A Digital IC Design for Low Power Capacitance-to-Digital Converter

---

A

Thesis

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

---

in partial fulfillment

of the requirements for the degree

Master of Science

by

Daehyun Lee

December 2023

APPROVAL SHEET

This

Thesis

is submitted in partial fulfillment of the requirements
for the degree of

Master of Science

Author: Daehyun Lee

This Thesis has been read and approved by the examing committee:

Advisor: Benton H. Calhoun

Advisor:

Committee Member: Brad Campbell

Committee Member: Todd A. DeLong

Committee Member:

Committee Member:

Committee Member:

Committee Member:

Accepted for the School of Engineering and Applied Science:

Jennifer L. West, School of Engineering and Applied Science

December 2023

Abstract

The need for increased power efficiency and performance is gaining traction in modern electronics, especially in applications such as wearable technology, medical monitoring devices, and Internet of Things (IoT) sensors. These applications require components that not only save power but also maintain high accuracy and reliability. Unlike the notion of home appliances, which receive a steady supply of power, semiconductors used in wearable and Internet of Things technologies are continuously shrinking in size while targeting energy efficiency through low power consumption. For instance, surgical medical devices must be compact enough for body insertion while ensuring clear and reliable functionality. For healthcare wearables and Internet of Things sensors, miniaturization of semiconductors and peripheral devices is essential for the development of small and lightweight devices, and low-power operation is required. To meet these demands, efforts are needed to improve analog and digital data processing and efficiency along with miniaturization. Therefore, it is becoming increasingly important to integrate mixed signal circuits based on analog signal processing.

This paper introduces the development of a digital integrated circuit (IC) design for a low-power Capacitance-to-digital converter (CDC), uniquely blending digital and analog methodologies to address these requirements. The proposed design is a fusion of innovative digital signal processing techniques and energy-efficient circuit design with the goal of minimizing power consumption while maintaining the accuracy and functionality of the Capacitance-to-digital converter. This mixed-circuit approach not only promotes reduced power usage but also ensures high accuracy and reliability, crucial for sensitive applications. Digital signal processing technology is integrated to complement analog components and maintain a balance between power savings and

signal integrity. Through simulation and experimental analysis, this design demonstrates multi-channel capabilities and improved power efficiency compared to existing Capacitance-to-digital converters, showing a significant increase in power efficiency at high operating frequencies. The paper also explores the broader implications of mixed circuit design, providing insight into potential applications and setting a precedent for future research on energy-efficient electronic design. The results of this study pave the way for further research in advancing the field of low-power mixed-signal integrated circuit design, especially in the context of multi-channel sensing technologies.

Acknowledgments

I would like to express my deepest gratitude to Professor Benton, who not only opened the door for me to pursue circuit research but also provided generous support throughout my journey. His guidance as my advisor during my master's program at UVA was invaluable, offering advice that was both challenging and empathetic. There were times of shared frustration and joy, which I will always remember fondly. Without his supervision and counsel, completing my master's degree would have been a far more difficult endeavor. I also extend my sincere thanks to my dissertation committee members, Professors Campbell and Todd, for their valuable time and insightful suggestions.

I would also like to thank Omar Faruqe for working with me on the project. Working with Omar on this project was a highly educational experience, greatly enhancing my understanding of analog circuits. The practical suggestions, questions, and insights we shared helped me navigate numerous challenges during the research process. I am sincerely grateful to the Bengroup family for their inspiration and encouragement, which eased my adaptation to life in the United States and at University of Virginia since joining the group in August 2021. I would like to thank Shourya for being a mentor from first day, Suprio for being a good friend, Peng and Rishika for helping with chip testing, and Katy and Natalie for being with me throughout this project. I would especially like to thank Peter, our PCB design expert. I would like to especially thank Terry and Brandon for their generous support of everything from experimental equipment to needs for research.

The two and a half years of my master's program will remain an unforgettable chapter in my life. My time at University of Virginia has not only been a period of academic achievement but also of significant personal growth. Thank you.

Contents

List of Figures

List of Tables

Acronyms

ADC Analog-to-Digital Converter.

AFE Analog-Front-End.

ASIC Application Specific Integrated Circuit.

CDAC Capacitor Digital-to-Analog Converter.

CDC Capacitance-to-Digital Converter.

CDN Capacitance Divider Network.

CLK div Clock Divider.

Cont Main Controller.

DAC Digital-to-Analog Converter.

E-nose Electronic nose.

E-skin Electronic skin.

IC Integrated Circuit.

IoT Internet of Thing.

Main out Main signal out.

Mon Monitoring Logic.

OTA Operational Trans-conductance Amplifier.

SA cont Sample and Analog Control.

SH clk Sample and Hold clock.

SPI Serial Peripheral Interface.

Chapter 1

Introduction

New wearable sensing technologies support the measurement of a wide range of physiological and physical metrics, detecting changes in the surrounding environment, including in medical monitoring applications. These physiological and physical indicators primarily consist of data that changes in real-time. Such sensing platforms require sensor interfaces capable of detecting, monitoring, and communicating physiological and physical data about both the human body and the environment [1–7]. Beginning with detecting a single data type through a single sensor, technologies are now being developed to extract multiple data types using a single sensor. Advancements in materials technology have also benefited the field of sensor technology. Leveraging flexible sensor interface technology, sensors with various bendable characteristics have been developed, enhancing wearable technologies. For instance, Electronic skin (E-skin) equipped with a series of sensors can detect environmental factors such as pressure, temperature, and air humidity [8]. Human skin, being curved, of various thicknesses, and with a smooth surface, necessitates a flexible sensor interface. These technologies enable E-skin to continuously and non-invasively monitor vital signs such as heart rate, blood pressure, and body temperature, showing promise in wearable health monitors and medical diagnostics. Electronic nose (E-nose) is another excellent example. E-nose devices utilize a series of chemical sensors to monitor and detect harmful gases and certain volatile organic compounds, which can cause bacterial infections, allergies, dizziness, and headaches [5]. Our skin senses various feelings, including temperature, humidity, and texture.

Similarly, the nose can detect multiple odors simultaneously, including those of food and burning. It is important to emphasize that the successful implementation of applications such as E-skin and E-nose requires multi-channel sensor interface circuitry. Recent advancements in wearable technology research not only highlight a variety of applications but also emphasize technologies that enable continuous, real-time monitoring and maximize energy efficiency. Continuous development in energy harvesting technologies is also improving battery life [9]. The emphasis of the argument is on enhancing user convenience through these technologies and adopting a low-power design approach. To extract signals from sensor arrays of multiple sensors, the latest readout circuits are being developed to interface with each sensor and sample the output of all channels for digitization. For example, a multichannel Capacitance-to-Digital Converter (CDC) supports 16 channels with shared Analog-Front-End (AFE) and Analog-to-Digital Converter (ADC) for all channels in [8]. However, it does not allow separate channel control, so power is wasted due to unused channels [8]. The CDC in [2] uses a 16-channel multiplexer with 16-bit resolution and consumes between 51 μW and 153 μW. Another proposed wireless sensor interface Integrated Circuit (IC) supports up to 32 channels (16 resistor and 16 capacitor) with a power of 61 μW [10]. These power figures far exceed the nW power requirements required for long-life wearable devices. Additionally, power consumption varies depending on the type of amplifier used. [11], which uses a charge amplifier based on Operational Trans-conductance Amplifier (OTA) that consumes a lot of power, shows low energy efficiency. Due to the characteristics of sensors and sensing environments, the development of re-configurable sensor interface ICs is an area of interest research. Both the sensing range and speed requirements of these ICs depend on the target application. Therefore, CDC should support the dynamic nature of sensors to extend their coverage. In this study, I will introduce a 10-channel capacitance-to-digital converter that can support multiple Internet of Thing (IoT) application such as wearable healthcare applications, and particularly propose research on digital design in mixed circuits [1]. The proposed CDC was developed targeting a capacitance sensor and converts 10 capacitance values into digital signals. The signal applied to the input of the capacitance sensor has periodicity and it is in the form of a clock with a duty cycle of 50:50. The name of this clock is sampling clock, and it is output from digital logic and transmitted to the off-chip sensor. The

capacitance signal input is connected to the internal capacitor group. By applying the capacitor voltage divider principle, the internal capacitor value is adjusted by targeting half of the input voltage. The voltage between the two capacitors passes through the analog buffer and is transferred to the input of the ADC, then the analog signal is converted to digital. The purpose of development was to measure small capacitance in pF units, and the target sensor application was a gas sensor in pF units. Due to the nature of multi-channel, a total of 10 inputs can be connected simultaneously, and measurements were possible from about 6 pF to 11 pF. CDC's performance can be programmed through internal registers. This thesis will cover the role, function, and stability of digital logic for programmable CDC design. The proposed design is a fusion of innovative digital signal processing techniques and energy-efficient circuit design with the goal of minimizing power consumption while maintaining the accuracy and functionality of the CDC. This thesis introduces challenges and motivation from a digital design perspective, and also includes a discussion of how the problems were approached to resolve.

Chapter 2

Analog and Digital Architecture overview

Before delving into the details of digital design, I would like to present an overview of the architecture for the Low Power Capacitance-to-Digital Converter (CDC) design project. This overview will be beneficial for understanding the placement and interconnection of analog and digital components within the fabricated Application Specific Integrated Circuit (ASIC) chip. As depicted in Figure 2.1, the CDC chip internally comprises significant analog and digital components. The yellow line indicates the digital top which includes an **Serial Peripheral Interface (SPI)** slave and **Sample and Analog Control (SA cont)**, while the analog top comprises an input channel **capacitor group**, **switch group**, **Sample and Hold circuit**, **Capacitor Digital-to-Analog Converter (CDAC)**, **on-chip reference**, and **Analog-to-Digital Converter (ADC)**. The blue line represents the Capacitance Divider Network (CDN), and the red line indicates the paths within the fabricated CDC chip. The CDC system encompasses a CDN, a Sample-and-Hold (S/H) circuit that samples the divider output, and an on-chip differential ADC followed by an analog buffer. It also features an on-chip adjustable voltage reference and an analog buffer to drive the ADC. Lastly, a digital circuit, labeled 'Control Logic & Counter,' is included to manage these analog components.

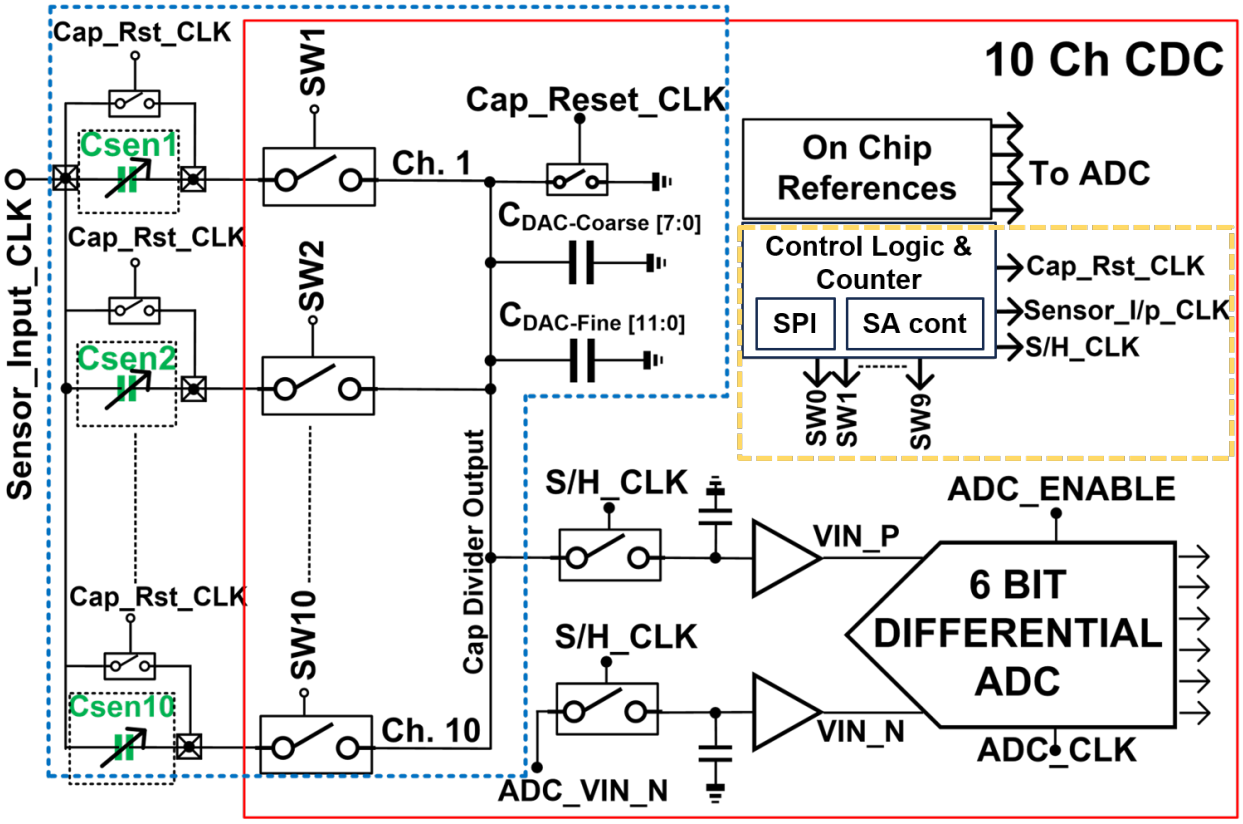


Figure 2.1: System Architecture of multi-channel Capacitance-to-Digital Converter (CDC)

