UNIVERSITY OF CALIFORNIA

Los Angeles

Oscillator-based Touch Sensor for Mobile Applications

A thesis submitted in partial satisfaction
of the requirements for the degree Master of Science
in Electrical Engineering

by

Li-Ting Huang

2012

ABSTRACT OF THE THESIS

Oscillator-based Touch Sensor
For Mobile Applications

by

Li-Ting Huang

Master of Science in Electrical Engineering
University of California, Los Angeles, 2012
Professor Frank M. C. Chang, Chair

In this thesis, an oscillator-based touch sensor system is presented. Existing technologies for touchscreen sensors include resistive, capacitive, infrared, and optical imaging; however, none of them is suited for remote sensing in mobile applications. The goal of this thesis is to explore the implementation of a remote sensor system targeted for mobile applications.

The proposed design uses an input-loaded oscillator that senses the ambient capacitance. The variation in ambient capacitance alters the operating frequency of the oscillator. A digital processing unit analyzes the change in frequency, and makes detection based on certain threshold.

The design is achieved by 65nm technology, having a core area of 335um x 230um, power consumption of 2.5mW, and runs on a clock frequency of 16MHz. The system's sensitivity is one order of magnitude higher than current touchscreen technologies, requiring a minimum ambient capacitance variation of 50fF.

The thesis of Li-Ting Huang is approved.

Lei He

Tatsuo Itoh

Frank M. C. Chang, Committee Chair

University of California, Los Angeles

2012

Table of Contents

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

1.1 Motivation

With the widespread use of smartphones and tablets, touch screens have become a very common way for humans to interact with machines. In the past, usually a keyboard or a mouse is required for users to establish an interface so as to communicate with the computer. Nowadays, with the technology becoming more and more advanced in scale, and users preferring devices in more compact and portable form factors, touch screens grow in popularity as an efficient solution for mobile applications. Touch screens removed the need for extra peripherals, and the integrated keyboard made the portable electronics even more compact and lightweight. In addition, because spare space is no longer required for physical keyboard, we can now have a much larger screen, and thus the readability and overall user experience has greatly improved through the wide use of touch screens.

Another factor that helped with the widespread of touchscreens is that most touchscreen patents were filed in the 1970s and 1980s, and due to the patent lifespan of 20 years, they have expired. Many companies developing the touchscreen technology no longer suffer from the expense of royalties. E. A. Johnson proposed the first touchscreen display using the “touch wire” concept in 1965 [1]. It uses passive conductive “touch wires” connected by insulated fine wires in a grid. When a user touches the “touch wire”, an operator then detects the capacitance variation, and senses the location of the touch. Through the past several decades, touchscreen technology has

been developed in many other implementations [2]. The renowned technologies include resistive touch, capacitive touch, surface acoustic wave (SAW), infrared, and optical imaging [3]. Each of these technologies has their advantages and drawbacks. The most common touchscreen technology used nowadays on mobile applications is capacitive touchscreen. However, this technology suffers from many aspects which users are unhappy about. Examples include, fingerprints on the screen, unresponsive when the hands are wet [4]. Problems with current capacitive touchscreens have inspired a remote-sensing solution. Using a remote-sensing mechanism eliminates the disadvantage of fingerprints on the screen, and if the mechanism is sensitive enough, the problem with wet hands can also be removed. In addition, remote sensing allows for a more flexible gesture sensing operation. The current touch technology uses a 2-dimensional sensing interface. With remote sensing, a 3-dimentional sensing interface can be achieved. Users not only can do a point-wise selection, a virtual cursor can be implemented and enhance the functionality one can realize. The increased flexibility and complexity will enable a much more diverse user experience.

1.2 Touchscreen Anatomy

The implementations of touchscreen displays involve development and assembly of many discrete components. A typical cross-section of a touchscreen system is shown in Figure. 1.1. The main contributing elements include the front panel, touch controller, touch sensor, liquid crystal display, and system software. The front panel serves as a protective shield for touch sensors. It often keeps either the moisture or outside obstacle from intervening the touch sensing detection. For some touch technology (e.g. infrared), it also functions as a cavity for touch sensor

Figure 1.1. Touchscreen system structure [5].

to be installed. The touch-controller lies between the front panel and the touch-sensor [6]. The controller is mostly a micro-controller based chip, which translates the touch action to a system language that embedded system software can understand [7]. The touch-controller is sometimes embedded on the system circuit board, or other times it is fabricated on a flexible printed circuit (FPC) that attaches to the touch-sensor itself [8]. The touch-controller is a system in operation by itself, and that is why it is solely developed by a company dedicated to the touch-controller design. The touchscreen sensor is a glass panel that has touch responsive sensors built on top of its surface [9]. This glass panel is often called the cover glass. The touch responsive material is often electrical conducting. Depending on the different technology used, the actual materials used and the arrangement of components on the cover glass would be different. The most popular technologies are resistive touch, capacitive touch, infrared, and optical imaging [10]. The detailed implementations of these technologies will be discussed in chapter 2. The touch-sensor

spread the whole area of the screen where touch action are desired to be sensed. The touch actions are usually sensed by a change in voltage or signal by the touch-controller, and then calculate the location of the finger [11]. The data is then sent to the embedded system software and implement the desired instruction communicated by the user. The liquid crystal display lies underneath the cover glass. This LCD is usually the same as the traditional display. However, due to the extra cover glass lying on top of the liquid crystal display, the requirement for light transmission and reflection of the touchscreen displays is higher. In order to have the same level of readability for the modern touchscreens, the brightness and resolution are demanded to meet a tougher specification. The last component of the touchscreen system is the embedded system software. The system software works together with the touch-controller to communicate to the operating system. The software helps to convert the touch action to an operation similar to a click on the mouse or typing on a keyboard in order for the device operating system to understand [12]. This software is sometime sold together with the operating system on a mobile device, or it can also be distributed separately and as an add-on to be installed independently.

1.3 Organization Of Thesis

Over the past few decades, many different technologies have been developed for the implementation of touchscreen systems. The different technologies are used for a different purpose in the design of touchscreens. In chapter 2, an overview of the popular touchscreen technologies will be given. These technologies include resistive touch, capacitive touch, infrared, and optical imaging. Detailed implementation of each of these technologies and their advantages and disadvantages will be discussed in the divided individual sections. The applications that

utilize these technologies will also be explored in chapter 2.

Chapter 3 will present the detailed structure and implementation of the proposed touch sensor design. Section 3.1 talks about the top level architecture and schematics, and where the analog and digital domains are divided. Section 3.2 introduces the implementation of the custom-designed block, which is the 13-bit synchronous counter, in the digital domain. Section 3.3 discusses the design of the digital computing unit, which was implemented using synthesized behavioral Verilog code and automatic place-and-route tool. Section 3.4 describes the self-test modes implemented for the digital computing unit to facilitate testing. Section 3.5 presents the design of the USART interface.

Chapter 4 will introduce the methods and tools used in the synthesizing flow. The digital computing unit and the USART interface are both first coded in Verilog representing its behavior. The behavioral code is then synthesized using RTL Compiler to generate gate-level netlist. The Cadence Encounter Tool is utilized to perform place and route and outputs a layout gds file. The layout is brought back into Cadence Design System to perform post-layout simulation and verification.

Finally, Chapter 5 presents the results and conclusion for the thesis. The performance and limitations of the proposed design is explored and discussed. The final chapter also investigates future integration and utilization of the proposed technology.

CHAPTER 2

Existing Touchscreen Technologies

2.1 Resistive Touch Sensor

The resistive touchscreens uses an overlay layer on top of the liquid crystal display. This overlay is composed of two interacting components. The top layer is flexible and the bottom layer is rigid, and the two layers are separated by an insulating substrate. The down side of the top flexible layer and the top of the bottom rigid layer is coated with a transparent metal oxide of Indium Tin Oxide (ITO) [13]. The ITO cover creates a surface with a predefined resistance distribution [14]. To sense the location of the finger selection, a voltage is applied either across X- or Y- direction. When a position of the resistive touch sensor is pressed, the flexible top layer gets bended and the two conducting surfaces touch each other and form a short circuit. The touch-controller can sense the position of the finger using calculated X and Y position based on voltage divider method [15]. A popular method called 4-wire resistive technology is discussed as follows:

2.1.1 4-Wire Resistive Technology

This technology uses uniform ITO distribution. Conducting silver bars are coated along the edges of the two conducting surfaces. A voltage of +5V is applied to the back layer first, when the user touches, a short circuit is formed [16]. The touch-controller has a voltmeter built in, after

applying a +5V difference to the back layer, the controller then switch on the voltmeter to measure the voltage difference on the top layer. Depending on the position of the touch, a different resistance ratio is formed, and each reading on the voltmeter will correspond to a specific position on the X-axis. Once X-position is determined, the touch-controller then switchs to apply +5V voltage difference across the silver bar on the top layer. The voltmeter then measures the voltage at the bottom layer and translates the voltage reading to a Y-axis position. The drawback of the 4-wire resistive technology is that both the top and bottom layers are used to apply a voltage across it [17]. The consistency of resistance gradient is very critical for accurate finger positioning. However, the outer layer is constantly exposed to environment factors such as humidity, temperature, and dust. In addition to that, the wear and tear of finger touching the screen also destructs the uniform coating of ITO, and thus the resistance distribution is deviated from ideal. This will create a less accurate reading on the detection of X- and Y-position.

In general, resistive touchscreen is a more economical solution, and it is also more durable because the sensing mechanism happens in an inner layer. However, the transparency of these resistive touchscreens is less desirable. The material used for the flexible top layer limits the clarity on these displays. In addition, the flexible top layer is rather fragile and can possibly be damaged by sharp objects. The resistive touchscreens are often used in applications where cost is a determining factor but clarity is not very crucial [10]. These applications include food-service computers, medical monitoring devices, industrial process control, and handheld devices.

