A LOW POWER LOW NOISE HIGH ACCURACY SENSOR IC

By

HAIDONG GUO

A dissertation submitted in partial fulfillment of
the requirements for the degree of

DOCTOR OF PHILOSOPHY

WASHINGTON STATE UNIVERSITY
School of Electrical Engineering and Computer Science

DECEMBER 2006

To the Faculty of Washington State University:

The members of the Committee appointed to examine the dissertation of HAIDONG GUO find it satisfactory and recommend that it be accepted.

Chair

ACKNOWLEDGEMENT

I would like to thank Professor David M. Rector in Department of Veterinary and Comparative Anatomy, Pharmacology, and Physiology for funding the research project. I would also like to thank Professor George La Rue for his advice and support on this dissertation.

I would also like to thank the School of Electrical Engineering and Computer Science for assistantship in completing my research.

A LOW POWER LOW NOISE HIGH ACCURACY SENSOR IC

ABSTRACT

By Haidong Guo, Ph.D.
Washington State University
December 2006

Chair: George S. La Rue

I investigated the design and implementation of low power low noise and high accuracy sensor IC for recording neural activity and studying sleep and other behavior in small animals. The sensor IC can acquire 16 electrophysiology signals in mice. It consists of 16 amplifier channels, a digital control circuit and a 16-bit 500 KSps charge redistribution self-calibrating successive approximation analog-to-digital converter (ADC). Each channel includes programmable gains from 12 to 250, a 7K Hz low-pass 2nd-order Butterworth filter and a track and hold. The integrated noise from 1 Hz to 7K Hz is 2.5 μV for 0 V DC offset input, 3.76 μV for 0.3 V DC offset input and 5.3 μV for –0.3 V DC offset input. The power supply rejection ratios (PSRR) for VDD and VSS are 61 db and 51 db at 1K Hz. The +/-0.3 V DC input offset of each channel is cancelled with two 5-bit DACs controlling the positive input node of the 2nd gain stage and 3rd gain stage op-amps. Total power dissipation is 1.2 mW for each amplifier channel with a +/- 1.5 V power supply. The 16-bit 500 KSps ADC has an input range of 2 V, a resolution of 16 bits, 6.2 mW power consumption and operates with +/- 1.5 V power supplies. Simulations show a signal-to-noise ratio of 90 dB for an effective accuracy of 15 bits in

TSMC's 0.25μ CMOS process. A novel interleaving architecture and an improved comparator design contribute to reducing the power while maintaining the accuracy and speed. The ADC is intended to digitize the amplified neurophysiological signals from the companion 16-amplifier-channel IC. The amplifier channel IC die area is 19 mm^2 and the ADC die area is 7 mm^2 in TSMC's 0.25μ CMOS process.

TABLE OF CONTENTS

LIST OF PUBLICATIONS

1. Haidong Guo, Rector, D.M., and La Rue, G.S., ”A low-power 16-bit 500 kS/s ADC,” in Microelectronics and Electron Devices, 2005. WMED '05. 2005 IEEE Workshop on April 15, 2005 Page(s):84 – 87
2. Haidong Guo, Champion, C.L., Rector, D.M., and La Rue, G.S., “A low-power low-noise sensor IC,” in Microelectronics and Electron Devices, 2004 IEEE Workshop on 2004 Page(s): 60 – 63

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

1. INTRODUCTION

Sensor interface circuits have applications in many areas to acquire analog signals from the real world and convert them to digital signals for storage and analysis by computer. Sensor circuits normally include amplification, filter and analog-to-digital conversion functions. A multi-channel low-power low-noise high-accuracy sensor integrated circuit (IC) is designed and implemented. Each channel includes gain stages, a low-pass filter, a track and hold circuit. An analog-to-digital converter design is also presented. The IC is designed to acquire neural signals of small animals for sleep research.

1.1 Background

An increasing number of electrophysiological channels are needed to record neural activity to study sleep and other behavior in small animals. Cellular-level neural network models make specific predictions about the effect of sleep on the behavior of neural networks. Testing these models can only be accomplished by recording many locations in the brain during natural sleep periods. New approaches to study the function of sleep involve mapping large brain regions with high temporal and spatial resolution.

Typically researchers have gathered data by burdening animals with wire tethers. High-impedance electrodes require amplifiers and low pass filters to be placed on the animal to provide gain and improve the signal-to-noise ratio. Small size and weight of the electronics as well as reducing cabling is important so as not to overburden small rodents.

Solutions range from single FET transistors [1] to miniature circuit boards with several amplifiers. Custom designed amplifiers ICs with several channels and some additional functionality have been reported [3-4]. Transoma Medical Company supplies remote sensors to provide monitoring in mice of only two biopotential (EEG, ECG or EMG) channels and are limited to a 100 Hz frequency response. Data Integrated Scientific System Company supplies remote telemetry products from with up to 6 biopotential channels. Unfortunately these multi-channel devices are too large for mice. Plexon offers preamplifiers on small printed circuit boards with 8, 16 and 32 channels and gain of 1 and 20 (www.plexoninc.com). Nicolelis has published results on 2 multi-channel integrated circuits for neural recording. These are small-volume 16-channel devices; one with gain of 2 and the other has high-pass filter and selectable gains of 250 and 500 (Obeid et al., 2003). However, these do not provide analog-to-digital conversion on the IC.

These solutions for many-channel electrophysiology still require significant cabling and relatively large size for small rodents, such as mice. Additionally, significant noise is introduced through the long cables before they can be converted into digital form. The most similar wireless implantable system to our needs was built by Akin in 1998 [5]. It uses an inductively coupled RF telemetry link to supply power and to passively transmit data from the implantable device. It can amplify neural signals by using on-chip 100 Hz to 3.1k Hz band limited amplifiers. It also includes an 8-bit analog-to-digital converter to digitize amplified neural signals. A brief comparison of those remote sensors is listed in table 1 along with our sensor IC chip specifications.

