates move enough to touch one another.

The MEMS structure actually creates *two* variable capacitors because of the symmetrical layout of stationary combs on either side of the oscillating mass. As one side moves to maximum capacitance (distance between mass fingers and stationary comb at a minimum) the other side moves to minimum capacitance. Therefore, the two resulting capacitive structures are 180 degrees out of phase with each other. The energy conversion process which will be described in section 2.1 may therefore be carried out in two phases, staggered by half of a vibration period, during steady-state operation of the converter.

2.1 Energy Conversion

There are two possible energy conversion cycles for the MEMS transducer as shown in Figure 2.3. Path A-B-D-A depicts charge constrained conversion, while path A-C-D-A depicts voltage constrained conversion. One basic constraint for both cycles is that there is some maximum allowable voltage, V_{max}, which is set by some process or system requirement. For example, the power switches which are employed in the converter will have oxide and channel breakdown limits which must be considered, and hot electron effects can become important at high voltages. Also, the MEMS

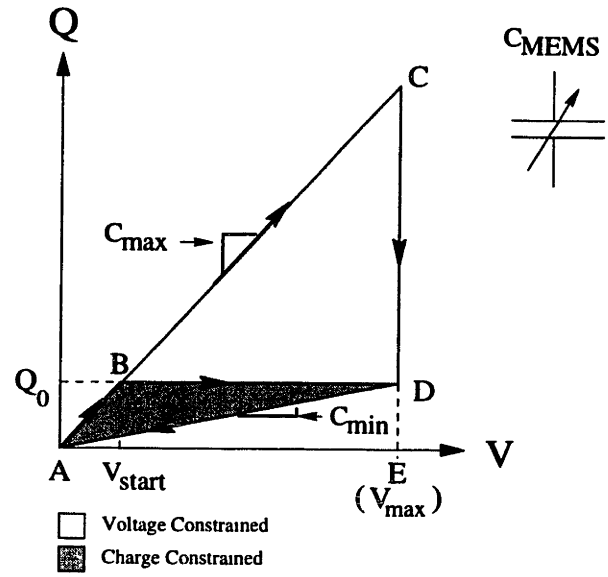


Figure 2.3: Conversion Cycles

device itself will have a maximum field limit which it can withstand when its plates are closest together.

2.1.1 Voltage Constrained Conversion

For the voltage constrained case, the cycle starts when the capacitor is charged up to V_{max} from a reservoir. This is done when the capacitance of the MEMS transducer is at a maximum (C_{max}). During this time, the value of C_{MEMS} is taken to be constant, and so segment A-C is a straight line. This is a valid assumption since the charge-up time to traverse path A-C (and discharge path D-A) is an electrical time near $0.6\mu s$, while path segment C-D, which corresponds to the plates moving, is traversed over a mechanical time near $400\mu s$. As the plates move and the capacitance decreases, path segment C-D is traversed to point D, where the capacitance is at a minimum. The mechanical force does work by causing charge to move from the capacitor back into the reservoir. If we think of the charge being removed as occurring in discrete packets, then we see that we remove these δQ packets at a higher voltage than that at which they were initially integrated onto the capacitor. The charge remaining on the plates is then recovered while $C_{MEMS} = C_{min}$ following path D-A. The energy input to the system is

$$E_{in} = \frac{1}{2} C_{max} V_{max}^2 \tag{2.1}$$

Since Q=CV, we can express this as

$$\frac{1}{2}Q_C V_{max} \tag{2.2}$$

where Q_C is the value of charge at point C, which corresponds to a voltage equal to V_{max}. The input energy is therefore the area of triangle ACE. The residual energy removed after energy harvesting has been performed is

$$E_{res} = \frac{1}{2}C_{min}V_{max}^2 = \frac{1}{2}Q_D V_{max}^2 \tag{2.3}$$

where Q_D is the charge remaining on the capacitor at point D. This is the area of a triangle formed by points ADE. The net energy gained, E, is the shaded area ACD in Figure 2.3. [12]

$$E = \frac{1}{2}(C_{max} - C_{min})V_{max}^2 \tag{2.4}$$

This method sets a maximum limit on the conversion process. The major problem with this approach is that some method must be employed to constrain the voltage across the MEMS device during the conversion process, which would require another source of value V_{max}. This supply is in addition to the conversion charge reservoir, which is of a lower voltage and powers the control electronics.

2.1.2 Charge Constrained Conversion

In the charge constrained case, C_{MEMS} is charged to some initial voltage while its capacitance is at a maximum, which corresponds to path segment A-B in Figure 2.3. As the capacitor separates, the voltage increases as capacitance decreases until the plate displacement is at a maximum ($C_{MEMS} = C_{min}$) at point D. The amount of charge initially placed on the plates, Q_o, was precalculated such that when C_{MEMS} reaches its minimum, the value of the voltage across the capacitor is V_{max}. The charge is then returned to the reservoir along path D-A. During this process the amount of charge on the plates does not change as the mechanical work is converted into electrical potential energy. The net energy out is the shaded area ABD. It is immediately obvious that this energy is less than what is possible with the voltage constrained conversion cycle. The advantage is that now only a single charge source is needed to complete the process, and its value can be much less than V_{max}.

We can derive an expression for the amount of energy available from charge constrained conversion in a similar manner as we did for voltage constrained conversion, where we showed that the ouput energy is the shaded area encompassed by the charge conversion process. For charge constrained conversion this is area ACD (which is the energy gained for voltage constrained conversion) minus area ABD.

$$E_{chrgcons} = E_{voltcons} - \frac{1}{2}(V_{max} - V_{start})(Q_C - Q_D) \tag{2.5}$$

where Q_C and Q_D are the values of charge at points C and D, respectively.

$$E_{chrgcons} = E_{voltcons} - \frac{1}{2}(C_{max}V_{max} - C_{min}V_{max}) \tag{2.6}$$

Using equations 2.4 and 2.6, we arrive at

$$E_{chrgcons} = \frac{1}{2}(C_{max} - C_{min})V_{max}V_{start} \tag{2.7}$$

If we compare equation 2.7 to equation 2.4 we see that the energy available from the charge constrained case is less than that available from the voltage constrained case by a factor V_{max}/V_{start}.

2.1.3 Modified Charge Constrained Conversion

Figure 2.4 depicts another alternative. Here, a capacitor of constant value, C_{par}, has been added in parallel to the MEMS device. Path a-b-d-a represents conversion without C_{par} and path a-b'-d'-a includes C_{par}. The energy converted in area acda, $E_{voltcons}$ equals the converted energy of shaded area ac'd'a, so no benefit for the voltage constrained cycle has been gained by incorporating C_{par}. However, if the energy converted for charge constrained cycles abda and ab'd'a are compared, it is evident that more energy, $E'_{chrgcons}$, is converted in area ab'd'a.

$$path\ \ a-c-d-a: \quad E_{voltcons} = \frac{Q_2V_{max}}{2} - \frac{Q_1V_{max}}{2} \tag{2.8}$$

$$path\ \ a-b-d-a: \quad E_{chrgcons} = E_{voltcons} - \frac{(Q_2-Q_1)(V_{max}-V_1)}{2} \tag{2.9}$$

let $Q_2 - Q_1 = \Delta Q$. Then,

$$E_{chrgcons} = E_{voltcons} - \frac{(\Delta Q)^2}{2C_{max}} \tag{2.10}$$

Equation 2.10 shows that there will be a significant difference between the amount of energy we get out for a given V_{max} depending on which approach we take. If a capacitor, C_{par}, is added in parallel to C_{MEMS}:

$$path\ \ a-c'-d'-a: \quad E'_{voltcons} = \frac{Q_4V_{max}}{2} - \frac{Q_3V_{max}}{2} \tag{2.11}$$

$$path\ \ a-b'-d'-a: \quad E'_{chrgcons} = E'_{voltcons} - \frac{(Q_4-Q_3)(V_{max}-V_2)}{2} \tag{2.12}$$

it can be shown that:

$$Q_4 - Q_3 = Q_2 - Q_1 = \Delta Q$$

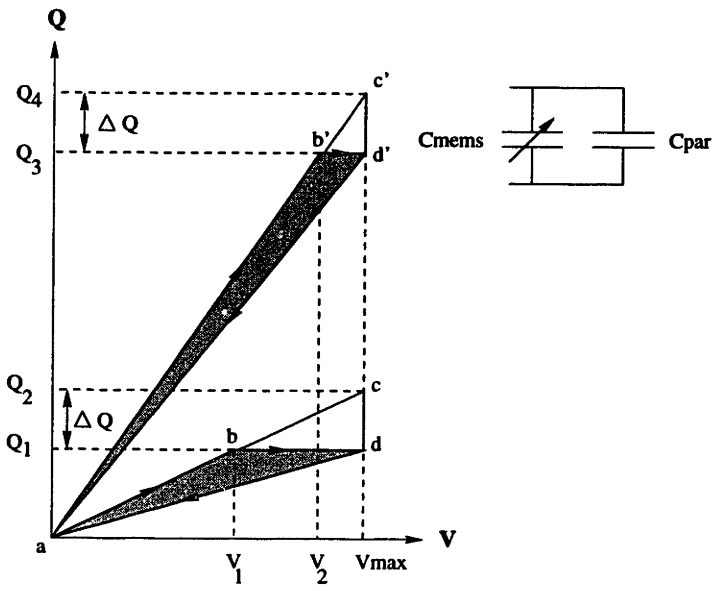


Figure 2.4: Modified Energy Conversion with C_{par}

therefore:

$$E'_{chrgcons} = E'_{voltcons} - \frac{(\Delta Q)^2}{2(C_{par} + C_{max})} \tag{2.13}$$

Equation 2.13 shows that in the limit as C_{par} approaches infinity, the modified charge constrained energy approaches that available through voltage constraint. Therefore, it is desirable to have a parallel capacitor to, in effect, "hold" the voltage across the MEMS device constant, mimicking the behavior of the voltage constrained condition. The disadvantage to adding C_{par} is that now more initial charge is required for the conversion process (Q_3 vs. Q_1). This means that the losses associated with energy flow in the system will be increased. The tradeoff between increasing the capacitance of C_{par} and the increase in losses will be discussed in detail in section 3.2.

An interesting point to note is that, as mentioned in the introduction to this chapter, the construction of the MEMS device creates a parasitic capacitance between the two wafers making up the transducer. This capacitance can be used to create the required value of C_{par} by tailoring the bonding oxide thickness to an appropriate value, eliminating the need for an off-chip parallel capacitor.

Chapter 3

Controller Implementation

This chapter discusses the design of a low power controller integrated circuit which orchestrates the energy conversion process. The design flow began with mathematical modeling of both system and component behavior at a low level of abstraction. From the models, a controller architecture composed of two subsystems (power electronics and control core logic) was designed, resulting in a novel DC-DC converter topology. Low power methodologies were utilized throughout the circuit design process. Finally, verification of the system was performed by simulating both the schematic level and layout level designs. The chip has been fabricated and tested. The results are presented in section 3.6.

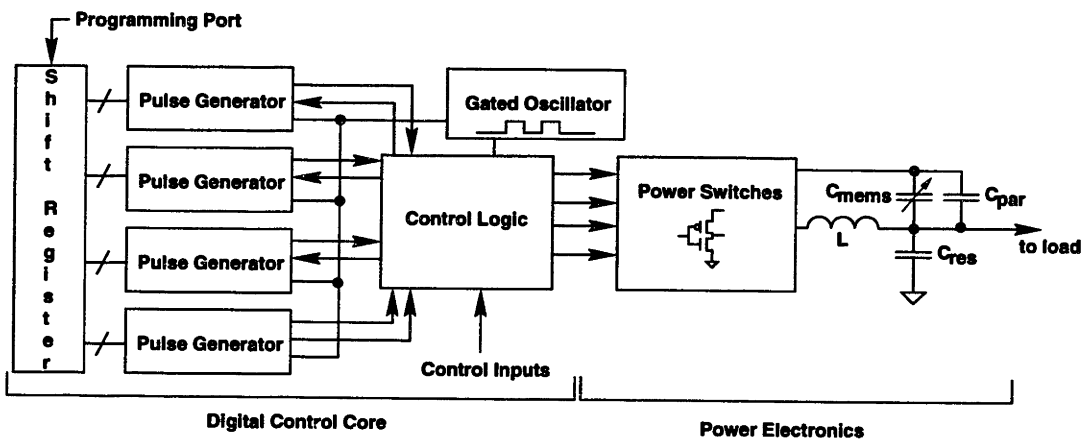


Figure 3.1: System Architecture (Single Conversion Phase)

The controller architecture is described as a novel DC-DC converter topology for several reasons. These will become more apparent in section 3.2, but are described

here in brief. Figure 3.1 presents the controller architecture. This is a novel architecture because the input filter (L and C_{res}) also are the output filter. The DC output voltage and input voltage are equal to the voltage across C_{res}.

The shift register is used to program the timing values into the digital control core. Pulse generators interact with the control logic to produce timing pulses which drive the power electronics section. The power electronics performs the energy harvesting through the conversion cycle described in chapter 2. The control inputs tell the digital core when the MEMS transducer is at maximum and minimum capacitance.

3.1 Mathematical Modeling

The system was modeled in MATLab using the Simulink package. Figure 3.2 is a block diagram representation of the model. The state variable elements are represented by simple integrators. The required timing pulses are generated by the timing block. The power switches are modeled as ideal switches with added loss terms in the switch set. The vibration source is included in the model through the variable capacitance. The capacitance is assumed to be sinusoidal. Since the MEMS device is tuned to the desired frequency [11], it has a bandpass frequency characteristic which aids to help filter out broadband vibrational modes. Further, the system operates electrically tuned to the desired vibration frequency via the C_{max} and C_{min} timing pulses, so these alternate vibrational modes are further filtered out. In the present controller, these pulses are external control signals, but in chapter 4, a feedback controller capable of performing this filtering will be described.

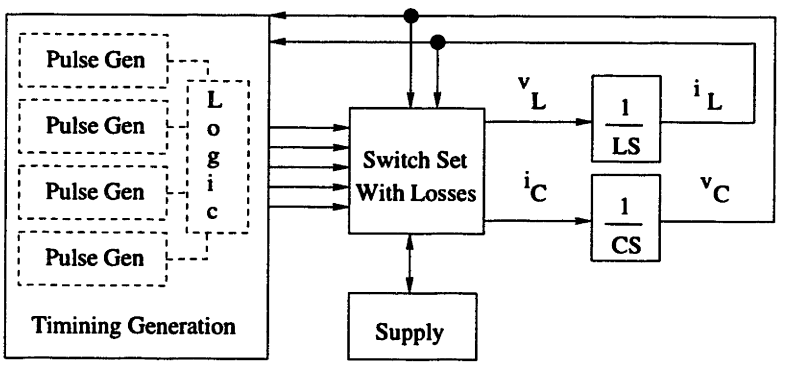


Figure 3.2: Block Diagram of MATLab Model

The timing values for the model are near ideal and based on direct examination of the state variables. Further, the assumption is made that it is known when the MEMS device will achieve maximum and minimum capacitance. This assumption is

carried through in the physical design of the controller. It is then possible to generate the timing pulses by first programming the controller and then triggering it off of the C_{min} and C_{max} control inputs.

The values needed to program the controller may be derived directly from examining the topology of the system during each state. Figure 3.3 shows the circuit configuration during each state, along with a simplified version of the power electronics subsystem (modifications to this basic circuit will be discussed in sections 3.3), and the state variable waveforms over a conversion cycle.

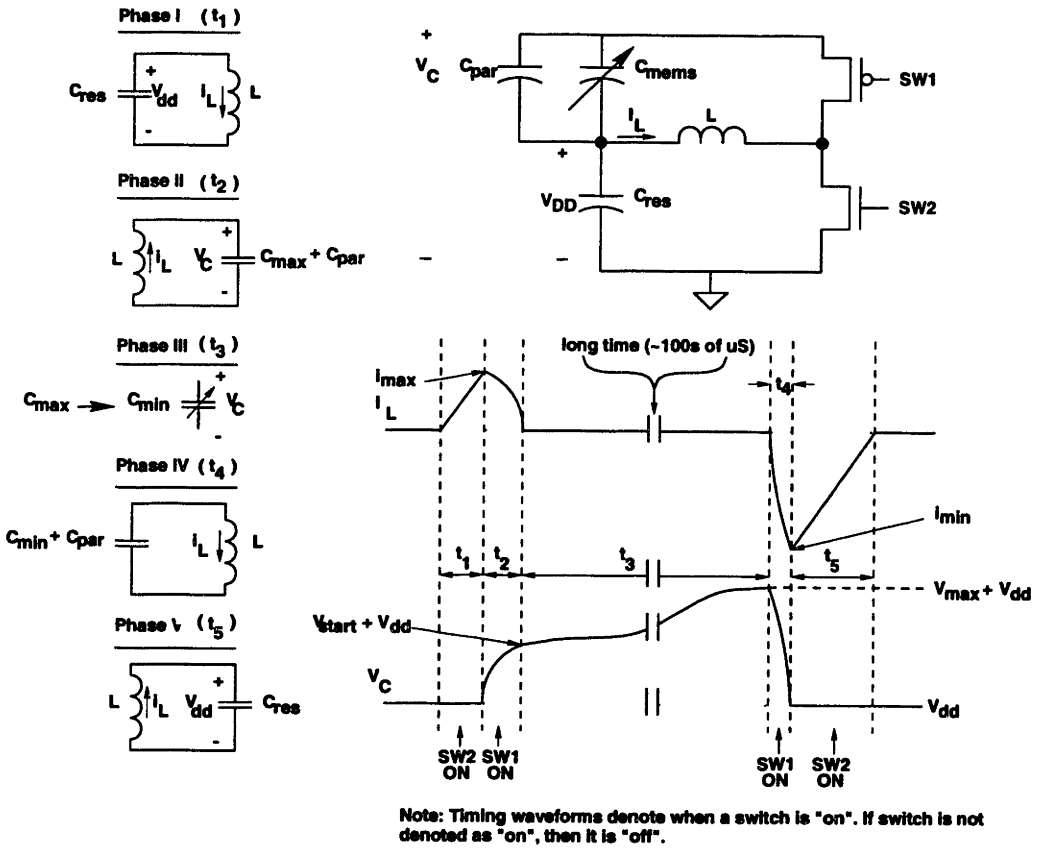


Figure 3.3: Circuit Topology in Various States During Energy Conversion

3.1.1 Derivation of Timing Values

During t_1, SW1 is off, SW2 is on, and C_{res} charges the inductor. C_{res} will be large, and so we may approximate it as a voltage source. (C_{res} is sized to satisfy filtering requirements of the input/output voltage, and therefore is approximately 1μF). The

inductor current ramps up linearly ($\sqrt{LC_{res}}$ is large). The goal is to charge the inductor up to some maximum current, i_{max}, such that when it transfers energy to the capacitor combination, V_{start} from Figure 2.3 will result across the capacitor combination. Recall that V_{start} is the required initial voltage when $C_{mems} = C_{max}$ that will result in a final voltage of V_{max} when C_{mems} moves to and reaches C_{min}. We can therefore calculate i_{max}, V_{start}, t_1, and t_2.