2.2 Capacitive Touch Sensor

Capacitive touchscreens uses a clear glass panel that has a layer of conductive material coated to the bottom of the glass. Fingers are partially conductive. When a finger touches the screen, it induces charge accumulation in the local area across the glass on the conductive material coating. Circuits located on the four corners of the screen measure the amount of charge accumulated and send these information to the touch-controller for post processing [18]. The touch-controller will be able to analyze each individual circuits data and pinpoint the location of the touch.

Capacitive touchscreens rely on the conductivity of human fingers; therefore, it cannot be replaced by stylus, and gloved fingers become unresponsive as well. However, capacitive touchscreens have many other advantages, which make capacitive technology among one of the most popular implementation in the current touchscreen market. The capacitive touch sensor is highly transparent. Even with the coated cover glass lying on top of the traditional liquid crystal display, the readability is still maintained. The technology provides great clarity, and at the same time, maintaining good sustainability to environmental factors because the sensor lies underneath the cover glass and is at all times getting protected. There are two main implementations of capacitive touchscreens, surface capacitive technology and projected capacitive technology.

2.2.1 Surface Capacitive Technology

In this technology, the bottom side of the insulating glass is coated with a uniform conductive

coat. When a voltage is applied to the edges of the glass, a uniform electrical field is created. When a finger touches the screen, the induced electrical field caused by the finger rearranges the charge distribution. An amount of charge accumulates on the other side of the cover glass around the local area of the finger [19]. The circuitry at the four corners of the screen can measure the change in capacitance. The touch-controller will use position calculation algorithm and estimate the location of the touch. The downside of the surface capacitive technology is that it is susceptible to parasitic capacitor coupling. A small fluctuation in the near field caused by environmental factors or nearby electric components can easily affect the accuracy of this touchscreen system. This low-cost solution is mainly used in simple applications such as industrial controls and public kiosks.

2.2.2 Projected Capacitive Technology

Projected capacitive technology uses conductive grid instead of a uniform surface for capacitive sensing. One variation of the conductive grid is by etching a crisscrossed conducting layer underneath the cover glass. The grid helps to achieve a better resolution for position sensing. When a finger touches the screen, only the capacitance around the proximity area changes. This grid helps the circuitry around the edge of the screen to better pinpoint the location of the touch. Each position on the screen is now given a coordinate, and when there is a touch action, the circuitry not only senses the difference in capacitance, it also attaches a coordinate of where the touch action happened [20]. Another variation is by forming the X-Y grid using two layers of conductive material, one in X-direction and one in Y-direction. This implementation achieves even better resolution because the X and Y layers are independent of each other [21]. When

trying to pinpoint the location of the touch, an X position is determined independent of the decision on Y-axis. The characteristics of good transparency and clarity along with high resolution allow the projected capacitive technology to be widely used in current products. The applications include a wide range of operating conditions, including industrial usage, outdoor handheld devices, and high-end mobile electronics.

2.3 Infrared Touch Sensor

The infrared touch sensor uses infrared LED transmitter and receiver on the edge of the screen. These transmitters and receivers are housed inside the bezel of the front panel. Due to the minimum required height for these transmitters and receivers to operate, the applications of this technology often has larger than normal front panel, and because of this reason, the variation in form factor is limited. The operation uses paired transmitters and receivers on the opposite side of the screen. When an object makes a touch action, the infrared beam is blocked, and the corresponding receiver will lose its connection from the master transmitter [22]. A touch-controller will detect this lost in connection and report the position of the finger touch [23]. The infrared touch technology provides very high clarity on the screen quality because nothing is overlaid on top of the liquid crystal display. Another advantage is that no conductive material is required to sense a touch, meaning any object such as stylus, pen, and gloved hands can be used. However, the drawback is that any undesired substance can cause a false reading. In addition, the resolution is limited by the number of infrared LEDs placed along the edges. Infrared touch technology is a low-cost solution, and it is highly durable. The applications include low-end mobile devices and game consoles, and outdoor applications that cannot rely on conductors to

sense a touch [24].

2.4 Optical Imaging Touch Sensor

The optical imaging touch sensor is a relatively new technology. It is similar to the infrared touch sensor technology but uses a newer and better technology elements. Instead of an array of infrared LEDs placed along the edges of the screen, light strips are installed along three edges of the screen. Several image sensors are placed at corners. The light strips are in the camera's field of view [25]. When finger is present, it shows up as a shadow in the camera's image sensor. The touch-controller can then combine the data from two or more cameras and use triangulation to determine the location of the finger [26]. Optical image sensing provides excellent screen clarity and transparency because no extra material is coated on top of the cover glass. In addition, this technology is highly scalable and any object can be used to activate the touchscreen. The optical imaging touch sensor technology is widely used in large-scale touchscreens, such as TVs and public information panels. The drawback is that space is needed to house the image sensor and light strips; therefore, the form factor is limited. For these applications, a bezel is needed on top of the screen. In addition, the resolution of these optical image touch sensors is limited by the specification on the camera. Also, another drawback is that undesirable objects in the camera's field of view can cause false readings.

CHAPTER 3

Proposed Oscillator-based Capacitive Touch Sensor Design

3.1 Overview of the Proposed Touch Sensor System

The goal of the proposed design is to implement a touchscreen system using remote sensing and the system must be suitable for mobile applications such as cellphones and tablets. The design concept originated from Professor Léon Theremin's invention of a musical instrument called "theremin" in 1928 [27]. This electrical instrument uses two antennas to sense the location of hands and alter the amplitude and frequency of the frequency oscillator. The actual implementation is done by capacitance sensing. When hands are moving in the field, it changes the capacitance characteristics on the frequency oscillator, and these alterations change the frequency and amplitude of the oscillator [27].

Inspired by the Theremin, the capacitive touch sensor also uses the property where the change in input capacitance changes the frequency on the oscillator. The proposed design uses a ring oscillator where the relationship between the frequency and input capacitance is shown by equation (3.1). The frequency is an inverted function of input capacitance, given the current

$$frequency = \frac{1}{2nR(C + \frac{\triangle C}{n})} \tag{1.1}$$

technology, the input capacitance of ring oscillators are in the orders of tens of farads [29]. Therefore, even a very small amount of change in capacitance can cause a visible effect in the

frequency difference. The top-level system architecture is shown in Figure 3.1.

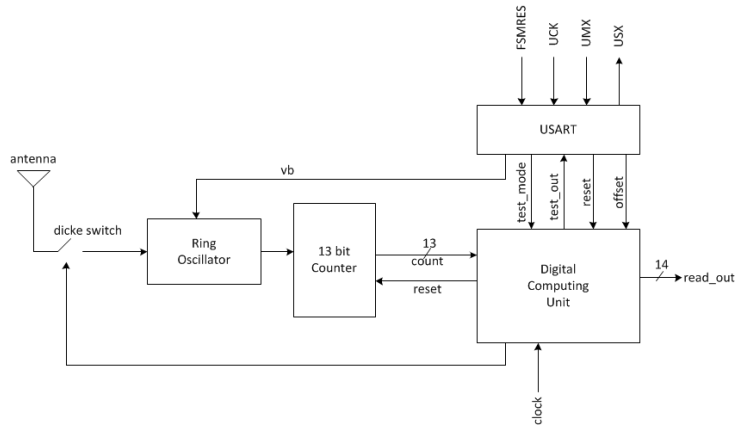

Figure 3.1. Top Level System Architecture

The antenna senses the ambient capacitance and loads it to the input device of the ring oscillator. The path from the antenna to the ring oscillator is gated by a dicke switch. As shown by Figure 3.2, the dicke switched antenna allows the touchscreen system to sense once with the ring oscillator free-running, and once with the sense capability enabled. These two reading are then subtracted to compare the difference in frequency, and also remove the flicker noise in the 1MHz range. The control signal for the dicke switch is configurable, and is provided by the digital block of this system.

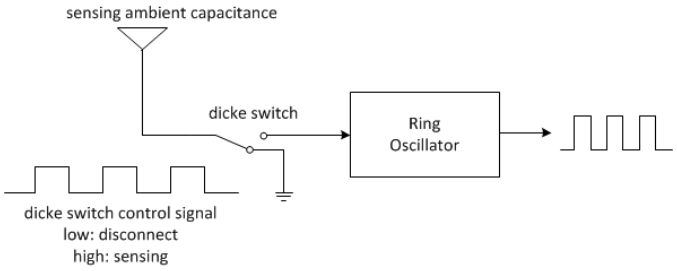

Figure 3.2. Dicke Switch and Antenna

The output of the ring oscillator feeds to a 13-bit digital counter. As illustrated by Figure 3.3, the counter resets every time when the dicke switch swithes. The digital counter counts the number of pulses in a given window of time. When frequency is lowered due to the presence of a finger, the count at the counter output will be different. By setting a threshold for detection, we can determine whether there is a finger or not, and by the value returned, the distance from the sensor can be calculated. The actual value used for threshold and distance calculation are uncertain currently, it will be determined once the chip is fabricated, and values are measured in a lab environment.

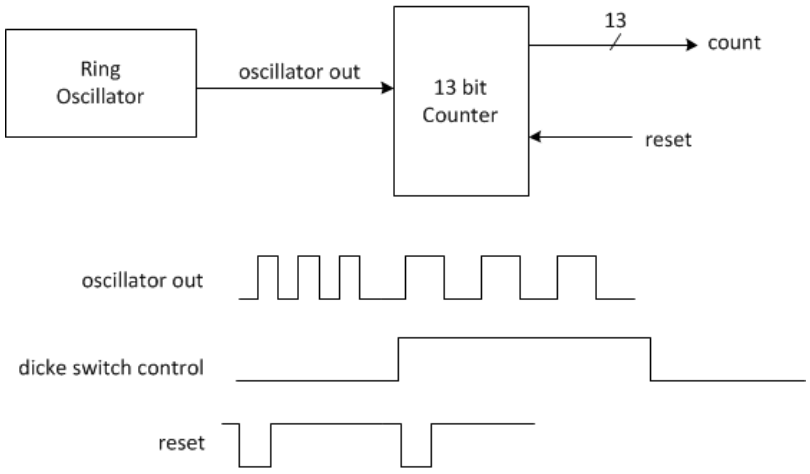

Figure 3.3. Oscillator Output and Counter Reset

The digital computing unit samples the counter output at the same frequency as the dicke switch. As depicted by Figure 3.4, every time before the counter resets, the digital computing unit grabs the count at the counter output, and saves it to a local register. At the end of every cycle, the digital computing unit takes the difference between the free-running count and the sensed count.