Table 1 Comparison of remote sensors

Project	Channels	Gain (V/V)	Noise (µVrms)	Band width (Hz)	Power /Channel (mW)	A/D (Bits)	Comments
P. Mohseni (2005)	1	100	7.8	0~9.1k	0.115	N/A	
Obeid (2003)	16	2/ 250/ 500	4.4	400~22k	1~1.4	N/A	Preamplifiers only
J. Morizio (2003)	16	1000	3.8	400-22k	9.25	N/A	
Akin (1998)	2	100	N/A	100-3.1k	2.8	8	Wireless data
Data Integrated	1~6	Up to 1000	N/A	0.1~125	N/A	7	Too large for mice
Transoma Medical	1~6	N/A	N/A	0~100	N/A	N/A	Wireless data
WSU	16	12 to 250	2.5	1~7k	1.6	16	Amplifier, Filter, A/D converter

The sensor integrated circuit we propose includes 16 channels. Each channel includes programmable gains from 12 to 250, a low-pass 2nd-order Butterworth filter with 7K Hz cutoff frequency, a track and hold circuit and a 16-bit 500 kSps analog-to-digital converter. The neural signal is amplified and filtered and then sampled by each channel. The held analog signals of 16 channels are sent alternately through a 16 to 1 multiplexer to a 16-bit resolution 500 KSps analog-to-digital converter. The on chip analog-to-digital converter provides a high-reliability digital data stream to a computer for analysis and archiving and avoids sending analog signals through cables, where noise from the outside world can be introduced. This single integrated circuit, with amplifiers, low pass filter, track and hold and analog-to-digital converter, can be implantable in small rodents to reduce the circuit size and weight as well as the cabling and noise.

1.2 Challenges

The first challenge comes from the DC offset of the neural signal. Typical neural signals from small animals have a DC offset up to +/- 0.3 V due to a difference in potential between the ground connection on the skull and the neural sensors in the brain. Most neural-sensor designs use ac coupling to remove the dc offset but this requires large off-chip capacitors to meet the bandwidth requirement of 0.1 Hz. Several alternative DC offset calibration designs were investigated to determine which one had the best performance. The first design that injected current into the feedback path of the op-amp to cancel the DC offset was found to inject too much noise into the channel. The second design that made the source degeneration resistors in an op-amp differential pair input transistors asymmetrical to cancel the offset had low noise but poor spurious-free dynamic range (SFDR). The third design that controlled the bulk voltage of the op-amp differential pair input transistors to cancel the DC offset had low noise, high SFDR but poor power supply rejection ratio. The fourth design using an instrumentation amplifier but dissipated more power than single op-amp designs. The final solution to remove the DC offset with high accuracy low power and low noise is to cancel the DC offset in the second and third gain stages.

The second challenge is to minimize power consumption, die area and noise. It is difficult to keep the noise and power low at the same time. Trade-offs between noise and power as well as between noise and die area are carefully made to minimize power consumption, die area and noise. The total input-referred noise voltage in the pass band integrated from 1 to 7K Hz is 2.5 μV for 0 V DC offset input. The power consumption is 1.2 mW per amplifier channel.

The third challenge is the analog-to-digital converter. To lower the total power consumption of the sensor chip, a low power ADC is required while maintaining sufficient conversion speed and high accuracy. A novel interleaving architecture and an improved comparator design are introduced to the charge redistribution self-calibrating successive approximation analog-to-digital converter (ADC). This technique contributes to reduce the power while maintaining the ADC accuracy and speed. The total power dissipation of ADC is 6.2 mW. The average power dissipation per channel is increased by 0.4 mW due to the ADC.

The fourth challenge comes from learning proper analog layout techniques and implementing the design into a working chip. The first amplifier channel IC was fabricated and tested. It had poor power supply noise rejection and poor power bus layout. The power bus layout was fixed using focused ion beam technology but the chip wasn’t usable because of its poor power supply noise rejection. The poor power supply noise rejection was due to controlling the bulk voltage of first op-amp input differential pair with a resistor DAC driven directly from the power supply. These problems were corrected and a second amplifier channel IC with a different DC offset cancellation technique was fabricated. The second amplifier channel IC had op-amp oscillation problems due to a poor bias circuit layout and without enough op-amp compensation to allow for large process variations. After modifying the design of the op-amp and layout of the amplifier channel IC, a third amplifier channel IC was fabricated with individual analog component test structures, a one channel test structure, and only 8 channels for the sensor IC. To keep costs under control with of the additional layout area required by the test structures, only 8-channels were implemented. Eight channels was enough to verify the design, and would provide benefits over existing technology In addition, the ADC could be added to the final 16 channel version. The third version 8-channel amplifier IC works properly in addition to all of the test structures. It matches the design targets except for cross talk between channels of about 54 dB. This cross talk is due to poor layout of the ground bus. The 16-channel amplifier IC with integrated ADC is our goal. Layout of the ground has been modified and the ADC layout is nearing completion.

1.3 Proposed Solution

A low-power low-noise high-accuracy sensor IC is presented. The sensor IC is implemented with a novel DC offset cancellation technique. The ADC of the sensor IC has a novel pipeline structure, novel gain error calibration and an improved comparator design. Compared to the aforementioned works, the first advantage means that no off-chip capacitors are required to implement a high-pass filter to remove the DC offset from the signal. The second advantage is the sensor IC has a large 85 db SFDR. Other advantages of the design is its low noise performance, which is 2.5 µV for 0 V DC offset input, its low power consumption, which is only 1.6 mW per channel, and it samples all of the channels simultaneously, which eliminates inter-channel timing differences. In addition it has multiple programmable gains from 12 to 250. This sensor IC chip is the state-of-the-art considering its power consumption, speed, accuracy and several novel designs.

1.4 Application

There are great demands in the clinical and animal research arena for high fidelity low-power low-noise high-accuracy amplifier sensor systems. This sensor chip can be used to study or monitor creature brain activity, muscle movement and heart activity. It also can be used for brain-computer and brain-machine interfaces. The sensor chip has other applications; anywhere the need exists to acquire multiple channels of weak analog signals with up to 7K Hz bandwidth.