2.1 Digital module block diagram

2.1.1 Overview

As shown in Figure 2.2, digital logic is comprised of two major components. The first is a communication protocol module, referred to as the **SPI slave**, and the second is the **Sample and Analog Control (SA cont)**, tasked with sampling and controlling the analog module. The SPI reference is predominantly used for chip-to-chip communication. This communication requires the configuration of a Master and a Slave. The Master generates and transmits Reset, Chip Select (CS), Serial Clock (SCLK), and Master Out Slave In (MOSI) signals to the Slave. Since the CS is an active-low signal, a high state indicates the slave is not selected, whereas a low state enables the slave, allowing it to receive data from the master. The **Main signal out (Main out)** module

plays a role in generating a sampling clock for the input sensor array, while the **Sample and Hold clock (SH clk)** module produces sampling and hold pulse signals to capture the voltage signal at the common node from the divider output. The **Clock Divider (CLK div)** module divides the clock frequency to support different sampling clock speeds. The **Main Controller (Cont)** module primarily generates and manages internal signals in the digital top. Finally, the **Monitoring Logic (Mon)** module is responsible for storing and displaying the ADC's output data via a SPI slave read operation.

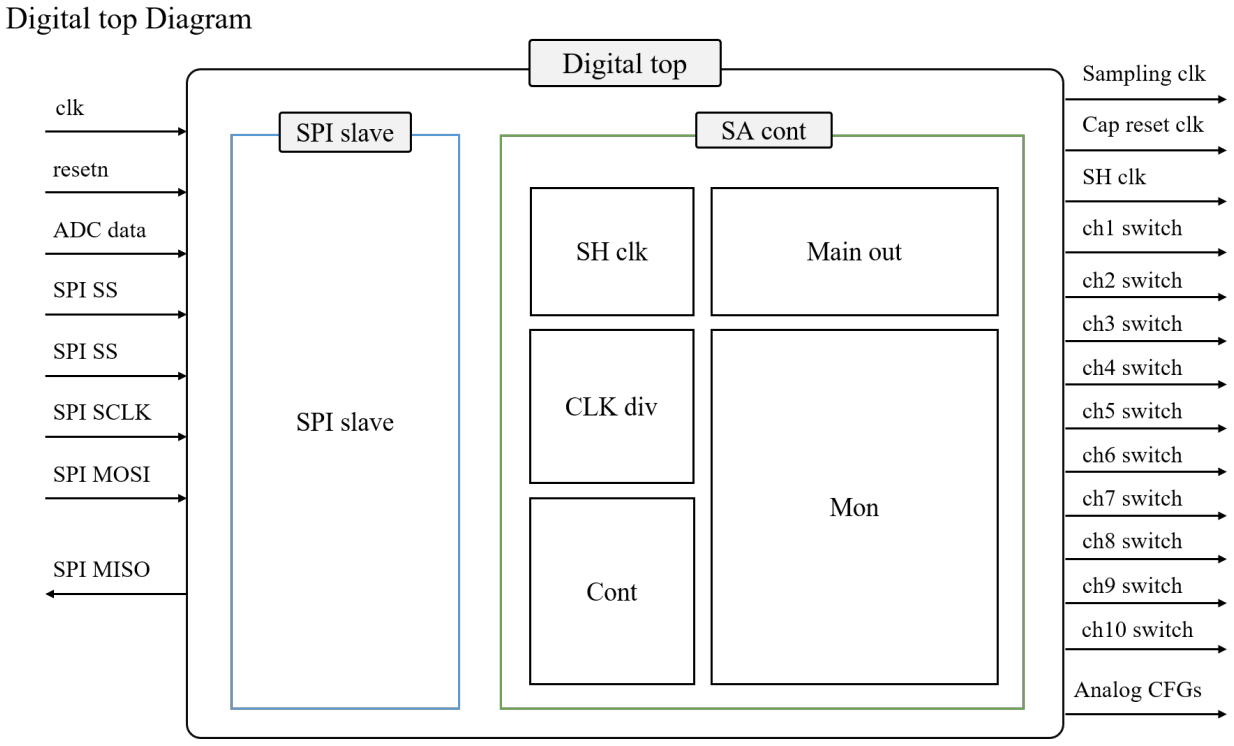


Figure 2.2: Digital top diagram

2.1.2 Digital module signal connection

The Figure 2.3 illustrates the connections within the digital top. The connection between the SPI slave and SA cont comprises wiring for configuration settings and five read paths for monitoring data. Signals directly linked from the SPI slave to the analog circuit are shown on the

left. Configuration signals are transmitted directly to the Capacitor Digital-to-Analog Converter (CDAC), analog buffer, and ADC. On the other hand, in areas where additional precise controls of digital logic are required, SA cont fulfills that role. The digital module is responsible for managing the enable timing of the ADC, including the switch, as well as handling Sample and Hold operations.

Digital module signal connection

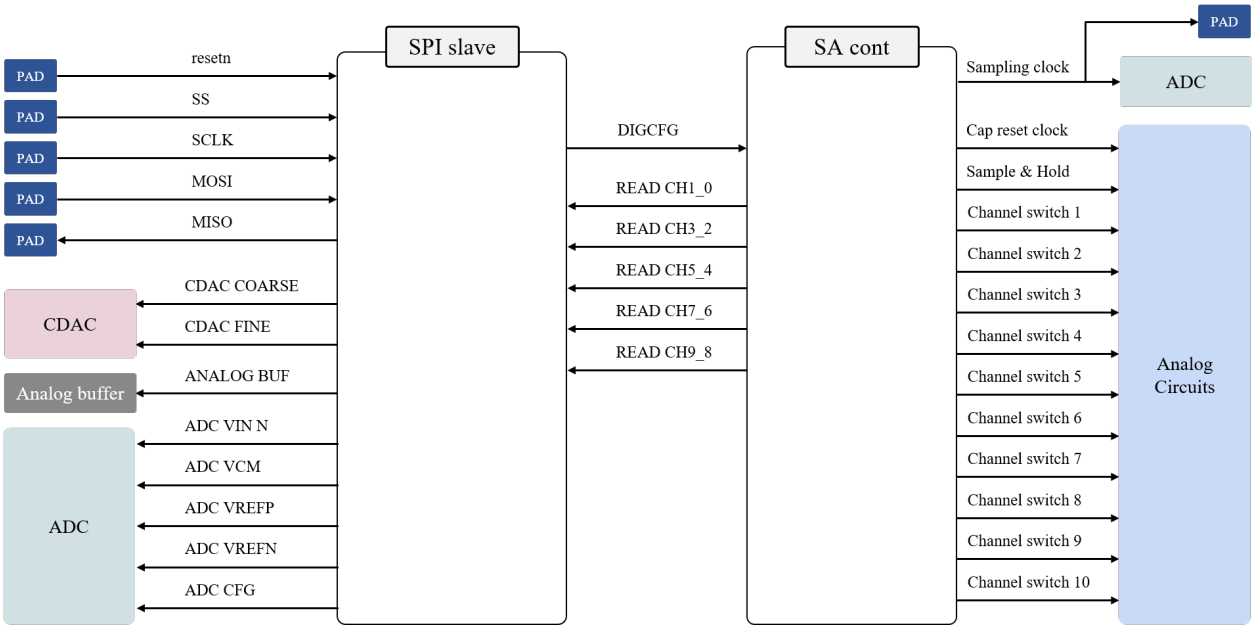


Figure 2.3: The illustration of inner connection in Digital top

2.1.3 Main digital block signal connection

The Figure 2.4 shows the internal connections of **SA cont** digital block in detail. As the main controller block, the **Cont** module generates major signals that are delivered to most internal modules. The **CLK div** plays the role of re-generating the periodic signals according to speed requirements through an internal clock distributor. This re-created clock information is delivered to **Main out** and **Cont**. The role of **SH clk** is to generate a pulse signal for transmission to the Sample and Hold circuit. The **Main out** module literally plays a significant role of generating main signals. Sampling clock and switch signals for 10 channels are examples. Lastly, the **Mon** module

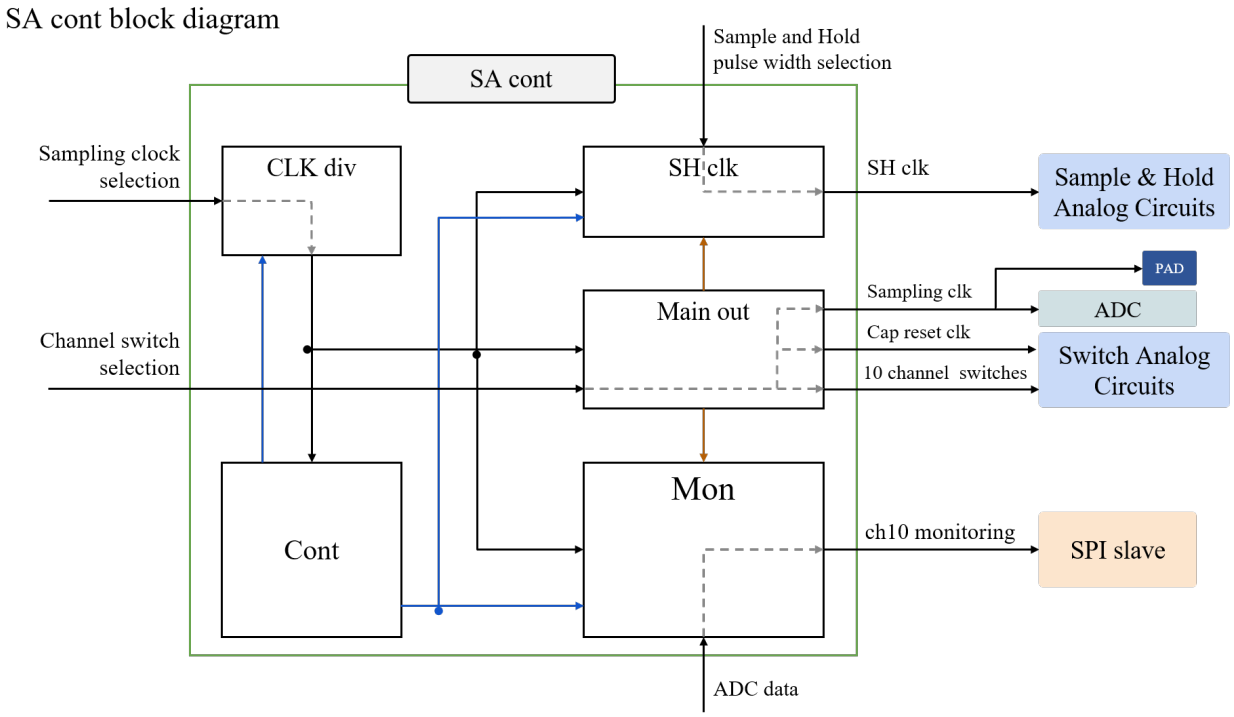


Figure 2.4: The Sample and Analog control block diagram

is a monitoring logic that stores the digit information of each channel using ADC output data, and outputs them according to the read command.

Chapter 3

Digital design

3.1 10-multi channels input control

3.1.1 Overview

In the picture below Figure 3.1, we can see the digital logic used for controlling multiple channels, which is a key component of the Low Power Capacitance-to-Digital Converter (CDC) design project. This is why it was named Main signal out (Main out). The Main out module generates a sampling clock that is transmitted to off-chip sensors. This clock is produced by combining internal signals derived from dividing a 32 KHz clock. The sampling clock operates on a basis of 10 channels; within this framework, it is further divided and controlled across these 10 channels. The module receives two input signals: the first is a 10-bit switch selection register information from the Serial Peripheral Interface (SPI) slave, and the second is from the main control logic managing configuration updates. A flip-flop receiving these signals outputs stable, programmed enable signals according to the updated configuration. The 10-bit signals at this point serve as channel information within the digital design. This switch enable information, indicating which channels are disabled, is transmitted to a flip-flop that generates a sampling clock based on this information. Concurrently, an enable signal, synchronized with the sampling clock, is sent to the switch logic of each channel through another flip-flop.