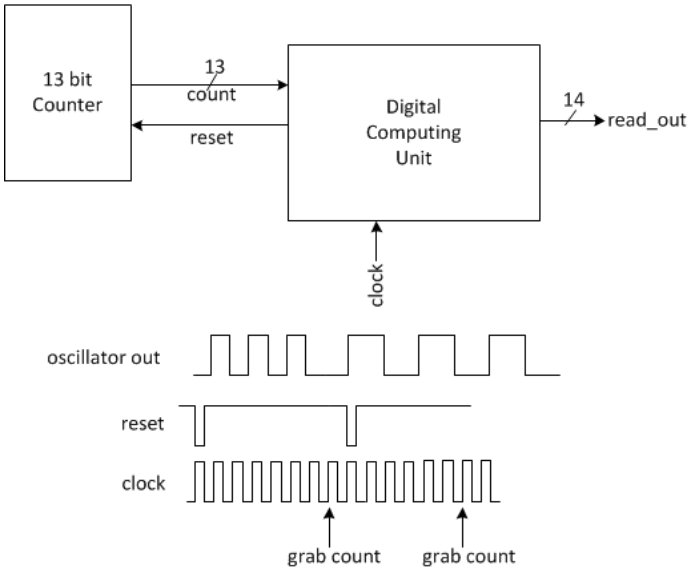

Figure 3.4. Counter Output and Digital Computing Unit

An offset mechanism is implemented at the end of the computing process. The offset block subtracts a configurable 14-bit number from the read-out. The offset accounts for average ambient noise, and static parasitic effects of the entire touchscreen system.

In order to facilitate testing complexity and minimize pin count on the system chip, a USART interface is also designed. The USART uses four pins, FSMRES, UCK, UMX, USX, to communicate to a PC. As shown in Figure 3.5, the control registers, offset, and reset for the digital computing unit are provided by the USART interface. This makes the system highly configurable, and the system overhead is still kept to a minimum.

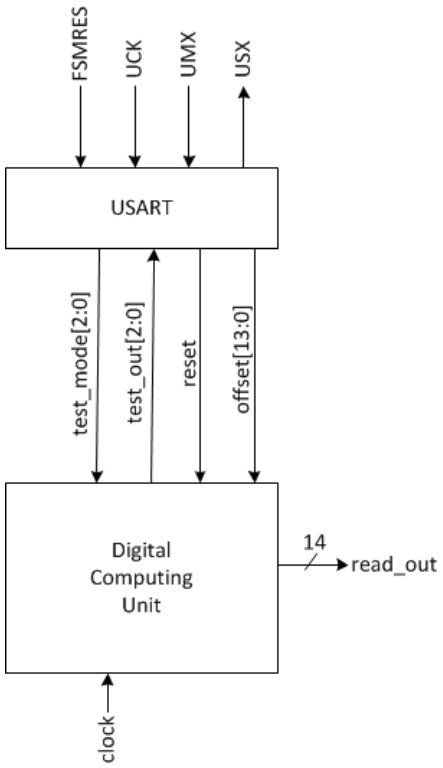

Figure 3.5. USART Interface and Digital Computing Unit Control

The whole touchscreen system is a group project. My main responsibility is on the implementation of the digital section. The digital domain of this system includes the 13-bit counter, the digital computing unit, and the USART interface. The following sections in Chapter 3 will go over the design of each of these three blocks in detail.

3.2 Custom-Designed 13-Bit Counter

The target frequency of the ring oscillator is 2GHz, and our preliminary decision for sample window size is 1 microsecond. As calculated by equation (3.2), these two numbers combined

$$count = frequency(\frac{numbers}{\text{sec}}) \times window(\text{sec}) \tag{1.2}$$

will require the counter to be able to count up to 2000. The required number of bits corresponding to 2000 is 11, by adding 2 bits for system flexibility and headroom, the counter design will support 13-bits count.

The 13-bit counter is implemented based on D flip-flops. The target requirement for this counter design is speed. The speed of the digital counter limits the speed for the ring oscillator. In order to meet the speed requirement, TSPC logic is used to implement the D flip-flop design. TSPC stands for true single-phase clocked logic family. In this logic family, only one clock is needed for the design, and minimum number of transistors is required to implement the logic [30]. The schematics of the TSPC D flip-flop is shown in Figure 3.6. The D flip-flop is composed of two segments, the master and the slave. When clock is low, the data at input D is passed by the master logic and stored in the intermediate node. When clock goes high, the data previously stored at the intermediate node is passed through the slave logic to the output Q. The flip-flop is positively triggered by a rising edge of the clock.

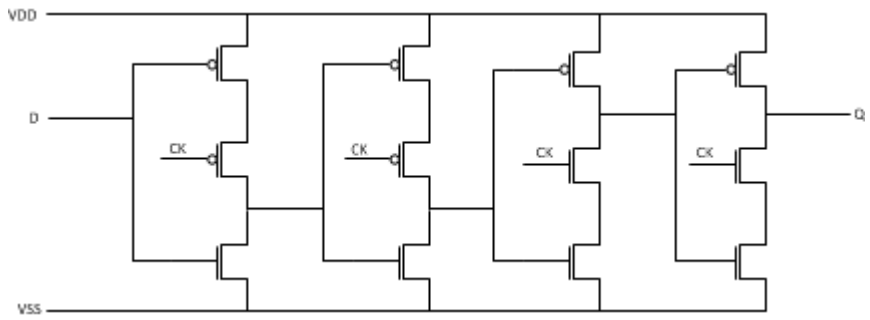

Figure 3.6. TSPC Logic D Flip-Flop

The counting mechanism is accomplished by toggling the next significant bit if all the previous bits are one. The pattern is illustrated by Table I. The first pulse toggles the least significant bit. For all the successive bits, it only gets toggled when all the previous bits are ones. In order to achieve this logic, a toggle flip-flop has to be implemented first. A toggle flip-flop is realized by inserting an XOR gate with the two inputs connecting to a toggle bit T and feedback Q output, as shown in Figure 3.7. The truth table in Table II shows that when T = 1, output Q flips, and when

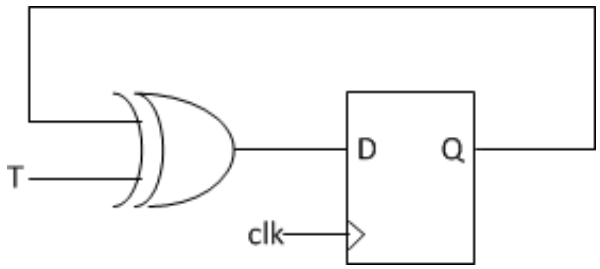

Figure 3.7. Toggle Flip-Flop Constructed by D Flip-Flop

Count	b0	b1	b2	b3	b4
0	0	0	0	0	0
1	1	0	0	0	0
2	0	1	0	0	0
3	1	1	0	0	0
4	0	0	1	0	0
. . .					
7	1	1	1	0	0
8	0	0	0	1	0

Table I. Digital Counter Pattern

T	Q	Qnext
0	0	0
0	1	1
1	0	1
1	1	0

Table II. Toggle Flip-Flop Truth Table

T = 0, the D flip-flop keeps its previous state. The least significant bit on the 13-bit counter uses clock as its toggle input because every time when clock is high, the LSB toggles its output. For all the successive counter output bits, the toggle bit uses a carry signal. The carry signal is an ANDed output of all previous counter output bits. The carry bit only switches high when all the previous bits are ones, which means, the current D flip-flop only toggles when all previous outputs are high. The top level counter architecture is shown in Figure 3.8.

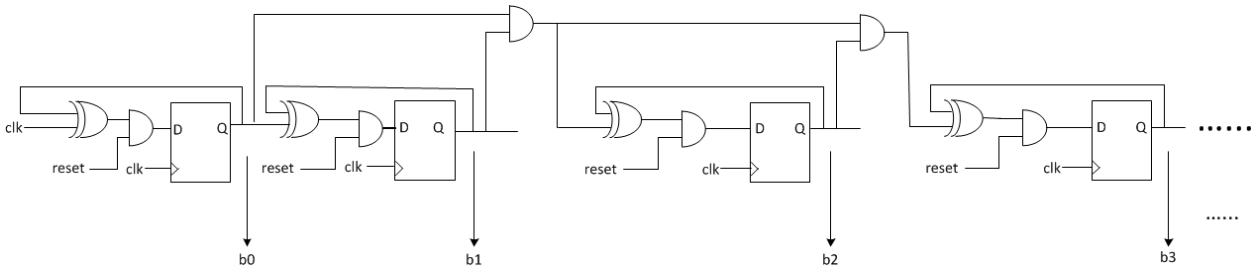

Figure 3.8. Top Level Counter Architecture

The reset of the counter is implemented asynchronously [31]. As shown in Figure 3.8, the D input to the flip-flop is gated by a 2-input AND gate connecting to the reset signal. When reset is zero, it forces the flip-flop to output a zero, and when reset is one, the flip-flop operates in its normal mode.