NOTE: In figure 3.3, the initial voltage is $V_{start} + V_{dd}$ and the maximum voltage is $V_{max} + V_{dd}$. In the above paragraph and discussion to follow, these are taken to be V_{start} and V_{max}. This is acceptable because the energy conversion process is based on ***changes*** *in voltage, so the common reference, whether V_{dd} or GND is not important, and making the reference zero volts rather than V_{dd} simplifies equation analysis.*

In the following analysis the system is treated as lossless. While this is not valid assumption for power dissipation analysis, it is acceptable approximation when deriving timing values because damping caused by the system resistive components is small. Sizing of the inductor will be discussed in section 3.2.2.

V_{max} is a known system constraint, as are C_{max}, C_{min}, and C_{par}. Through charge conservation we know

$$(C_{max} + C_{par})V_{start} = (C_{min} + C_{par})V_{max} \tag{3.1}$$

Therefore

$$V_{start} = \frac{(C_{min} + C_{par})}{(C_{max} + C_{par})}V_{max} \tag{3.2}$$

i_{max} may be calculated by looking at the energy transfer relationship between the inductor and capacitor combination during t_2 and using equation 3.2.

$$\frac{1}{2}Li_{max}^2 = \frac{1}{2}(C_{max} + C_{par})V_{start}^2 \tag{3.3}$$

$$i_{max} = \sqrt{\frac{1}{L}\frac{(C_{min} + C_{par})^2}{C_{max} + C_{par}}}V_{max} \tag{3.4}$$

Phase I : t_1

t_1 may be found by using the linear inductor current relationship to V_{dd}. (Again, the relationship is approximately linear because C_{res} is large, so for the length of time the circuit is in this state its behavior looks linear.)

$$i_L = \frac{V_{dd}t}{L} \tag{3.5}$$

$$t_1 = \frac{i_{max}L}{V_{dd}} \tag{3.6}$$

$$t_1 = \sqrt{\frac{1}{L}\frac{(C_{min}+C_{par})^2}{C_{max}+C_{par}}\frac{V_{max}}{V_{dd}}L} \tag{3.7}$$

Phase II : t_2

During t_2, the inductor transfers its energy to the capacitor combination. t_2 is simply one quarter of the resonant L($C_{max}+C_{par}$) period. The inductor is acting as a current source to charge the capacitor combination. The use of the inductor in this way allows the creation of a V_{start} which is greater than V_{dd}.

$$t_2 = \frac{\pi}{2}\sqrt{L(C_{max}+C_{par})} \tag{3.8}$$

Phase III : t_3

t_3 is not a timing pulse that is programmed into the controller. Rather, the controller receives a pulse at $C_{mems} = C_{max}$ telling it when to start t_1. After generating t_1 and then t_2, the controller waits for a pulse telling it that $C_{mems} = C_{min}$, and proceeds to generate t_4 and t_5 pulses. In the fabricated controller, the MEMS transducer capacitance control pulses are inputs to the digital core. In chapter 4 a feedback controller capable of generating the C_{max} and C_{min} control pulses is presented. For completeness, t_3 is derived as one half of the vibration period

$$t_3 = \frac{T_{vib}}{2} = \frac{1}{2f_{vib}} \tag{3.9}$$

Phase IV : t_4

Once the MEMS capacitor plates have moved to $C_{mems} = C_{min}$, the system is ready for energy harvesting to be performed. During t_4, $C_{min}+C_{par}$ transfers $\frac{1}{2}(C_{min}+C_{par})V_{max}^2$ joules of energy to the inductor. The time required to do this is

$$t_4 = \frac{\pi}{2}\sqrt{L(C_{min}+C_{par})} \tag{3.10}$$

Phase V : t_5

After phase IV has been completed, the inductor returns the energy to the source for t_5. The time it takes to complete this final step of the energy conversion process is determined by the final value of current that the inductor is charged to, i_{min}.

$$i_{min} = \sqrt{\frac{C_{min}+C_{par}}{L}}V_{max} \tag{3.11}$$

$$t_5 = \frac{i_{min}L}{V_{dd}} \tag{3.12}$$

$$t_5 = \sqrt{(C_{min} + C_{par})L}\frac{V_{max}}{V_{dd}} \tag{3.13}$$

Finally, it should be noted that a basic assumption mentioned in the preceding analysis was that, when the controller was active during t_1, t_2,t_4, and t_5, the value of the MEMS transducer capacitance was constant. This is a good assumption because the vibration period is approximately 400μs, while the controller timing pulses are $\sim$ 500ns. Since the transducer plate motion is assumed to be sinusoidal, the change in capacitance around the maximum and minimum values will be small during t_1, t_2,t_4, and t_5.

3.2 Power Electronics

This section describes the design issues relevant to the power electronics subsystem. These deal mainly with optimization of the system components for minimum power loss. First, issues of startup charge for the energy conversion process will be discussed. This leads into an optimization for the parallel capacitance of section 2.1.3. An approach to power switch sizing is presented based on previous work [4], but with modifications appropriate for the energy converter system. Finally, issues related to functionality of the circuit topology will be discussed.

3.2.1 Q_o

As stated in the energy conversion chapter, there is a constraint on how much initial charge we may place on the parallel capacitor configuration. This has to do with the losses accrued by the inductor and power switches. Each element will have some series resistance, R_L and R_{ds} respectively, which will limit the maximum value of current that the inductor will charge to, and therefore the initial energy we can place into the system. This maximum current will be $I_{Lmax} = V_{dd}/(R_L + R_{ds})$, where V_{dd} is the supply voltage across C_{res}. We also must take into consideration the assumption that C_{res} is very large. In reality, the supply can be modeled as some large, but finite, capacitor (and resistor), so if we choose L or C_{res} too small, the assumption of a linear inductor current characteristic during t_1 and t_5 is not valid. Increasing C_{res} means that the filter volume becomes physically large. As we increase L, the linear charging behavior becomes a better approximation, but higher value inductors will also exhibit a higher series resistance, limiting the value of I_{Lmax}. In either case we need to be sure that we can supply the necessary initial charge Q_o to the capacitor combination. In fact, the other system requirements for C_{res} as described in section 3.3.3 ensure that the inductor current during t_1 and t_5 looks linear.

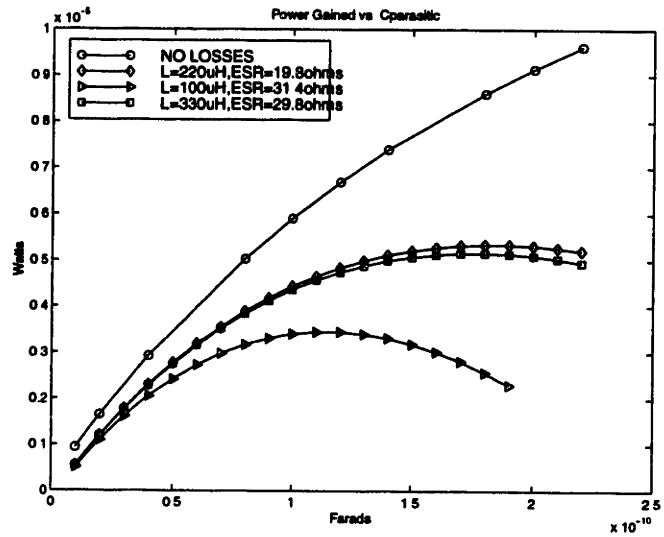


Figure 3.4: Optimization Curves for Various L and C_{par}

3.2.2 Parallel Capacitor Optimization

From the above discussion, we see that a desirable scenario is to have a large valued inductor with low series resistance. Since the main job of the inductor is to act as a charge source for the capacitor combination, ($C_{combo} = C_{MEMS} + C_{par}$) we take this into account when sizing L. The limits on the MEMS capacitor, C_{min} and C_{max} are fixed, so what we really need to look at is the relationship between L and C_{par}. To do this, the mathematical model of the system, discussed in section ?? , was used along with some real inductor specifications from an electronic parts distributor's catalog. L was modeled as the nominal values from the catalog, and C_{par} was varied to observe performance. Figure 3.4 shows optimization curves for C_{par} for three values of L.

The peaking in the curves show that after some optimal value of C_{par} we begin to gain less energy. This is due to the increased series losses in the inductor and power FETs (which are also included in the MATLab model). We arrive, therefore, at $C_{par} = 180pF$ and $L = 220\mu H$. The model suggests we can expect to get $\sim 5\mu W$ out of the converter. HSpice simulations of the controller predict that it will use $\sim .5\mu W \rightarrow 1\mu W$, so overall we expect to have an average of $\sim 4\mu W$ available for the load for a single phase of energy conversion. The lossless case for larger values of C_{par} asymptotically approaches the value predicted by the discussion of section 2.1.3, $E'_{chrgcons} = \frac{1}{2}(C_{max} - C_{min})V_{max}^2$. This condition was included to verify the accuracy of the model.

3.2.3 Design of Switches

A major source of loss associated with the converter relates to the power FETs, so it is desirable to size them appropriately. A straightforward approach to optimizing transistor width has been developed [4]. (In general we assume that the FETs will be sized to minimum or near minimum lengths and appropriate widths. The choice between minimum or non-minimum gate length depends on the voltages in the system and whether or not short channel effects are a concern.) This approach models the FETs as being in the linear region during operation with some constant gate drive V_{GS}. This model forms a useful basis for analysis of the converter, but will offer better results if we modify the assumptions to provide for a gate drive that varies over the course of one conversion period. This modified model more accurately reflects our converter as shown in Figure 3.3, where the PFET gate drive is given by V_C.

The on resistance of a FET in the linear region is given by:

$$R_{DS} = \frac{L_G}{\mu C_{ox} W_G (V_{GS} - V_T)} \tag{3.14}$$

where L_G is the gate length, μ is the channel mobility, W_G is the gate width (which we seek to optimize), C_{ox} in the oxide capacitance per unit area, and V_T is the threshold voltage. The power dissipated by a FET over the course of one switching period (assuming that the FET switches twice per period) is given by:

$$P_{tot} = P_{switch} + P_{series} \tag{3.15}$$

where

$$P_{series} = I_{rms}^2 R_{DS} \tag{3.16}$$

$$P_{switch} = C_{GS} V_{GS}^2 f + C_{dyn} V_{dyn}^2 f \tag{3.17}$$

Here, I_{rms} is the rms current through the switch over one switching period, C_{GS} is the gate capacitance, V_{GS} is the gate drive voltage, C_{dyn} is the capacitance of the switched drain or source node, and V_{dyn} is the voltage that the dynamic node is switched at. It is possible as in [4] to make some simplifications to the power loss optimization problem. The first is to to combine the gate and drain or source capacitances if the drain and source extensions are roughly equivalent to the gate area. Also, if a further restriction on the system is that the gate drive voltage is always the same and is roughly equal to the switched voltage at the dynamic node, then the two terms in equation 3.17 may be linearly combined to produce a simpler equation. For our system, this is not true. If we refer again to Figure 3.3, we see that the PFET causes special conditions to occur. For proper operation, we must ensure that during t_3 the PFET is off. This requires that there be a level converter driving its gate since the dynamic capacitor voltage, V_C, will start at V_{DD} and increase to some V_{max}. This means that the PFET has a much higher gate drive during t_4 than during

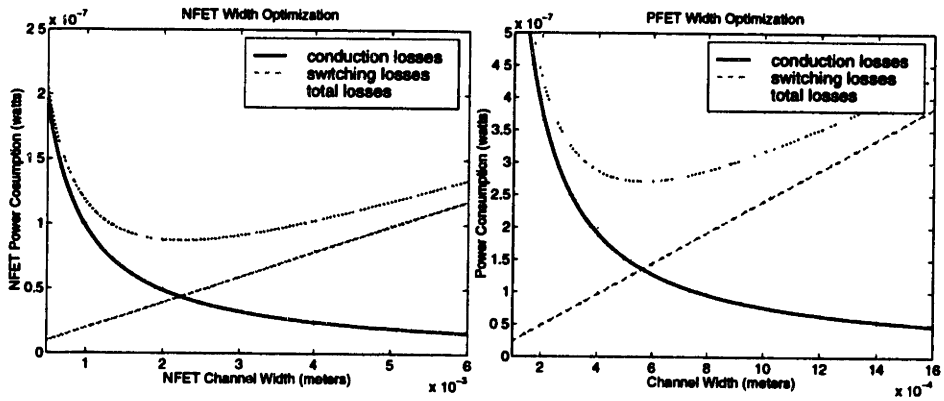


Figure 3.5: Power FET Sizing Curves

t_2. Figure 3.5 shows optimization curves for the power switches with the difference in gate drive accounted for.

We can see that unlike conventional complementary switcher designs [7, 4] the PFET is actually smaller than the NFET for minimal power losses. This is due not only to the fact that the PFET experiences higher gate drive for half of its switching duties, but also because of the fact that the NFET passes higher rms currents, since it is charging and discharging the inductor alone, which takes more time than charging and discharging the capacitor combination. One characteristic to note is that the curves of Figure 3.5, especially the NFET, exhibit shallow troughs. This means that we can take a value away from the absolute optimum and save in area at a very low power cost. Also, for the PFET, some of the power we "lose" by going to a shorter width is actually saved because the level converters which we will need to drive the PFET can become smaller. (The level converter losses were not included in the optimization program because their sizing is dependent on the PFET's width). The chosen sizes for each switch (W/L) are NFET=1399/1, PFET=538/1.

3.3 Other Considerations for the Power Electronics Circuitry

As previously mentioned, Figure 3.3 does not represent the full power electronics circuit, but serves as a functional model. The topology used has some practical consequences which must be accounted for.

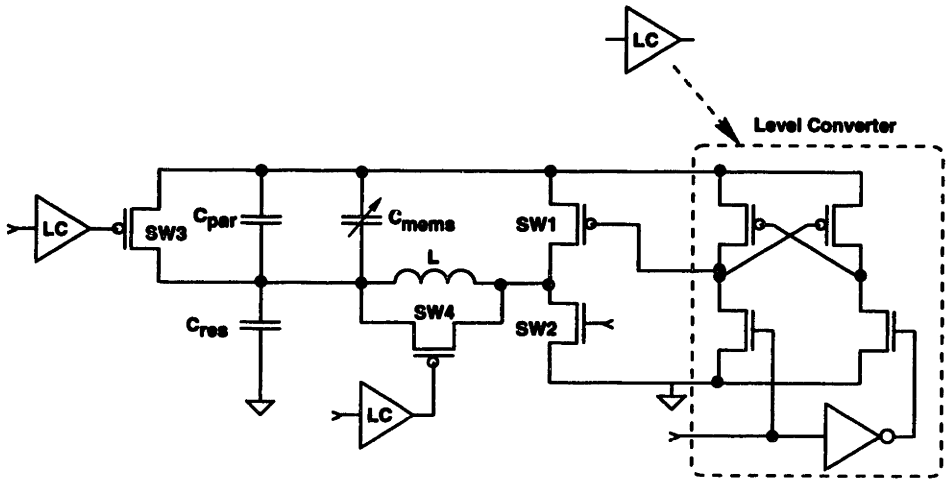


Figure 3.6: Complete Power Electronics Circuit (Single Phase)

3.3.1 Reactive Ringing

After the inductor charges $C_{mems} + C_{par}$ (during t_2), SW1 is turned off, and SW2 remains off. If the inductor current is not zero at the end of t_2, then the RLC formed by the inductor and the switch drain parasitics will reactively ring until this remaining energy is dissipated. This will cause conduction in the drain-substrate diodes, which will inject carriers into the substrate. This could lead to latch-up. For this reason, a shunt switch, SW4, has been added across the inductor as shown in Figure 3.6. This shunt is turned on at all times except during t_1, t_2, t_4, and t_5. It acts to dissipate any left-over energy in the inductor and prevent undesirable ringing. In addition, guard rings were used to surround the power switches as a whole, and well (for PMOS) and substrate (for NMOS) plugs were extended to provide some degree of isolation of one switch from another. Figure 3.7 shows the layout of the power electronics including the level converters. Figure 3.8 is a zoom in to show the guard ring and isolation structures. Note that both Figures 3.7 and 3.8 do not show all of the connections explicitly. Metal 2 has been left out of the figures for ease of visibility.

3.3.2 Capacitor Resetting

If the $C_{mems} + C_{par}$ capacitance is not completely discharged during t_4, there will be some initial voltage across it when it is charged at the next cycle during t_2. This incomplete discharge may occur if t_4 is not ideal. The effect of this residual voltage will be to accumulate over many cycles, and will eventually lead to dangerous voltage levels being developed in the system. For this reason, a clamp transistor has been added across the $C_{mems} + C_{par}$ pair. This PFET, SW3, also shown in Figure 3.6, is active for

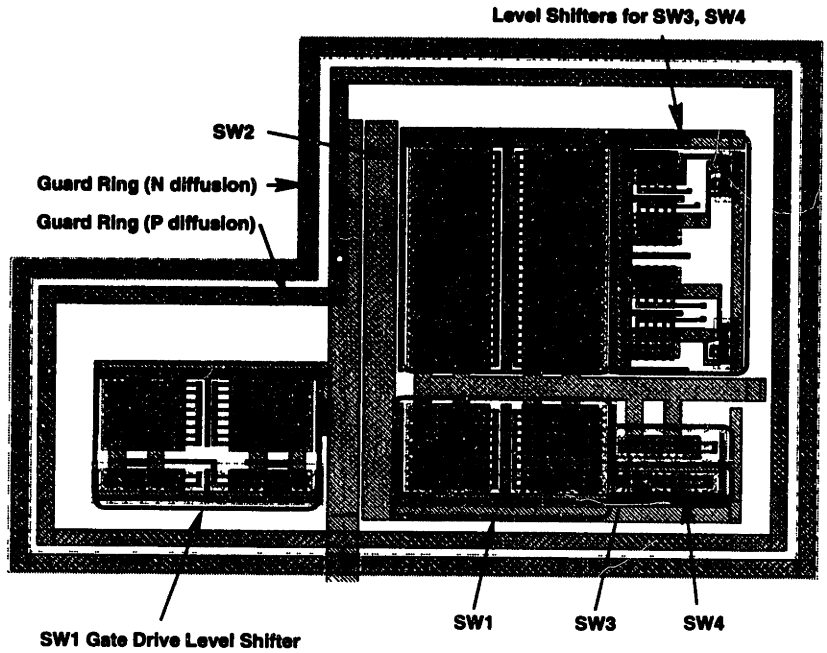


Figure 3.7: Layout of the Power Switches Section

the time after t_5 and before the next conversion cycle's t_1. Figure 3.9 demonstrates the functionality of SW3. A non-ideal t_4 results in the capacitor combination not being completely discharged. This voltage is held on the capacitance until after t_5, when the shunt switch, SW3, is turned on to dissipate the remnant voltage.

3.3.3 Sizing C_{res}

Since C_{res} acts as the charge source for the system, it must be capable of supplying enough charge for the controller and energy conversion startup process without exhibiting unacceptable droop. A simple calculation may be done to determine the appropriate value of C_res.