CHAPTER TWO

2. AMPLIFIER CHANNEL

2.1 Introduction

This sensor IC is designed for acquisition of 16 electrophysiology signals in mice. It includes 2 major blocks, 16 amplifier channels and a 16-bit 500 kSps analog-to-digital converter (ADC). Each amplifier channel includes programmable gains from 12 to 250, a 7K Hz 2nd-order low-pass Butterworth filter and a track and hold circuit. Experimental results show the amplifier channel has minimal 79 dB SFDR up to 7K Hz for a differential DC offset input signal. The input referred noise integrated from 1 Hz to 7K Hz is 2.5 μV. The +/- 0.3 V DC input offset of each channel is cancelled with two 5-bit digital-to-analog converters (DACs) controlling the positive input of the second and third gain stage op-amps. Total power dissipation for each amplifier channel is 1.2 mW with a 3 V power supply. The ADC [9] [10] is a charge-redistribution self-calibration successive approximation analog-to-digital converter with low power, low noise and 16-bit resolution and 15-bit accuracy at 500 kSps conversion speed. By using a self-calibration circuit, the accuracy is increased beyond the device matching limits. The power dissipation is 6.2 mW with a 3 V power supply.

Communication with the sensor IC is performed serially to reduce the number of interconnections. Each channel has 10-bit registers for offset cancellation and 3-bit registers for programmable gain. The serial input word has a three start bits, followed by a command, data and address field. A digital control circuit, which is formed using a state machine converts the serial input words into signals that control sampling, multiplexer timing and storage of data into registers to cancel the DC offset and to set the programmable gain. Figure 1 shows a block diagram of the sensor IC.

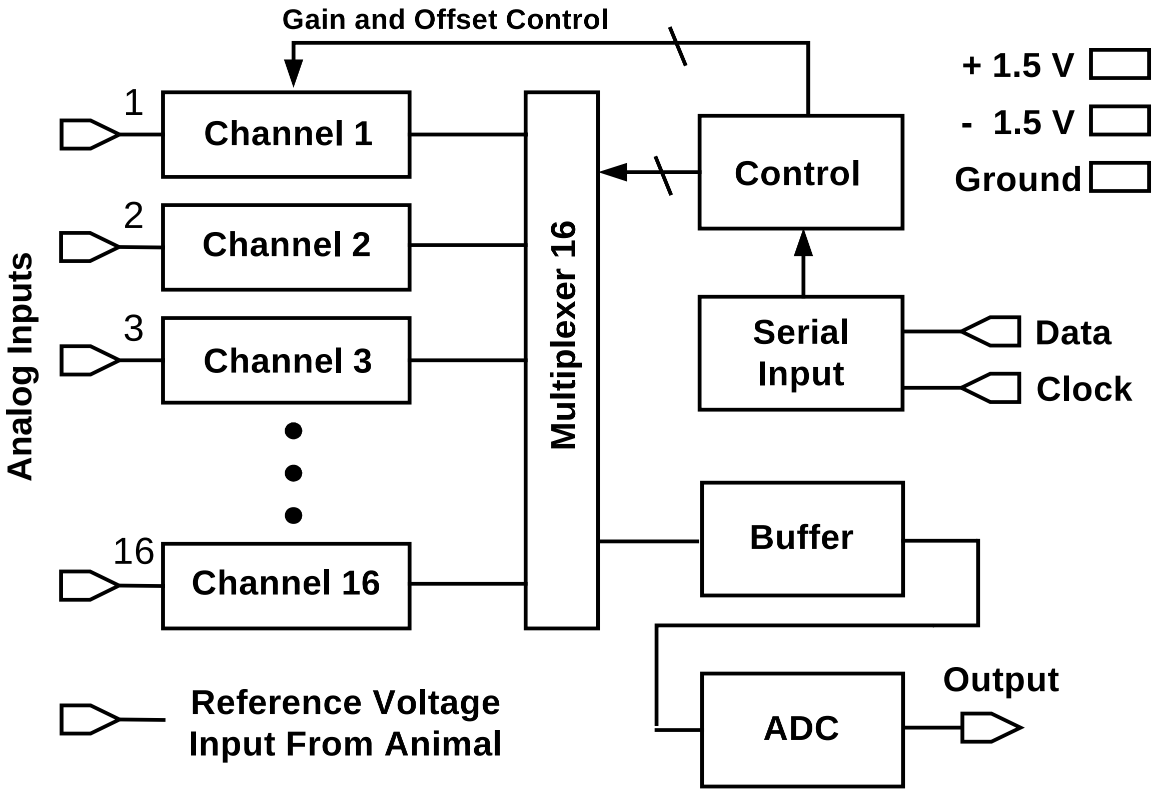


Figure 1 Sensor IC chip

The digital circuits operate from –1.5 V to 1.5 V. The analog circuitry operates from separate power supply pads at –1.5 V, ground and 1.5 V. Bypassing of the supplies can be performed external to the chip so that three power supply lines are required when operating with a tether. The number of wires in the tether is therefore 3 times the number of sensor ICs plus the 3 power supply lines. Although it is desirable to remotely power the sensor chip and wirelessly transmit the data, keeping the number of wires low is important to reduce weight when a tether is used. The design of the different circuits comprising the amplifier channel is discussed below.

2.2 Amplifier channels

The amplifier channel provides superior performance and functionality for recording electrophysiological data. Sixteen electrode inputs are provided along with a reference input from the skull of the animal. Each of the 16 channels amplifies and filters the difference between one of the 16 electrode inputs and the reference input. A DC offset may exist between the reference input and the electrode potentials. This offset is cancelled with DACs in each channel. Most neurosensor designs use ac coupling to remove the dc offset but this requires off-chip capacitors to meet the bandwidth requirement of 0.1 Hz or lower. Since input amplitudes can vary widely in different experiments the channels have programmable gains of 12, 30, 48, 100, 120 and 250. After filtering, all channels sample their signals simultaneously with separate track-and-hold circuits at rates up to 30 KHz. Sampling simultaneously eliminates inter-channel timing differences. An analog multiplexer selects the output from each channel one at a time to be routed to the output buffer amplifier and then to the ADC. The ADC converts the output signal from the output buffer amplifier into a digital signal. Having only one output reduces the number of wires in the tether.