The entire 13-bit counter is hand-drawn in schematics and layout using Cadence Virtuoso. The flow involving custom design of this block is shown in Figure 3.9. A bit-by-bit construction in schematic is verified individually in Cadence Analog Design Environment. A bit-wise simulation is verified before the counter designed is duplicated for 13-bit operation. Checking the functionality in each design phase ensures that the top-level design is accurate and robust. After the schematics are verified in functionality and performance, the design layout is then hand-drawn. While composing the layout of a circuit, it is very crucial to analyze the critical path of the circuit, and ensure that the most efficient path is allocated for this critical path in layout. The layout of the circuit must pass the DRC (Design Rule Check) test. DRC verifies that the layout drawn can be properly fabricated by a factory. It checks for the minimum spacing between

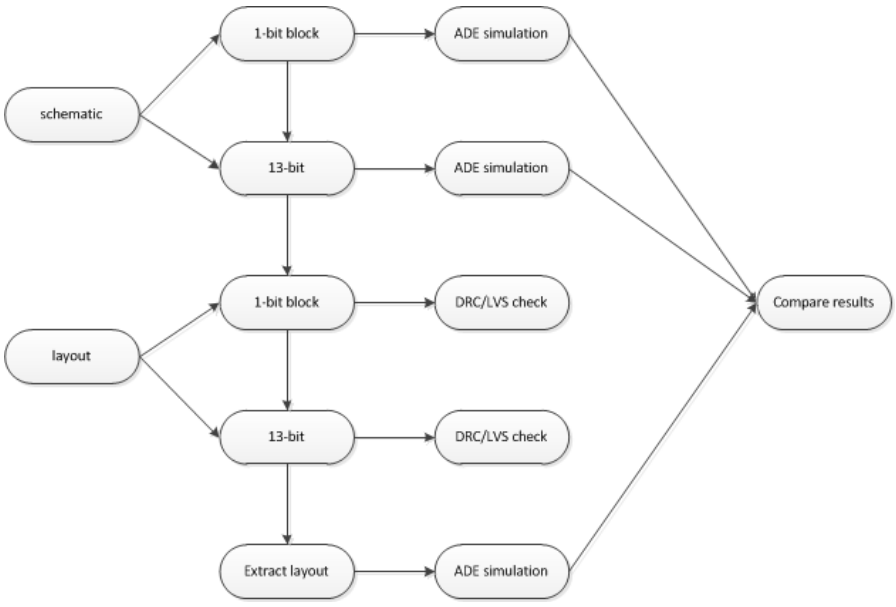

Figure 3.9. Counter Custom-Design Flow Chart

metals, width of wires, and metal densities in each layer. Doing frequent DRC checks ensures that mistakes be found early in the design phase. If the errors are found late in the design phase,

and large amount of metals are already drawn, it becomes very difficult to make a modification. Another very important check is LVS (layout versus schematics). This ensures that the circuitry drawn by layout matches the functionality and connections in the schematics. After the layout has been finished, and DRC, LVS passed, the layout gets extracted with parasitic capacitances and brought back into schematics with parasitic included. Another round of circuit simulation is done using the extracted view to verify that the functionality and system performance is still within specification.

3.3 Synthesized Digital Computing Unit

The digital computing unit samples the counter output at clock transition before counter reset, take the difference in counts, adjusts offset, and outputs the difference in read-out. The relationship between the digital computing unit, counter and USART are illustrated by Figure 3.10. The controlling signals are provided by USART and is set using a serial interface from the PC. The functionality of the digital computing unit is characterized by behavioral coding in Verilog. The project has gone through two versions of Verilog module architecture, and each of them will be discussed in the following sections.

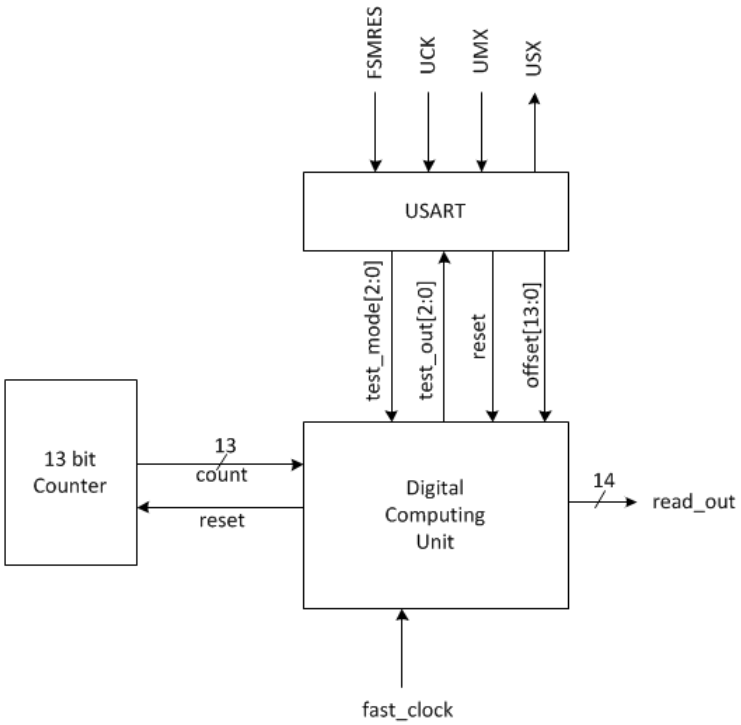

Figure 3.10. Counter, Digital Computing Unit, and USART Relationship

3.3.1 Block-Wise Implementation

The first implementation of the digital computing unit is by using discrete modules to fulfill the functionality in each step, and the modules are then combined by a top module. The architecture is depicted in Figure 3.11. The counter outputs are first demux-ed into two paths, one storing the free-running count, and one storing the sensed count. The demux-ed outputs are clocked and

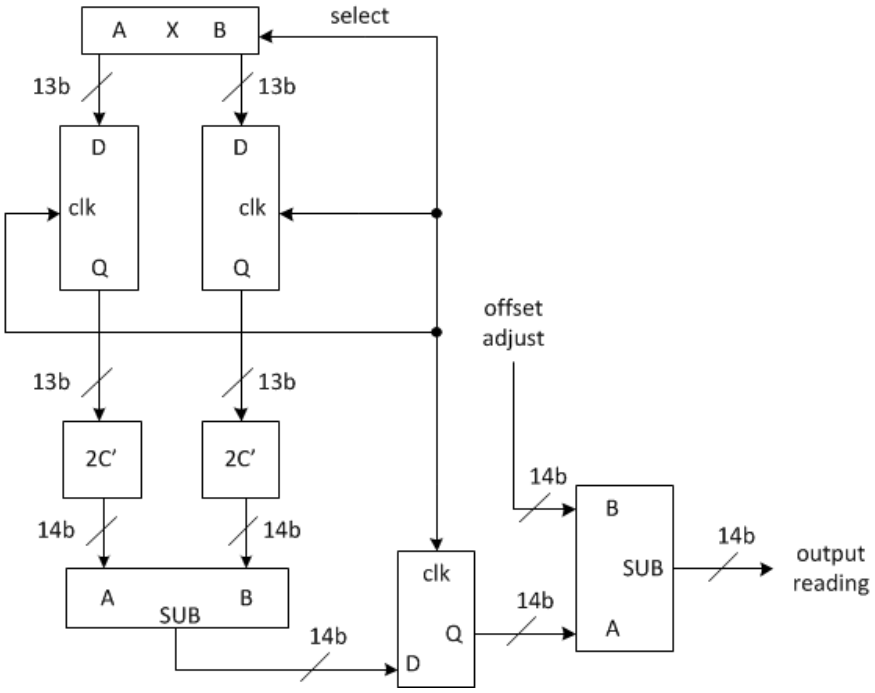

Figure 3.11 Digital Computing Unit Design Version 1

stored by two registers triggered by the same clock frequency. The registers ensure that the data at this point are synchronized. Before the data performs subtraction, it has to be converted into two's compliment form for proper handling of negative numbers and overflow. The output of the subtractor is again stored in a register before it is adjusted for offset. The final output register clocks the output of the second subtractor that adjusts for system offsets. Using this architecture, a module for each individual block is solely designed and verified with a corresponding testbench. Once each individual block is verified in functionality, the top module combines all the sub-modules, and verifies for system functionality again. During the implementation of this design, many difficulties revealed. Because each individual block is implemented using a dedicated module, they each have their own clock frequencies. Some are clocked by positive

edges, and some are clocked by both edges. The problem arises when all the modules are combined, and it becomes very difficult to properly time the top module. Timing violations become very hard to track down the hierarchy. One signal is triggered by another in another hierarchy and when one violation is fixed then another violation relating to a different signal shows up. Even though the design was verified for functionality in the end, the implementation is not very "clean" and robust. Simulations in ModelSim shows correct results but with process variations and interaction with other components, these extra uncertainties make the design rather susceptible to non-idealities. After serious consideration upon this issue, this design is consequently scrapped.

3.3.2 Clock Divider Implementation

The second version of the digital computing unit uses clock divider architecture. The entire block is implemented using only one Verilog module. The Verilog module takes in one clock, and divides it by 16 times. The divided by 16 clock is sent to the dicke switch as the control signal. As shown by Figure 3.12, the top module is a finite state machine. Initially, all the registers are set to zero. The finite state machine increments when the always block is triggered by positive edge on the external clock signal. When the count increments 8 times, the output clock to the