First we examine the lower ripple peak, the *droop* voltage. If we look at the amount of charge that C_{res} must supply to the system, it has two dominant components.

$$\Delta Q_{droop} = \Delta Q_L + \Delta Q_C \tag{3.18}$$

ΔQ_L is due to the current that charges up the inductor during t_1 and ΔQ_C is associated with the rms current needed to run the control core. ΔQ_L may be calculated as

$$\Delta Q_L = \int_0^{t_1} i_L \, dt = \int_0^{t_1} \frac{V_{dd} t}{L} \tag{3.19}$$

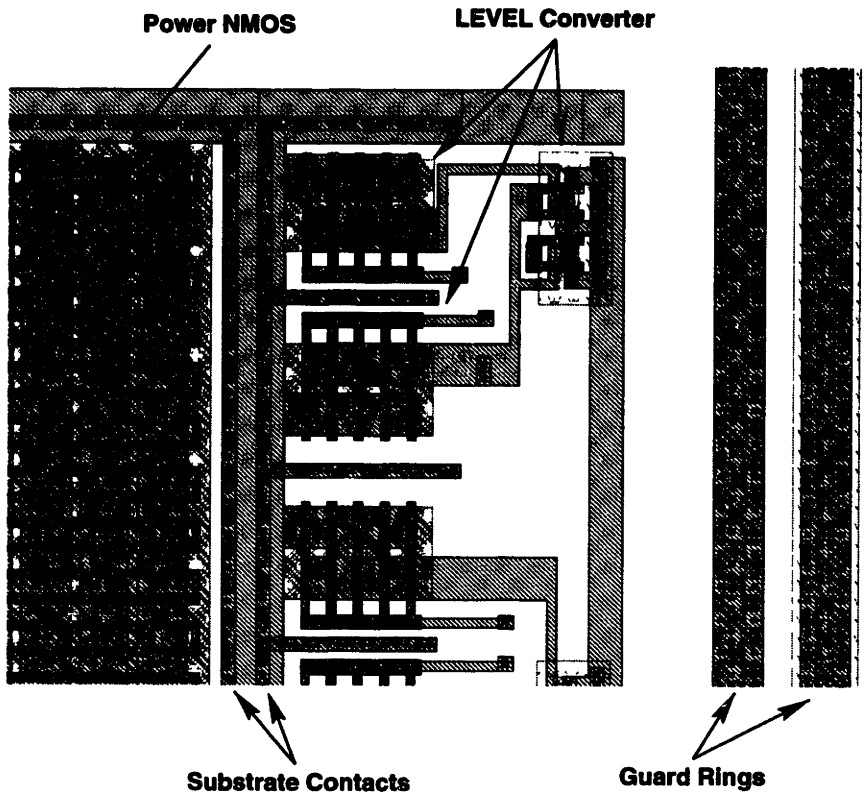


Figure 3.8: Zoom in showing guard rings and isolation structures

therefore

$$\Delta Q_L = \frac{V_{dd} t_1^2}{2L} \tag{3.20}$$

In a similar way we can calculate Q_C.

$$\Delta Q_C = \int_0^{T_{vib}} i_{core-rms}\, dt = i_{core-rms} T_{vib} \tag{3.21}$$

We can use the charge relationship $\Delta Q_{tot} = C_{res} \Delta V$ and solve for C_{res} for a given droop voltage, ΔV_{droop}.

$$C_{res} = \frac{(\frac{V_{dd}}{2L} t_1^2 + i_{core-rms} T_{vib})}{\Delta V_{droop}} \tag{3.22}$$

Using values from the actual system, $V_{dd} = 1.5V$, $i_{core-rms} \approx 500nA$, $L = 220\mu H$, $t_1 = 800ns$, $T_{vib} = 400us$, for a $\Delta V_{droop} = 10mV$, we arrive at $C_{res} = 240nF$.

The upper peak of the ripple voltage, in the absence of load and regulation, is determined by the amount of energy we get out of the system.

$$\Delta Q_{peak} = \int_0^{t_5} i_L\, dt = \frac{V_{dd} t_5^2}{2L} \tag{3.23}$$

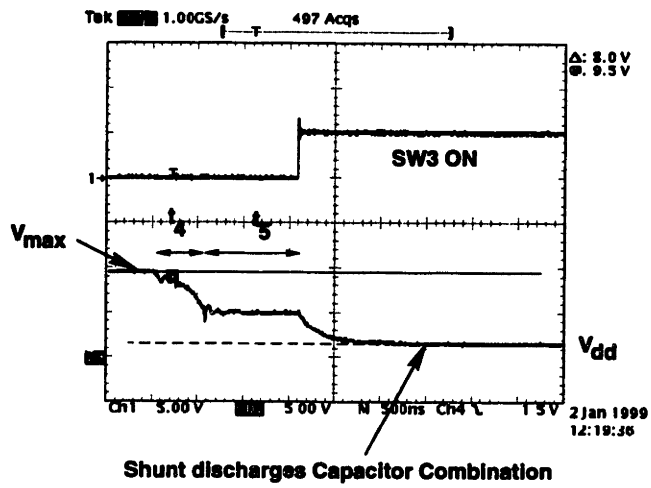


Figure 3.9: SW3 Discharges the Capacitor Combination for Non-Ideal t4

Again, using the charge-voltage relationship for a capacitor, $C_{res} = \Delta Q_{peak}/\Delta V_{peak}$. For the implemented circuit, $t_5 \approx 1\mu s$. For a $\Delta V_{peak} = 10mV$, we require $C_{res} = 340nF$. This upper bound sets a more conservative requirement on C_{res}, and so should be the equation used for sizing. We will revisit the charge source requirements of C_{res} in chapter 4, when additional strain is placed on ΔV_{droop} because of the use of feedback.

3.4 Digital Controller

It is straightforward to implement a programmable digital controller based on the MATLab model. Because the important system parameters $L, C_{par}, C_{max}, C_{min}$ and f_{sw} are known, we can realize the timing pulses through the use of a programmable delay line. The basic block diagram for a single pulse generator is shown in Figure 3.10.

This is a hybrid delay line approach which combines the area savings of a fast clocked counter approach with the resolution flexibility of a tapped delay line.

In a tapped delay line approach, N buffer delays are serially connected and tapped off into an N-to-1 multiplexer, with $\log_2 N$ selects. For large N, the multiplexer becomes area intensive. If we represent the power consumed by one buffer as $CV^2 f_{tap}$ where C is the switched capacitance, V is the supply voltage and f_{tap} is the switching frequency, then the total power consumed by the delay line is [1]:

$$P_{tap} = KCV^2 f_{tap} \tag{3.24}$$

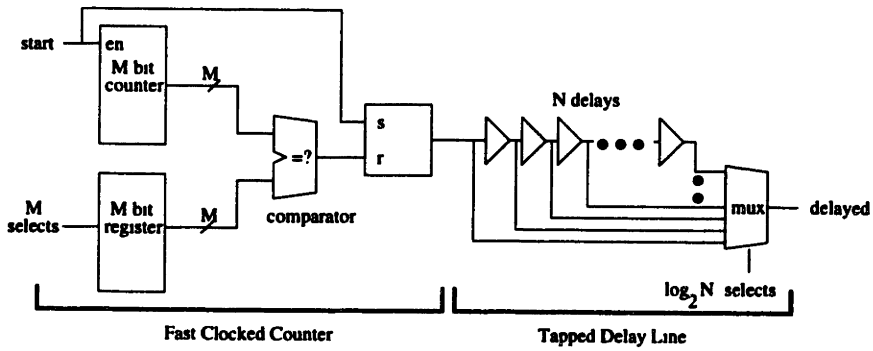


Figure 3.10: Hybrid Delay Line

Where K is the number of delay buffer through which the pulse propagates. This assumes that we have a means to vary the length of the delay line to prevent any buffers downstream from the desired tap point from switching unnecessarily. This can be accomplished through the addition of appropriate gating logic.

Another approach, shown in the left half of Figure 3.10 is to use a fast clocked counter and comparator to implement the delay line. Now we have an $M = \log_2 N$ bit counter clocked at $f_{count} = Nf_{tap}$. We assume that the supply voltage does not need to be raised to support this higher frequency. This is true for our converter, where V_{dd} is constrained by a lower bound not by the control electronics requirements, but by the required charge necessary for conversion as described in the Q_o section. As long as the counter can be clocked at the required f at the designated V_{dd}, our assumption is valid.

We can see that the fast clocked counter approach is more power intensive than the tapped delay line, even without taking into account the power that will be required to generate a clock. The buffer in a tapped delay line is made up of two inverters sized to give the appropriate delay. The counter is made up of flip flops which we will assume have the equivalent of $2r$ inverters of switching capacitance, or r times that of a buffer. The activity factor for the counter (how often the flops change value and consume energy) is dependent on the value of the delay we wish to implement. For an M bit ripple counter, we can assume that the number of times that the LSB will switch is K, where K is the desired final value of the delay. The second LSB would switch $\frac{K}{2}$ times, and, continuing this pattern, the MSB $\frac{K}{2^M}$ times. Of course, all of these are rounded down to the nearest whole number, so that if the fraction is less than one, that bit switches zero times. So, combining all of the bits we arrive at:

$$P_{count} = K(\frac{1}{2} + \frac{1}{4} + \ldots + \frac{1}{2^M})rCV^2Nf_{tap} \tag{3.25}$$

As M increases the series converges to one, and we then have:

$$P_{count} = rKNCV^2f_{tap} = k'P_{tap}; \quad k' > 1 \tag{3.26}$$

One advantage of this approach is that area is saved since the M bit counter and comparator logic will take up much less area than an $N = 2^M$ buffer delay line and N-to-1 mux. Another advantage is that a simple external crystal oscillator may be used to clock the counter, providing an accurate time step. The tapped delay line will suffer more from device mismatch and unbalanced loading, requiring calibration to generate delays of a known value.

Figure 3.10 shows the hybrid approach where both a counter and tapped delay line are used. The counter provides large time steps and the delay line is used to fine tune the delay. The requirement for this approach to work well is that the sum of delays in the delay line must be approximately equal to one clock period. One way to ensure this would be to use the delay line itself as a clocking mechanism for the counter. After a number of large time steps represented by the period of the counter occur, the line can be tapped to the appropriate fractional period. This requires that the buffer delays be reasonably well known and matched to some known time performance criteria, or that some form of calibration be performed.

Once the delay line has been designed, it may be duplicated four times to generate the four necessary timing pulses of duration t_1, t_2, t_4 and t_5. The individual pulse generators are joined by control logic. The delay line has been designed for 10 bit input words, four bits for the delay line fine adjust, and 6 bits for the fast clocked counter coarse adjust. The counter clock frequency is nominally 20MHz based on the values chosen for L and C_{par}. The control words for each pulse generation stage are programmed in through a 40 bit serial shift register interface. The first ten bits control t_1, the second ten t_2, the third ten t_4, and the final ten t_5. The number of bits for each stage was chosen based on the expected range of frequencies available from the counter clock design to be described in the following section, given a range of reasonable values for L and C_{par}.

3.5 Synchronous Gated Oscillator

Since the hybrid counter delay line architecture was chosen, it is necessary to generate a clock signal for the counter. One possibility is to bring in an external, high frequency clock, and then gate the signal on chip to save power. However, the power that would be wasted in the IO pads would be constant and significant. Another possibility would be to integrate a high frequency, low power oscillator on-chip, such as in [13]. In this case, the oscillator would consume static power, which, because of the high frequency (20MHz) required, would be large (many μWs). The use of an integrated low power, *low* frequency oscillator to generate a reference clock for a novel delay locked loop (DLL) will be discussed in chapter 4.

Since the pulsewidths generated by the controller are small (100s of ns) compared to the vibration period (400μs), it would be beneficial for power savings to only have

the clock operational when timing pulses are being generated. Also, it is desirable, for optimal conversion, to have the rising edge of the clk signal synchronous with the start of conversion C_{max} pulse, and this would not be possible with an external oscillator. (An external oscillator would have no information about when C_{max} occurs and so would be asynchronous with respect to C_{max}.) Instead, a resettable, synchronous, gated oscillator was used in order to conserve power. The oscillator circuit is shown in Figure 3.11. M1 serves to assure that the clk will have a rising edge synchronous

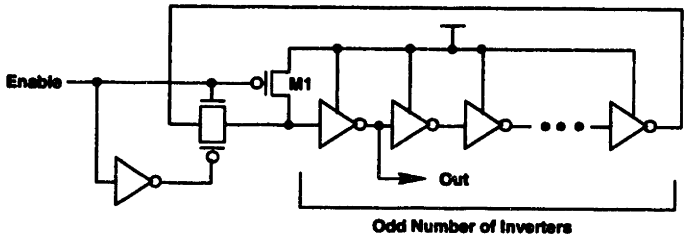


Figure 3.11: Gated Oscillator Circuit

with the enable signal. The enabling gate assures that oscillations will only occur during timing pulses, reducing power consumption. The clock period may be well defined by using a digital calibration scheme which compares its period to that of a reference and adjusts the number of inverters in the ring accordingly. Such a technique is discussed in section 4.4.1. Figure 3.12 shows output waveforms from the fabricated IC, demonstrating the oscillator's behavior. The oscillator is designed to

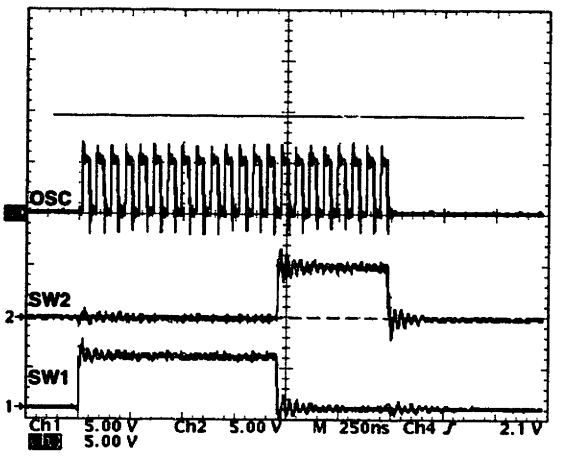


Figure 3.12: Gated Oscillator Waveform

have a nominal frequency at $V_{DD} = 1.5V$ of 20MHz. The period of Figure 3.12 is 18MHz at $V_{DD} = 1.5V$. This is acceptable given the full scale range of the counter in the hybrid delay line.

To give flexibility, the possibility of using an external oscillator was created by multiplexing the on-chip gated oscillator output with an off-chip oscillator input. The gated oscillator output is brought out to pin OscOut for diagnostic purposes. An external input, ClkIn, allows a crystal generator to act as an input to the chip. The ClkSel control input, when high, selects the off-chip oscillator, When ClkSel is low, the on-chip gated oscillator is used. OscEn must be high for the gated oscillator to function. The OscEn and ClkSel pins were made separate functions for flexibility in testing.

3.6 Fabricated IC Testing and Results

Figure 3.13 is a chip photograph of the fabricated controller.

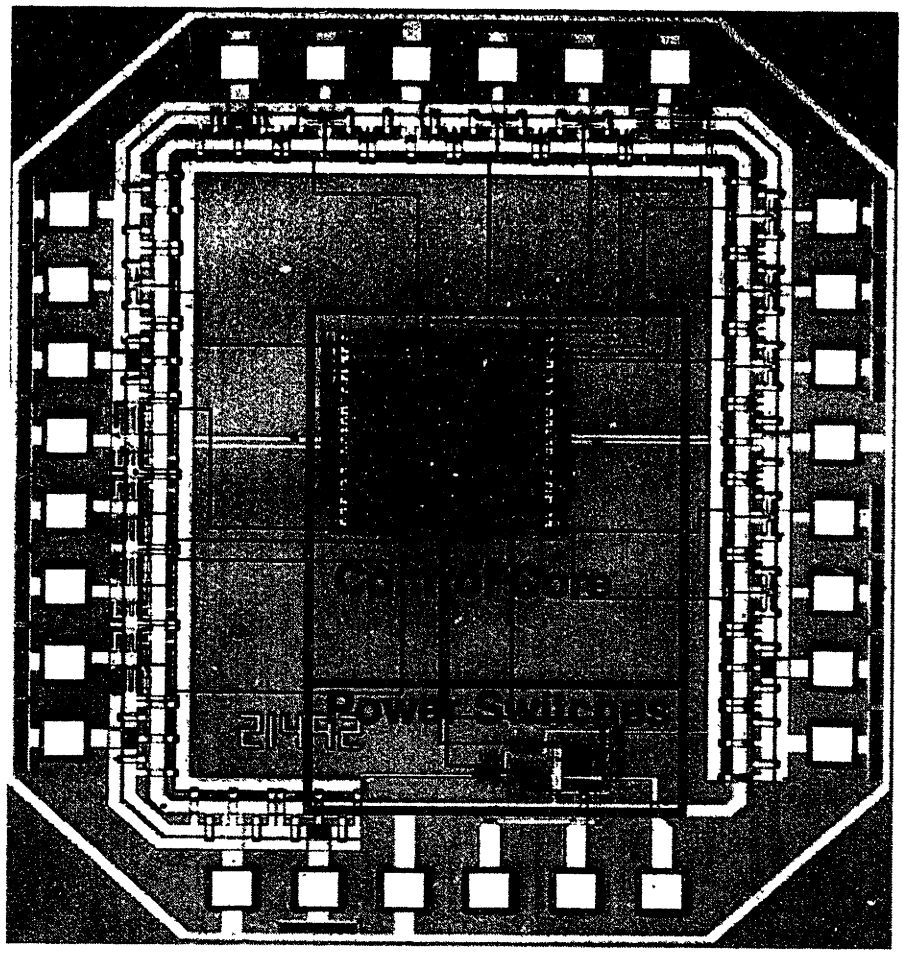


Figure 3.13: Chip Photograph

The programmable controller and integrated power switches have been implemented in a $0.6\mu m$ CMOS process. Presently the MEMS transducer is in fabrication, so test circuitry was designed to emulate its effects on the system. Table 3.1 presents the results of testing for a single phase of energy conversion; meaning one side of the variable capacitance. Since the MEMS structure creates two capacitors 180 degrees out of phase with each other, we can double the numbers to account for both phases of conversion.

Area	2163μm x 2554μm
Transistor Count	2661
Power NFET	W/L = 1399/1
Power PFET	W/L = 538/1
Process	0.6μm CMOS
Core Power	500nW (f_{vib} = 2.5kHz, V_{dd} = 1.5V)
Switch Loss Power	3.87μW (V_{max} = 8.0V)
Predicted Power Out	4.29μW (V_{max} = 8.0V)

Table 3.1
Controller Specifications (half circuit)

Figure 3.14 shows the test circuit used for system testing. Since the MEMS is not presently available, a constant value capacitor of value $C_{test} = C_{par} + C_{max}$ is used in its place. The NFET serves to emulate the behavior of the MEMS device. It is turned on during t_3. This causes the DC test source, V_{MEMS}, to be switched onto C_{test}, which allows testing of the level converters used in the power electronics section.