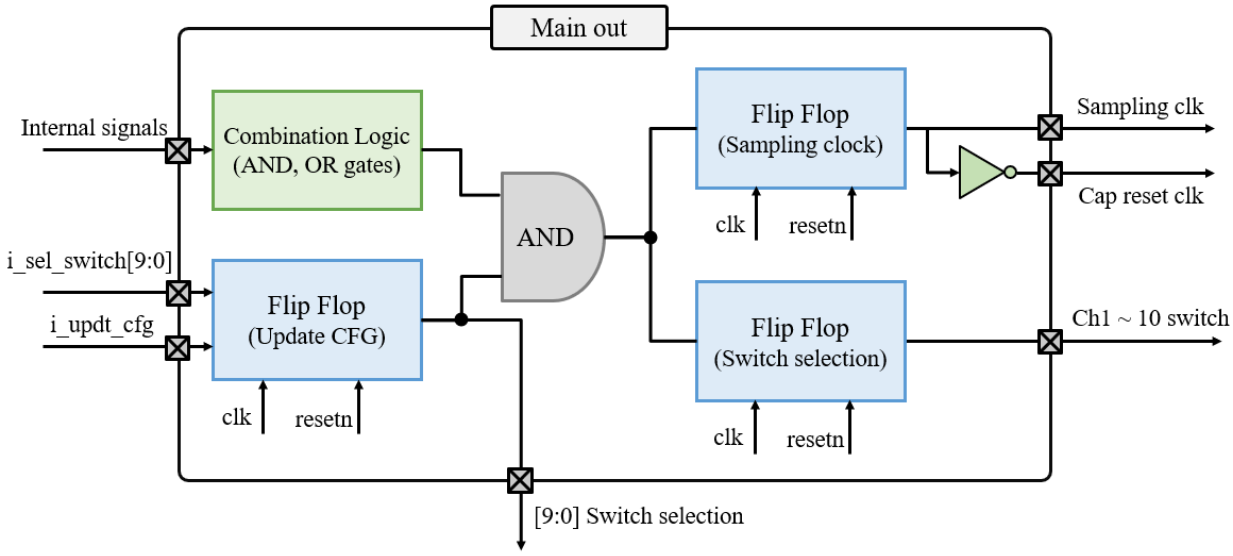


Figure 3.1: Main out module which is for 10 channels control

3.1.2 Motivation and Challenge

While working on the Low Power CDC design project, I successfully submitted a conference paper and was invited to contribute to a journal by the same organization, on the condition of adding more content. I want to highlight that a key factor in achieving these milestones was the effective control of 10 channels. In the early development stages, a major point of discussion was distinguishing between multi-channel and single-channel configurations. There was a deep discussion about whether activating only one channel and turning off the others was qualified as multi-channel operation. An opposing view was that the essence of multi-channel existence was to receive input data with all channels active. Another perspective considered was the idea of parallel processing, where data from all channels was processed simultaneously. Among these ideas, I chose the approach of receiving input data with all channels turned on. However, exploring and resolving the challenges of processing input data simultaneously from these multiple channels proved to be a significant hurdle, primarily due to the structure of the analog design. The reason was from the fact that the architecture structure was already set up so that all inputs were merged into one common node. We considered ways to solve the problem through digital design, but the

complexity was expected to increase significantly, and I determined that parallel processing would not be possible without fundamental changes to the architecture. So the method chosen was time distribution technique which was a time multiplexing to prevent analog signals from overlapping each other during merging.

3.1.3 Design options and selection

One of the main argument outlines for this project is multi-channel control. As explained earlier, as a digital logic designer, I had a variety of choices. The first option was to control the input models of the 10 channels one by one. The method was to assign on/off settings for all channels to a register, so that only one channel out of all channels was turned on and the other 9 channels were turned off. The biggest reason why I came up with this method was because the input signals were eventually merged into one node. At this time, because the merged signal was an analog signal, we did not have an analog circuit that could effectively separate multiple signals when they overlapped. Therefore, we considered developing a digital controller that would have no choice but to turn off the other 9 channels. However, this was not multi-channel. This structure was no different from a single channel. This simply meant that only several modules were physically present in the input. Another option was to implement parallel processing. The goal was to develop logic to manage 10 signals input in parallel inside the Analog-Front-End (AFE) chip with all channels turned on, under the condition of not knowing when the 10 input signals were input. This type of structure was attractive in that it could maximize the benefits of multi-channel, and the variety of applications that could be applied was also expanded. However, the development of additional logic was essential, and the development period had to be extended. Logic complexity was also a consideration. Since the ultimate goal was to support sensors with 10 different functions, the midpoint between the two options above was set as the goal for the first tape-out. In the end, the current digital design in the form of time distribution was created as the last option. Figure 3.2 showed the simulation results of in/output ports about the main out digital module.

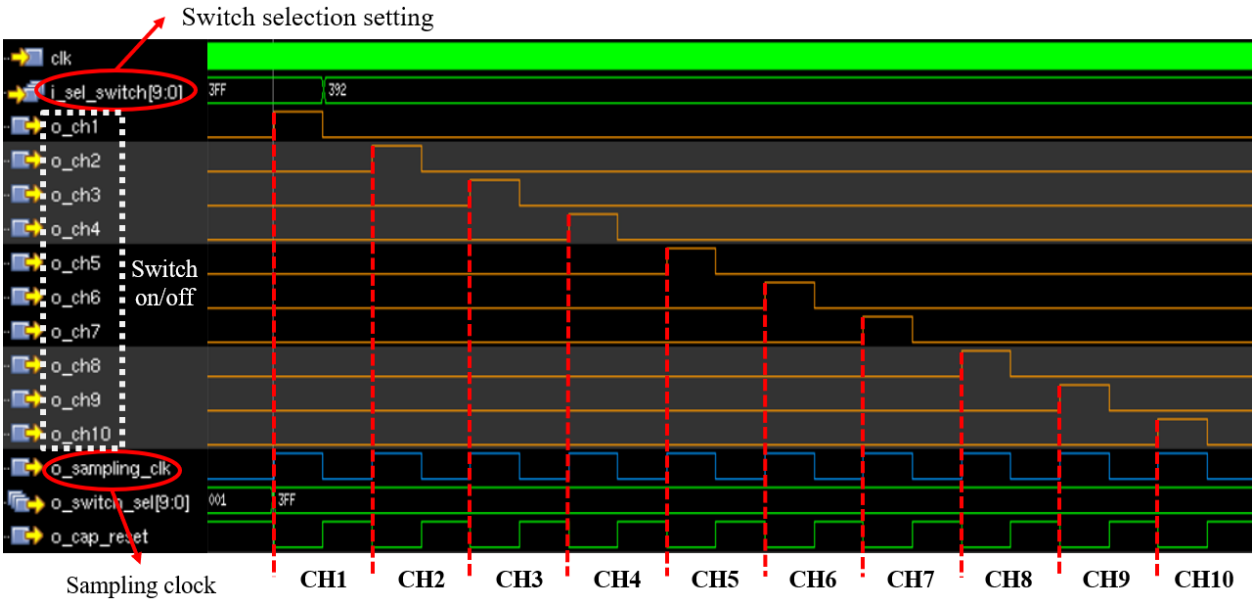


Figure 3.2: Main out module in/out ports waveform

3.1.4 Design principle

Counter and clock divider logic was designed using a 32 KHz clock, and logic to count 10 channels was added. Each counter operates based on the input clock, so it shows continuous operation. In addition, OR gate combination logic was used to distinguish between enabled and disabled channels. For disabled channels, a zero value was assigned that does not toggle, and for enabled channels, it was designed to have a signal that transitions from high to low. Therefore, the final form of the sampling clock is as shown below Figure 3.3.

3.2 Sample and hold block

3.2.1 Overview

Sample and hold clock digital module is meant to quickly sample an analog voltage signal before being passed on to the Analog-to-Digital Converter (ADC). Here, ”Hold” is not a function of digital logic. However, because digital logic helps maintain quickly sampled signals, it is called

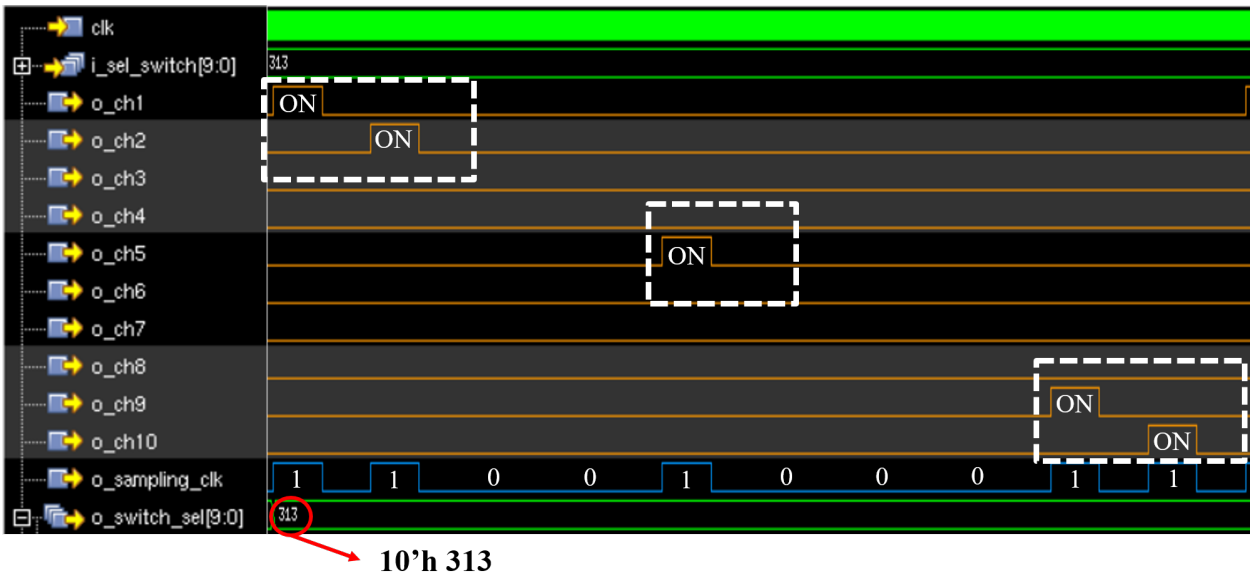


Figure 3.3: Channel selection results in Main out module

sample and hold. As shown in the following Figure 3.4, configuration register information is received from SPI slave, and synchronized to the configuration update signal, and the configuration is transmitted to the Mux. The input of Mux consists of 8-bits. This input is divided into several speeds depending on the sample & hold register options. The logic Flip Flop (Internal Signal) and another Flip Flop (Inner Count) create each speed. In the end, the output of the Mux will be a signal at the speed selected by the user, which is then finally output through the Flip-Flop (Out). The design ensures that this output signal remains active unless the channel select is configured to zero.

3.2.2 Motivation and Challenge

Before designing the sample and hold block, I had several conversations to coordinate opinions with a doctoral student in charge of analog and top integration. As a result, the conclusion of the conversation was that a signal with a narrow enable period was needed. The signals that pass through the Capacitor Digital-to-Analog Converter (CDAC) were gathered into one point. This place was referred to as a common node. A change in voltage was detected at the common node.