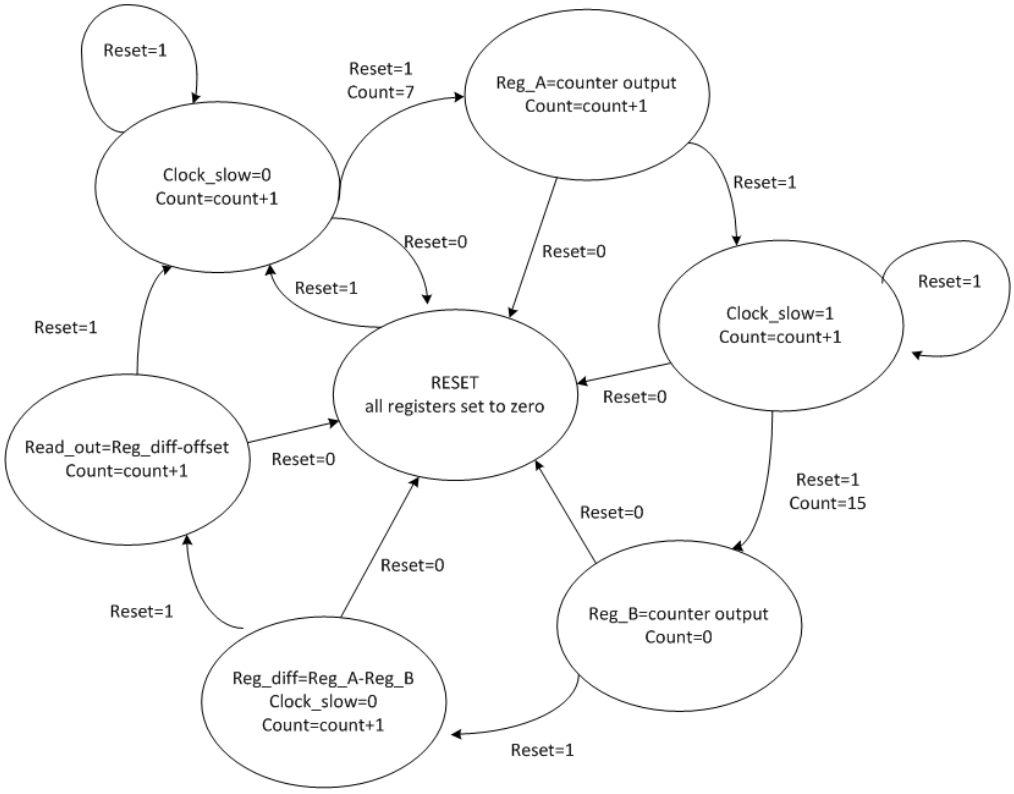

Figure 3.12 Digital Computing Unit Design (Version 2) Finite State Machine

dicke switch flips. For the operation of the count comparison, the digital computing unit samples the counter output at the 7th pulse, and saves the count in register A, this is the oscillator free-running count. At the 15th pulse, it samples the counter output again and saves the data in register B, this is the sensed count. At the next rising edge, a “diff register” saves the difference of register A and B. At the next edge again, the digital computing unit subtracts offset from difference register and outputs the readout.

3.4 Self-Test Modes

The digital computing unit has a 14-bit output. While the block is running, it becomes very difficult to monitor the data output and covert the 14-bit binary output to a decimal number in real-time. When running simulation, the designer has to first find out what the free-running count and sensed count are, take the difference, and then verify if the number matches what the system is outputting. By having such a long data bus, visually verifying the functionality is not very practical. In order to test the functionality accurately and efficiently, self-test modes are created. The self-test modes allow the digital computing unit to enter a state where it tests the functionality itself. The states of the test modes are specified by a 3-bit "test_mode" signal. The operation of the self-test modes are described by Table III. Test_mode 000 sets the digital computing unit to run in normal mode. Test_mode 001 sets the registers A and B to pre-programmed numbers, takes the offset communicated from the USART interface, computes the difference, and compares this number with a pre-programmed number. If the two numbers match, the digital computing unit will output a 3-bit "test_out" signal corresponding to the test-mode input. If the test fails, then the test_out signal will output 000. Test_modes 010 and 011 tests the digital computing unit without applying reset. Test_mode 100 puts the digital computing unit partially in normal operation. The objective is to set the oscillator running in constant frequency. If it is in normal operation (test_mode 000), the read_out output should be zero. In test_mode 100, the register B samples the counter output exactly one cycle short of the register A. In other words, if the counter and digital computing unit are both running correctly, the read_out output should provide a fixed number, which represents the constant offset between register A and B samples.

Test_mode	Mode	Reg_A	Reg_B	Test_out
000	normal			000
001	test with system offset	8191	1	001
010	test without offset	1111111111111	0101010101010	010
011	test without offset	1111111111111	0000000000000	011
100	constant count offset			100

Table III. Digital Computing Unit Test Modes

After the chip is fabricated, it has to be verified in the lab environment. When the system is running in real time, even with the test modes implemented, it is still hard to capture the signals at a specific point of time. There is no pause button on a system like this. After the system resets, it keeps running on its own and it is hard to grasp a certain state at a certain point in time. The state in which the system operates at a certain time is undefined and unclear to the test engineer. In order to facilitate testing, a "lock" signal is added. As shown in the finite state machine diagram in Figure 3.13, the system starts with lock signal set to zero. When count is 2 and lock is

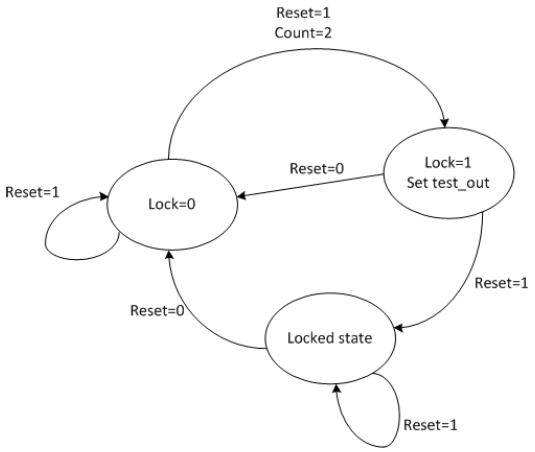

Figure 3.13. Lock Signal Finite State Machine

0, if the output matches the pre-programmed number, then the test_out is set to test_mode number and lock signal to 1. After the test_out loop is executed once, it will no longer be executed again because the lock signal is now fixed to 1. In other words, once the test_out signal gets changed, the test_out signal is locked until the next reset. In a lab environment, the logic analyzer can be easily observed at any point in time for test_out signal verification. The test engineer can observe a static signal instead of a fluctuating signal which state is unclear.

3.5 USART Interface

USART stands for Universal Synchronous/Asynchronous Receiver/Transmitter. The USART is a serial-to-parallel interface, which communicates to a PC through serial interface, and to internal control signals through parallel interface. The handshaking operation between USART and the PC is shown in Figure 3.14. The PC initiates a communication by padding a "1" to a master packet, and sending the data through UMX port. The "1" added before master packet is recognized as the "packet ready" signal. The on-chip USART sees the "packet ready" signal, and

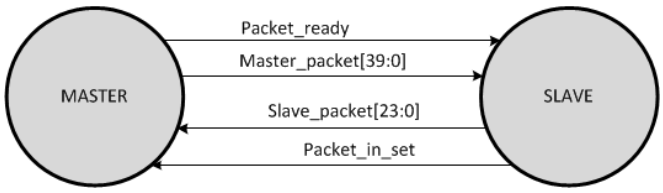

Figure 3.14. USART and PC Handshake

starts to put the master packet into a queue. Once the on-chip USART finishes reading in the whole packet, it starts to implement the functions that are instructed by the PC. As soon as the

slave completes its operation, it instantiates a “1” padded before its slave packet. This “1” added is recognized as the “packet in set” signal. When the PC sees the “packet in set” signal on the USX port, it starts to read in the slave packet stream, and the user can analyze the data that is returned by the on-chip USART to verify whether the instruction was completed successfully.

The master packet has 40 bits, and the construction is depicted in Figure 3.15. The packet is divided into three sections. The 8 MSBs are mode_in bits. The mode-in bits tell the USART which operation to execute. The total number of modes the USART can support is 256. In this application, it only uses 4 modes. The USART block is universal structure, and can be easily

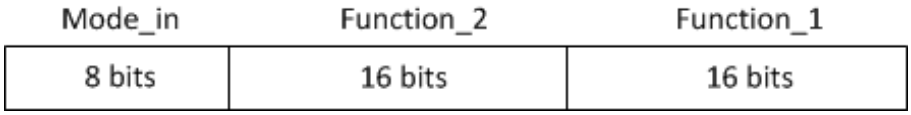

Figure 3.15. Master Packet Distribution

modified for other applications. The next 16 MSBs are function_2 bits, and the last 16 LSBs are function_1 bits. Depending on the mode specified, the USART would execute the instruction defined in the data stream sent in function_1 and function_2.

The slave packet is 24-bit long, and is divided into sections of 8-bit long. The 8 MSBs are set to ones for easier detection. The middle 8 bits returns the mode_in number that was instructed by the PC. The lower 8 LSBs returns the data that was modified in this particular instruction. The only exception is for offset mode, the data return needs 14 bits, so the 14 LSBs returns the data, and the 8 MSBs return the mode_in number instead of all ones.

Table IV shows the instruction of each mode_in defined by the PC. For mode 0, the USART sets the reset bit on the digital computing unit. For mode 1, it sets the test_mode bits for digital computing unit. For mode 2, the USART outputs the test_out bits from the digital computing unit. For mode 3, it sets the system offset value. For mode 4, it sets the internal bias voltage for the ring oscillator in the analog domain. The bias voltage is a constant 1V when in operation. Table V summarizes the slave packets for each mode of instruction.

Mode	Mode_in	Function_2	Function_1
	8 bits	16 bits	16 bits
set reset	0	x	reset bit
set test_mode	1	x	test_mode[2:0]
read test_out	2	x	x
set offset	3	x	offset[13:0]
set bias voltage	4	x	1 or 0

Table IV. Master Packet

Mode	Preceding	Mode	Data
	8 bits	8 bits	8 bits
set reset	11111111	0	reset bit
set test_mode	11111111	1	test_mode[2:0]
read test_out	11111111	2	test_out[2:0]
set offset	3	offset[13:8]	offset[7:0]
set bias voltage	11111111	4	bias bit

Table V. Slave Packet

CHAPTER 4

Design and Verification

4.1 CAD Design and Verification Flow

This chapter explores the design and verification flow used for the digital computing unit. The majority of digital circuit designs nowadays uses the same process to generate the layout. The design and verification are both carried out in each step, and on top of that, the verification is done again at the system level after integration. Figure 4.1 shows the CAD design and verification flow. ModelSim, Cadence Encounter RTL Complier, Cadence SoC Encounter, and Cadence Design System were the tools used in this flow. The details of each step will be explained in this Chapter.