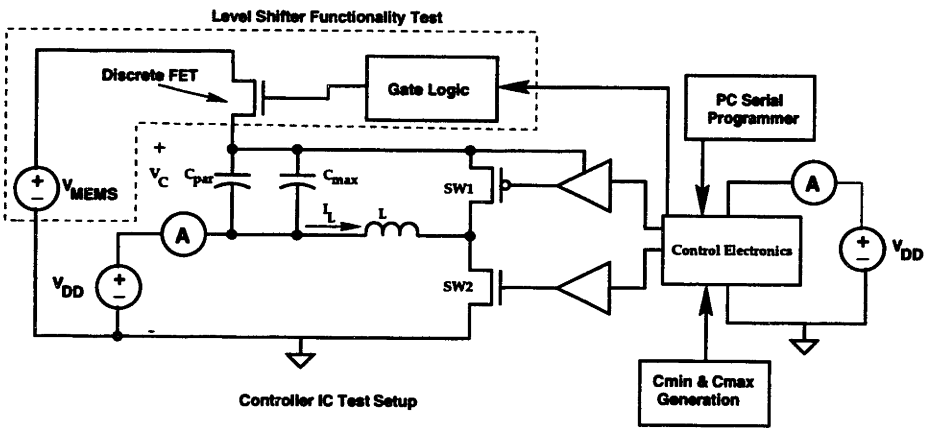


Figure 3.14: Test Circuit for Fabricated Controller

Losses may also be measured. The layout of the controller chip was designed such that the digital core supply could be separated from the charge reservoir, so that losses in the power electronics subsystem could be measured separately from the losses associated with the digital core. Losses in the power electronics are measured by simply sending energy through the system without switching in the V_{MEMS} test source. This effectively moves the charge through the system with no external energy source, and measures the electrical losses through the inductor, power switches, and level converters. Table 3.2 shows results of testing for several different values of desired V_{max} (and therefore different values of V_{start}). Recall that for a given change in capacitor voltage there is a corresponding initial voltage which must be present on the capacitor before t_3. (This is V_{start} from Figure 2.3). Since the energy gained is a function of the change in voltage developed by the MEMS plate motion, it is desirable to have a large change in voltage during the conversion cycle. The maximum final voltage is restricted by process constraints. The power electronics were tested for several values of V_{max} in order to demonstrate the gains of larger starting voltages.

V_{max}	Digital Core Losses	Power Set Losses	Converted Power (Predicted)	Power Out (Predicted)
8V	456nW	3.87μW	8.62μW	4.29μW
9V	471nW	4.95μW	10.91μW	5.49μW
10V	513nW	6.30μW	13.46μW	6.65μW
11V	537nW	7.74μW	16.26μW	8.53μW

Table 3.2
Power Outputs for Various V_{max} (Single Phase)

The system was tested up to a $V_{max} = 11V$. This creates maximum voltages in the system of $V_{dd} + V_{max} = 12.5V$. It may be possible to run the system at higher voltages but this would add more stress to the power switches and may lead to long term reliability problems. However, it is desirable to run the system at the highest possible V_{max} since increasing V_{max} increases the available output power. A limitation on the upper bound of V_{max} will be due to physical limitations of the system: FET breakdown (either gate oxide or reverse bias diode), dielectric breakdown in the transducer, or increased losses through the power electronics, particularly in the level converters whose lower rail drive voltage is 1.5V.

All of the reported measurements are for a single phase of conversion. For both phases the numbers in the table may be doubled, so that, in a full system with a $V_{max} = 10V$, the predicted power available to a load with both phases accounted for would be $P_{out} = 13.3\mu$W.

3.7 Testing Requirements

In testing the controller IC there are three procedures which must be followed.

1. The input C_{max} and C_{min} pulses must be of a shorter duration than t_1 and t_4, respectively. This is because of an RS flip-flop used to initiate the conversion cycle. If the input pulses are longer than allowed, both switches may activate repeatedly, which will cause incorrect conversion. As a precaution, it is therefore a requirement that the C_{min} and C_{max} pulses be <100ns wide. This is easily handled at the test board level.

2. The upper six bits of each ten bit timing pulse control word (as described in section 3.4) cannot all be zero. The requirement is that at least one of these six bits be high. The reason for this is to simplify the comparator design in the fast clocked counter section. This is not viewed as a practical problem. In reality, the timing values needed to generate the pulses will require that the counter count to a value greater than zero since the LC time constants of the system will be greater than the maximum value available from the delay line section of the hybrid pulse generator. However, if the chip is being evaluated, this behavior should be taken into consideration in order that a false failure not be detected.

3. The final, and most important, point to note is that since the timing pulse values are programmed in a serial fashion, and because this is a DC-DC converter with reactive components, it is necessary that the chip be in RESET mode (all flops held in a reset state) during programming. This may be done by holding the POR signal high during programming. If this procedure is not followed, the power switches may be blown because of dangerous voltage and current levels developed due to erroneous switching events created as the programming values propagate down the shift register.

3.7.1 Serial Interface

The timing pulses are programmed into the control core through the use of a three pin serial interface. The control pins are SerClk, SerClr, and SerIn. SerClk is the shift register clock. SerClr clears the contents of the shift register and should be low during programming and normal operation. SerIn is the data input port. Figure 3.15 demonstrates programming via the serial port. The timing values are programmed starting with t_1, then t_2, t_4, and t_5. The LSB for each value is first, and MSB last. The first 4 bits of each code word correspond to the tapped delay line control bits, LSB first, and the last 6 bits to the fast clocked counter control bits, again LSB first. To program in all four timing values requires 40 SerClk cycles. This was accomplished by writing a C program which used a PC parallel port to control SerClk, SerClr, and SerIn.

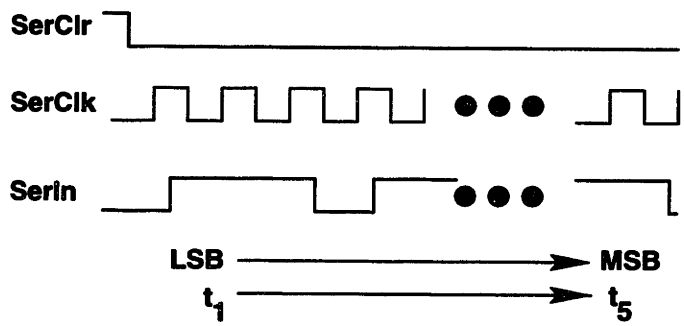


Figure 3.15: Serial Port Timing Waveforms

3.8 Comparisons with Modeling

A great deal of analysis was done before the controller was fabricated. The mathematical model of the system was used along with HSpice simulations of the controller circuitry to ensure that the system was feasible in terms of power output given control overhead costs. This section compares results obtained by the mathematical models, HSpice simulations, and actual measurements.

3.8.1 Power Electronics Functionality

Figure 3.16 is an HSpice plot of a simulation of the power electronics subsystem layout. The vibration period was decreased so that the entire conversion process would be visible. This does not create any problems for comparisons with real data, since this alteration only effects how quickly the MEMS capacitance changes. By carefully setting up the spice deck, a variable capacitance with the desired behavior may be created. This simulation results compare well with the MATLab model. The

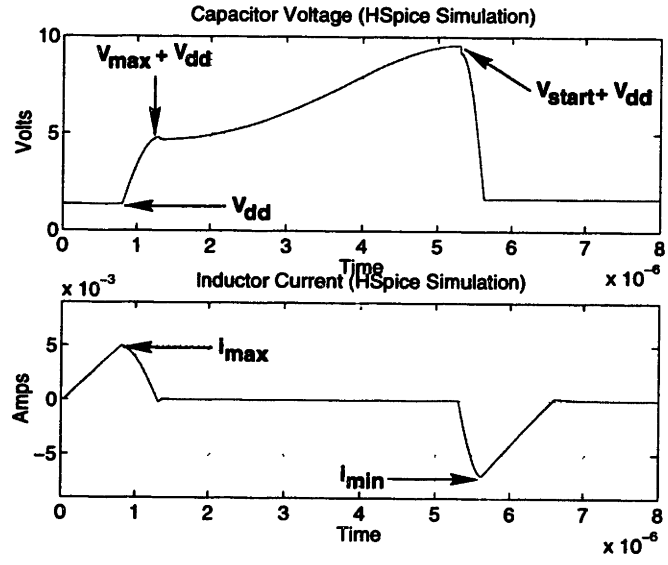


Figure 3.16: HSpice Simulation Results

switch timing pulses were set up to generate the desired value of V_{start} to test the level converters. Loss and output power predictions may be performed to determine the validity of the models.

3.8.2 Loss Estimation

Figure 3.17 shows a comparison between data measured in the laboratory, HSpice simulations, and the system mathematical model. The power numbers are for a single phase of energy conversion. The two models, HSpice and MATLab, were within 20%

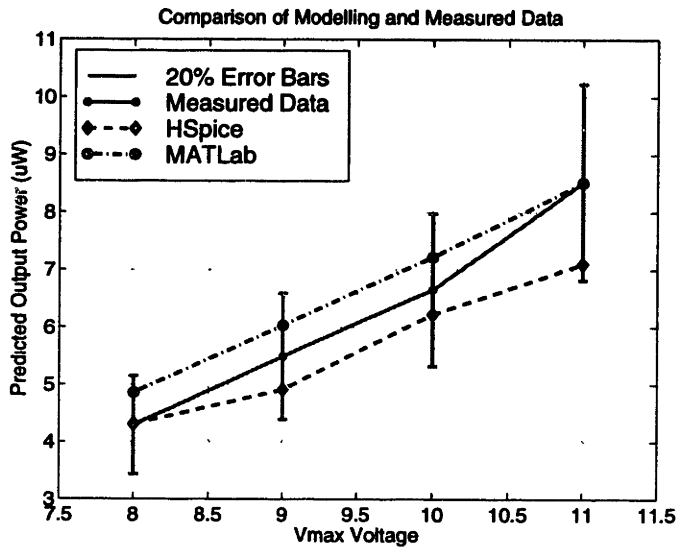


Figure 3.17: Data Comparison

of the measured data for the four data points shown, demonstrating a good correlation between the models and reality. As expected, output power increases with increasing V_{max}. The behavior appears approximately linear as predicted by equation 2.7, which can be modified from an energy equation to a power equation to read

$$P_{out} = \frac{1}{2} f_{vib}(C_{max} - C_{min}) V_{max} V_{start} \tag{3.27}$$

where f_{vib} is the mechanical vibration frequency. It is clear that equation 3.27 has a linear dependence on V_{max}.

Chapter 4

Energy Feedback Controller

The fabricated controller does not have the ability to lock to the capacitor motion automatically. While this could be a requirement placed on the load electronics (if the load is an accelerometer, for instance, then it would have information about the plate motion phase), in order to create a more general energy converter for use by varying low power applications, it is necessary to give the controller the ability to determine when to initiate conversion. This implies some sort of feedback which the controller can use to lock its conversion process to the phase of the plate motion.

Since the energy available for conversion in the system is limited to an ideal maximum value of 42μW, the overhead associated with the controller feedback must be minimized. In fact, as we have seen from section 3.6, even with the system optimized to provide for maximal energy output, the converted mechanical power is only on the order of 10μW, so the restrictions on controller consumption are very strict.

Traditional feedback techniques utilize a reference signal, usually either in the form of a voltage, current, or clock signal, and a comparator to control an output variable. In DC-DC converters, this reference is normally either a bandgap voltage, a current

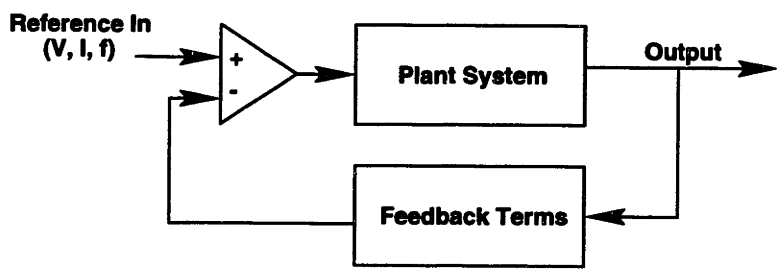


Figure 4.1: Traditional Feedback

transformed into a measurable voltage [14, 15], or a reference clock which gives a measure of the computational capability requirement of the load [16] . Depending on the

load requirements, the conversion rate will change, by employing either pulse width modulation (PWM) or pulse frequency modulation (PFM) schemes. This creates the need for continuous feedback, meaning that the output must be sampled either continuously, or at a rate near the corner frequency of the output filter dynamics. Efficiency is a major metric for evaluating desirability of a particular architecture.

In the case of the MEMS energy converter, this is not true. The load requirements must be satisfied, but beyond this efficiency is not a concern. The energy conversion process not only provides a power source for use by the load circuitry, but also provides damping for the mechanical system consisting of the MEMS capacitor. This means that the conversion process should be continuous in order to provide for stability of the mechanical system.

4.1 Delay Locked Loop: Overview

The architecture for generating timing pulses used to drive the power FETs has been discussed in section 3.4. What is now desired is a way to generate conversion initialization pulses which correspond to C_{mems} being at a maximum and at a minimum. The method chosen to do this is to use a modified form of a delay locked loop (DLL), where a reference clock with a period equal to the desired mechanical vibration frequency, T_{vib}, is phase locked to the capacitor plate motion. This is a unique approach because, traditionally, DLL's have been designed to lock to an integer number of reference clock periods, whereas in this case ultimate lock will most likely result in a fractional period delay which corresponds to zero phase difference between the mechanical vibration and reference clock. A block diagram of the loop is shown in Figure 4.2.

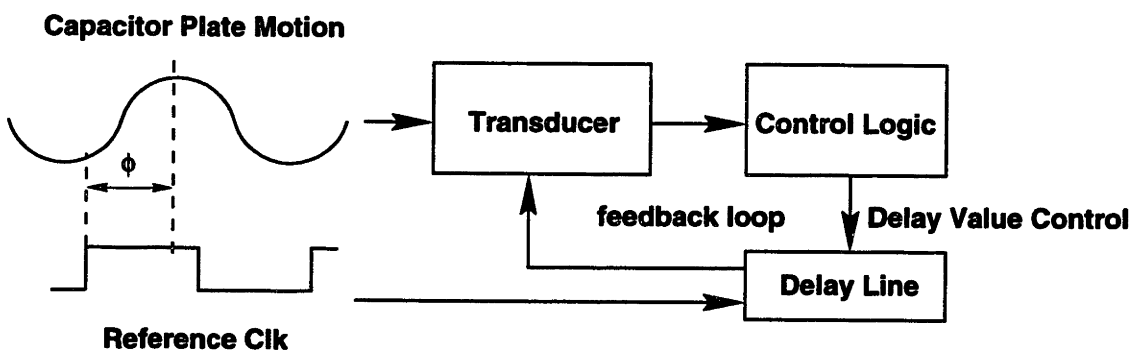


Figure 4.2: DLL Block Diagram

Basically, there is a tapped delay line which delays the reference clock by an amount set by the feedback mechanism. The feedback employed is to measure the

energy converted in a given period at a given delay, and compare it to the energy converted over a period at a different delay. By using the delay value which corresponds to the global maximum value of energy converted over a period, we can attain phase lock.

There are several points to note about the delay line implementation. First, the length of the line is variable in two respects, which will be discussed in section 4.4. This is in order to save power during both locking of the loop and steady state operation. Second, the delay of a single tap is constant. Some DLL implementations [17, 18], which use analog current starved inverters or variable capacitive loads to vary the delay of the line to achieve lock, require large static power capabilities relative to this system's control overhead budget. This is due in large part to the variable gate voltages that must be generated in order to vary the delay and the analog overhead associated with doing this. It is possible to dynamically manage the analog portions of the circuit as in [4], but this requires that some form of sample and hold action be performed, which in turn means that the gate voltage is stored on a dynamic node and must be periodically refreshed, necessitating the need for periodic analog operations (in this case in the form of operating the charge pump over many cycles to change the gate voltage). In section 4.5, the value of using the current starved inverter approach and allowing some DC power dissipation as a tradeoff for large per-buffer delays will be discussed. The difference between this discussion and that of [4] is that in the latter the gate control voltage is dynamically controlled to change the delay of the entire line, and therefore each buffer, while in section 4.5 the length of the delay line is also variable, but each delay is implemented by current starved inverters with a *fixed* gate voltage. Consequently, the delay line's delay value is varied by choosing different taps off of the line.

4.2 Energy Based Feedback

In order to attain phase lock between the reference clock and capacitor motion, energy feedback is utilized. For the lossless case, the relationship between energy out and phase delay between the reference clock and vibration is

$$E_{out} = \frac{1}{2}\frac{C_1(t_d) - C_2(t_d)}{C_{par} + C_2(t_d)} L i_{max}^2 \tag{4.1}$$

Where $C_1(t_d)$ and $C_2(t_d)$ are the values of the MEMS capacitance when the conversion process is started and completed as functions of delay time, t_d, and ideally $C_1(t_{nom}) = C_{max}$ and $C_2(t_{nom}) = C_{min}$ where t_{nom} is the optimal value. This relationship is depicted in Figure 4.3. The rule here is that the values for t_1, t_2, t_3, t_4 and t_5 are constant. These timing pulsewidths are dependent on the values of the inductor, external capacitor, C_{par}, and vibration frequency and are independent of phase. By

moving this window of pulses in phase, different output energies are obtained. In order to give a qualitative feel for how the energy output varies with delay, the value

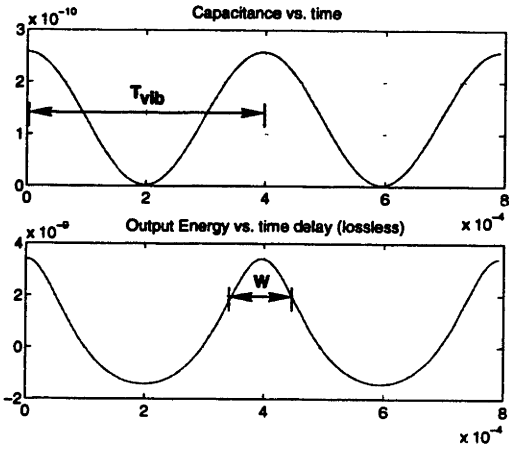


Figure 4.3: Energy Out vs. Delay

of capacitance of the MEMS device has also been plotted. As expected, at integer multiples of delay equal to the vibration period, T_{vib}, energy output is at a maximum, and at integer multiples of $T_{vib}/2$, energy output at a minimum. Figure 4.3 shows that there is a window, W, of delay values about the optimum value for which the energy out is still relatively large. This suggests that if the system is initially locked to the maximum value, phase and/or frequency drift can be tolerated in the reference clock. Once the energy converted drifts out of the acceptable window area, the system should be re-locked. This eliminates the need for continuous feedback. Ideally, if the reference clock matches the vibration frequency exactly, the delay value will never need to be updated and feedback control is only implemented once. In a practical system, the feedback update time will be dependent on the frequency difference and phase drift between the reference clock and mechanical vibration. This will be discussed in section 4.6.4.