Figure 2 shows a block diagram of an amplifier channel. The first operational amplifier is a 2-stage op-amp with p-channel input transistors. The first stage has a fixed gain of 2.5. The 2nd and 3rd gain stages have gains that are adjustable by controlling FET switches. Two 5-bit DACs cancel the DC offset to less than 0.2 V at the output of the 3rd gain stage by adjusting op-amp positive input node voltages in the 2nd and 3rd gain stages. 2nd-order low pass Butterworth filter with 7 KHz cutoff frequency follows the gain stages. The filtered signal is sampled simultaneously with other channels by a track and hold circuit. The output voltage from the track and hold circuit is held while the channels are output serially. A 1.5 gain buffer amplifies the held signal. The 1.5 gain is put at the end of the channel and not at the gain stages because this can lower the DC offset cancellation range and reduces the common-mode signal to these gain stage op-amps, which would lower the performance of these op-amps. Another reason for putting this 1.5 gain at the end of channel is to increase the linearity of the track and hold circuit.

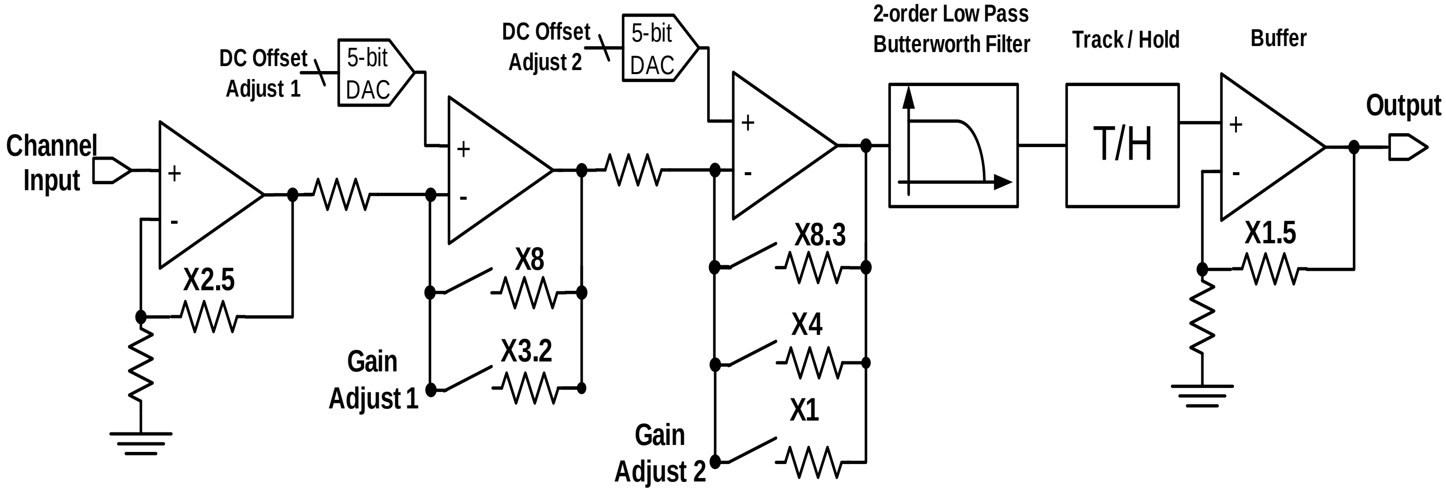


Figure 2 One amplifier channel

2.2.1 First stage operational amplifier

Requirements of first gain stage amplifier are low noise, low power, high gain, high SFDR and high power supply rejection ratio. Complicating the first stage design is that there is a DC offset signal as large as +/- 0.3 V due to a difference in potential between the ground connection to the skull and the neural sensors in the brain. Most neural sensor designs use ac coupling to remove the DC offset but this would require off-chip capacitors. Due to the large DC offset input the gain of the first stage is at most about 3 since the supplies are +/- 1.5 V, unless this offset is cancelled or partially cancelled at the input. It is desirable for the first stage amplifier to have a gain higher than 3 to reduce the noise contribution of the following amplifier stages. Several designs were investigated to try to archive low noise, low power, high gain, high SFDR and high power supply rejection ratio while canceling the DC offset. A design that injected current into the resistor feedback path of the op-amp to cancel the DC offset was found to injecting too much noise into the channel. This design requires a very low noise current source, with output noise below 1.1 μV, at least 4 times the die area and 20% more power consumption to reach the same performance as the current design. A design that made the source degeneration resistors in the differential input pair transistors of op-amp asymmetrical to cancel the offset had low power, low noise, nice power supply rejection ratio but poor SFDR. Its SFDR, which is an important specification, is 20 db lower than the current design. A design, which uses an instrumentation amplifier, dissipated twice the power and layout area compared to using a single op-amp design. A design that controls the bulk voltage of the differential input pair transistors of an op-amp to cancel the DC offset, shown in figure 3, had low noise, low power, high SFDR, high gain but poor power supply rejection ratio of only 10 db. The best solution for canceling DC offset and archiving high gain, high SFDR, high power supply rejection ratio low power and low noise is to set the gain of first gain stage to 2.5 and cancel the amplified DC offset in the second and third gain stage with two 5-bit DACs. The DACs control the positive input node of op-amps in the 2nd and 3rd gain stages. All of the gain stages are formed by standard two stages op-amps and feedback resistors.