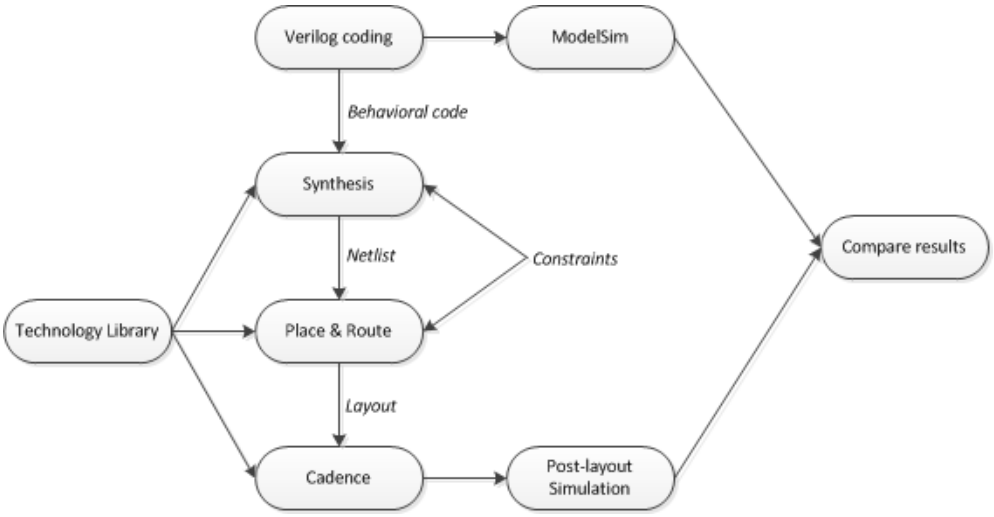

Figure 4.1. CAD Design and Verification Flow

4.2 Verilog Coding and Verification

The implementation of both the digital computing unit and USART were done using Verilog behavioral coding. They both only used one module. The basic architecture of the digital computing unit is to use a divide-by-16 clock divider and carry out the functionality using a finite state machine. The detailed structure and realization was described in Chapter 3.3. The behavior of the USART interface was explained in Chapter 3.5. Both the functionality of the digital computing unit and the USART interface were verified using ModelSim. The testbench uses registers for module inputs, and wires for outputs. The top module is called inside the testbench module. In the initial block, the time domain transient input signals are defined as time increments. For example, at time zero, all registers are set to zero. After 10 time instances, the reset is set to one. After another 10 time instances, the test modes are toggled. In the always block, the clock signal toggles at every time instance, creating a constant frequency signal. At the end of the module, add another initial block to specify when for the testbench to exit simulation. If the stop time is not properly set, the testbench would run consistently can cause trouble in the simulation tool.

One trick for efficient Verilog coding is to use the Replace function in the text editor. When coding in Verilog, inputs and outputs are first defined, and then either registers or wires are assigned to each input and output. When the design gets large, re-typing the signals because tedious and inefficient. Copying and replacing corresponding signal definitions with proper assignments saves design time and makes the code text neater.

ModelSim is the tool used for Verilog code simulation. Both the top level module and the testbench module are imported together into the work library. ModelSim compiles the two

Verilog file together and then the testbench is used to launch the simulation. Once the simulation completes, add all signals to waveform viewer. The waveforms can be viewed as binary numbers or decimal numbers. This allows for significantly easier reading and verification. Once the digital computing unit has been verified with no timing violations, it is ready to be synthesized.

4.3 Synthesis

The tool Cadence Encounter RTL Compiler was used to synthesize and generate the gate level netlist from the Verilog code. Two individual blocks are used in this design, digital computing unit and USART interface. In order for RTL Compiler to operate, a tcl script is used to specify the Verilog input file, clock frequency, output loading capacitance, and timing constraint files to be saved. The Cadence Encounter RTL Compiler loads the tcl script, which calls the corresponding Verilog file and generates the gate level netlist. This design uses the 65nm, regular V_T technology library because the frequency is low, and design constraint is relatively relaxed. The timing constraint is specified in the tcl script indicated by a requirement for clock frequency. When the RTL Compiler runs the script, it generates a gate level netlist that meets the timing requirement, while keeping area and power optimized. Power is not a major problem in this design since the complexity is very low. Even though we aim to create a system with low power, the requirement need not to be taken care of explicitly since the requirement essentially meets itself. However, area is of major concern because the remote touch system is to be placed inside a portable mobile application. Space in these applications is highly limited. The lesser area this touch system occupies, the more likely it would be adapted in future mobile application products.

Since for both the digital computing unit and the USART interface, the complexity is very low, and register count is relatively small. The remaining time it takes to use the gate level netlist and generate layout utilizing automatic place and route tools is short. Another round of gate level simulation was skipped. The design will be entirely verified once the layout is generated and extracted, and then run simulations in Cadence Design System.

4.4 Place and Route

Cadence SoC Encounter tool is utilized to perform automatic place and route. This Cadence tool takes in a gate level netlist and generates a layout gds file. The steps and flow involved in automatic place and route using Cadence Encounter is described in Figure 4.2. Floorplanning is the first step in place and route, and it sets the fundamental constraints and requirements for this design. When a small area is specified for the design, the density in between individual components becomes higher, and that makes the placing and route the design tougher to meet the specified timing constraint. There is a tradeoff between the area and the timing constraint. After the design is laid out meeting all the timing constraints, optimization can then be performed to improve the timing on the system. If at this time, meeting the timing constraints is impossible with the tool, area must be increased to relax the design.

In this design, power rings around the core are implemented. The power rings help to isolate the

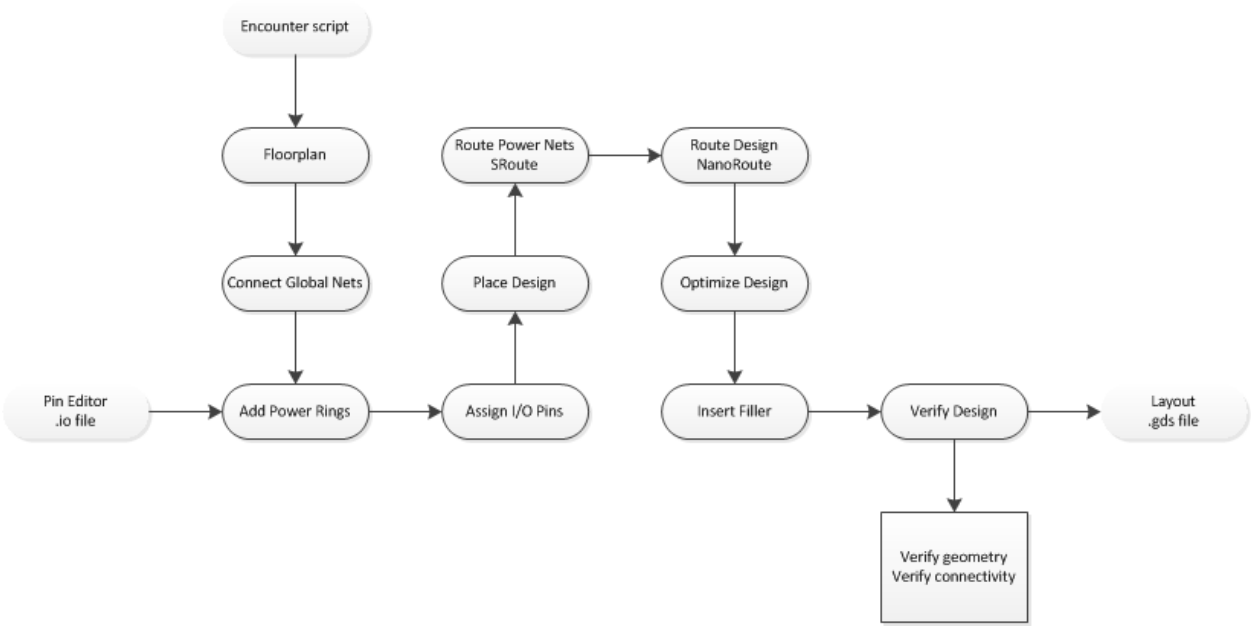

Figure 4.2. Encounter SoC Tool Flow

signal from other circuit blocks on the chip. In addition, the power rings also make it easier for circuit integrations with the global power net. In the next step, connecting global nets maps the lows and highs (“0”s and “1”s) in the gate level netlist to power and ground on a physical circuit layout. Due to the fact that the top level signal paths will use metal layer 5 and above, the internal routing layers were limited to metal layer 1 to 4. Figure 4.3 shows the finished layout for digital computing unit, and Figure 4.4 presents the layout for the USART interface in the Cadence SoC Encounter tool. The Cadence Encounter tool saves a gds file of the layout and this is then used to import the layout into Cadence Design System for post-layout simulation and top-level integration.