4.2.1 Energy Measurement

The feedback variable is the energy converted over a period. The method proposed to measure the energy converted is depicted in figure 4.4. During t_5 of the energy conversion cycle as described in section 3.1.1, the inductor would normally return the converted energy to the source. During the phase lock sequence, the converted energy is instead integrated onto a measurement capacitor, C_{meas}. After t_5 is over, C_{meas} is used to power a ring oscillator which in turn clocks a counter. The final value of the count over a sampling window gives a measure of the energy converted. This

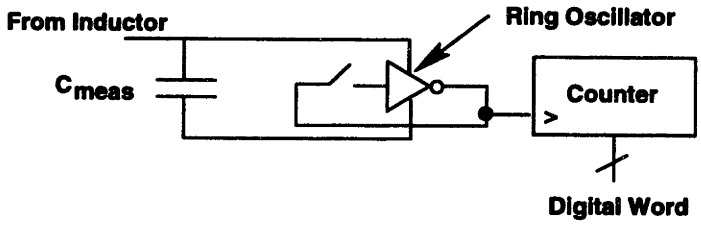


Figure 4.4: Energy Measurement Circuit

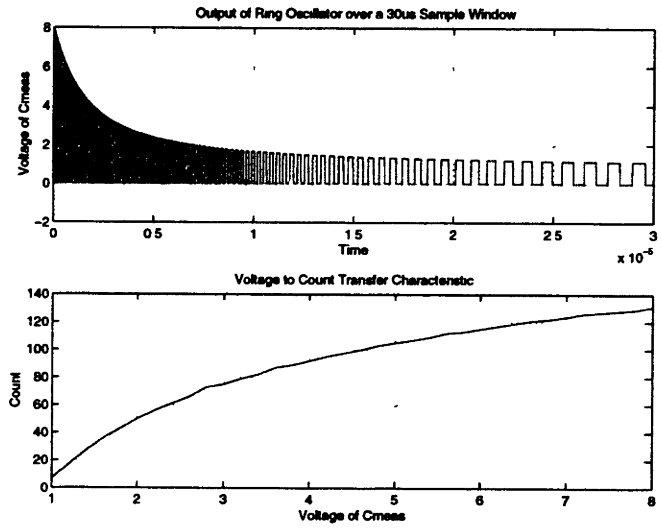


Figure 4.5: Energy to Count transfer Characteristic

can then be used to lock to some maximum value of energy converted, corresponding to the reference clock being in phase with the mechanical vibration.

The transfer characteristic of voltage to count is shown in Figure 4.5. The top sub-plot demonstrates the decaying nature of the feedback capacitor voltage as the oscillator rings. The lower plot is the voltage to count transfer characteristic for $C_{meas} = 100pF$, with a 30us sample window of the count. It is apparent that there is a high gain region for low values of voltage and a lower gain region for higher voltages. This is a highly desirable characteristic if continuous feedback were used, since decreasing gain with increasing voltage can be exploited as the system converges to phase lock.

The value of C_{meas} will determine what range of voltages the ring oscillator will see. This, coupled with the sampling window utilized, will constrain the dynamic range measurement capabilities of the feedback mechanism. Figure 4.6 is the same as Figure 4.3, with C_{meas} voltage as functions of delay and capacitance shown. The main

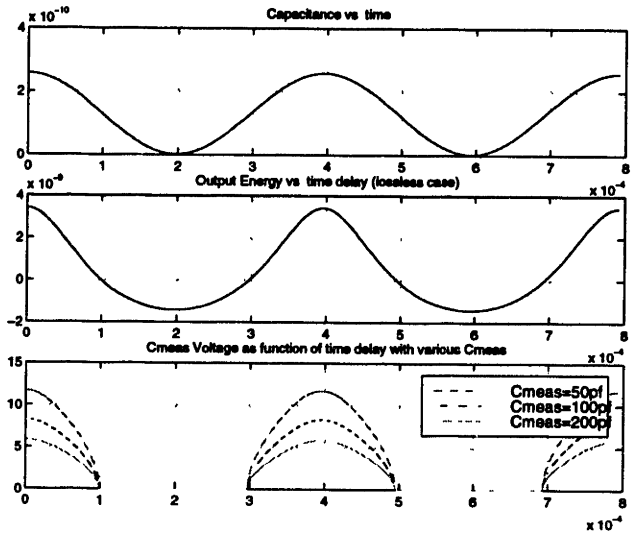
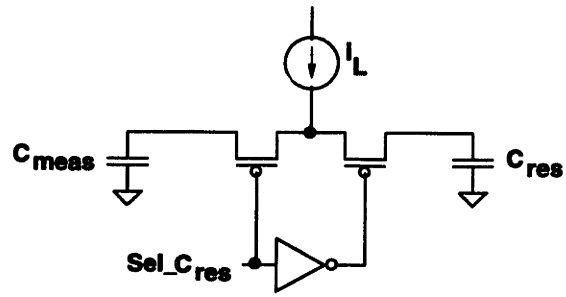


Figure 4.6: Energy and Capacitor Voltage vs. Delay

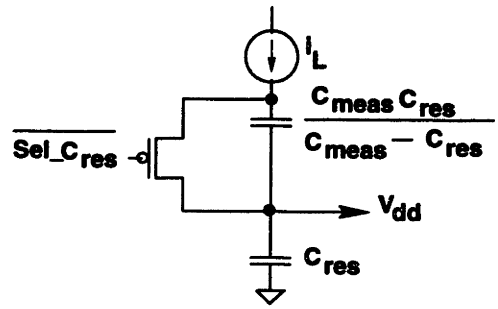
constraint on sizing C_{meas} is that it must be small enough so that a reasonable range of voltages (and therefore counts) are developed, but large enough so that process breakdown limitations are not exceeded. Based on the 5V process used to design the controller chip, a value of $C_{meas} = 100pF$ was chosen as being reasonable to meet the desired criteria. A more subtle point about the measurement process is to note that t_5 must be changed from its locked value. This is caused by the fact that $C_{meas} << C_{res}$, so the time for C_{meas} to integrate up to its maximum value is less than t_5. In order to prevent reactive ringing of the LC_{meas} circuit, t_5 must be decreased to $\frac{\pi}{2}\sqrt{LC_{meas}}$.

An analog multiplexor is needed to choose which of C_{res} and C_{meas} receives the converted energy from the inductor during t_5. Since the inductor is acting as a current source, a simple PMOS pass gate mux is not sufficient. This is because as the voltage on C_{meas} rises past V_{dd} during t_5, it it not possible to keep the unselected PMOS off with V_{dd}. Figure 4.7 depicts two mux topologies. Figure 4.7 a) is a standard PMOS pass gate mux. This is a useful circuit when a *voltage* source is switched between two circuits, or two different voltage sources are switched to a single node. The PMOS devices are used so that the full supply voltage passes to the load. The requirement for correct operation is that the unselected transistor remains off. In reference to Figure 4.7 a), this means that when Sel_C_{res} is high, the gate of the left PFET remains high enough to keep it off. This is not true when a current source, such as an inductor, charges C_{res} past V_{dd}. In this case the left PFET will begin to conduct when $V_{res} > V_{dd} + |V_{tp}|$.

A solution to this problem is shown in Figure 4.7 b). Here a capacitive divider is used for the measurement process. When a smaller capacitance is needed for



a) Pass Gate Mux



b) Capacitive Divider Mux

Figure 4.7: Analog Multiplexor

measurement, the PFET is switched off and the series capacitor network creates an effective capacitance of C_{meas}. V_{dd} may still be tapped by the rest of the circuit unaffected because of the large difference between the two capacitor values. ($C_{res} \sim \mu F$, $C_{meas} \sim 100pF$) Of course, level converters whose upper rail is set to the voltage at the current source node must be used to drive the gate of the PFET to ensure proper functionality when the PFET is off.

4.3 Feedback Approaches

Once the measurement technique has been established, it is necessary to evaluate different methods for utilizing this information to achieve phase lock. Figure 4.8 shows one possible approach. The top trace represents the total value of the delay, the sinusoidal wave is the capacitor value, and the square wave is the reference clock. Here, a counter is used to keep track of the desired delay. The amount by which the delay is increased is dependent on the count output from the counter. When the count is below a certain threshold value, a large increment is used. This is equivalent

to a proportional (the gain control) plus integral (the delay counter) control method. Once the count is past the lower threshold, the gain is decreased until the count passes an upper threshold value, above which the system is determined to be locked. The delay increment was made large enough so that phase capture occurs over few enough periods so as to be discernible. This approach is sensitive to the value of the

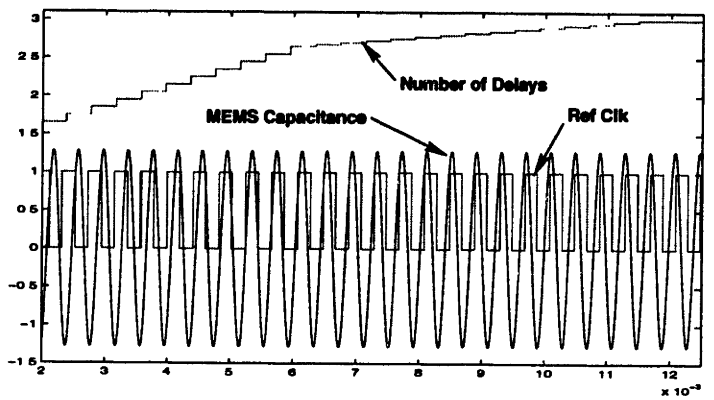


Figure 4.8: Locking based on a count threshold

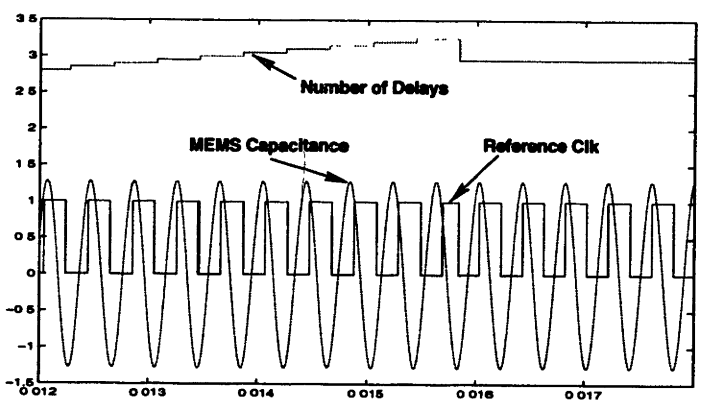


Figure 4.9: Locking based on examining a full period

upper threshold. If it is chosen too low, the system will have to re-lock sooner for a given phase drift, because it will be just on the edge of acceptable energy conversion. If the upper threshold is set too high, the system will never lock and the delay line length will saturate.

Another approach is to measure the values of energy converted at each possible delay and, after a period is over, choose the value of delay corresponding to the maximum energy converted. This is shown in Figure 4.9. The step down in the delay

value trace denotes that the controller has sensed that a complete period has been sampled, so it switches to the stored value of delay matching to the maximum energy converted. Once again, the delay step size has been exaggerated to make the plot more readable. This approach has several advantages. It is not dependent on a-priori knowledge of the count that will be produced under ideal lock, in order to obtain ideal lock, as the previous approach does. Also, because the entire period of the reference clock is sampled over, some distortion in the capacitor plate motion can be filtered out. This is true because the controller looks for a *global* maximum over a period and so will filter out local maxima as the period is sampled. Of course, the degree to which the distortion filtering is effective (and therefore the amount of distortion allowed) is dependent on the resolution of the sampling process, which is determined by the delay line length and single delay duration.

The best solution is to use a combination of the two approaches. In order to allow for distortion in the plate motion and a attain true phase lock, the entire reference period is sampled over. In order to decrease the number of cycles to accomplish this, a lower threshold of count is set. When the feedback counter output is below this value, the delay increment is made large. In this way, regions of low output energy may be rapidly stepped through. Also, after the system has captured phase lock, the output energy is sub-sampled at a lower frequency and compared to a lower bound of acceptable feedback count. The value of the lower bound is based on the stored value of maximum count when phase lock was previously achieved. If it is above this minimum upper bound, the system is allowed to continue at the set delay. If the value is below this threshold, the system undergoes the phase lock process again. In this way, phase drift and frequency differences between the reference clock and the mechanical vibration can be compensated for.

4.4 Delay Line Length

Power may be saved if the length of the line is variable such that the unused taps are prevented from switching. This requires some gating logic around each delay, or around stages of equal numbers of delays. Another way to vary the line length is to use a variant of the fast clocked counter method. A circuit which is similar to the circuit of Figure 3.10 may be used. A ring oscillator period is matched to the desired delay of a single buffer tap. The oscillator runs until the count is equal to a reference value corresponding to the desired delay. This method automatically assures that no extra transitions take place at a much lower control overhead circuitry cost than a simple tapped delay line. However, the extra power used in toggling the counter bits is highly undesirable. This creates an area-and-complexity versus power tradeoff in implementation. Because power is the limiting factor, the simple delay line with enabling logic has been chosen.

This power saving technique is useful once the system has attained phase lock. The algorithm used to achieve lock has been briefly described in section 4.1. Recall that the basic algorithm is to look at the energy converted at each possible delay increment and determine what the required delay is to achieve the maximum converted energy. The requirement to ensure that a true maximum is found is to look over an entire period of the reference clock. This implies that the total delay of the tapped delay line must be at least equal to a reference clock period, so some form of calibration of the line length is necessary.

4.4.1 Delay Line Length Calibration

Since the power budget for the system is low, analog techniques cannot be used to calibrate the total delay of the line (and therefore the delay of each buffer) to the reference clock period. As previously mentioned, these techniques require the ability to generate continuous values of gate voltage for use by current starved inverters or capacitive loads. This necessitates the presence of analog amplifiers, whose biasing dissipate static power. The possibility of digitally calibrating the delay line does exist. The requirement for the total line delay is that it be equal to or greater than the reference clock period. (Or in the case of how the controller was actually implemented, half of the reference clock period). One way of achieving this is to make sure that, at all process corners, the total line delay is greater than the required time. This implies a penalty is paid in that we need to provide extra buffers as a security measure to account for fast process corners, which will be unnecessary at a slow corner. If calibration is desired, so that the total delay at the last buffer which is tapped is not greater than a single period (or half period), traditional DLL techniques may be used. The controller can be designed to implement a calibration sequence where the reference clock is allowed to propagate to a particular tap, and a phase detector is used to determine whether the particular tapped delay chosen represents more or less than a reference clock period. An UP/DOWN signal then determines whether the tap selection increases or decreases, and, over time, calibration is achieved, with oscillations of the line length (phase jitter) about an average value approximately equal to one period. In order to decrease jitter, the continuous feedback may be cut off after sensing that these steady-state oscillations have occurred, giving an oscillation-free steady-state error equal to some fraction of a single delay. This traditional approach requires some duration of continuous feedback (meaning that the feedback measurement for phase adjustment is done over consecutive periods), which has two major repercussions. The first is that the time required for initial phase lock from startup, and during periodic re-calibrations, is increased by the calibration time. The second, and more important from a low power systems perspective, is that this calibration represents more overhead power cost, and so a method for limiting its impact is necessary.

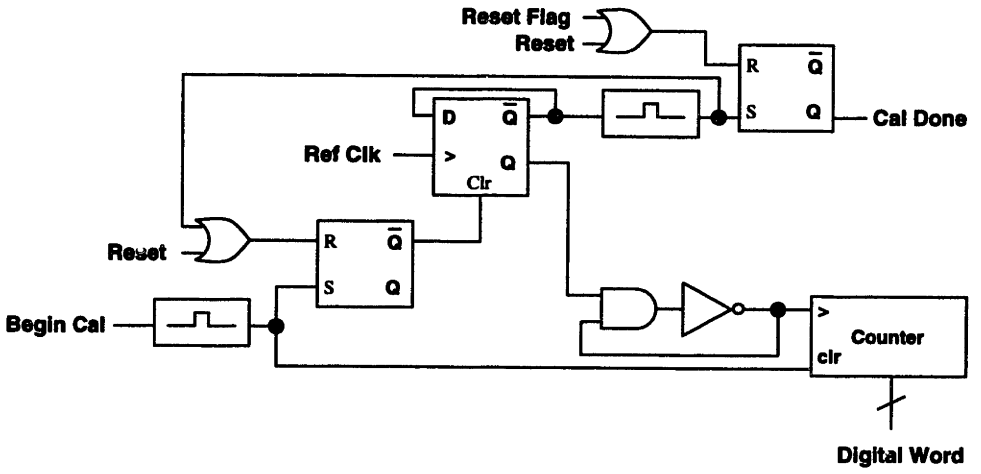


Figure 4.10: Single Period Calibration Circuit

Figure 4.10 shows a circuit that allows the calibration of the delay line length to occur in one reference period. The flip-flop combination creates a pulse of a duration equal to one clock period (note that both flops must be cleared prior to calibration occurring) which enables a ring oscillator. This oscillator consists of a ring of inverters whose period equals one tap of the delay line. The counter then tracks how many taps are required to equal one reference clock period. There will be steady state error, but since the oscillator is enabled by the reference clk, we are assured that the error will be some fraction of a single delay tap. This means that we will, at worst, only be off by a single measurement each time we perform an energy phase lock as described in section 4.2.1. The calibration code word is stored in a register, to be used by the phase locking algorithm. The major benefit from calibration is that the effects of process, supply and temperature variations may be eliminated. In addition, calibration assures that when phase locking is performed, only a single reference clock period is sampled, saving power.

4.5 Delay Line Architecture

In section 4.4 the simple fixed delay line was chosen over the less area intensive fast clocked counter approach because of the power savings that it offered. Even after the general form has been chosen, the architecture of the line in terms of where and how tapping is performed, and how a single delay is constructed has many possibilities.

4.5.1 Duty Cycle Requirements of the Delay Line

This design is aggressive because it is necessary to generate a worst case delay of hundreds of μs, to cover the entire range of possible phase shift between the reference clock and vibration, with a steady-state power consumption of only a few μW. Further, the output of the delay line has to be of 50% duty cycle in order that the controller can distinguish both maximum and minimum capacitance of the MEMS transducer. This has the requirement that if the reference clock itself is sent down the delay line and tapped accordingly, the loading of each buffer must be identical so that the duty cycle does not change during propagation down the line. (This phenomena has been reported in [4]). Figure 4.11 shows a general implementation of a tapped

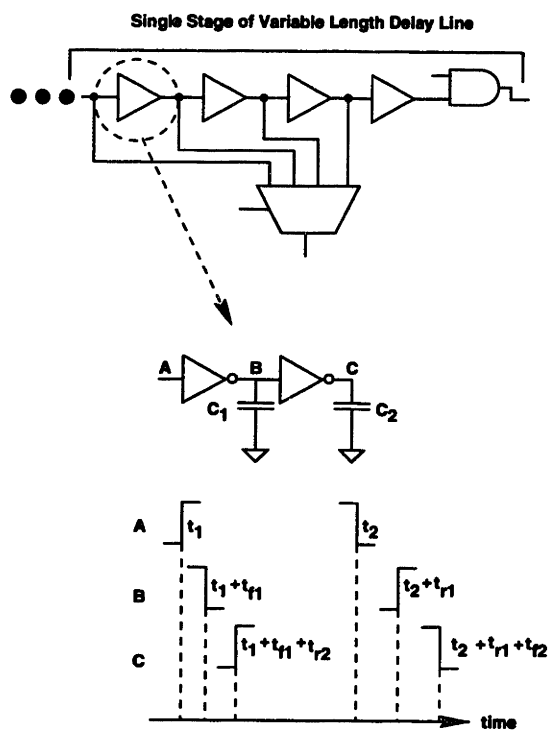


Figure 4.11: Pulse Width Growth Phenomena

delay architecture. The method used to create a buffer delay is to cascade inverters. These may be simply misratioed transistors or current starved inverters to create the required delay resolution (or may be odd numbers of a smaller unit inverters cascaded together); the actual buffer style is not important. However, because the two inverters making up a buffer see different loading, represented by C_1 and C_2, their respective rise and fall times will be different (The first inverter only sees the gate of the second, while the second sees the gate of the next buffer and a mux input.