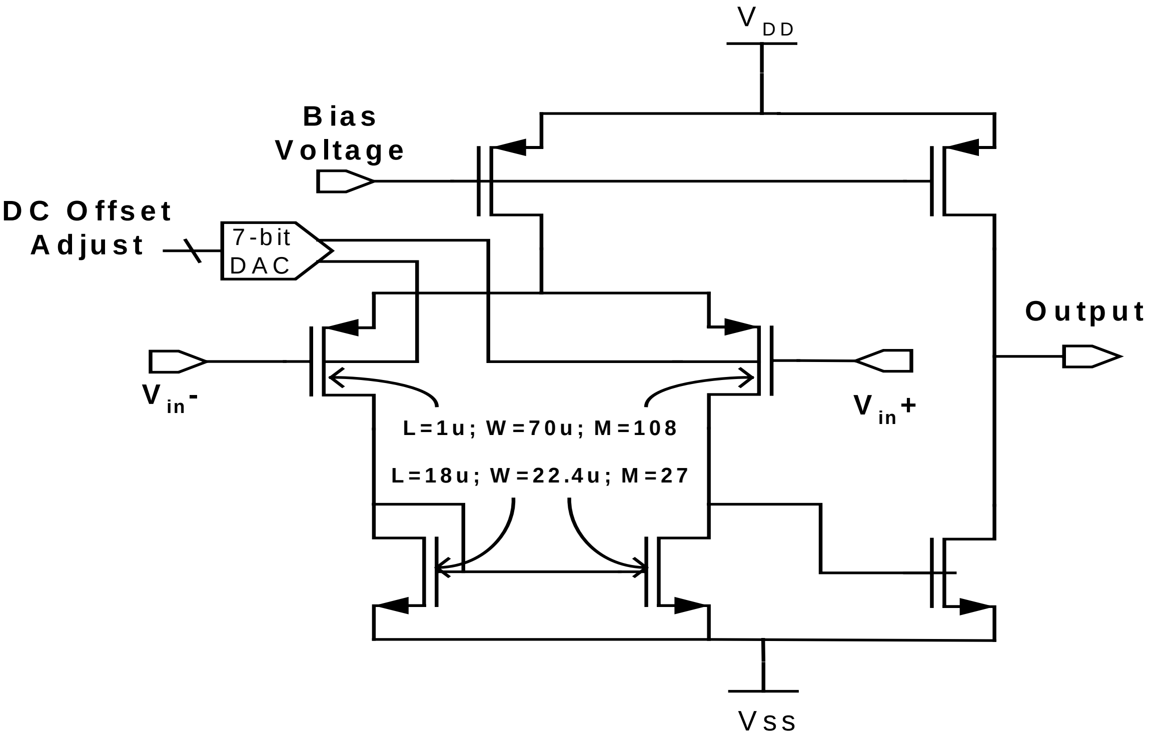


Figure 3 Input DC offset cancellation

The requirement of input referred noise of the amplifier channel is less than 2.2 µV in the band pass of 7 KHz. Flicker noise, which is inversely proportional to the active area of the transistors, dominates at low frequencies. This necessitates that the size of differential transistors and active load transistors in the first op-amp be made huge. At higher frequencies, thermal noise, which decreases with increasing current, dominates. This puts a lower bound on the power dissipation. So the first stage operational amplifier is designed with very large differential input transistors and active load transistors as well as relatively high current. The gain of the first stage was chosen to be 2.5 to ensure not to amplify the DC offset to which the second stage can't handle.

Switch capacitor gain amplifier as an alternative method to reduce the noise was discussed. The possibility of this method was researched. It uses correlated double sampling to minimize the flicker noise, which can save lots of die area without increasing the noise. Switch capacitor gain amplifier will be designed for the new version sensor IC

The foundry noise model overestimates the noise significantly at low frequencies and low gate-source voltages compared to measured noise data. The model parameter, flicker noise coefficient KF, was modified to provide a more accurate estimate of the expected noise. Measured amplifier noise confirms that the modification was appropriate.

2.2.2 2nd and 3rd gain stage operational amplifiers

The second gain stage amplifier provides a selectable gain of either 3.2 or 8 while the third gain stage amplifier provides a selectable gain of 1, 4 or 8.3. The variable gain is selected by switching resistors to different sizes. High value resistors are used to allow low-power op-amps to be used without limiting the output swing. High-valued resistors have more noise so there is a limit on how large they can be. The second amplifier has large-area differential transistors and active load transistors for lowering the flicker noise because the first gain stage has relatively low gain. The offset at the second stage input is up to +/- 0.75 V and can be cancelled within +/- 0.375 V at the second gain stage output. This large offset voltage can cause a large common mode signal at the second op-amp. This requires a NMOS type op-amp for 2nd gain stage, because the threshold voltage of the PFET in TSMC 0.25µ CMOS process is quite a bit larger than the NFET. After three gain stages the offset is cancelled within +/- 0.2 V in order not to exceed the range of the filter and track and hold circuit.

2.2.3 Trade off between noise, power consumption, output swing and layout area

The noise and power consumption are important specifications for this multi-channel amplifier IC. Two factors affect the total noise, flicker noise and thermal noise of MOSFETs. The first op-amp noise is the dominant noise source. The thermal noise of MOSFETs inside first op-amp contribute 15% of total noise of first op-amp. The rest of the noise is from MOSFETs flicker noise. Since the thermal noise is not the major noise source, power consumption per channel could be reduced to 0.3 mW thereby increasing thermal noise to 50% of the total noise of the first op-amp. But this would require very large resistor values to supply enough output swing, which in turn would increase the die area. A trade off between die area, noise and power consumption is made. The first op-amp power consumption is set to 0.3 mW. The feedback resistors are set to 21 KΩ and 14 KΩ to match the driving ability of first op-amp to supply sufficient output swing. The second op-amp feedback resistor values are determined by the driving ability of the first op-amp, with a minimum value is set to 110 KΩ. So the third op-amp feedback resistor values are determined by second op-amp driving ability, with a minimum value of 125 KΩ. The flicker noise of the input differential pair and active load MOSFETs of in the first op-amp contributes 75% of total noise of first op-amp. The flicker noise of the active load MOSFETs is divided by the gain when referred back to input so adjusting the MOSFET length ratio between differential input pair and active load to 1:3 reduces the noise contribution of active load by a factor of 9. A smaller MOSFET size is used for active load transistors to save die area without impacting noise adversely. The differential input pair MOSFETs size is set to match the noise design target, which has 1μ length, 30μ width and 216 fingers.

The two stage op-amp is selected because it requires fewer MOSFETs. Having more MOSFETs in an op-amp, such as a folded cascode op-amp, increases the size of chip when the noise is kept at same level.

2.2.4 5-bit DAC and bandgap voltage reference

Two 16-segment resistors form the 5-bit DAC. The bandgap voltage reference [2] supplies the constant voltage for the resistor DAC. First and second resistors DACs share the constant voltage reference to reduce power consumption. Because the first DAC is at the input of the second gain stage and the gain of the first gain stage is only 2.5, the noise of the bandgap voltage reference must be low enough to meet the noise design requirement. So the op-amp in the bandgap voltage reference is similar in size to the second gain stage op-amp to reduce flicker noise. The op-amp of the bandgap voltage reference also has relative high current to drive the resistor string DAC and have sufficiently low thermal noise. The bandgap voltage reference has high power supply rejection ratio to block the power noise going into the signal path.