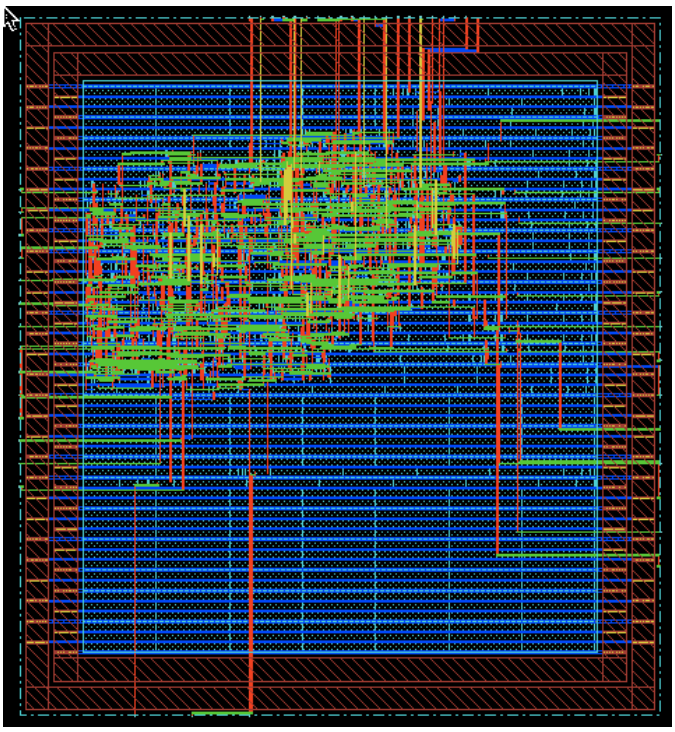

Figure 4.3. Digital Computing Unit Layout in Encounter SoC Tool

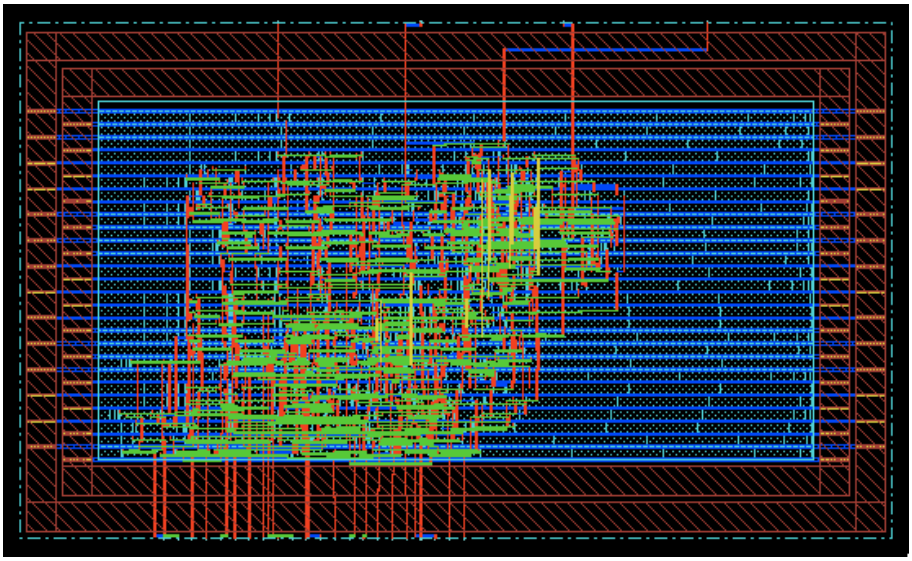

Figure 4.4. USART Interface Layout in Encounter SoC Tool

4.5 Cadence Design System

The exported layout gds files from Cadence Encounter tool were then imported into the Cadence Design System for post-layout simulations and verifications. The layout can be opened from Cadence Virtuoso and two rule checks have to be passed. First, DRC (design rule check) ensures that no layout violations against foundry rules were produced in the automatic place and route phase. LVS (layout vs. schematic) checks for the consistency in design specifications. LVS makes sure that the layout performs the same functionality that was specified in the Verilog behavioral code. Once the two tests are passed, the layout is then extracted with parasitic resistances and capacitances. The extracted view is consisted of transistors and parasitic capacitors as explicit components. Running simulations with the extracted view makes sure that the system runs correctly with the modeling of the real physical structure of the layout after fabrication.

The digital computing unit and the USART interface are integrated with the analog portion of this system and the custom-designed 13-bit digital counter. Figure 4.5 shows the digital domain integration of counter, digital computing unit, and USART interface. The lower left block is the counter. Its 13-bit output feeds the digital computing unit on the right, and the 14-bit read out are located on the right side of the block. These 14 bits are passed to output pads directly for system testing. The USART interface is place on the top, and the control signals are sent to the digital computing unit in parallel vertical wires. Figure 4.6 shows the whole system integration with input/output pads. The analog domain is on the left and digital on the right, and they each have a separate power ground. As can be seen from the system layout, this design is pad-limited. The chip area including the pad frame is 1.5mm by 1.5mm. The core of this system only takes an area

of 335µm by 230µm.

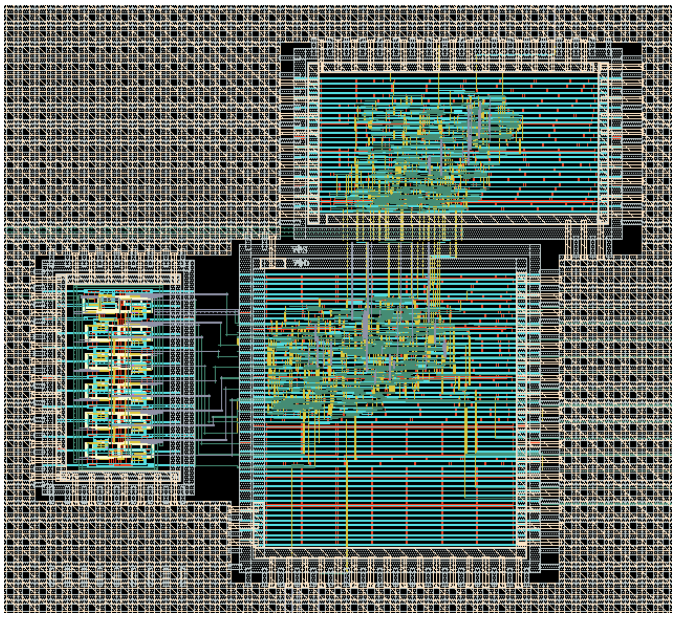

Figure 4.5. Digital Domain Integration Layout

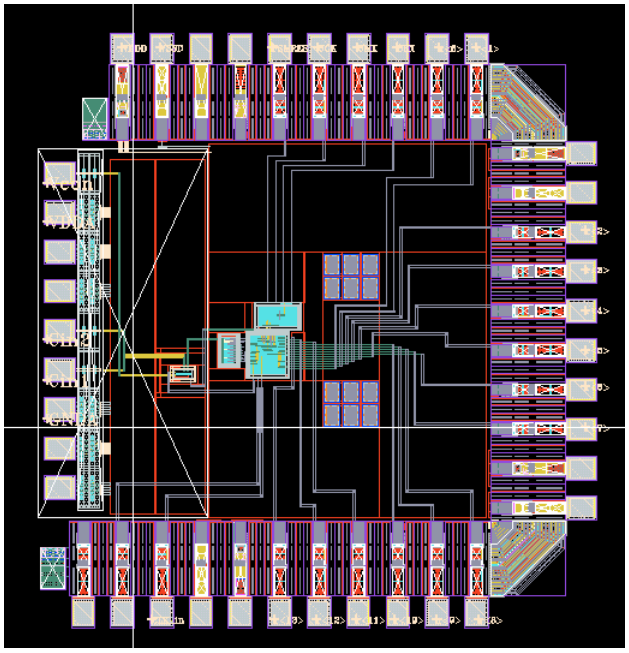

Figure 4.6. Touch Sensor Chip Final Layout

CHAPTER 5

Results and Conclusion

5.1 Post-Layout Simulations and Results

After the final layout gets extracted, and brought back in schematic view, post-layout simulations are then be performed in Cadence Design System. The digital I/O pads cannot be simulated in Cadence so the system level simulations were done with the digital I/O pads removed. During extraction, both parasitic resistance and capacitance were included to properly model the system behavior after fabrication. A clock signal with 10ps rise and fall times were used to drive the system. The entire system uses one external clock, which is set to 16 times faster than the clock that drives the dicke switch. The USART cannot be tested explicitly; therefore, it is also removed from the top level simulation. The USART interface has been used in many other circuit applications in the High Speed Electronics Lab, and the functionality has been verified in real silicon. In this touch sensor system, we did not have to verify it again. The signals that are supposed to be provided by USART would be generated by DC voltage sources instead. The antenna will be an off-chip antenna. In order to model the capacitance loading from the environment sensed by the antenna, a pair of discrete capacitors is used to load the differential input ports. The schematic of the testbench is shown in Figure 5.1.

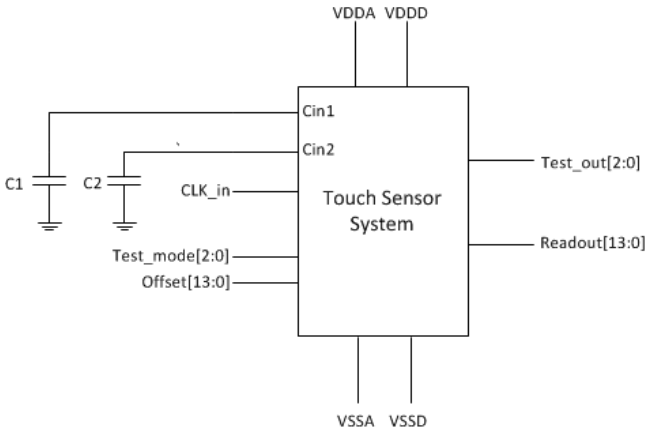

Figure 5.1. Touch Sensor System Post-Layout Simulation Testbench

The performance of the oscillator-based touch sensor system is indicated by its sensitivity to input loading capacitance. The testbench uses a pair of discrete capacitors, C1 and C2, as shown in Figure 5.1, to model the ambient capacitance. When running simulation, the output of the oscillator is probed. As can be seen from Figure 5.2, the frequency changes when clock_slow switches on and off. The C1 and C2 values are reduced until the system loses its sensitivity to

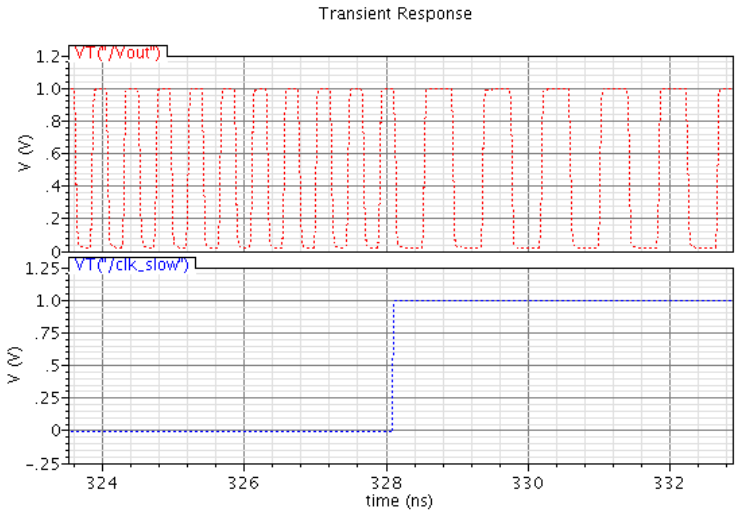

Figure 5.2. Oscillator Output Waveform

detect the difference. The target dicke switch frequency is 1MHz, which is equivalent to a sampling window length of 0.5u seconds. The operating frequency of the oscillator is chosen to be 1.5GHz to ensure the proper operation of the digital counter. The capacitor values are reduced until the system cannot detect a difference in pulse count using a window of 0.5u seconds.