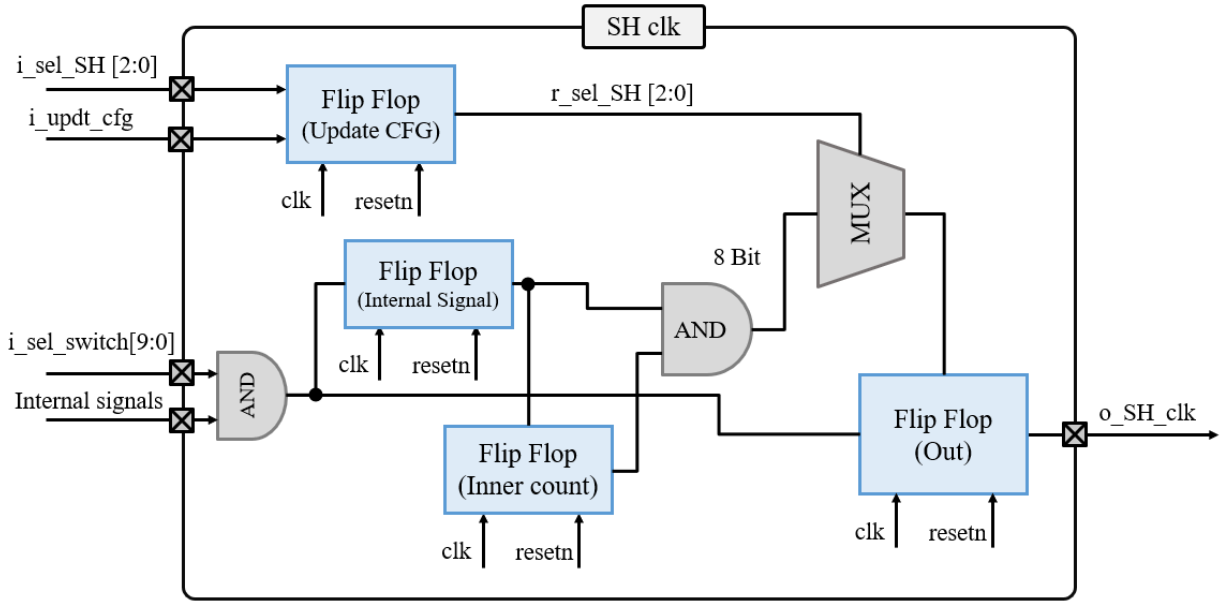


Figure 3.4: Sample and Hold digital controller module diagram

The voltage level that had to be input through the ADC input was about 300 mV, and this voltage had to be maintained during the enable time. However, this voltage appeared to increase linearly. Our expectation was too ideal to estimate this problem. Attempting structural changes was fraught with risk, and we decided that it was important to quickly capture the voltage before it increased continuously. As a result, the specification for the error rate of the CDC chip was determined according to the pulse period of the signal generated by the sample and hold module. Intuitively, I could realize that the shorter the trigger time, the closer to the ideal value. The second challenge was to determine a starting point for capturing. It was expected that capturing the same phase as the moment of rising voltage at the common node would eventually result in inaccurate values being output on the initial output of the ADC. Since there was a rising time due to capacitance in the analog design domain, we predicted that the sharp edge signal generated by digital logic may not capture the voltage value of the part we want. But it was predicted that these concerns could be resolved naturally. The internal digital signal, Sample and hold pulse, would be transmitted with almost no delay. The valid data input from the actual sensor was transmitted from the output of the digital module through the off chip metal wire to the common node through the analog circuit. By

naturally using this delay, the trigger started before the voltage value of the common node rises, and the rising edge and level of the voltage were designed to stably enter the pulse window of digital module.

3.2.3 Conceptual Approach of Digital design

Sample & Hold block design within digital logic focuses on generating pulse signals with a narrow period. The design issue was resolving the question of how narrow it should be. This is because with the input clock set to 32 KHz, it was impossible for digital logic to produce a signal faster than this. I wanted to provide a more flexible programmable design by providing users with the option to select the pulse period. I aimed to provide a more flexible, programmable design by giving users the option to select the pulse period. This decision stemmed from concerns about inherent delays. The pulse signal, when transmitted inside the chip, reaches the analog circuit with almost no delay. However, valid data input from the actual sensor undergoes a longer path: it is transmitted from the output of the digital module, through the off-chip metal wire, then through the sensor, and finally to the analog circuit at the common node. Predicting the total delay was challenging until the actual chip test was conducted. Additionally, it was difficult to determine precisely when a valid section would begin based on the capacitance slope. Therefore, the possibility that the shortest period of the pulse signal might not be long enough to effectively capture a valid section could not be ignored. With this intention, the Sample & Hold digital design can support 8 different pulse periods through registers of SPI slave. The Figure 3.5 shows the simulation results of Sample & Hold digital module about different register setting for various speed. The table for the relevant settings is as follows Table.3.1. In the initial model, the signal from this module was always toggle. Regardless of channel selection, it was always active. This design resulted in unnecessary power consumption. Thus, at the end, the design has been changed to be inactive when the Channel is not selected. However, I missed the detailed setting here. In one set consisting of 10 channels merged from a common node, the voltage level for the disabled channel is zero. Therefore, even if the digital logic toggles at that timing, it appears to be operating normally from a functional perspective because no signal is captured. However, in reality, the sample

and hold digital design did not fulfill the purpose of low power design. For disabled channels, the sample-and-hold digital logic should also have been designed to avoid capturing the action. It is anticipated that in the next version, additional modification of this part will effectively reduce power consumption by eliminating unnecessary operations.

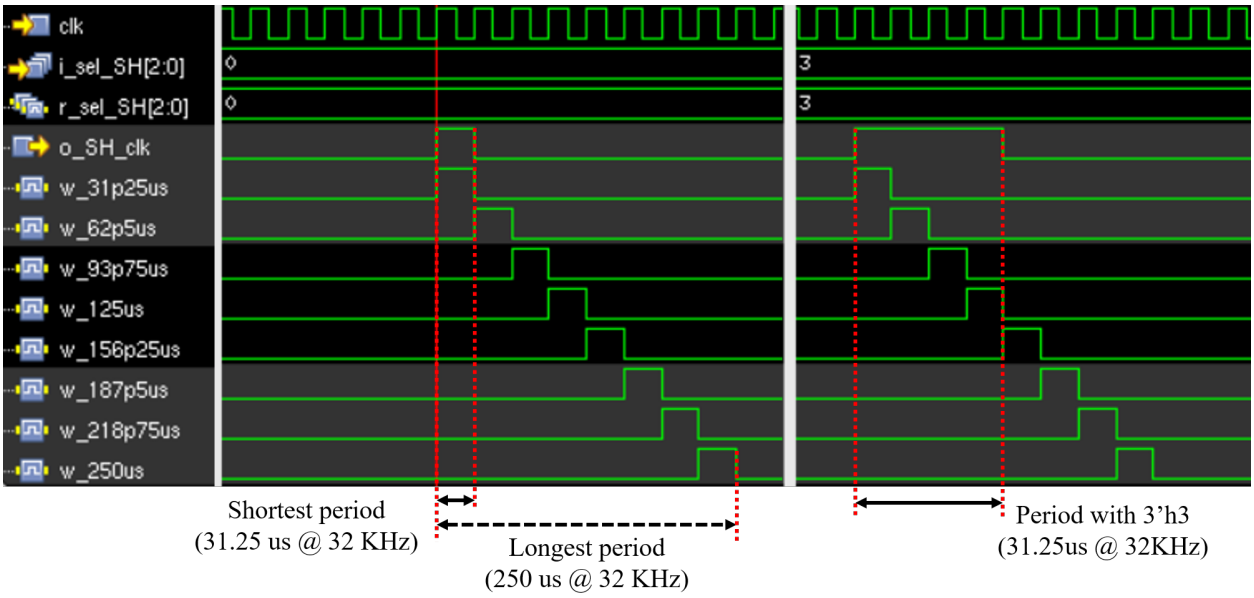


Figure 3.5: Different pulse period waveform at 32KHz

Features	Mux	one sample (µs)
Sample and Hold clock selection	0 (default)	30.5176
	1	61.0352
	2	91.5528
	3	122.0704
	4	152.588
	5	183.1056
	6	213.6232
	7	244.1408

Table 3.1: Single pulse period setting table at 32KHz core clock frequency

3.3 SPI block

3.3.1 Overview

SPI stands for Serial Peripheral Interface, a protocol technology used for wired communication. The SPI signal comprises several components: Serial Clock (SCLK), Chip Select (CS), Master Out Slave In (MOSI), and Master In Slave Out (MISO), which are responsible for data transmission. SCLK literally plays the role of a clock. CS stands for chip select and is characterized as active-low. It is a signal that allows the master to specify a slave address and initiate transmission. MOSI is used by the Master to deliver read and write commands, register settings, and data. This indicates that the direction of data flow is from Master to Slave. On the other hand, MISO is used to transmit data from Slave to Master in response to a Master's command. It is primarily used for transmitting read data, and depending on the design, it can also be employed to transmit status and register information.

3.3.2 Motivation and Challenge

When designing the AFE chip, as an engineer, I identified two main considerations. The first was enabling the AFE chip to perform computations and display results independently. The second was to allow the CPU to control the AFE chip. The former, it was capable of computation and displaying results. Also, it allowed for the removal of the communication protocol module, reducing a chip size. However, it should process data on its own which could lead to increase chip size. The latter provided full control to the microprocessor via SPI slave, and setting the AFE chip in a slave mode to receive commands. This method could have a smaller chip area and lower power consumption. Consequently, I concluded that for a CDC design project aimed at low power consumption, the best approach would be to utilize SPI slaves.

3.3.3 Conceptual Approach of Digital design

The SPI slave module played an important role as a communication protocol for interfacing with off-chip components. In the architecture, the CDC was designed as a sub-chiplet without a processor. This approach was a method of control from outside the chip through the Central Processing Unit (CPU). As a result, external communication was essential for setting registers. To facilitate this, SPI slave logic was implemented into the CDC design. This SPI slave logic is a part of Bengroup's in-house Intellectual Property (IP), and its stability has been proven through the works of previous graduate students. For the register configuration, digital data used to program the analog circuit were systematically organized in the documentation. Subsequently, multiple 16-bit registers were constructed to allocate data associated with each module in their respective 16-bit registers. Unused bits in the 16-bit registers were marked as reserved. A total of nine write registers and five read registers were implemented, with write addresses allocated from 0 to 8 and read addresses from 9 to D. Consequently, the SPI slave design has a total of 4-bit addresses. Since the output of the ADC is 6 bits, a total of 12 bits are mapped from 0 to 11 in one register, with the 5 Most-Significant Bit (MSB) bits allocated as reserved. As there are a total of 10 channels, five read registers were required. The following tables, Table.3.2 and Table.3.3, show the SPI slave write and read register maps, respectively.

#	Command	Name	Address&Reset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Write	DIGCFG	0	i_sel_sample_clk[2:0]			i_sel_SH[2:0]			i_sel_switch [9:0]									
			0001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2	Write	CDAC_COARSE	1	reserved	XTAL_BIT [5:0]						sel_32K_EX_CLK	cdac_coarse [7:0]							
			01F8		0	0	0	0	0	0	1	1	1	1	1	1	0	0	0
3	Write	CDAC_FINE	2	reserved				cdac_fine [11:0]											
			0000					0	0	0	0	0	0	0	0	0	0	0	0
4	Write	ANALOG_BUF	3	reserved	analog_buf_Ikcr [4:0]					analog_buf_ofc [4:0]					analog_buf_onc [4:0]				
			1FE7		0	0	1	1	1	1	1	1	1	1	0	0	1	1	1
5	Write	ADC_VIN_N	4	reserved	ADC_vinn_buf [4:0]					ADC_vinn [9:0]									
			1D5E		0	0	1	1	1	0	1	0	1	0	1	1	1	1	0
6	Write	ADC_VCM	5	reserved	ADC_vcm_buf [4:0]					ADC_vcm [9:0]									
			1D5E		0	0	1	1	1	0	1	0	1	0	1	1	1	1	0
7	Write	ADC_VREFP	6	reserved	ADC_vrefp_buf [4:0]					ADC_vrefp [9:0]									
			1FF4		0	0	1	1	1	1	1	1	1	1	1	0	1	0	0
8	Write	ADC_VREFN	7	reserved	ADC_vrefn_buf [4:0]					ADC_vrefn [9:0]									
			1D5E		0	0	1	1	1	0	1	0	1	0	1	1	1	1	0
9	Write	ADC_CFG	8	reserved						DLY_sel [1:0]		DAC_sel	threshold [4:0]					sel_adc_clk	ADC_reset
			001D							0	0	0	0	0	1	1	1	0	1

Table 3.2: SPI slave write register map

10	Read	READ_CH1_0	9 -	reserved	ADC_ch1	ADC_ch0
11	Read	READ_CH3_2	A -	reserved	ADC_ch3	ADC_ch2
12	Read	READ_CH5_4	B -	reserved	ADC_ch5	ADC_ch4
13	Read	READ_CH7_6	C -	reserved	ADC_ch7	ADC_ch6
14	Read	READ_CH9_8	D -	reserved	ADC_ch9	ADC_ch8

Table 3.3: SPI slave read register map

One difference from a typical SPI slave was that the initial value was set. The SPI slave was controlled through Pattern Generator Logic Analyzer (PGLA) during actual chip testing, and at this time, the initial value was set for the purpose of checking chip operation and for debugging. When creating a Comma-Separated Values (CSV) file for a PGLA test, there might be errors in the CSV file, so we found a way to check the chip operation just by applying voltage. When Reset was input correctly, first channel was enabled without setting in PGLA, and the value was intended to be confirmed through the output of the ADC. Through this initial function, we expected to demonstrate that the internal signal connections were well established. This function was helpful in identifying the starting point for debugging if the CVS file was incorrect. The following waveform Figure 3.6 shows writing operation of SPI slave digital design. Subsequently, the Figure 3.7 shows the read operation of SPI slave.