Also, further nonlinearity is introduced by the AND gate which is the enabling logic used to vary the length of the delay line and stop unnecessary downstream switching events.) As is evident from the figure, the input pulsewidth is $t_2 - t_1$ and the output pulsewidth is

$$t_{out} = t_2 - t_1 + ((t_{f2} - t_{f1}) + (t_{r1} - t_{r2})) \tag{4.2}$$

In the ideal case the rise and fall times of the inverters are equal and the output pulsewidth is the same as the input. However, because of the added loading on the second inverter, its rise and fall times will be longer, and so the pulse will grow. One solution would be to try and exactly match the rise and fall times of each particular gate, even if the absolute value of rise time or fall time is different from buffer to buffer; the two terms in parenthesis would have opposite signs and cancel. However, matching rise and fall times for a gate is difficult given process variation, and any error would integrate over the length of the delay line. Another solution is to balance the loading so that rather than trying to make rise and fall times equal for each buffer, the requirement is for the rise times to be consistent from buffer to buffer. The same constraint is placed on fall times. In this case, the two terms in parenthesis in equation 4.2 would be zero. The balanced loading approach requires more switched capacitance when a variable length delay line is desired, and therefore more power. Finally, the brute force method would be to not send the reference clock down the delay line at all. Instead, a pulse is generated both at the rising and falling clock edge and sent down the line to the appropriate tap. While this means that the power spent in the delay line has doubled because now there are four edge transitions as opposed to two propagating down the line, the time between the rising edges of each pulse is guaranteed to be one half of the reference period because the two pulses will travel the line in the same fashion. The requirement in this case is that the pulse growth phenomena does not cause the pulse to grow so much that, in the case where a large value of delay is needed and so a very far downstream tap is used, the pulse remains narrow enough to not swamp the delay line.

4.5.2 Decoding the Line Length

So far it has been established that it is desirable to vary the length of the delay line in two ways. First the line is calibrated to make sure that during the phase lock algorithm only a single period is sampled over. Once calibration establishes this maximum required line length, power savings will occur if we vary the line length to only cause switching events as far down the line as we need to tap it. Figure 4.11 shows a way of preventing propagation by using an AND gate in series with a stage of delays. The maximum number of extra switching transitions is then constrained to be one less than the number of buffers in a stage. For small values of delay this offers large energy savings. However, there is a control overhead associated with implementing the enabling logic. The delay tap selection signals must be decoded to

control the enable logic. From a circuit design standpoint, it is desirable to make the decode logic as simple as possible. An alternative to the method of Figure 4.11, where a large multiplexer is ultimately required to select a delay, is shown in Figure 4.12. In this instance, binarily weighted delays are used to vary the line length. The

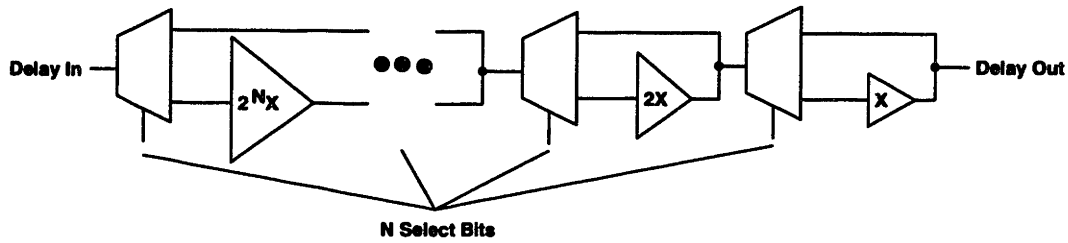


Figure 4.12: Binary Weighted Delay Line

decode logic is simple here, and in steady state no extra transitions will occur. In [4] this method was discarded in favor of that of Figure 4.11 because of the difficulty in generating binarily weighted delay accurately. It was proposed that such delays might be implemented using binarily weighted current sources or capacitor ratios. However, it is possible to simply use multiple unit delays to created the binary weights. This is exactly the same approach as in Figure 4.11, except that the multiplexer is not longer needed, and the decode logic is extremely simple in comparison. There will be some small amount of error through the decode logic depending on which path is chosen, but this error will be on the order of ns (a few gate delays), whereas the delay resolution is chosen to be approximately $5\mu s$. There is no need to use a delay matching network to ensure that all possible decoder paths see the same delay since the error is insignificant in relation to a buffer delay.

4.5.3 Single Buffer Delay Design

A single buffer in the delay line may be implemented as cascaded unit inverters, misratioed inverters of increasing delay (in this case the pulse growth phenomena would be very prominent because of different $\frac{W}{L}$ ratios which create different loading), current starved inverters, or capacitively loaded inverters. Since it has been determined that fixed delays will be used in this design, the last two approaches, which use analog gate voltages to derive a delay, will require DC voltage references which dissipate static power. If the references can be constructed to dissipate very low static power, it may be possible that these approaches will have an advantage over simple inverter cascades because of the large delays that they can generate through a single inverter.

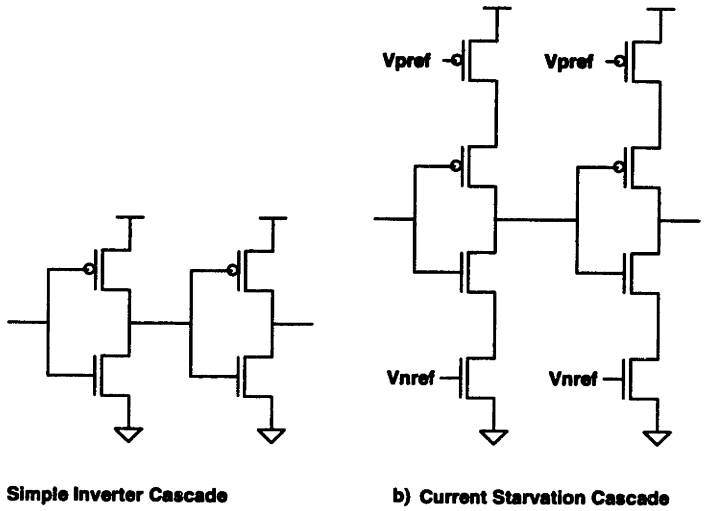


Figure 4.13: Buffer Delay Options

Simple Cascades

Figure 4.13 depicts two alternatives for buffer implementation. 4.13 a) is a simple inverter cascade. In this case delays may be made larger by either cascading an even number of many inverters or by misratioing individual inverters. As the gate length is made longer gate capacitance increases and current drive capability decreases, so delay will increase.

Current Starvation

In Figure 4.13 b), starvation transistors are used to control the current drive capability of the inverter. This results in an exponential relationship between gate voltage and delay time. [4, 17] At small DC bias voltages near threshold, delays per inverter for this method can be made much larger than for simple cascades. However, DC power is dissipated in the reference circuitry. Since the relationship between gate voltage and delay is exponential, the variation of gate voltage with supply, process, and temperature variations must be carefully examined during design of such buffer stages to assure that the delay per stage is within the resolution requirements of the delay line under worst case circumstances.

Ultimately, the cascaded simple delay approach was chosen. Although simulations showed that current starved inverters were able to reduce delay line power consumption over the simple cascade case, the exponential relationship between the starvation voltage and delay size requires too much control. The main problem in using constant valued current starved inverters is in generating a reasonably constant reference

for the gate control voltage while dissipating static power. For this reason, a simple cascade of misratioed (long channel length) inverters was chosen.

A point worthy of some discussion is the problem of overlap (short circuit) current in misratioed inverters. Overlap current occurs when both devices in an inverter structure are on at the same time creating a purely resistive path between Vdd and GND. Figure 4.14 shows this phenomena.

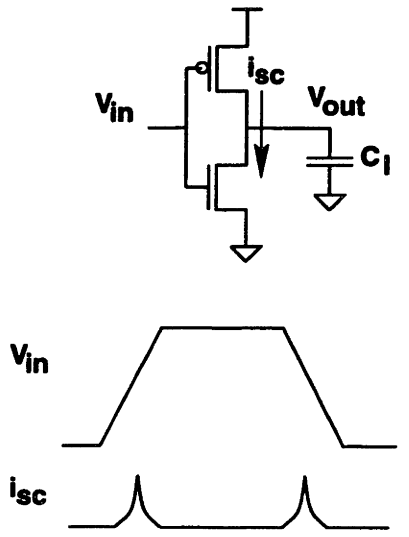


Figure 4.14: Overlap Current [1]

This behavior becomes more relevant when the signal driving the device gates is slow, or ramp-like. One of the side effects of using long channel devices is that the rise and fall times of the output will be slower than for minimum sized gates. So the question arises as to whether overlap power, which is usually a small percentage of dissipation in digital gates, becomes dominant under this type of circuit construction.

It can be shown that overlap current becomes a problem when the rise and fall times at the input of an inverter is much greater than the rise and fall times at the output.[1]:

$$P_{short-circuit} = \frac{\beta}{12}(V_{dd} - 2V_t)^3 \frac{\tau}{T_{clk}} \tag{4.3}$$

Here, β is the transistor strength, V_t is the threshold voltage (assuming equal values for N and P devices), τ is the input rise time, and T_{clk} is the system operating frequency.

If the long inverter drives an inverter of equal dimensions (the self-loading condition) then overlap issues are a wash. This is intuitive because even though the input

of the inverter is driven with a slower, more ramp-like waveform, the current drive capabilities of the output have also decreased, as compared to the case where minimum length devices are used. A problem arises, however, when the long channel length gate is used to drive a minimum length gate. In this case overlap power dissipation in the minimum gate becomes a significant portion of its overall power dissipation because of the low channel resistance these gates exhibit when they are on.

A possible solution to this problem is to follow a similar approach to that used in clock buffering circuits [19]. When it is necessary to distribute a clock to drive a large number of fan out gates across large interconnect, the inverting buffers are scaled up from minimum width devices to wider devices by a scaling factor, α, until the final output driver is wide enough to satisfy the design constraints. In the case of a cascade of long devices, a similar approach can be taken, where the channel lengths of the output drivers are gradually shortened until a minimum feature length is reached. While this will help decrease the overlap power dissipation at the output stage, it will add more delay to the output signal (nonlinearity), and will add switched capacitance (power). Since the delay line proposed only consists of six binarily weighted stages, requiring six interfaces between long and short channel length gates, no tapering of the channel length was performed. Instead the overlap power penalty is paid as it amounts to a small amount of power loss compared to the power associated with generating such large delays.

4.5.4 Reference Clock Polarity

A simple circuit trick to decrease the required delay line length is to use an XOR gate in series with the reference clock as shown in Figure 4.15. The XOR allows

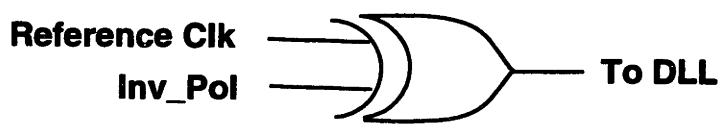


Figure 4.15: Reference Clock Inverter

a modification to the lock algorithm. Now, a half-period of the reference period is sampled over with the clock simply buffered by the XOR gate. After this half-period window is over, the clock polarity is flipped by asserting InvPol, which gives a half period phase shift "for free." A second half-period is sampled with this negative clock phase. After this window is over, a comparison is performed between the maximum counts of each clock polarity, and the overall maximum is chosen. This halves the required length of the delay line.

4.6 Implementation of Lock Algorithm

4.6.1 Peak Detection

The peak detector works to filter out local maxima form the global maximum. In

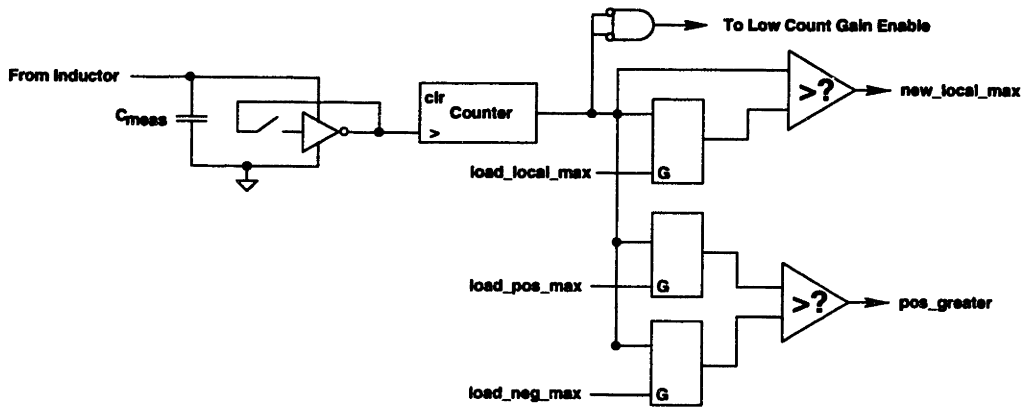


Figure 4.16: Peak Energy Detector Circuitry

section 4.2.1 the energy-to-count circuit was described. The peak detector compares the current value of count to the previous maximum. If a new, local, maximum is found, it is loaded into the local_max register. As described in section 4.5.4, there are two half periods to be sampled, each of opposite reference clock polarities. For this reason, two registers are used to store the maximum counts for each clock polarity. After an entire period has been sampled over, these two maxima are compared to see which clock polarity resulted in the largest energy. Of course, in reality, only two total registers are required. The local_max register is redundant. The controller can be made intelligent enough to reuse the local max detection register for one of the clock polarities. It was made a separate register to clarify the circuit explanation. Whichever polarity resulted in maximum count sets the steady-state clock polarity.

The peak detector works in conjunction with the delay line length control circuitry through the gain enable gating. If the present value of count is below threshold (in Figure 4.16 an example of simply taking the two MSBs from the count is used), the delay control circuitry increases the delay increment so that region of low converted energy may be stepped through quickly. If the present count is above threshold, the output of the threshold detection gate will be low and the value of delay is incremented by a smaller amount.

4.6.2 Delay Line Length Control

Figure 4.17 shows the delay line length control circuit. The adder-accumulator keeps

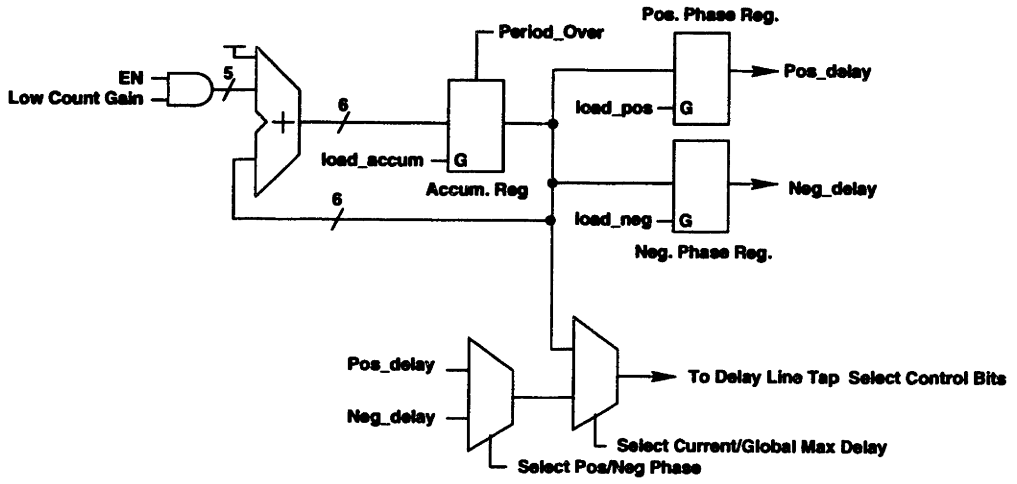


Figure 4.17: Delay Line Control Circuitry

track of the values delay. As the controller steps through values of delay, the amount by which delay is incremented is controlled by the gain inputs. For regions of low count, as determined by the energy-to-count circuit of Figure 4.4, the gain is increased to some larger value. During regions of high count (which means that the system is near phase lock), the gain is lowered, with the lower bound of gain being one, as represented by the high bit input to the adder-accumulator tied to V_{dd}. During the phase capture process, the outputs of the adder-accumulator control the selects of the variable length delay line. Also, during phase capture, as the peak detection circuit locates new local maxima of count (and therefore energy), the value of delay which generated that count is stored in either the Positive Phase Register or Negative Phase Register, depending on whether the reference clock has been inverted or not.

After a complete period has been sampled, the controller determines which reference clock polarity corresponds to maximum count via the peak detection circuit. It uses this information to select the appropriate clock polarity delay register to control the delay line length selects. The system is therefore locked to peak energy.

4.6.3 Lock Time

The power consumed by the controller during phase locking is directly proportional to the lock time. For a simple algorithm where the count is incremented by one after

each energy measurement is performed, the lock time is

$$t_{lock} = NT_{ref} = \frac{T_{ref}^2}{t_{d1}} \tag{4.4}$$

Where N is the number of buffer delays in a reference clock period, T_{ref}, and t_{d1} is the delay of a single buffer. The lock time of the controller in the proposed implementation will be

$$t_{lock} = T_{ref}\frac{t_{thr-}}{Kt_{d1}} + T_{ref}\frac{t_{thr+}}{t_{d1}} \tag{4.5}$$

Where t_{thr-} and t_{thr+} are the delays for which the count is below and above threshold, respectively, and K is the gain when the count is below threshold. Making simple assumptions that the count is above and below threshold for equal times, we arrive at

$$t_{lock} = \frac{T_{ref}^2}{2t_{d1}}(1 + \frac{1}{K}) \tag{4.6}$$

By comparing equations 4.4 and 4.6 it is obvious that for values of $K > 1$, the lock time is improved. Equation 4.6 also suggests, given these assumptions, that the lock time will never be less than $NT_{ref}/2$ no matter what K is chosen. Of course, the threshold count may be chosen such that the system utilizes the high gain region more than half of the time. Also, rather than having two regions, one of high gain and one unity, it is possible to use more strictly proportional gain, but this creates the risk of jumping past the optimum lock point if the waveform is very distorted.

4.6.4 Lock Updating

In order to compensate for phase drift and frequency differences between the reference clock and the mechanical vibration, the converted energy is checked at a low sampling rate. If it is above threshold the system continues normal operation. If drift has caused the count to fall below threshold, the system is re-locked.