The output frequency of the oscillator measured when capacitors are loaded indicates how sensitive the system is. Depending on the free-running frequency of the oscillator, the larger the frequency difference is, the more sensitive the system is. Different capacitor values are tested in this design, and the results are shown in Table VI. The oscillator free-running frequency is assumed to be 1.5GHz for proper digital counter operation, and the initial sampling window length is assumed to be 0.5u seconds. As indicated by the table below, if the capacitor value continues to decrease below 1fF, the count difference can be easily below 100, and the design becomes less robust. For input capacitance of 50fF, the system is sensitive enough for proper detection.

Oscillator free-running frequency = 1.5GHz Sampling window = 0.5u seconds			
Delta Input Capacitance	Output Frequency	Frequency Difference	Count Difference
50fF	1.07GHz	430MHz	215
10fF	1.18GHz	320MHz	160
1fF	1.22GHz	280MHz	140

Table VI. Touch Sensor System Sensitivity Test

5.2 Conclusions

The existing touch sensor technologies use actual finger contact to make activation. The reason why they are unable to carry on the same concept to implement remote sensing stems from its limited sensitivity. For current capacitor touch sensors, the capacitance threshold for detection is 0.5-1 pF. The proposed design in this paper has sensitivity of 50fF, which is one order of magnitude below the current capacitor touch sensors. The sensitivity advantage of the proposed oscillator-based touch sensor enables its remote sensing ability. The design is compact and low-power. The core size of the proposed design is 335um by 230um. The power consumption is 2.5mW. The integrated digital counter has 13 bits, therefore, the system has the ability to count to 8191. This allows for a lot of room for flexibility and reconfiguration. The initial design setup uses 1MHz clock for dicke switch, which requires an external clock of 16MHz. The clock frequency can be adjusted once design modification in demand.

This paper discusses the "proof of concept" phase of a product that has great potential in improving users' experience with mobile applications. The proposed design is still yet pending in post-silicon verification, and final optimization after testing. Once the individual chip has been proved in its functionality, multiple chips can be put together in a system to triangulate a finger position. The system integration is still in process, however, the potential is phenomenal. Smart phones and tablets have become essential digital companions in human lives. The utilization of the proposed technology could change the way how human interact with digital medias significantly. The Air-Touch system will certainly enrich people's lives in both variety and diversity.

References

[1] E. A. Johnson, "Touch Display - A novel input/output device for computers," *Electronics Letters*, vol. 1, no. 8, pp. 219-220, 1965.

[2] Nicole Cohen. (2012, March) National Public Radio. [Online]. http://www.npr.org/2011/12/23/144185699/timeline-a-history-of-touch-screen-technology

[3] Timothy Hoye and Joseph Kozak, "Touch Screens: A Pressing Technology," in *Tenth Annual Freshman Conference*, Pittsburgh, 2010, p. B9.

[4] Yongsuk Park, Jinwoong Bae, Eungsoo Kim, and Taejoon Park, "Maximizing responsiveness of touch sensing via charge multiplexing in touchscreen devices," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1905-1910, August 2010.

[5] Cypress Semiconductor. (2009, June) Cypress Perform. [Online]. http://www.cypress.com/?docID=17212

[6] Brian Dipert. Digi-key Corporation. [Online]. http://www.digikey.com/us/en/techzone/microcontroller/resources/articles/selecting-a-touch-screen-controller-implementation.html

[7] R. N. Aguilar and G.C.M. Meijer, "Fast interface electronics for a resistive touch-screen," in *IEEE Sensors*, 2002, pp. 1360-1363.

[8] Seung-Ho Kim, Kiwon Lee, and Kyung-Wook Paik, "High speed touch screen panels (TSPs) assembly using anisotropic conductive adhesives (ACAs) vertical ultrasonic bonding method," in *Electronic Components and Technology Conference (ECTC)*, 2010, pp. 1964-1967.

[9] Adam Ellison and Ivan A. Cornejo, "Glass Substrates for Liquid Crystal Displays," *International Journal of Applied Glass Science*, vol. 1, no. 1, pp. 87-103, March 2010.

[10] Mudit Ratana Bhalla and Anand Vardhan Bhalla, "Comparative Study of Various Touchscreen Technologies," *International Journal of Computer Applications*, vol. 6, no. 8, pp. 12-18, September 2010.

[11] J. A. Paradiso, K. Hsiao, J. Strickon, J. Lifton, and A. Adler, "Sensor systems for interactive surfaces," *IBM Systems Journal*, vol. 39, no. 3.4, pp. 892-914, 2000.

[12] Daniel Sonntag, Matthieu Deru, and Simon Bergweiler, "Design and Implementation of Combined Mobile and Touchscreen-based Multimodal Web 3.0 Interfaces," in *International Conference on Artificial Intelligence (ICAI)*, 2009.

[13] Darran R. Cairns, David C. Paine, and Gregory P. Crawford, "The Mechanical Reliability of Sputter-Coated Indium Tin Oxide Polyester Substrates for Flexible Display and Touchscreen Applications," in *MRS Proceedings*, 2001, pp. F3.24.1-12.

[14] William H. Mangione-Smith, "Technical Challenges for Designing Personal Digital Assistants," *Design Automation for Embedded Systems*, vol. 4, pp. 23-39, 1999.

[15] Jing Gao, Lian Qing Zhu, Yang Kuan Guo, and Hao Meng, "A LCD Touch Screen Controller Based on LPC2138," *Advanced Materials Research*, vol. 383-390, no. 2012, pp. 6914-6919, 2012.

[16] Tzu-Ming Wang and Ming-Dou Ker, "Design and Implementation of Readout Circuit on Glass Substrate for Touch Panel Applications," *Journal of Display Technology*, vol. 6, no. 8, pp. 290-297, August 2010.

[17] Neal Brenner and Shawn Sullivan. (2008, February) EE Times Asia. [Online].

http://www.eetasia.com/STATIC/PDF/200808/EEOL_2008AUG27_STECH_EMS_AN_01.pdf?SOURCES=DOWNLOAD

[18] Ik-Seok Yang and Oh-Kyong Kwon, "A touch controller using differential sensing method for on-cell capacitive touch screen panel systems," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 3, pp. 1027-1032, 2011.

[19] P. T. Krein and R. D. Meadows, "The electroquasistatics of the capacitive touch panel," *Industry Applications Society Annual Meeting*, vol. 2, no. 2-7, pp. 1712-1716, October 1988.

[20] Tong-Hun Hwang, Wen-Hai Cui, Ik-Seok Yang, and Oh-Kyong Kwon, "A highly area-efficient controller for capacitive touch screen panel systems," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 2, pp. 1115-1122, 2010.

[21] Hyoung-Rae Kim et al., "A mobile-display-driver IC embedding a capacitive-touch-screen controller system," in *2010 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 2010, pp. 114-115.

[22] Zheng Wei et al., "The design of infrared touch screen based on MCU," in *2011 IEEE International Conference on Information and Automation (ICIA)*, 2011, pp. 485-489.

[23] Li Mingwei and Liu Bintao, "Design and implementation of a multiple-touch system based on infrared technology," in *2011 3rd International Conference on Awareness Science and Technology (iCAST)*, 2011, pp. 233-236.

[24] V. Rana, N. Paliwal, and A. Chahar, "Touch Sensor Assembly Using Infrared Radiations and its Applications," in *Computational Intelligence, Communication Systems and Networks, 2009. CICSYN '09. First International Conference on* , 2009, pp. 96-98.

[25] Zhang Song, Ma Jianshe, Zhou Qian, Guo Hongfend, and Cheng Xuemin, "An economical

large area touch panel utilizing a compact optical configuration and linear optical sensors," in *Imaging Systems and Techniques, 2009. IST '09. IEEE International Workshop on*, 2009, pp. 123-126.

[26] Li Hua, Zhang Shi-chao, Han Chao, Zheng Ming, and Meng Xiao-feng, "A near infrared imaging detection system based on davinci platform," in *Electronic Measurement & Instruments, 2009. ICEMI '09. 9th International Conference on*, 2009, pp. 4-154-4-159.

[27] Gerlinda Grimes. (2011, July) howstuffworks.com. [Online]. http://electronics.howstuffworks.com/gadgets/audio-music/theremin1.htm

[28] W. Buller and B. Wilson, "Measurement and Modeling Mutual Capacitance of Electrical Wiring and Humans," *Instrumentation and Measurement, IEEE Transactions on*, vol. 55, no. 5, pp. 1519-1522, 2006.

[29] A.A. Alsharef, M.J.T. Marvast, M.A.M. Ali, and H. Sanusi, "A high speed and low power voltage controlled ring oscillator for phase locked loop circuits," in *Micro and Nanoelectronics (RSM), 2011 IEEE Regional Symposium on*, 2011, pp. 132-134.

[30] M. Jung, J. Fuhrmann, A. Ferizi, G. Fischer, and R. Weigel, "Design of a 12 GHz Low-Power Extended True Single Phase Clock (E-TSPC) Prescaler in 0.13µm CMOS technology," in *Microwave Conference Proceedings (APMC), 2011 Asia-Pacific*, 2011, pp. 1238-1241.

[31] Clifford E. Cummings and Don Mills. (2002, April) engsoc.org. [Online]. http://www.engsoc.org/~jvd/docs/hdl/CummingsSNUG2002SJ_Resets.pdf