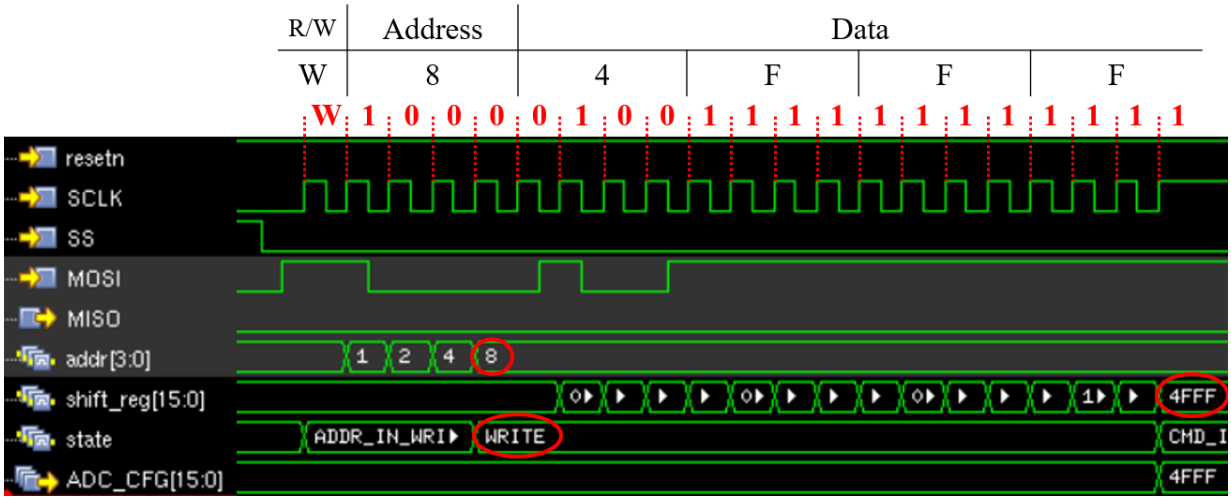


Figure 3.6: SPI slave write operation waveform

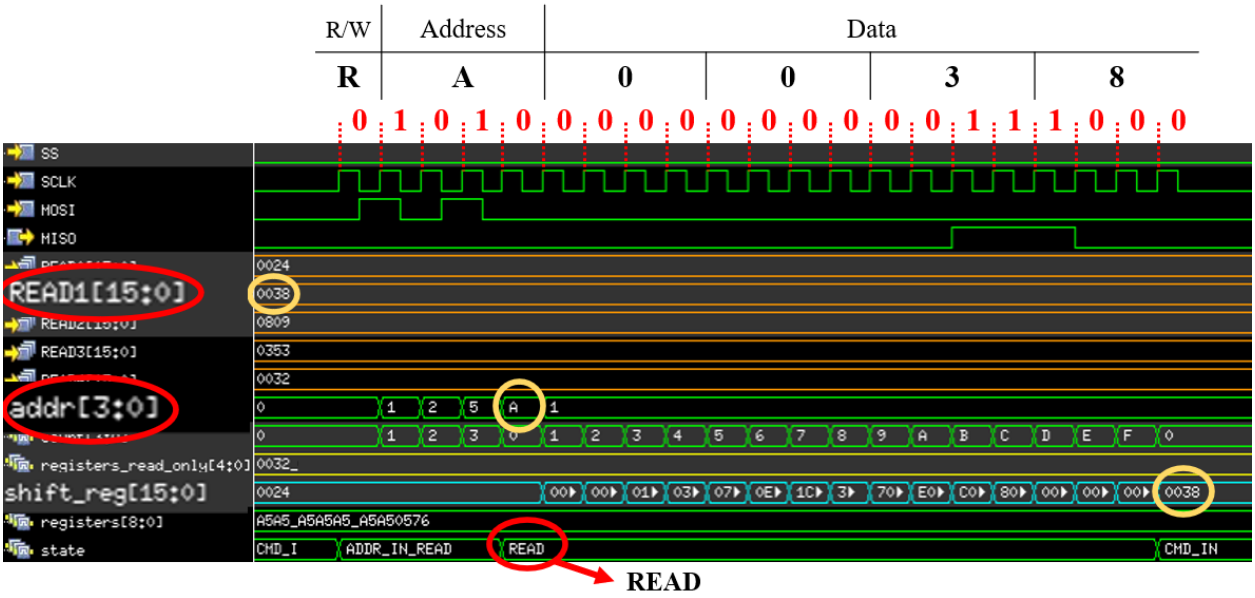


Figure 3.7: SPI slave read operation waveform

3.4 Adjustable clock frequency

3.4.1 Overview

Adjustable clock frequency literally means that the clock speed can be adjusted. Here, "adjust" means that the frequency can be set through register. The clock speed could be increased or decreased. It was not the meaning of changing core clock frequency which input into the digital logic. This clock indicated the output signal from the digital block. Because the signal had periodic nature and a 50:50 duty cycle, it was defined as a clock.

3.4.2 Motivation and challenge

This generated clock signal refers to the sampling clock, which was the output of the Main out digital module introduced earlier. It was connected from the digital top to the output pad, and was transmitted outside the chip. This output signal is used as the input for the Capacitance

Sensor. Therefore, the primary design motivation for the adjustable clock function was to support various sensor applications. Although the sensor interface was fixed to Capacitor, the required clock frequency might vary depending on the type of sensor application. To support these variables, clock sources of various speeds had to exist internally. Additional logic was required to control multiple clock sources, and the area was also expected to increase. To resolve these problems, a digital logic circuit was designed to generate various clock frequencies. For instance, the sensors that do not need to operate quickly, a reduction in power consumption can be expected by lowering the clock frequency. The Table.3.4 presents the relevant setting for the Sampling Frequency option.

Features	Mux	one sample (ms)	Sampling rate (ms)
Sampling clock selection	0 (default)	9.765632	97.65632
	1	4.882816	48.82816
	2	2.441408	24.41408
	3	1.220704	12.20704
	4	0.610352	6.10352

Table 3.4: Sampling Frequency setting option

3.4.3 Conceptual Approach of Digital design

The project's target frequency was 10 Hz. So, the architecture had a data structure in which 10 channel data were updated per 0.1 seconds. In other word, one channel had a period of 10 ms, and the next sample was input with a total period of 100 ms. Other than the target frequency required by the project, a faster or slower clock frequency might be needed. Power consumption could be reduced by lowering the operating frequency, and as a result, overall system power could be reduced and battery life time would be expected to increase in Internet of Thing (IoT) devices. Depending on the type of input sensor at the application level, higher performance could be required. In this case, a faster clock may be needed, and I wanted to provide an option to increase the clock speed according to these need. The Figure 3.8 shows the dynamic sampling frequency.

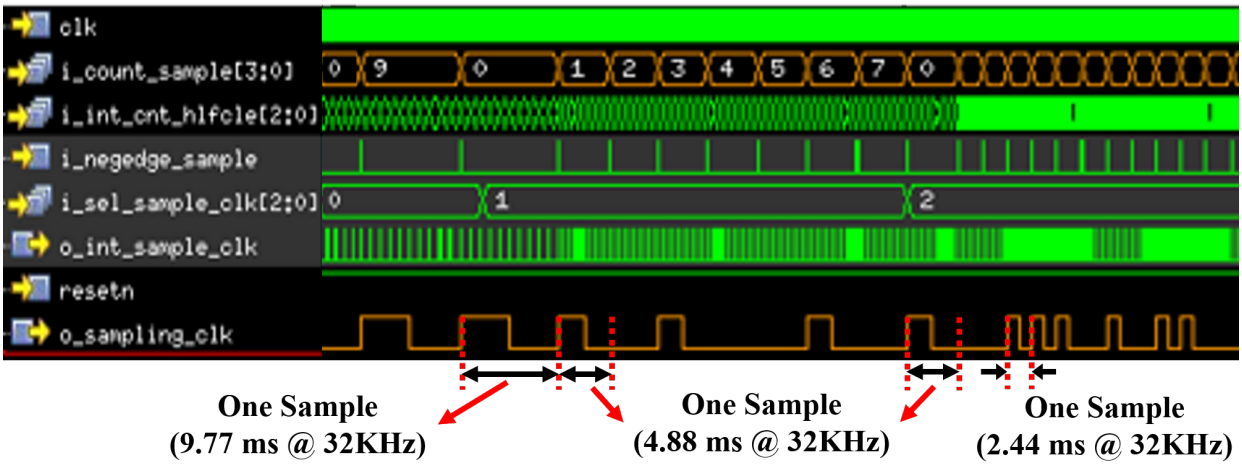


Figure 3.8: Different Sampling Frequency

In digital design, sampling clock speed can be implemented as a divider. However, there was one part that required careful attention. It was the asynchronization between the point when the channel was enabled and the part where the sampling clock was delivered into the input sensor. The sampling clock was transmitted outside the chip, but the enable signal going to the switch was transmitted inside the chip. Although it is very subtle, signal transmission outside the chip generally have a larger delay than inside the chip. This is because it has to pass through the PCB and metal wire. However, in the Low Power CDC design project, the delay would be significantly smaller than the enable period of the sampling clock, and we determined that there would be no problem with the function of receiving input data stably. Another clock shifted by 180 degrees was created to reset the internal capacitors. Figure 3.9 shows the dynamic sampling frequency and sampling clock status based on switch on/off status.

3.5 User-friendly setup

3.5.1 Overview

I would like to first define the term user-friendly. I want to insist that being user-friendly means the user should be able to utilize the logic I designed without concerning about it malfunction-

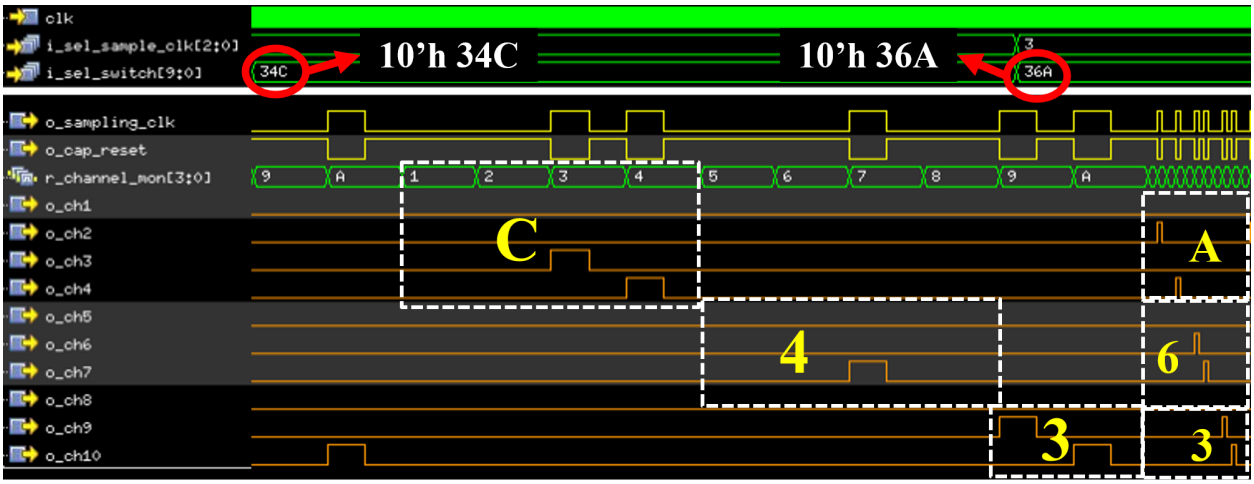


Figure 3.9: Different Sampling Frequency with dynamic channel selection

ing, and the user's commands should be clearly conveyed. The user primarily focuses on user commands transmitted through the Master. As shown in the following Figure 3.10, it receives an internal signal as input and outputs the 'o_updt_cfg' signal through several stages of flip-flop and combination logic. This signal served to notify the digital logic when the configuration was updated. This role was also used in protecting the digital design from Master random interrupts. Other than that, all other output signals were control signals used within the digital top.

3.5.2 Motivation and Challenge - 1

The first digital design was designed to update the relevant information each time the user's command was delivered to the digital logic. For example, the register setting for the adjustable clock speed as introduced above was an option that could be switched according to the user's change of sampling clock frequency requirement. The CPU was able to set the register through SPI slave, and then selected the speed as needed. The speed was immediately applied to the entire digital design. The speed of the digital logic immediately responded to the interrupt transmitted from the master side, resulting in an unintended effect on the clock's duty cycle. A phenomenon was discovered where data was broken in an unintended direction as the clock cycle of each channel collapsed. It was determined that the user's command interrupt was likely to cause glitch effects.