The bandgap voltage reference is designed carefully to avoid oscillation due to the positive feedback path of the op-amp. Resistors of bandgap value are selected appropriately to make sure that the negative feedback of the op-amp is larger than the positive feedback of the op-amp.

The inverting op-amp buffer, which creates the negative voltage reference from the bandgap voltage reference, is well compensated to prevent oscillation. Good compensation of those two op-amps prevents interference between them, which might causes oscillation. Figure 4 shows detail structure of the 5-bit DAC, bandgap voltage reference and inverting op-amp buffer.

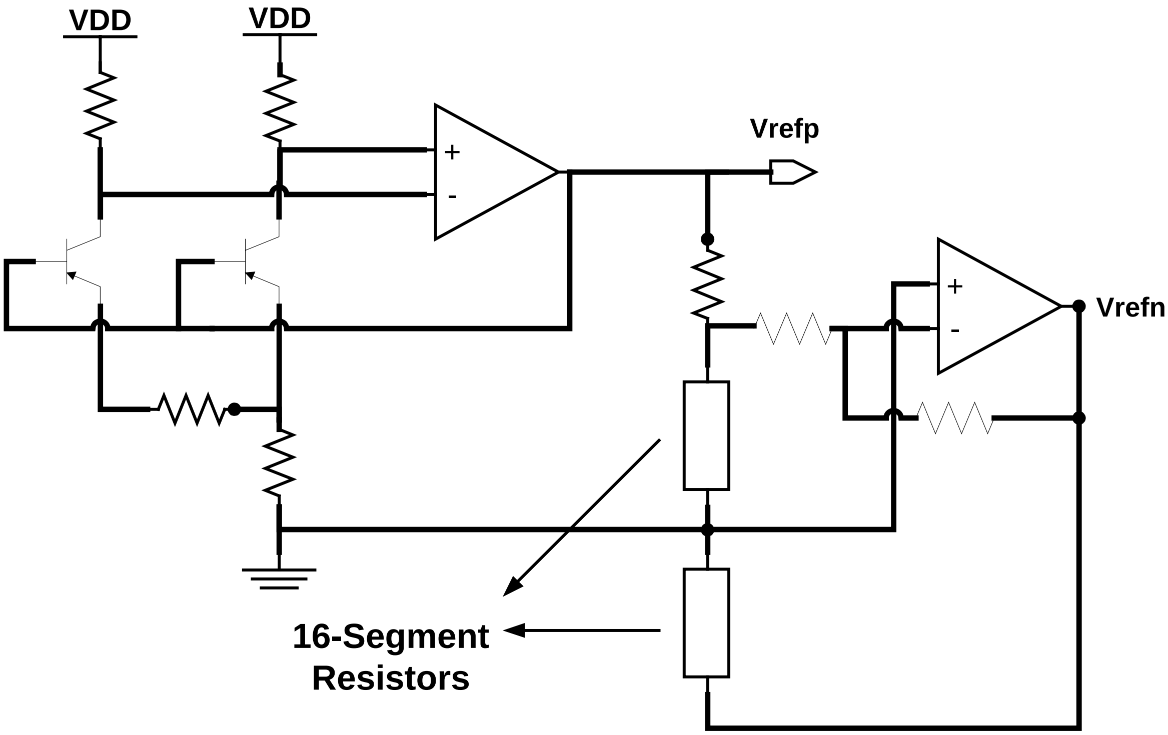


Figure 4 Bandgap voltage reference and inverting op-amp buffer

2.2.5 The 2nd -order low-pass Butterworth filter

The low-pass 2nd order Butterworth filter [7] showed in figure 5 filters the 3rd gain stage output signal. The design is a standard one. The op-amp was optimized for maximum gain value and minimal power dissipation in a frequency range of 1 to 7 KHz. A high-gain op-amp increases the linearity of the 2nd-order Butterworth filter. The sizes of capacitor and resistor were optimized to reduce the layout area.

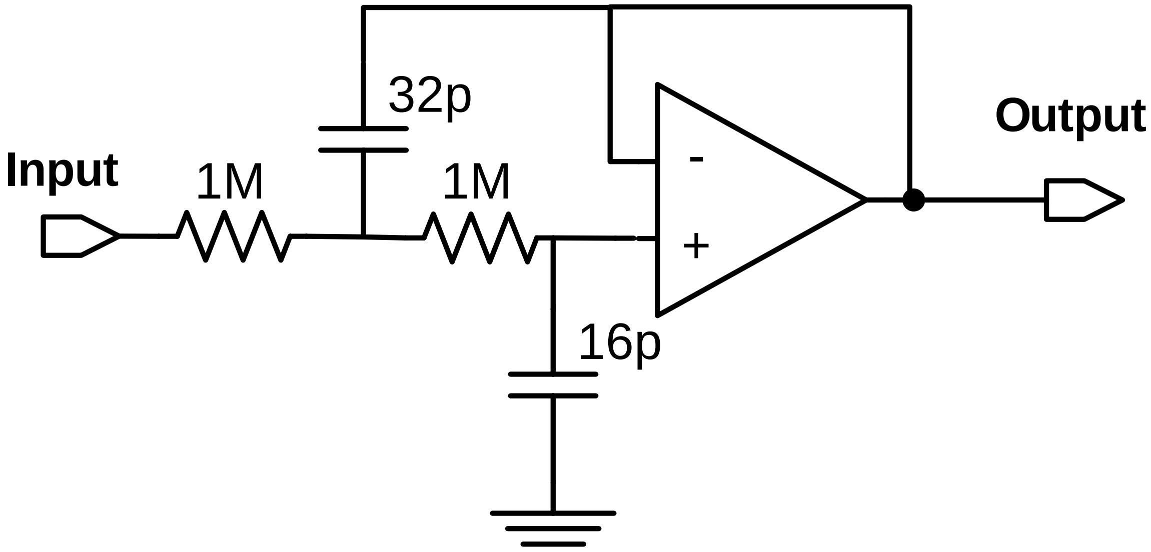


Figure 5 2nd-order low pass Butterworth filter

2.2.6 Track and hold circuit

The output of the Butterworth filter is sampled by a simple non-inverting track and hold with clock-feed through cancellation circuit [8], which is shown in figure 6.