Reference Clock

The system is very application specific. The mechanical system has been tuned to a resonance centered around 2520 Hz. Ideally, the reference clock frequency will match this exactly. Crystal oscillator circuits have been implemented in CMOS process operating at low V_{dd} and power levels [3, 13]. The frequency stability of these oscillators is more than adequate for the MEMS energy converter system (on the order of 100ppm). Even using 1970's technology [20], power levels under 1μW have been demonstrated. Therefore, the reference clock is derived from a standard (32768 Hz) crystal oscillator reference. If the crystal output is divided by 13, a reference clock of 2520.6 Hz can

be generated, resulting in a frequency difference of 0.02%. The design of the crystal divider must create a reference which has a 50% duty cycle, which means that the clock divider must divide by 6.5 twice. A clock divider was designed and simulated with power dissipation of 8pJ/period, corresponding to 262nW. The total power consumption for the crystal and divider is expected to be less than 1μW based on state of the art oscillator design. If it is desired to lock to a different vibration frequencies, then a more complex, variable, clock divider can be implemented.

Number of Cycles Between Check for Re-lock

The sub-sample rate at which a check for the re-lock condition is performed can be set based on an estimate of the number of cycles required for the delayed reference clock to drift far enough out of phase with the mechanical vibration such that an unacceptably low amount of energy is harvested. The window of acceptable conversion energies is based on Figure 4.3, which shows output energy as a function of delay between the reference and mechanical vibration. (Note that this assumes that the frequency of the reference is equal to that of the vibration. In reality, it will be the difference in their respective frequencies, which causes the time delay between them to accumulate over many periods. However, the energy converted over a *single* period will not be noticeably different than the model predicts since the absolute frequency difference is much less than 1%) If a window of width W is chosen based on Figure 4.3, the number of conversion cycles for the system to drift out of acceptable phase can be calculated.

$$N_{drift} = \frac{W}{T_{ref} - Tvib} \tag{4.7}$$

From Figure 4.3 we can see that for $W = \pm 50\mu s$, the output power has a minimum value of approximately 60% of the maximum value. $N_{drift} = 516 cycles$. By using this relationship, a sub-sampling rate to check for the re-lock condition may be determined. It is important that the re-lock cycle number, N_{drift} be greater than the number of cycles needed to initially achieve lock so that, in the worst case where re-lock is necessary every N_{drift} cycles, most time is spent converting useful energy. For this reason, the value of K from section 4.6.3 should be chosen to be as large as possible. It is possible that re-lock won't be necessary, if the count measured at N_{drift} is greater than the set threshold.

Effects of Delay Line The preceding analysis assumed that the reference clock propagates through the delay line undistorted. In reality, if there is a difference between the rise and fall times of the inverters used the delay chain the reference clock period may change, which will decrease the number of cycles allowed to occur before the re-lock condition is checked. This phenomena has been described in section 4.5.1.

There are two possible solutions to this problem. The first is to simply use edge-to-pulse circuitry to transform both edges of the reference clock into pulses that can then be sent down the line. These are then used as C_{max} and C_{min} pulses directly. This has the effect of doubling the switched power down the delay line since four transitions now occur rather than two. However, since the controller timing pulse circuitry will be sensitive to the edges of these signals, and the rising edges will propagate the delay line at the same rate, this approach will be insensitive to the pulse growth phenomena.

4.6.5 Re-lock

If the count at check for re-lock is below threshold, the system must be re-locked. The value of the threshold check for re-lock is made independent of process variation by basing it not on an absolute number from simulations, but on the maximum count value determined by the phase lock algorithm. The maximum count value is stored in the local_max register during phase locking. It is simple to then base the re-lock threshold on some fraction of this value, based on the general shape of the Energy-to-count transfer function of Figure 4.5.

4.7 Ancillary Issues

This section completes the discussion of the energy feedback controller. The effects of feedback control on energy storage requirements of C_{res} will be explained. A summarized algorithm will be presented.

4.7.1 Conversion Rate and C_{res}

In section 3.3.3, a method for determining the required value of C_{res} was presented based on the energy needs of the system. For the feedback control case, the energy requirement increases by three factors. The first is the additional controller current required by the FSM and control logic during feedback. The second is the energy needed to operate the delay line. The final additional factor is the additional burden of the lock time, since during feedback operation, no energy is returned to the source.

The additional feedback overhead is difficult to calculate because it is not a steady-state factor. Once the system is determined to be locked, feedback is only intermittently performed at the sub-sample rate determined by clock drift. If the re-lock condition is false, the system continues in steady-state operation until the next sub-sample interval is completed. Good engineering practice dictates that the worst case be used in analysis, so it is assumed that the re-lock condition is always true, and the system always undergoes re-calibration at the sub-sample rate.

Even under these worst case circumstances, the feedback load will be amortized over the duty cycle of steady-state converter operation. As discussed in sections 4.6.3 and 4.6.4, there will be lock time and a re-lock check sub-sample rate associated with the feedback controller. These performance metrics will determine the duty cycle of the converter in terms of percentages of operation useful energy is converted vs. percentage of operation that is required for feedback analysis to be performed. This is analogous to the conversion rates associated with more traditional DC-DC converters [14]. The conversion rate may be determined as

$$D = \frac{t_{drift}}{t_{drift} + t_{lock}} \tag{4.8}$$

where t_{drift} is the time after lock has been established before check for the re-lock condition, and t_{lock} is the time required for lock to be established. Using equations 4.5 and 4.7, we find

$$D = \frac{W}{W + (T_{ref} - T_{vib})(\frac{t_{thr-}}{Kt_{d1}} + \frac{t_{thr+}}{t_{d1}})} \tag{4.9}$$

Equation 4.9 shows that as we increase the gain, K, when the system is below a preset threshold, t_{thr-}, the conversion ratio approaches 1. Further, if the vibration period, T_{vib} matches the reference clock period,T_{ref} exactly, the conversion ratio is 1 (or actually, approximately 1 since one conversion cycle will be lost when the sub-sampled feedback check is performed). We can make simple assumptions based on previous discussions. A single buffer delay is set to $5\mu s$. This means that for a vibration period of approximately $400\mu s$, there will be 80 cycles required to perform phase lock with $K = 1$. We can set K such that we take only half this many cycles to lock. Using the result of section 4.6.4 that the number of cycles after lock before checking for the re-lock condition is approximately 500 cycles, we arrive at $D = .93$, which means that useful energy is converted 93% of the time. D may be made larger by setting K to be larger.

A conservative estimate for the control overhead associated with implementing the feedback FSM and associated random logic is that it will be on the same order as the current controller. This is conservative because most of the power in the present controller goes into generating the gated oscillator, while the additional feedback circuitry will consists of simple FSMs and logic. However, a stringent upper bound of 0.5μW, amortized over the duty cycle of the feedback controller, resulting in negligible additional power.

The power consumed by the delay line in steady-state will be determined by the number of taps activated. A worst case condition is that all taps toggle every period. Simulations of a delay cell show that, in steady-state operation of all delays, the entire line would consume 1μW, where a buffer cell is implemented as long inverters.

The number of cycles required to attain phase lock has been estimated to be 40 cycles. In a brute-force approach to feedback control, this means that to maintain a given V_{dd} droop voltage as described in section 3.3.3, C_{res} would have to be increased by a factor of 40. A more intelligent approach would be to increase the lock time by adding some cycles during phase lock where the energy converted is steered to the reservoir, C_{res}, rather than to the measurement capacitor, C_{meas}. This would allow C_{res} to be small (a few μF), rather than 10 to 20 μF.

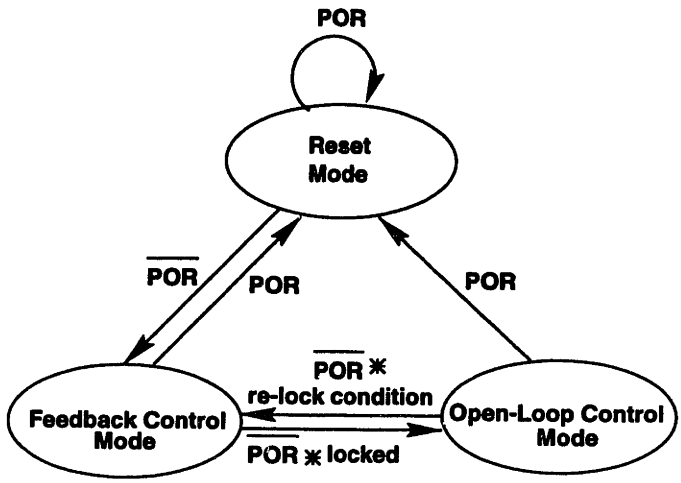


Figure 4.18: Energy Converter Operating Modes

4.7.2 Proposed Control Algorithm

In this final section, a control algorithm will be proposed. The algorithm incorporates all system considerations discussed throughout this chapter.

Figure 4.18 shows the three possible operating modes of the controller. In RESET mode, all flops are reset and the power electronics does not operate. This mode is also when the serial timing pulse registers are programmed with t_1, t_2, t_4, and t_5. This is therefore the start-up mode of the controller. Once POR is released, the system goes into FEEDBACK mode and performs the measurements required to lock the reference clock to the mechanical vibration. Once the feedback measurement is completed, the system moves into OPEN-LOOP mode, where the reference clock is delayed by an amount determined during FEEDBACK mode. While in OPEN-LOOP mode, the system periodically measures output energy to see if it is at an acceptable level. If not, the system returns to FEEDBACK mode to perform the phase locking measurements again. POR always returns the system to RESET mode.

OPEN-LOOP Mode

OPEN-LOOP mode is a subset of FEEDBACK mode, so it will be described first. This mode is depicted in Figure 4.19. The operation sequence starts with the controller waiting for the C_{max} input. In the fabricated controller, this is an external input. In the case of the energy feedback controller, this corresponds to the rising edge of the delayed reference clock. Upon receiving this stimulus, the controller generates pulse t_1. When t_1 completes, it triggers generation of pulse t_2. The controller then waits for control signal C_{min}. Again, this may be an external signal, or, in the

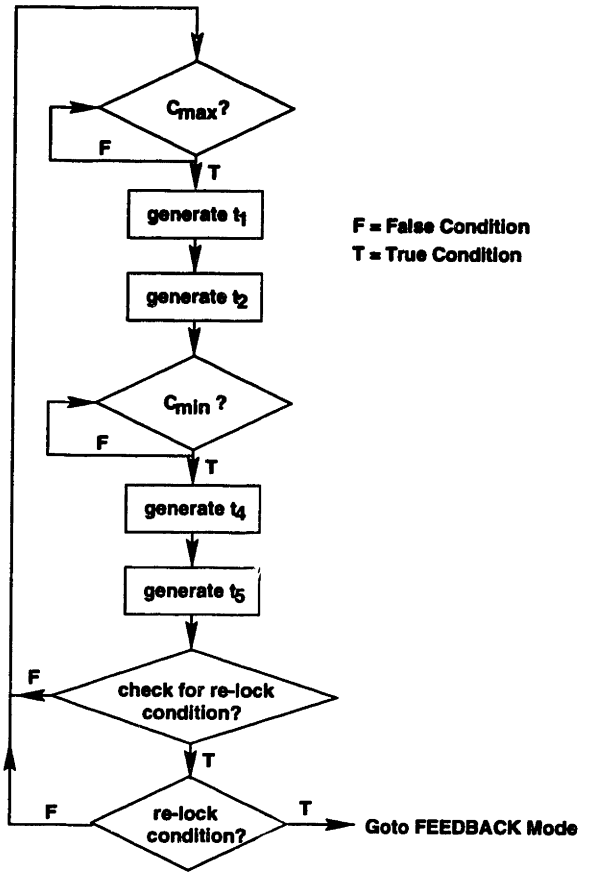


Figure 4.19: Open Loop Operation

case of the energy feedback controller, corresponds to the falling edge of the reference clock. After t_4 and t_5 are generated, two control paths are possible. If it is time to check for the re-lock condition, the controller performs one energy measurement. If the measurement reveals an unacceptably low level of harvested energy, the controller returns to feedback mode to re-lock the system. If it is not time to check for the re-lock condition, or if a check has been performed and energy harvesting is at an acceptable level, the controller waits for the next C_{max} to start the next energy conversion cycle.

FEEDBACK Mode

At startup, and during periodic system re-lock, the controller is in FEEDBACK mode. At the start of the control flow, the local max registers, for both count (energy) and delay, are reset. The reference clock polarity is initially set to be positive. Next, the

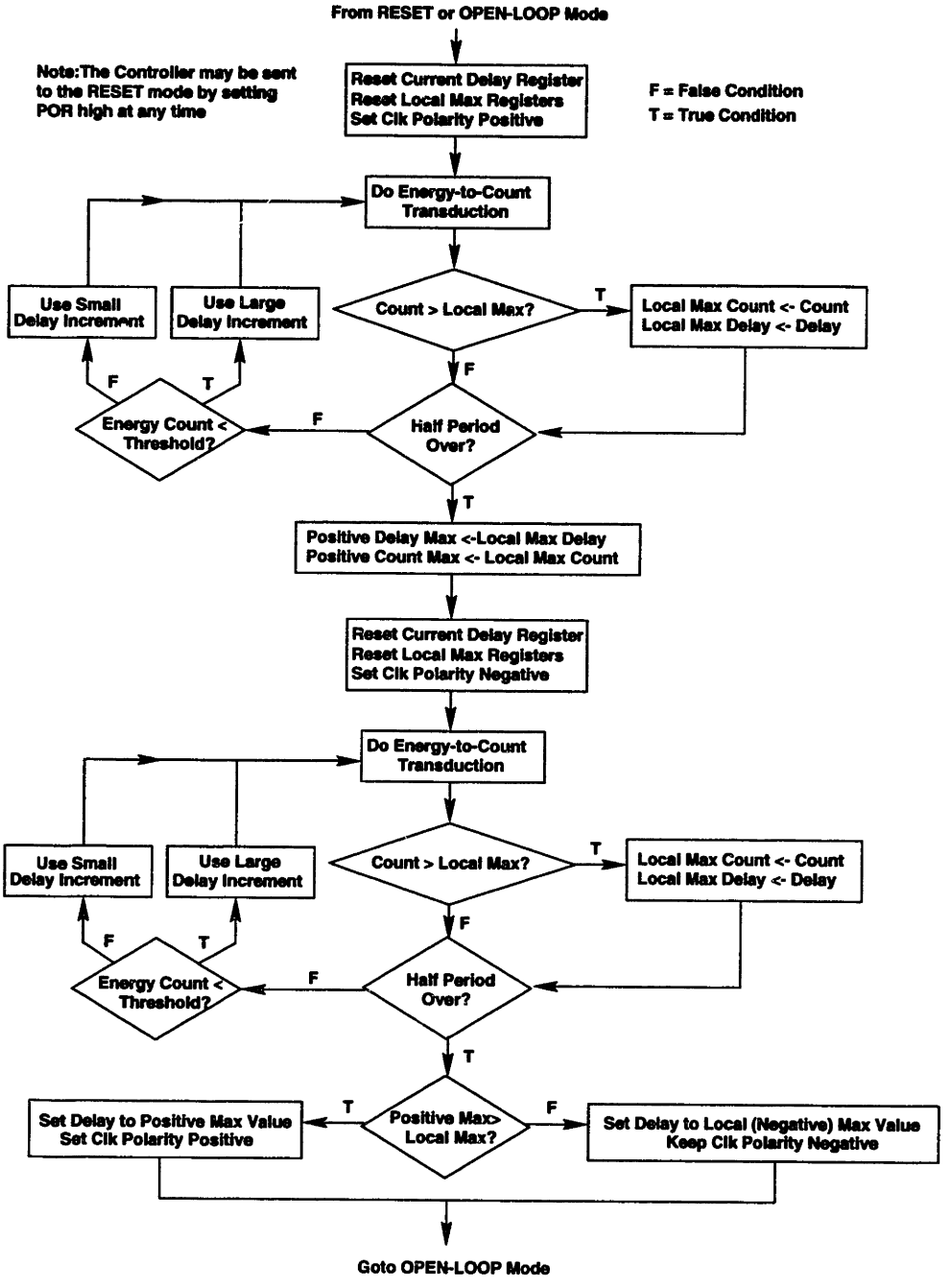


Figure 4.20: Feedback Operation

controller performs energy conversion. This consists of the control sequence described above for the open loop controller, where, on a rising edge of the delayed reference clock, the feedback control generates a C_{max} pulse, and on the falling edge a C_{min} pulse. The energy transduction is done through use of a ring oscillator and counter as described in section 4.2.1. The result is a digital word indicative of the amount of energy harvested by the system. Initially the register keeping track of delay is reset to initialize it to zero.

The count is compared to the previous maximum value (which is initially zero). If the present count is greater than the previous maximum, the present count is stored in the local_max register, and the corresponding delay value is stored in the positive_polarity register. If the present count is not larger than the previous max no register values are changed. Next, the controller checks to see if a half period of the reference has been completely sampled. If this is true, the values in the local_max registers for count and delay are moved to the positive_polarity_max registers, the local_max registers are reset, and the reference clock polarity is flipped. The controller then proceeds down a similar control path for negative clock polarity. If a half period is not over, the controller looks to see what the harvested energy count is. If it is below the preset threshold, the delay increment is set to be a large value, so that regions of delay corresponding to small output energy may be rapidly stepped through. If the count is above threshold, a small delay increment is used since the converter is approaching the value of delay corresponding to phase lock.

Once both polarities of reference clock have been examined, the controller compares the maximum count values from each phase. For the negative phase, the local_max_count register may be used. Whichever polarity resulted in a higher count is then set to be the steady-state polarity, and the corresponding delay for that polarity is used to act as the delay line control word. The system then enters OPEN-LOOP mode.

4.7.3 Possible Modifications to Proposed Algorithm

Some subtle modifications to the proposed algorithm may be explored to improve DLL performance. Once alteration would be to make the controller sensitive to the clock polarity and delay value that last result in phase lock when a new phase lock sequence is performed. If the previous delay value was within a certain range, a simple guess would be to say if it was within some window in the middle of the delay range or perhaps at the beginning, the next re-lock sequence only need sample that phase rather than perform a complete, dual polarity, control sequence. This would save power and shorten the re-lock time.

Another possibility to look into has already been mentioned in section 4.2. In order to speed up the lock time, a more proportional, variable gain control for the

delay increment can be used, rather than the simple two value (high and low) gain setting previously proposed. Differential control, to speed up the lock time, may be achieved by looking at the difference between the last count and the present count. Combinations of these, as well as using a counter to keep track of delay (integral control), may be combined to form PD (proportional-differential), and PID (proportional-integral-differential) controllers which are standard in continuous time control systems. [21, 15]

Chapter 5

Conclusions and Future Work

5.1 Conclusions

We have presented a system capable of harvesting ambient mechanical vibration energy and converting it into electrical form. An open loop controller has been fabricated and verified to function correctly. Losses for the digital core and power electronics were measured and found to be within acceptable limits (20%) of the values predicted by earlier models. A second, closed loop controller has been proposed based on a new technique of measuring harvested energy. This revised design has been aggressively optimized for low power. By employing discontinuous feedback, satisfactory performance is estimated to be attainable at an overhead control cost of only a few μW. This cost may be further amortized by paralleling many converters and having one controller.

5.2 Future Work

5.2.1 Current Controller

Once the MEMS transducer is fabricated it can be tested in the system to verify that actual levels of attainable output power match predictions. Further examination into the process limits of maximum voltage stress levels will give more insight into the upper bounds of output power.