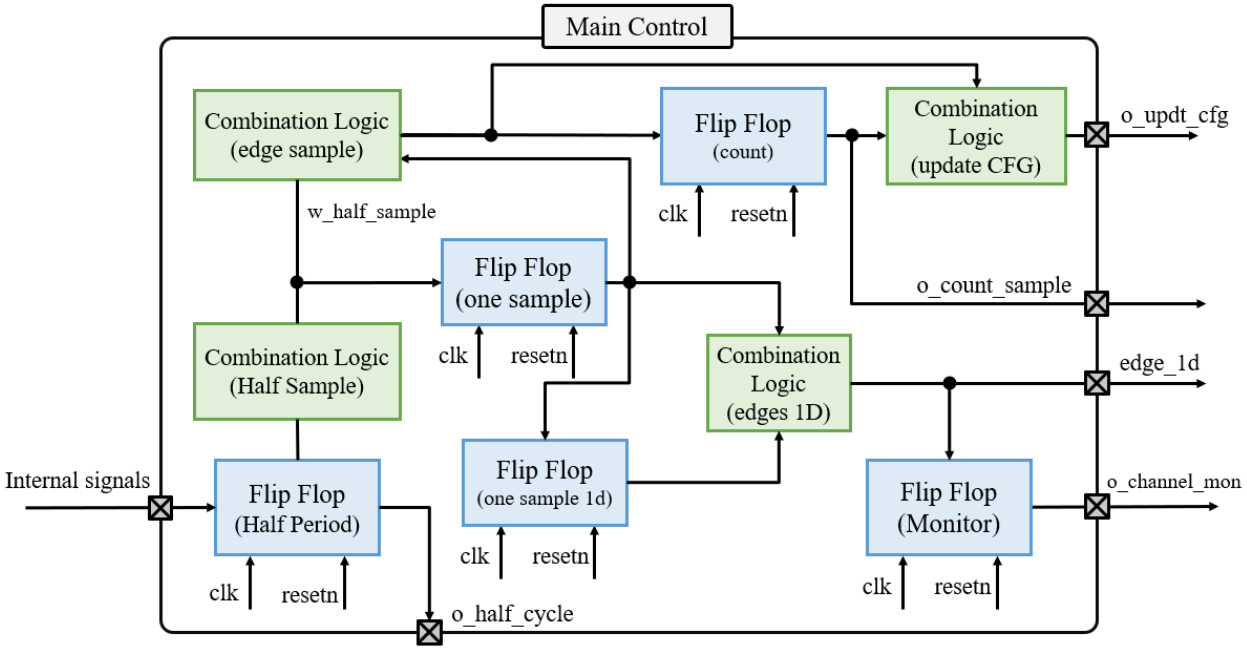


Figure 3.10: Main Control block diagram for user-friendly function

Fast response with reacting every interrupt could relieve frustration, but it also caused unintended problems.

3.5.3 Motivation and Challenge - 2

Another reason was that the concept of the initial architecture was designed with the premise that 10 channels operate as one set. As a result, the internal counter logic was reset due to a sudden interrupt, resulting in an unbalance in the digital signal. If 100 master interrupts occur while 10 channels are operating, 100 register settings are applied to the digital logic.

3.5.4 Questions and Conclusion

At this time, I asked the question why the user needed to change the clock speed 100 times. Also, a question arose as to which of the 100 values did I want to use. I thought that the same value could have been used repeatedly, and that the reason for overwriting a new value over the previously written value was the dominant. Therefore, digital logic was designed to apply the

value written just before the last 10th channel. After 10 channels, the digital logic controls to apply the newly updated clock speed value at the turn where the channel 0 value is processed again. This scheme allows the entire analog circuit as well as the digital logic circuit to operate normally without interference from numerous interrupts. This specification can be seen clearly and concisely from the user's perspective, thereby enabling easy and simple control.

3.5.5 Design Option and Selection

As a designer, I had two design options. One was to set each channel as the target of one command and apply that setting to the next channel operation whenever a master interrupt occurred. The complexity of the logic will increase slightly in the sense that internal logic operations and signals need to be interpreted more finely. In terms of feasibility, it was determined to be highly likely. However, in the current design, it was not possible to provide the user with an option to choose which channel's speed they would like to change. Without this function, I thought it would not be very persuasive to apply logic to change the operation speed of each channel while increasing the complexity of the logic. The other method was to apply new settings when the operation of 10 channels was completed, like the current digital design. The biggest reason for choosing this design method was that it was relatively the simplest of the two design methods. In addition, it was predicted to operate more reliably. The number of required logic was small, and the concept that new settings would be applied when the next sampling cycle returned was also considered to be persuasive.

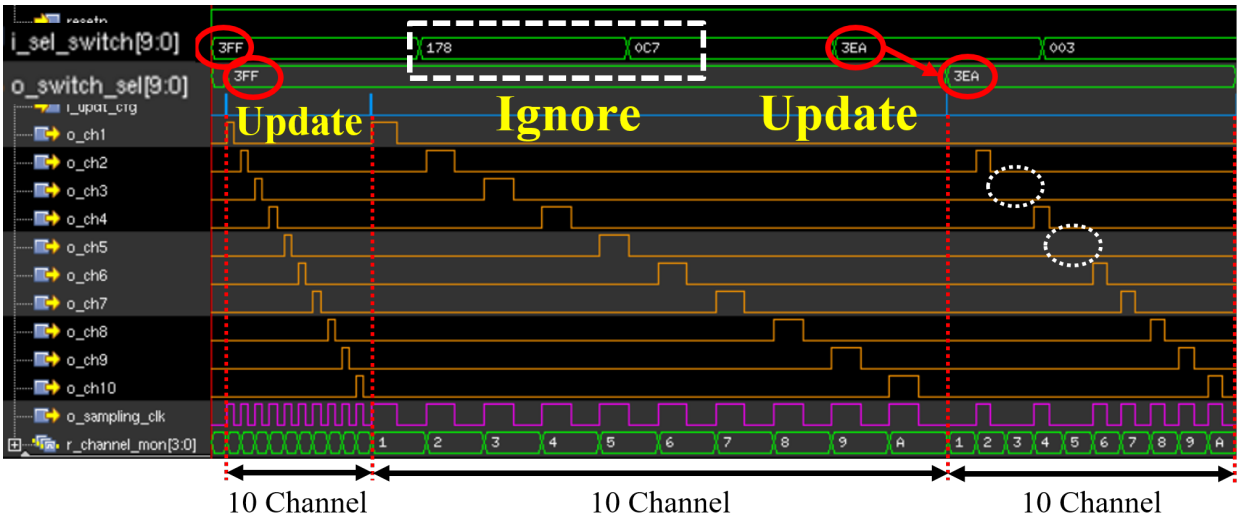


Figure 3.11: The configuration setting applies to the digital module every 10 channels cycle

3.6 Monitoring logic

3.6.1 Motivation and Challenge

The digital logic of the Low Power CDC design project includes monitoring logic. The primary reason for developing this logic was the unclear boundaries of the channels. In other words, although multi-channel data was continuously input to the ADC, there was no mechanism to provide information about each individual channel. The ADC's output was digital and operated on a 32KHz clock base. It received an enable signal and outputs valid data only during the enabled period. During the disabled period, where the enable signal falls to zero, there was no transfer from analog to digital. Then, the transfer resumed once it was enabled again. At this point, the ADC's internal logic kept the last data from the previously enabled state. Then, based on the re-enabled point, this last data started to be transmitted again. This phenomenon occurred because the previous data was held until the transfer of new data was completed. The Figure 3.12 shows this phenomenon via simulation waveform results. This design mechanism ensured stable transmission without data loss in single-channel scenarios. Furthermore, since all result signals did not drop to zero while disabled, there was no additional transients occurrence, which benefits power

consumption perspective. However, what was a design advantage in single-channel applications had the effect of collapsing the boundaries between each channel in multi-channel design. The structure outputs all channel information continuously, and this ADC was applied to the layout as a Graphic Design System (GDS) image type. This led me to realize the necessity of a design that separates the information of each channel.

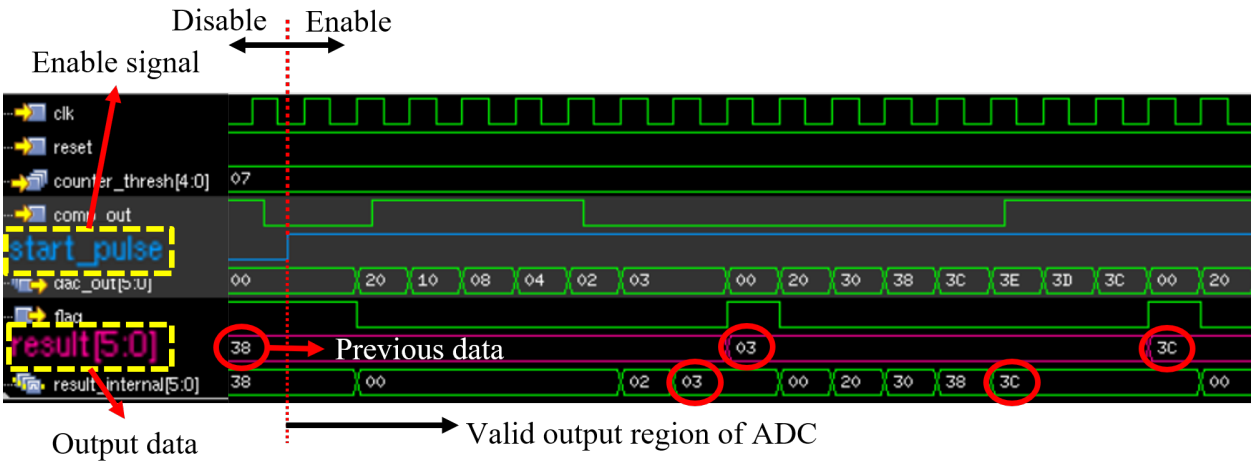


Figure 3.12: The phenomenon of keeping last data in digital block of ADC

3.6.2 Intention of Design and Reason

Multi-channel can only be called a multi-channel if the data of each channel is fundamentally differentiated. To overcome this, monitoring logic was created. In addition to the reason that there was no room to modify the ADC directly, it was more risky to modify the ADC that was already proven working well. However, I could not compromise on the perspective that the correct data should be delivered on user, and I wanted to resolve this problem by creating a monitoring logic. The issue of size naturally followed. Because there was an enough room for entire digital part area, I decided to develop it despite the size increase. As the size increased, an increase in power was inevitable, and an on/off mux was added to avoid power consumption as much as possible. Thus, when the user wants to know the exact digit information, the monitoring logic would be turned on to check the digit information through the SPI slave read communication protocol. The Figure 3.13

presents the stored real data of ADC's output in monitoring digital logic. In addition, the Figure 3.14 illustrates the SPI slave reads the monitoring data via five read ports which has 16 bit width.

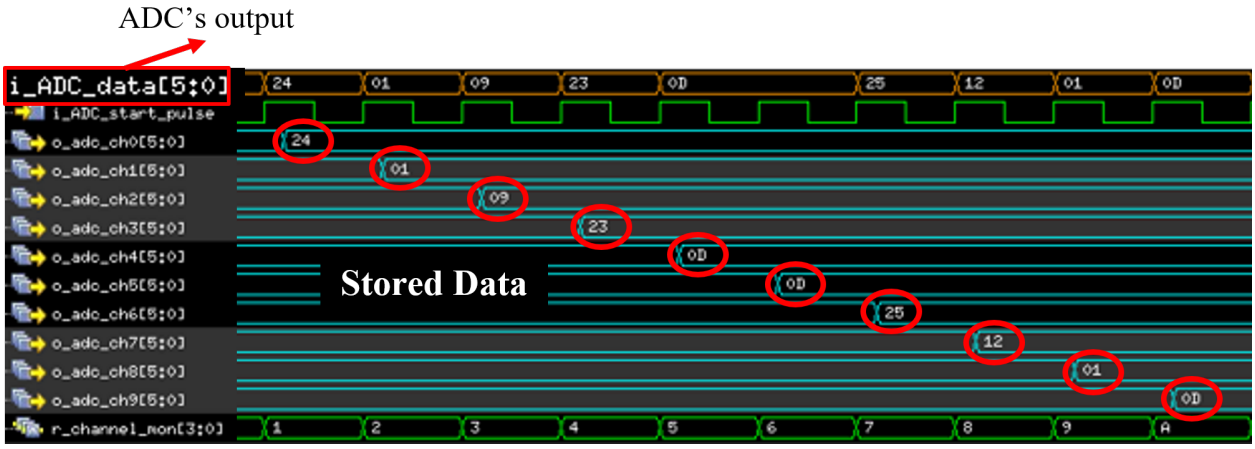


Figure 3.13: The simulation result of monitoring logic

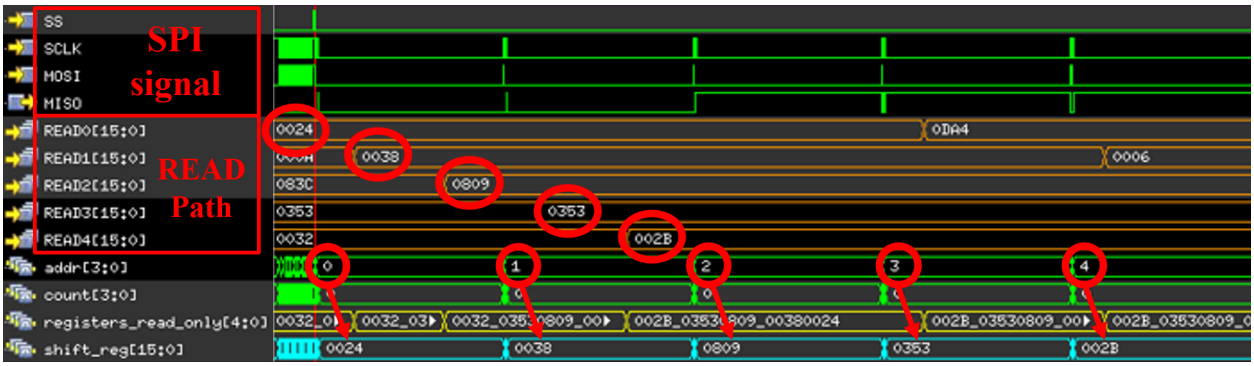


Figure 3.14: The output of monitoring logic and read operation via SPI slave

3.6.3 The operating principle

The operating principle of monitoring logic is surprisingly simple. As explained earlier, the designed digital logic operates in a time distribution method. The principle is that 10 input signals are placed in 10 slots within the sampling period. One channel information is updated after 10 samples. In other words, the information of each single channel is updated at sampling period intervals. During this update time, the monitoring logic holds each channel information. This data