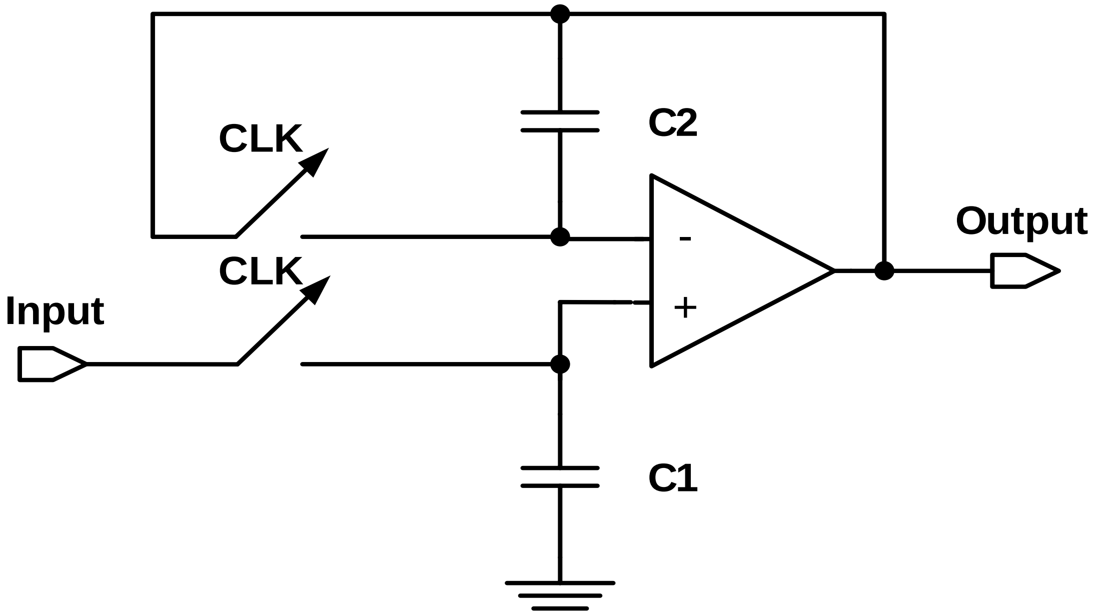


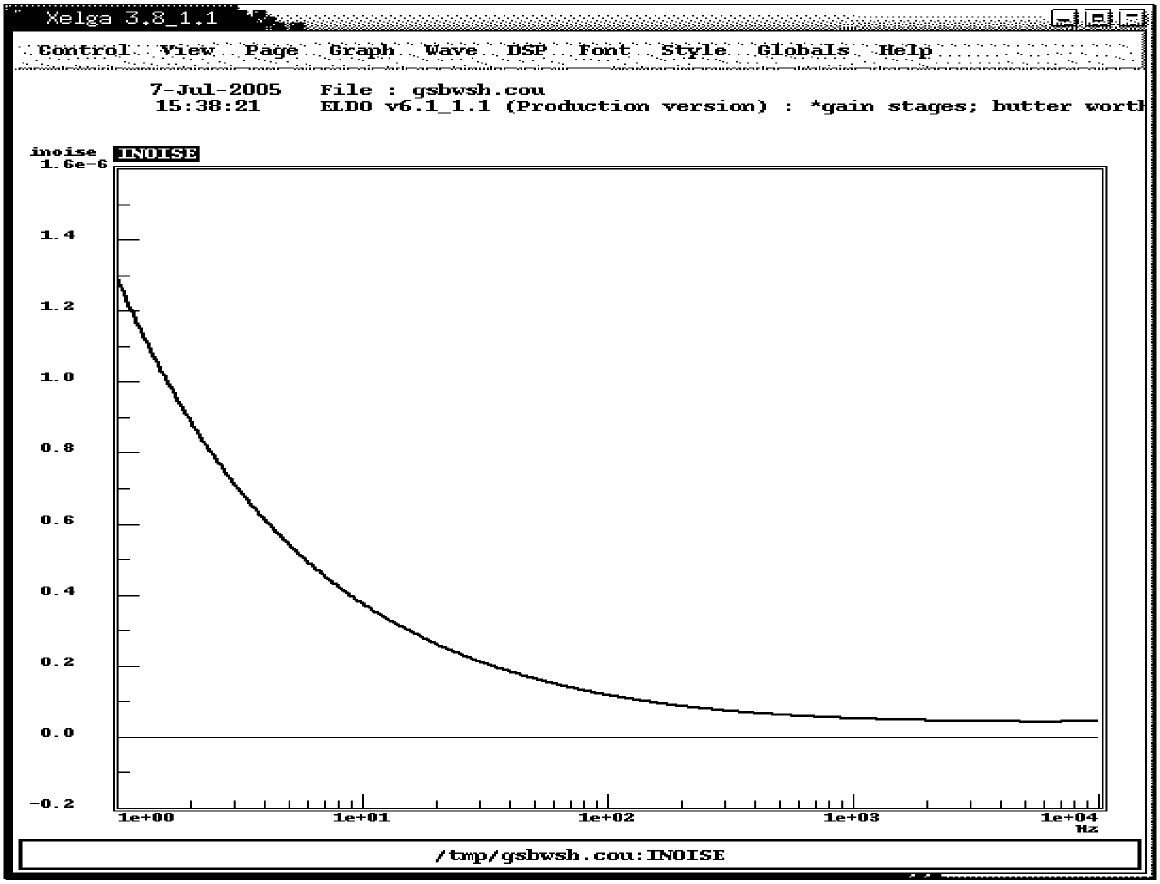
Figure 6 Track and hold circuit

All of the signals are sampled simultaneously to avoid timing errors between channels. The op-amp bandwidth is adjusted to exactly meet the 16-bit settling accuracy in its minimal holding time of 2 μs to save power.