5.2.2 Feedback Controller

The design of a feedback controller has been explained. The constituent sub-systems which together create the feedback scheme have been designed. The system has

been modeled, but further study into stability and optimal performance in terms of decreased lock time and phase drift should be explored in order to decrease the control overhead power costs. The brains of the feedback controller has not been designed because it offers little room for innovation, and may be implemented with a traditional dedicated FSM or more general purpose micro-controller. It is the author's instinct that the FSM is a better approach to take because of its dedicated nature. Depending on the complexity of the design the FSM architecture may best be implemented using random logic or a PLA structure. If a PLA is used, it would be desirable to use a self-timed approach to limit power consumption. Also, techniques such as PLA folding, can make this design style desirable because of added area savings.

Analog techniques were abandoned due to the undesirable costs associated with static power dissipation. It may be possible to pursue more analog designs if either the power budget for the controller can be increased (by perhaps operating many energy converters in parallel with a single controller, thus amortizing control cost), or if more aggressive analog design, requiring less bias power, can be achieved. Sub-threshold circuits may be amenable to this criteria.

Regulation of the input/output voltage, V_{dd} was not discussed. Sizing of the C_{res} capacitor has been explained in terms of restricting the droop voltage. However, in steady-state the capacitor voltage will have a tendency to rise, since energy is being supplied to it. The upper bound of V_{dd} can easily be set using passive rectification. By utilizing a low leakage zener diode in parallel with C_{res}, the upper bound of V_{dd} may be constrained. If more active voltage regulation techniques are desired, the simplest way to embed this function into the controller would be by either controlling t_5, which determines how much energy is returned to the source, or by adding a dummy load switch to dump excess energy.

5.3 Feedback

Two useful feedback techniques were developed in this thesis which merit further research. The first is the concept of using system energy as a feedback variable through use of a ring oscillator circuit. This, however, will have limited applications as it is simpler and more straightforward to measure system voltage levels or currents.

The more promising area is that of using mixed open-loop/closed-loop control approaches to save power in systems where precision and steady-state error are not the most important metrics. By employing intelligent digital control algorithms, acceptable performance can be achieved at a small fraction of power cost for an analog system which over-satisfies the same specifications.

5.4 Digital Calibration

Digital calibration was employed generously in the feedback controller design as a power-friendly way to compensate for process, temperature, and supply drift. This is a useful technique in general and has already been employed, to varying degrees, in many systems, including analog-to-digital and other data converter design. Opportunities exist to more aggressively apply this method to systems where low power is a main concern.

Bibliography

[1] A.P. Chandrakasan and R.W. Broderson, *Low Power Digital CMOS Design*, Kluwer Academic Publishers, Norwell, MA, 1995.

[2] A.P. Chandrakasan, R. Amirtharajah, J. Goodman, and W. Rabiner, "Trends in low power digital signal processing," *1998 IEEE International Symposium on Circuits and Systems Vol. 4*, 1998.

[3] Eric Vittoz, "Micropower IC," *Proceedings of the IEEE European Solid State CIrcuits Conference*, 1980.

[4] Abram Dancy, "Power supplies for ultra low power applications," *Massachusetts Institute of Technology, Masters Thesis*, 1996.

[5] Joshua Bretz, "DC-DC converters with high efficiency over wide load ranges," *Massachusetts Institute of Technology, Masters Thesis*, 1999.

[6] S.M. Wu, J.R. Yang, and T.Y. Liu, "An ASIC transponder for radio frequency identification system," *Proceedings Ninth Annual IEEE International ASIC Conference and Exhibit*, 1996.

[7] R. Amirtharajah and A.P. Chandrakasan, "Self-powered signal processing using vibration-based power generation," *IEEE Journal of Solid State Circuits, vol.33, no.5 May*, 1998.

[8] M.J. Konak, I.G. Powlesland, S.P. van der Velden, and S.C. Galea, "A self-powered discrete time piezoelectric vibration damper," *Proceedings of SPIE - The International Society for Optical Engineering Vol. 3241*, 1997.

[9] K. Bult et al., "Low power systems for wireless microsensors," *IEEE/ACM International Symposium on Low Power Electronics and Design, August*, 1996.

[10] G. Asada, M. Dong, T.S. Lin, F. Newberg, G. Pottie, W.J. Kaiser, and H.O. Marcy, "Wireless integrated network sensors: Low power systems on a chip," *IEEE ESSIRC*, 1998.

[11] Jose Oscar Mur-Miranda, "Design of a mems transducer," *Massachusetts Institute of Technology, PhD Thesis In Progress*, 2001.

[12] H.H. Woodson and J.R. Melcher, *Electromechanical Dynamics, Vol. 1*, John Wiley, 1968.

[13] D. Aebischer, H.J. Oguey, and V.R. Kaenel, "A 2.1Mhz crystal oscillator time base with a current consumption under 500nA," *IEEE Journal of Solid-State Circuits, Vol 32, No.7, July*, 1997.

[14] J.G. Kassakian, M.F. Schlect, and G.C. Verghese, *Principles of Power Electronics*, Addison Wesley, Reading, MA, 1992.

[15] J.K. Roberge, *Operational Amplifiers:Theory and Practice*, John Wiley and Sons, Boston, MA, 1975.

[16] V. Gutnik and A.P. Chandrakasan, "Embedded power supply for low-power DSP," *IEEE Transactions on VLSI Systems*, 1997.

[17] M.G. Johnson and E.L. Hudson, "A variable delay line PLL for CPU coprocessor synchronization," *IEEE JSSC*, 1992.

[18] A. Dancy and A.P. Chandrakasan, "A reconfigurable dual output low power digital PWM power converter," *Proceedings of the ISLPED*, 1998.

[19] N. Weste and K. Eshragian, *Principles of CMOS VLSI Design, 2nd Edition*, Addison Wesley, Reading, MA., 1992.

[20] Eric Vittoz, "Logical circuit for the wrist watch," *IEEE Eurocon 71 digest*, 1971.

[21] R.C. Dorf and R.H. Bishop, *Modern Control Systems*, Addison Wesley, Reading, MA, 1995.

Appendix A

Bonding Diagram and Pin-out

Figure A.1 has been included to visually relate the power electronics analog IO pins ToL, ToC, and Vdd:Ahi to the test circuit. Note that in the Pin-out list, VddCore is the digital core Vdd, VddPad is the digital IO pin Vdd, set to 5V to interface to standard logic families, and Vdd:Ahi is the voltage across C_{res}. The analog and digital supply voltages were separated so that losses could be measured separately.

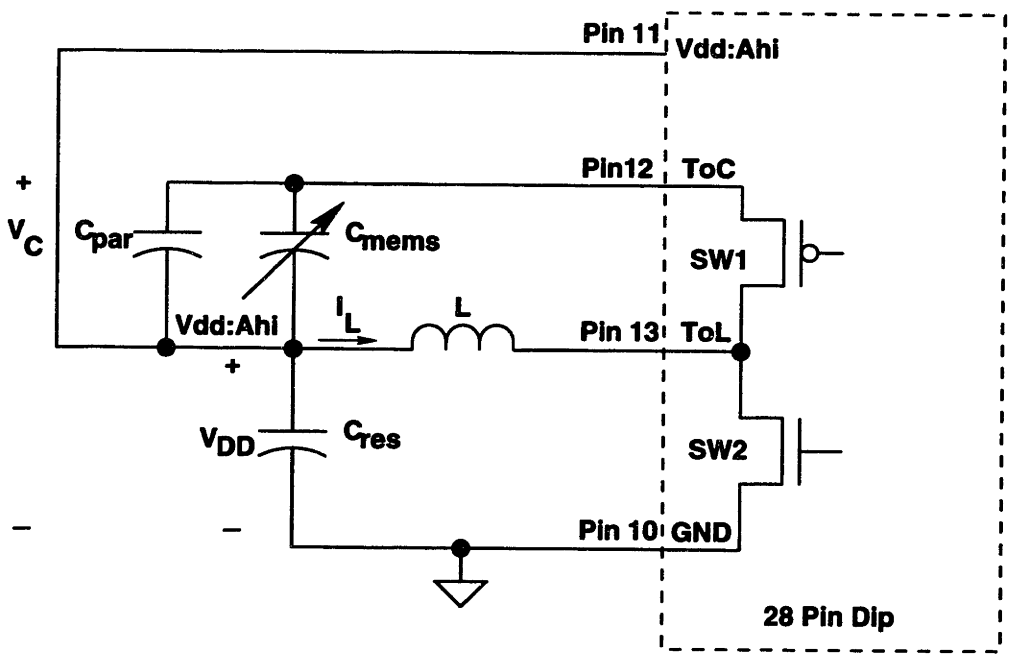


Figure A.1: Pin-out Schematic

Pin No.	Name	Function
1	SW2	Diagnostic Output
2	SW3	Diagnostic Output
3	SW1	Diagnostic Output
4	Vddcore	Digital Core Vdd (1.5V)
5	NC	No Connection
6	NC	No Connection
7	NC	No Connection
8	GND	GND
9	VddCore	Digital Core Vdd (1.5V)
10	GND	GND
11	Vdd:AHI	Analog Vdd (1.5V)
12	ToC	Connects to MEMS Cap
13	ToL	Connects to Inductor
14	NC	No Connection
15	NC	No Connection
16	NC	No Connection
17	VddPad	Pad Vdd (5V)
18	VddCore	Digital Core Vdd (1.5V)
19	Cmax	Cmax Control Input
20	Cmin	Cmin Control Input
21	GND	GND
22	SerIn	Serial Input
23	SerClr	Serial Clear (asynchronous)
24	SerClk	Serial Port Clk
25	NC	No Connection
26	NC	No Connection
27	NC	No Connection
28	OscOut	Diagnostic Output
29	OscEn	Must be high for gated osc to function
30	ClkSel	Low selects gated osc, high selects external osc
31	GND	GND
32	ClkIn	External Clk In
33	VddCore	Digital Core Vdd(1.5V)
34	NC	No Connection
35	NC	No Connection
36	NC	No Connection
37	VddPad	Pad Vdd (5V)
38	POR	Chip Global Reset (asynchronous)
39	VddCore	Digital Core Vdd(1.5V)
40	SW4	Diagnostic Output

Table A.1
Chip Pin-out

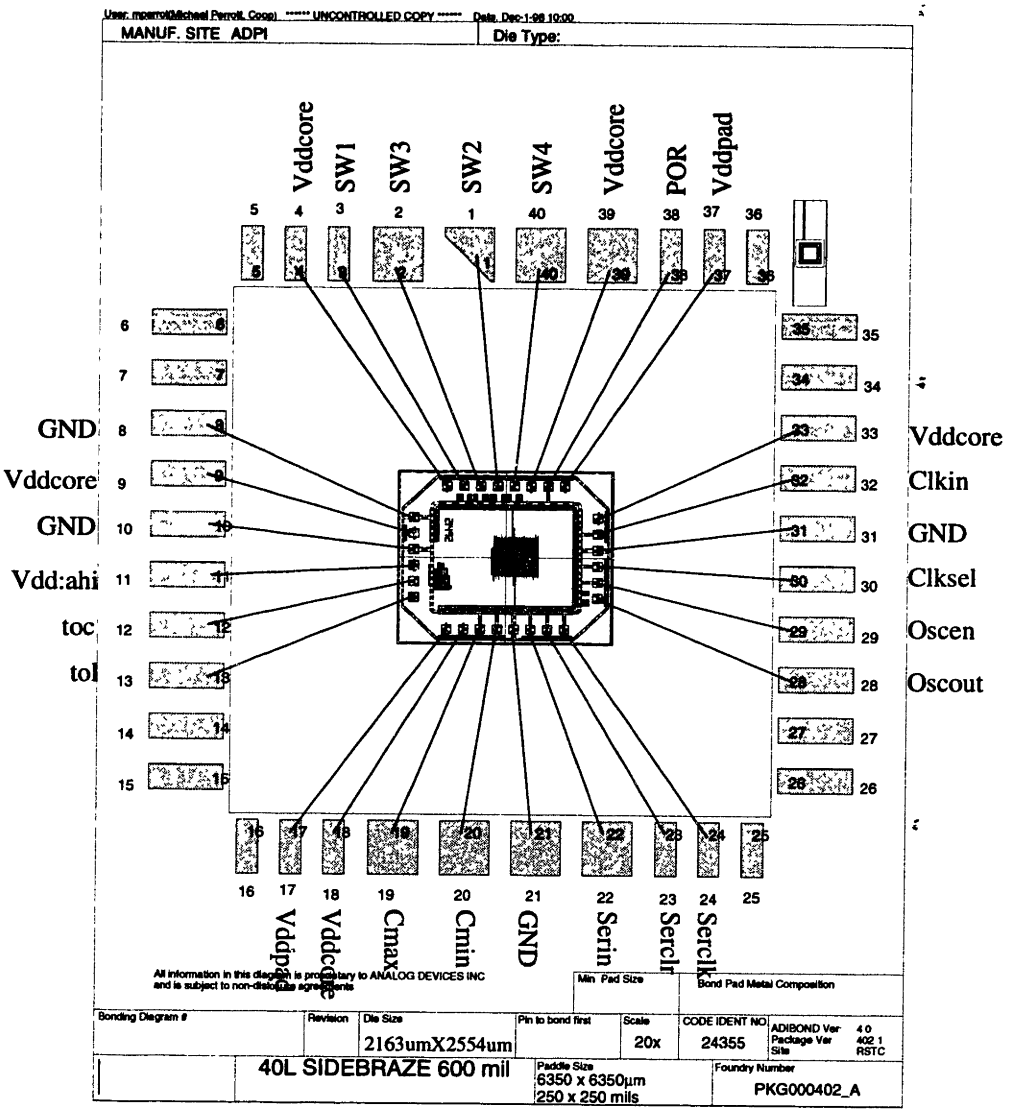


Figure A.2: Bonding Diagram

Appendix B

Serial Interface Code

This is the code used to program in the timing values t_1, t_2, t_4, and t_5 in a serial fashion. Each timing value is a ten bit word, so 40 bits total must be programmed. t_1 is programmed first, with its LSB as the first bit. The first four bits of each word correspond to the four bits controlling the delay line selects of the controller architecture, from LSB to MSB. The last six bits of each word correspond to the six control bits, LSB first and MSB last, of the fast clocked counter control bits. The restriction is that, for each timing value 10 bit control word, the six upper bits corresponding to the fast clocked counter control bits cannot all be zero. At least one of these upper six bits of each control word must be high. The reason for this, as well as conditions necessary for successful programming were discussed in section 3.7.

```
/* mems_driver.c */

/* software for programming MEMS Converter IC */



#include <stdio.h>
#include <conio.h>
#include <ctype.h>
#include <math.h>

#include "parallel.h"
#include "spi.h"
#include "delay.h"

#include "cls.h"
```

```
/* pin definitions for the parallel port        */

/* output pins */
#define kCLK                    (int) 2         /* SerClk */
#define kDATA                   (int) 3         /* SerData */
#define kSTB                    (int) 4         /* _SerEn */
/* #define kPOR                 (int) 5         /* POR */


#define kLPT 1


/* define to invert the spi signals since the board has */
/* inverting buffers */

#define SPI_INV



void loadRegister(unsigned char ser_add, unsigned char ser_data);
unsigned char loadField(unsigned char field_number,unsigned char
                        field_value);


main()
{
        int     done=kFalse;

        unsigned char time1=128;
        unsigned char time2=0;
        unsigned char time3=2;
        unsigned char time4=8;
        unsigned char time5=32;

        unsigned char field_number=0;
        unsigned char field_value=0;

        unsigned char valid_value;

        unsigned char reg_data;
        const unsigned char field_mask_1 = 0xFF;
        const unsigned char field_mask_2 = 0xFF;
```

```
        const unsigned char field_mask_4 = 0xFF;
        const unsigned char field_mask_5 = 0xFF;
        /* select which LPT port */
        pSelectLPT(kLPT);

        /* select clock polarities for MOSI, SCLK */
        /* these are set to 1 (inverting) due to inverting level */
        /* shift */
        /* circuitry on the board */
#ifdef SPI_INV
        spiClkPol(1),
        spiMOSIPol(1);
#else
        spiClkPol(0);
        spiMOSIPol(0);
#endif

        /* initialize the spi interface */
        spiInit(kDATA,10,kCLK);

  while(done == kFalse)
  {

        pDataDirIn();
        pDataDirOut();

        cls();


        printf(" 1: T1          (0-255)\n");
        printf(" 2: T2          (0-255)\n");
        printf(" 3: T3          (0-255)\n");
        printf(" 4: T4          (0-255)\n");
        printf(" 5: T5          (0-255)\n");
        printf("Enter Field Number to Modify: ");

        scanf("%d",&field_number);

        if (field_number >=1 & field_number <=5 ) {
          printf("\nEnter Value for Field #%d: ",field_number);
          scanf("%d",&field_value);
```

```
valid_value=0;

switch (field_number)
{
        case 1:
                reg_data = (field_value & field_mask_1);
                if (field_value>=0 && field_value<=255){
                        valid_value=1;
                        time1=reg_data,
                        }
                break;
        case 2:
                reg_data = (field_value & field_mask_2);
                if (field_value>=0 && field_value <=255) {
                        valid_value=1;
                        time2 = reg_data;
                }
                break;
        case 3:
                reg_data = (field_value & field_mask_4);
                if (field_value>=0 && field_value <=255) {
                        valid_value=1;
                        time3 = reg_data;
                }
                break;
        case 4:
                reg_data = (field_value & field_mask_4);
                if (field_value>=0 && field_value <=255) {
                        valid_value=1;
                        time4 = reg_data;
                }
                break;
        case 5:
                reg_data = (field_value & field_mask_5);
                if (field_value >= 0 && field_value <= 255) {
                        valid_value=1;
                        time5 = reg_data;
                }
                break;
}
```

```
    if (valid_value == 1) loadRegister(time1,time2,time3,time4,time5);
         } else if (field_number == 0) {
         } else {
                done = kTrue;
         }

  }
  return(0);
}


#define kSign   1


void loadRegister(unsigned char t1,unsigned char t2,unsigned char t3,
unsigned char t4,unsigned char t5)
{

/*
        The TX data is sent MSB first:

        A7 A6 A5 A4 A3 A2 A1 A0 D7 D6 D5 D4 D3 D2 D1 D0

*/


#ifdef SPI_INV
        pSetPin(kSTB);
#else
        pClrPin(kSTB);
#endif

        spiWriteRev(t1);
        spiWriteRev(t2);
        spiWriteRev(t3);
        spiWriteRev(t4);
        spiWriteRev(t5);
#ifdef SPI_INV
        pClrPin(kSTB);
#else
        pSetPin(kSTB);
```

```
#endif

}
```

THESIS PROCESSING SLIP

FIXED FIELD: ill _____ name _____

index _____ biblio _____

► COPIES: (Archives) Aero Dewey (Eng) Hum

Lindgren Music Rotch Science

TITLE VARIES: ► ☐ _____

NAME VARIES: ► ☑ Edward

middle name

IMPRINT: (COPYRIGHT) _____

► COLLATION: 92P

► ADD: DEGREE: _____ ► DEPT.: _____

SUPERVISORS: _____

NOTES:

cat'r. date

page

► DEPT: E.E. ► J156

► YEAR: 1999 ► DEGREE: S.M.

► NAME: MENINGER, Scott