from ADC is input into 6-bit registers, and the 6-bit register stores the value during the sampling period and then updates it again. At this time, the existing value is overwritten with a new value. The Figure 3.15 shows the block diagram of monitoring logic in the digital design. It consists of AND gate and Flip Flop that receive several internal signals and process multi bit width data internally.

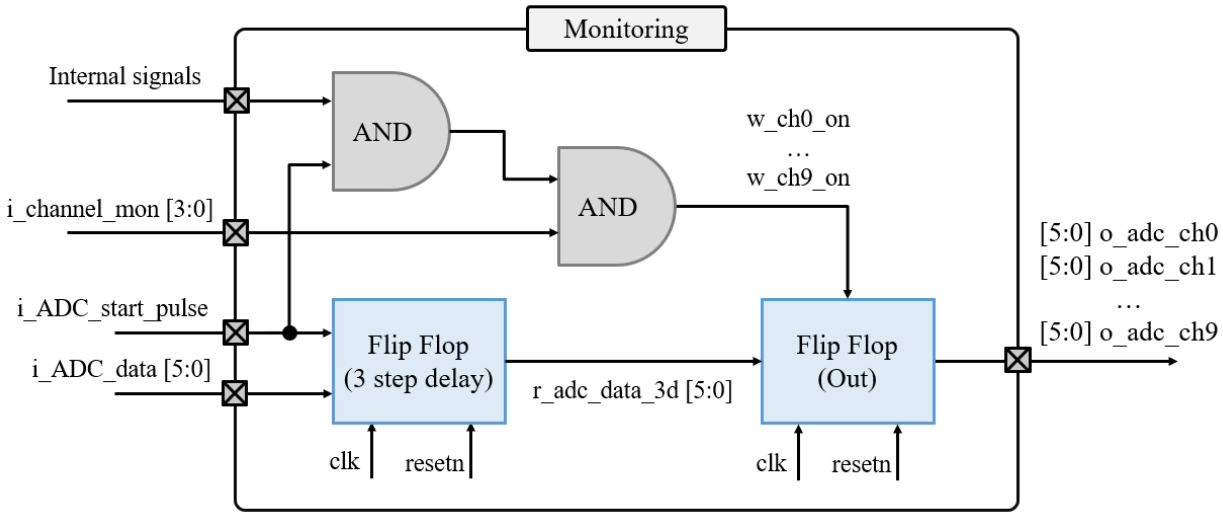


Figure 3.15: The block diagram of ADC output's Monitoring logic

3.6.4 Pros and cons of monitoring logic

3.6.4.1 Pros: presenting the digital output real time data

The biggest advantage is that user can read sufficiently accurate values just by reading the data from the microprocessor rather than directly measuring it with a measuring device such as an oscilloscope. Ultimately, the role of the ADC is to deliver accurate channel information, and monitoring logic exists to resolve the inaccuracy of the interaction section between each channel. From the perspective of a microprocessor, the advantage is that the ADC output value can be known with just a command line that reads the value in the monitoring logic without making an effort to capture the value for each sampling period.

3.6.4.2 Cons: occupied a big number of gates and redundant

It doesn't necessarily have only advantages. When it is necessary to design a very small chip, monitoring logic is clearly a major obstacle in reducing the chip size. If the CPU is capable of handling the ADC's output data until it is determined to be stable, then employing monitoring logic becomes unnecessary. The best design is to achieve the highest efficiency with the smallest logic, from this design perspective, monitoring logic is clearly a negative factor. However, from a functional perspective of delivering accurate data, it is sufficient to offset these negative factors.

Chapter 4

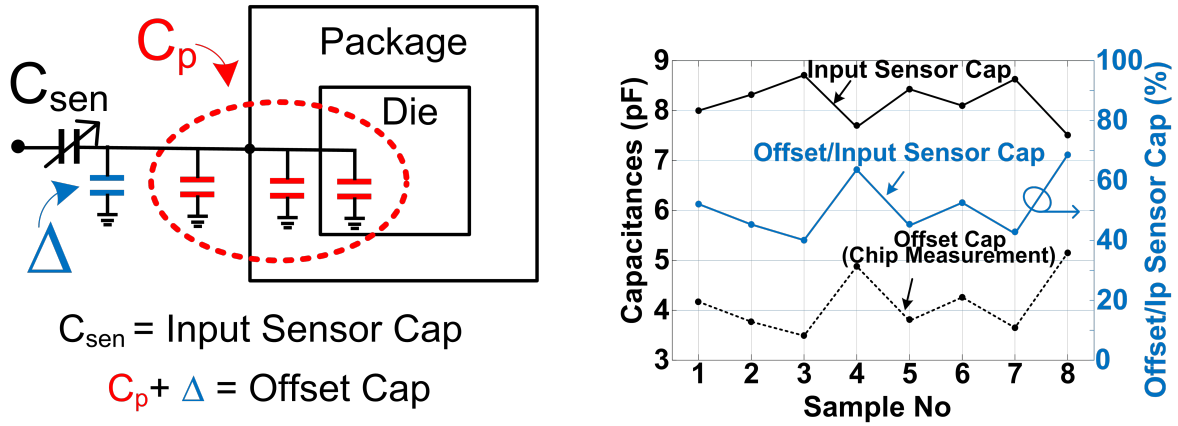
Real chip experiment result

4.1 Overview

In this chapter, I would like to show the results of actual chip testing. The number of chips produced was about 20, and surprisingly, the first chip tested worked well. After that, it continued to show stable operation. It was necessary to determine which of the 20 chips were operating stably, and again, surprisingly, the test results showed that all 20 chips operated well. When installing a chip into a socket during a PCB board test, the pressing process was essential. If we press too hard at this time, there was a risk of damaging the chip. Regrettably, one chip was damaged during the test while repeatedly attaching and detaching the chip. After inferring the reason, I think that the structural connection of the bonding wire could be the reason. Several bonding wires, which are very thin, pass over the pad. It can be inferred that if the chip is pressed too firmly, these wires might be damaged. The experiments conducted at the University of Virginia testing lab involved testing with capacitors of various capacity ranges in pF units, connected to the input. We attempted to purchase an actual sensor which had pF capacitance unit, but we were unable to find one with small capacitance matching the targeted capacitor capacity range in pF units for this project.

4.2 Introduction

The Figure 4.1(a) shows the top level chip view of the proposed Capacitance-to-Digital Converter (CDC) structure. One of the intentions of the project was to explore how the input value measurement can be distorted by unintended parasitic capacitors that may exist between the capacitor value of the input capacitance sensor, the chip package, and the chip die. Offset error arises when unintended capacitance are included in the inherent capacitance of a sensing path of CDC. Figure 4.1(a) depicts the source of offset capacitance, and Figure 4.1(b) illustrates the measurement of offset capacitance on a chip and its influence (offset/input sensor cap = $40 \sim 70\%$) on sensor capacitance when the input sensor capacitance is low $(8 \sim 9pF)$.



(a) Source of offset capacitance

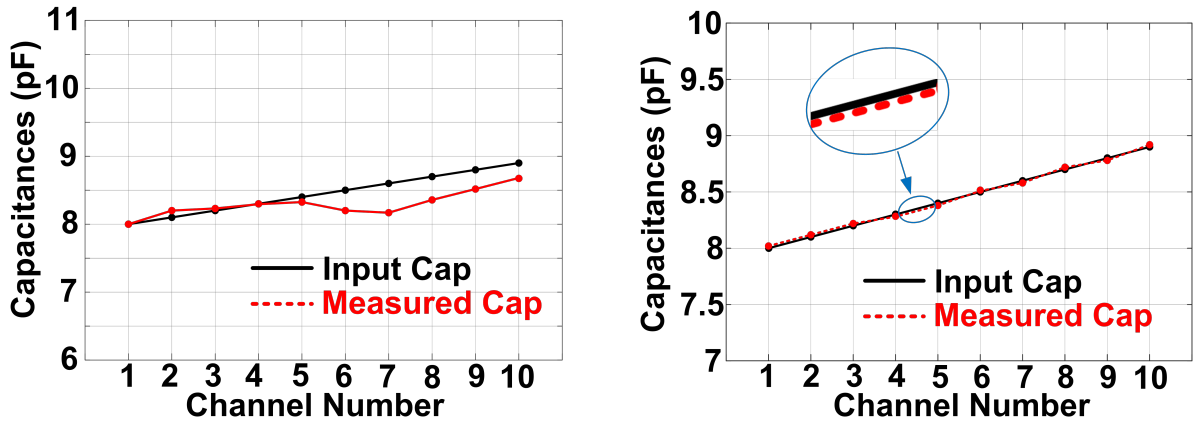
(b) Impact of offset capacitance

Figure 4.1: Top level chip overview and parasitic capacitance impact

4.3 Measurement Results

This CDC system underwent testing with both fixed capacitors and adjustable capacitors, used as off-chip sensors. For all measurements, the sensor baseline was maintained at $8pF$ to align with the specifications of the target sensors. Figure 4.2(a) illustrates capacitance measurements for channels 1 through 10 when only channel 1 is calibrated. Measurement errors occur in channels 2 to 10 due to varying offset capacitance from tolerances (Δ) and parasitic capacitance (C_p).

However, the system design permits calibration of each channel independently, allowing for adjustments of their specific offsets and baselines. Each channel calibration significantly improves the measurement accuracy, as shown in Figure 4.2(b). For example, with an input capacitance of 8.7 pF, the measurement error decreases from 3.94% to 0.37%. The capacitance variation (ΔC) ranges from $0pF$ to $0.9pF$ throughout the measurement, and the proposed CDC system successfully calibrates the baseline and other offsets to measure only this variation.

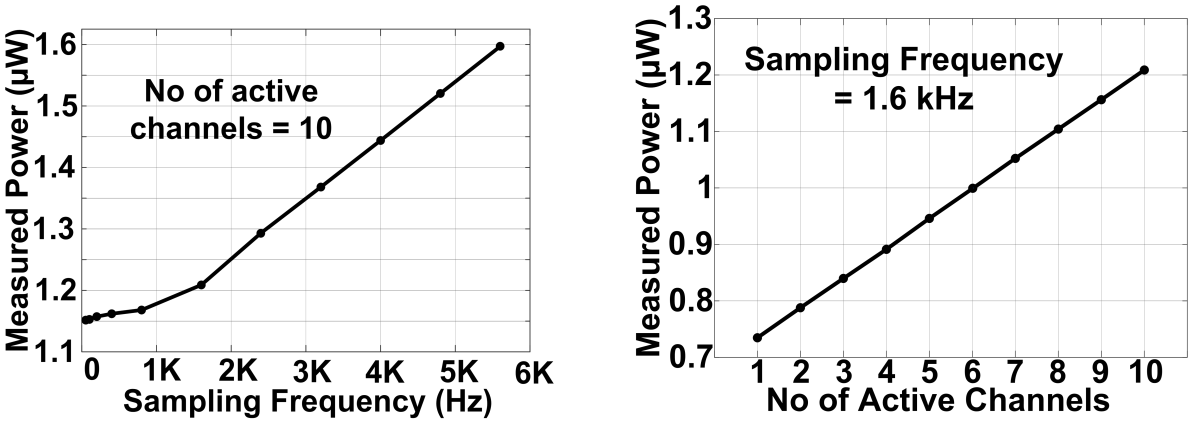


(a) Measurement results when only ch1 is calibrated (b) Measurement results per channel calibration

Figure 4.2: Top level chip overview and parasitic capacitance impact

Figure 4.3(a) shows the total power vs. sampling frequency with all channels active, and Figure 4.3(b) shows the total power vs. the number of active channels. Power consumption increases proportionally with transient times, and the measurement results also reflect this power trend. Similarly, a greater number of active channels results in higher power consumption.

Earlier, when explaining the reason for the development of monitoring logic, the Analog-to-Digital Converter (ADC) module was mentioned. Although it's implemented in Graphic Design System (GDS) format, the input signal range of the ADC can be determined by choosing from three reference input voltage options. Adjusting this voltage range influences the resolution of capacitance measurement. A larger input range offers broader coverage, but it also means a higher value per unit bit for capacitance which can be expressed in 6 bits. Conversely, a smaller input range reduces coverage but can lead to finer measurements due to decreased value per unit bit for

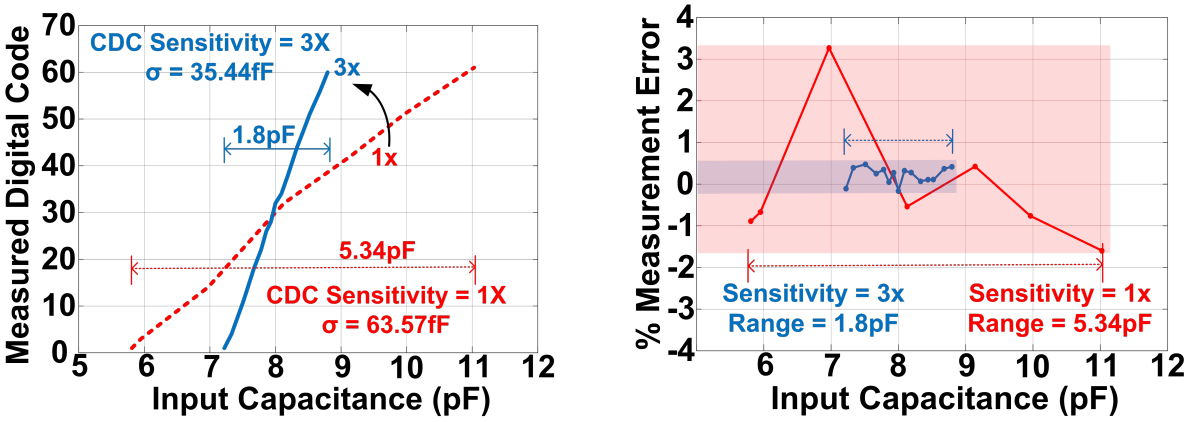


(a) Power consumption vs Sampling frequency

(b) Power consumption vs active channels

Figure 4.3: Power measurement results with different sampling frequency and active channel numbers

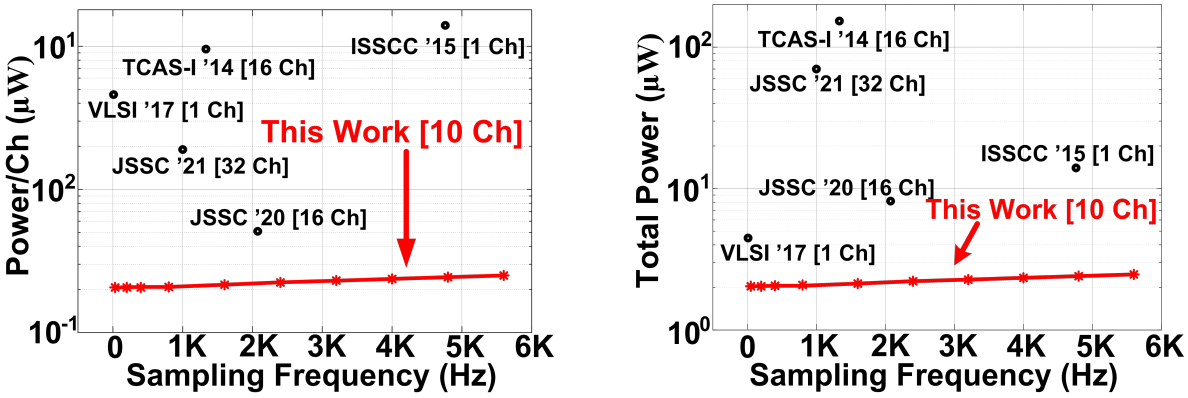
capacitance. To validate the adaptive CDC's sensitivity and sensing range, measurements were conducted across various ADC conversion ranges by adjusting the voltage references. For CDC sensitivity set at 1x, the measured sensing range was $5.34pF$, compared to $1.8pF$ at 3x sensitivity, as illustrated in Figure 4.4(a). However, the resolution at 1x sensitivity $(63.57fF)$ is lower than at 3x sensitivity $(35.44fF)$. This impact on resolution is also demonstrated in Figure 4.4(b), where 1x sensitivity exhibits a greater measurement error compared to 3x CDC sensitivity. The x1 and x3 labels illustrates that the CDC sensing range value of $1.8pF$ is three times smaller than the sensing range value of $5.4pF$. However, the former provides better resolution than the latter. Consequently, the x1 label corresponds to the $5.4pF$ case, while the x3 label denotes the $1.8pF$ sensitivity case.



(a) Sensing range based on CDC sensitivity (b) Measurement error by CDC sensitivity

Figure 4.4: Adaptive sensing range and error range by different sensitivity

Figure 4.5(a) shows the power consumption per channel by frequency compared to State-of-the-Art. The interesting point of view is that the power consumption per channel does not increase noticeably as the frequency increases. Figure 4.5(b) also shows total power consumption by frequency compared to State-of-the-Art. Even though all 10 channels are activated, it shows better performance in terms of low power design than previous work.



(a) Comparison of Power Consumption Per Channel with State-of-the-Art (b) Comparison of total Power Consumption with State-of-the-Art

Figure 4.5: Comparison of Power Consumption with State-of-the-Art

Chapter 5

Conclusion

5.1 Overview

Wearable technology, medical monitoring devices, and Internet of Thing (IoT) sensors are fields of interest for many engineers and are powerful systems. Moreover, they are constantly growing fields of technology, but has created many new problems for researchers to resolve. The goal of this thesis is to emphasize the role of digital design related to mixed circuit systems, and explain the potential to solve problems in analog circuits through the development of digital logic. Collaboration between analog and digital design is essential to maintain high accuracy and reliability in applications that require improved power efficiency and performance. Solving seemingly complex analog circuit problems using the high flexibility and simple logic of digital design is a truly fascinating study. From the Analog-Front-End (AFE) mixed circuit system perspective, digital does not play a role in directly handling physical signals. However, behind the scenes, the digital circuit has been in charge of supporting analog design so that it can perform its role well. In the project introduced in this thesis, digital design sets up and supports analog operation. However, it is important to note that since technologies have developed, the field that digital logic must perform is becoming increasingly larger. The role of digital logic is also important in the verification field. This thesis described that the initial test of an analog circuit can be completely performed by setting the initial value of the Serial Peripheral Interface (SPI) slave. For healthcare

wearables and IoT sensors, miniaturization of semiconductors and peripheral devices is essential for the development of small and lightweight devices, and low-power operation is required. To meet these demands, efforts will be needed to improve data processing and efficiency of digital logic. Therefore, the multi-channel Capacitance-to-Digital Converter (CDC) model introduced in this thesis will be a good reference for AFE development that can support various applications and can provide insight into digital design exploration.

5.2 Future work

5.2.1 Duty cycling

The duty cycle of the current sampling clock, set at a 50:50 ratio, can be modified and developed to support low-power design by providing a flexible option. We anticipate that this can be accomplished by altering the clock scaling logic in the previously mentioned digital design.

5.2.2 Multiple input frequency

Beyond the current step of adjusting the speed of the sampling clock, the goal is to change the clock speed for each channel individually. To achieve this, I need to investigate how to reduce the unit size responsible for scaling the clock speed. Additionally, when the Master transmits a command to set the clock speed, the command should be implemented in the digital logic with as little delay as possible. The current digital design reflects the register setting per 10 channel period, and this setting should be changing to single channel processing unit.

5.2.3 Function to automatically detect sensor failure

Auto-sensing technology is a highly useful function from an application perspective. To implement this, we must first define about a malfunction. It is essential to develop logic that can switch off by sending a feedback signal to the digital logic upon detecting a failure. Digital design,

which processes continuous analog signals, is expected to require the development of logic that can precisely recognize the boundaries of each channel.

5.2.4 Parallel processing

I would like to emphasize that this represents the pinnacle and ultimate function of multi-channel technology. To achieve this, all architectures must be re-configured to focus on parallel processing. Given that the 10 input signals are input randomly, the valid point in time for each signal will be different. Appropriate response to this random input will be one of challenge. Additionally, the response should be precise and have a minimum delay as an interface design perspective. To store the data temporarily, the internal register design can be considered. Furthermore, the direction of internal design changes and development will depend on whether the output is in parallel form or as a single output.

Bibliography

[1] O. Faruqe, D. Lee, N. Ownby, and B. H. Calhoun, "A 10-channel, 1.2 μw, reconfigurable capacitanceto-digital converter for low-power, wearable healthcare applications," in *2023 IEEE Biomedical Circuits and Systems Conference*, vol. 1, 2023.

[2] X. Zhang, M. Liu, B. Wang, H. Chen, and Z. Wang, "A wide measurement range and fast update rate integrated interface for capacitive sensors array," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 1, pp. 2–11, 2014.

[3] Y. He, Z.-y. Chang, L. Pakula, S. H. Shalmany, and M. Pertijs, "27.7 a $0.05mm^2$ 1v capacitance-to-digital converter based on period modulation," in *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*, 2015, pp. 1–3.

[4] B. Yousefzadeh, W. Wu, B. Buter, K. Makinwa, and M. Pertijs, "A compact sensor readout circuit with combined temperature, capacitance and voltage sensing functionality," in *2017 Symposium on VLSI Circuits*, 2017, pp. C78–C79.

[5] M. M. Mahmud, C. Seok, X. Wu, E. Şennik, A. Biliroğlu, O. J. Adelegan, I. Kim, J. S. Jur, F. Y. Yamaner, and Oralkan, "A low-power wearable e-nose system based on a capacitive micromachined ultrasonic transducer (cmut) array for indoor voc monitoring," *IEEE Sensors Journal*, vol. 21, no. 18, pp. 19 684–19 696, 2021.

[6] J. C. Yang, J.-O. Kim, J. Oh, S. Y. Kwon, J. Y. Sim, D. W. Kim, H. B. Choi, and S. Park, "Microstructured porous pyramid-based ultrahigh sensitive pressure sensor insensitive to strain

and temperature," *ACS Applied Materials & Interfaces*, vol. 11, no. 21, pp. 19 472–19 480, 2019.

[7] A. dos Santos, E. Fortunato, R. Martins, H. Águas, and R. Igreja, "Transduction mechanisms, micro-structuring techniques, and applications of electronic skin pressure sensors: A review of recent advances." *Sensors*, vol. 20, no. 16, p. 4407, 2020.

[8] Y. Luo, Y. Li, A. V.-Y. Thean, and C.-H. Heng, "An 8.2- μw $0.14 - mm^2$ 16-channel cdma-like capacitance-to-digital converter," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 5, pp. 1361–1373, 2020.

[9] X. Liu, S. Kamineni, J. Breiholz, B. H. Calhoun, and S. Li, "A 194nw energy-performance-aware lot soc employing a 5.2nw 92.6% peak efficiency power management unit for system performance scaling, fast dvfs and energy minimization," in *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 65, 2022, pp. 1–3.

[10] Y. Luo, Y. Li, A. V.-Y. Thean, and C.-H. Heng, "A 70-μw $1.35-mm^2$ wireless sensor with 32 channels of resistive and capacitive sensors and edge-encoded pwm uwb transceiver," *IEEE Journal of Solid-State Circuits*, vol. 56, no. 7, pp. 2065–2076, 2021.

[11] X. Tang, S. Li, L. Shen, W. Zhao, X. Yang, R. Williams, J. Liu, Z. Tan, N. Hall, and N. Sun, "18.2 a 16fj/conversion-step time-domain two-step capacitance-to-digital converter," in *2019 IEEE International Solid-State Circuits Conference - (ISSCC)*, 2019, pp. 296–297.