2.3 Simulation results

2.3.1 Noise

Figure 7 shows noise curves for one channel vs. frequency with no DC voltage offset at the input and maximum channel gain 250. From the figure 7, the flicker noise is dominates at low frequency and thermal noise dominates at high frequency. The noise increases as the absolute value of the DC offset increases because the DAC, which is used for canceling the DC offset signal, injects noise into the channel.



```
7-Jul-2005    File : gsbwsh.cou
15:38:21      ELDO v6.1_1.1 (Production version) : *gain stages; butter wortl
```

Figure 7 Output noise vs. frequency

Table 2 DC Offset vs. simulation noise (Channel gain is 250)

DC Offset (V)	0.3	0	-0.3
Input Referred Noise Integrated in 7K Hz Bandwidth (μV)	2.7	2.2	3.76

2.3.2 SFDR vs. frequency and DC offset

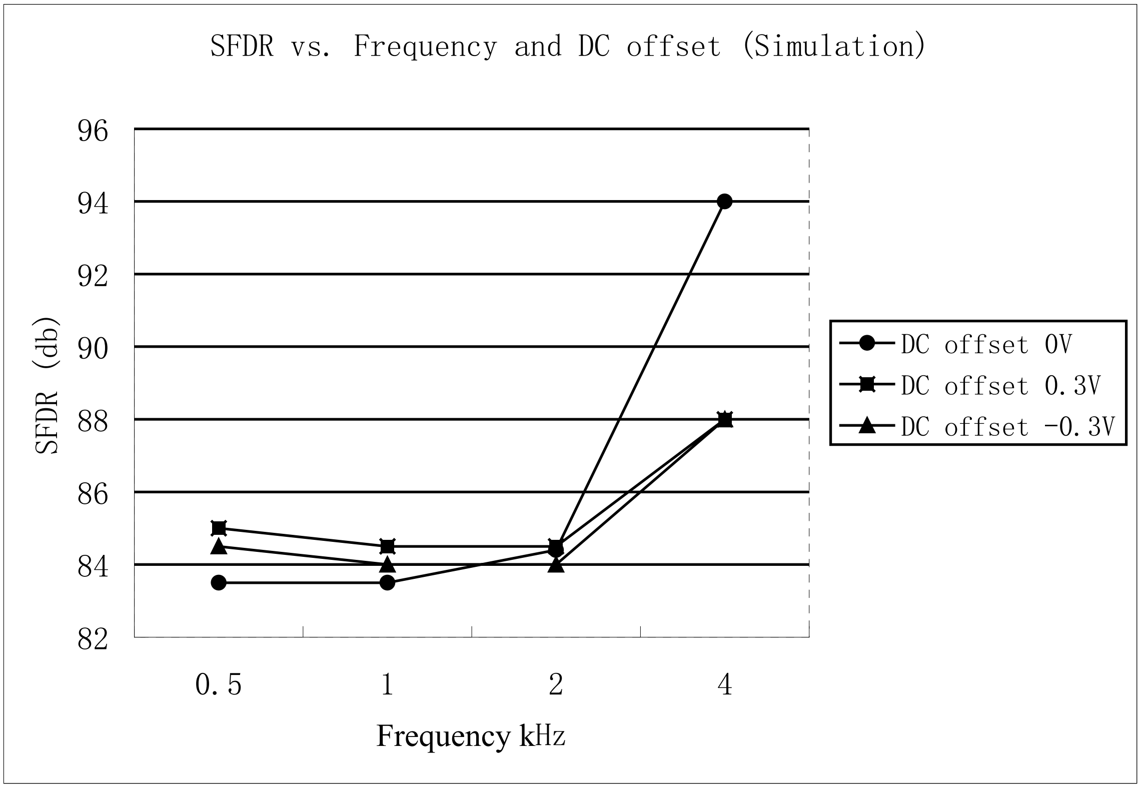


Figure 8 SFDR vs. frequency and DC offset

Figure 8 shows the SFDR vs. frequency and DC offset for the output of one channel at a channel gain 250. An SFDR of 84 db is obtained below 3 KHz. The SFDR is higher, with a minimum of 88 db, between 3.5 KHz to 7 KHz, because the second harmonic of the input signals is above 7 KHz and is filtered out by the on-chip low-pass filter.

The major limitations for SFDR comes from nonlinearity of resistors and capacitors. The op-amp open loop gain can also affect the SFDR of the channel if the op-amp open loop gain is not sufficiently high at high frequencies.

2.3.3 Power supply rejection ratio

The simulation results for power supply rejection ratios are shown in figure 9 and figure 10. The channel PSRR decreases as frequency increase, due to decreasing of the channel op-amps PSRR. There are 4 op-amps, which contribute most of the PSRR of channel. These are the 1st gain stage PMOS type two stage op-amp, the 2nd gain stage NMOS type two stage op-amp, the bandgap voltage reference PMOS type folded cascode op-amp and the inverting PMOS type folded cascode op-amp for creating the negative voltage reference for the resistor DACs. Folded cascode and two stage PMOS type op-amp have better PSRR+ than PSRR-. Two stage NMOS type op-amp has better PSRR- than PSRR+. Folded cascode op-amps normally have better PSRR than two stage op-amp. Therefore, the 1st gain stage op-amp PSRR+ has dominative effect on the whole channel PSRR+. Because the gain of 2.5, the 1st gain stage mitigates the poor PSSR+ of 2nd gain stage op-amp on whole channel PSRR+. When the DC offset increases, the effective voltage differential input pair of first gain stage op-amp, V_{eff}, reduces causing the transconductance, G_m, of the differential pair and the gain of the op-amp to reduce. This reduces the PSRR+ of the op-amp and is the reason that as the DC offset increases, the channel PSRR+ decreases. Because PSRR- is worse for both the folded cascode and two stage op-amp, the channel PSRR- is lower than the channel PSRR+ at the same frequency. The 2nd gain stage op-amp dominates the channel PSRR-. At high frequencies, the 1st gain stage op-amp PSRR- is mostly related to the compensation capacitor and not the open loop gain. So when the DC offset increases, the Gm of the differential pair and the gain of the second gain stage op-amp both increase increasing the PSRR- of the op-amp. This is the reason that as the DC offset increases, the channel PSRR- increases. Trade offs between PSRR and power consumption is made for signal bandwidth.

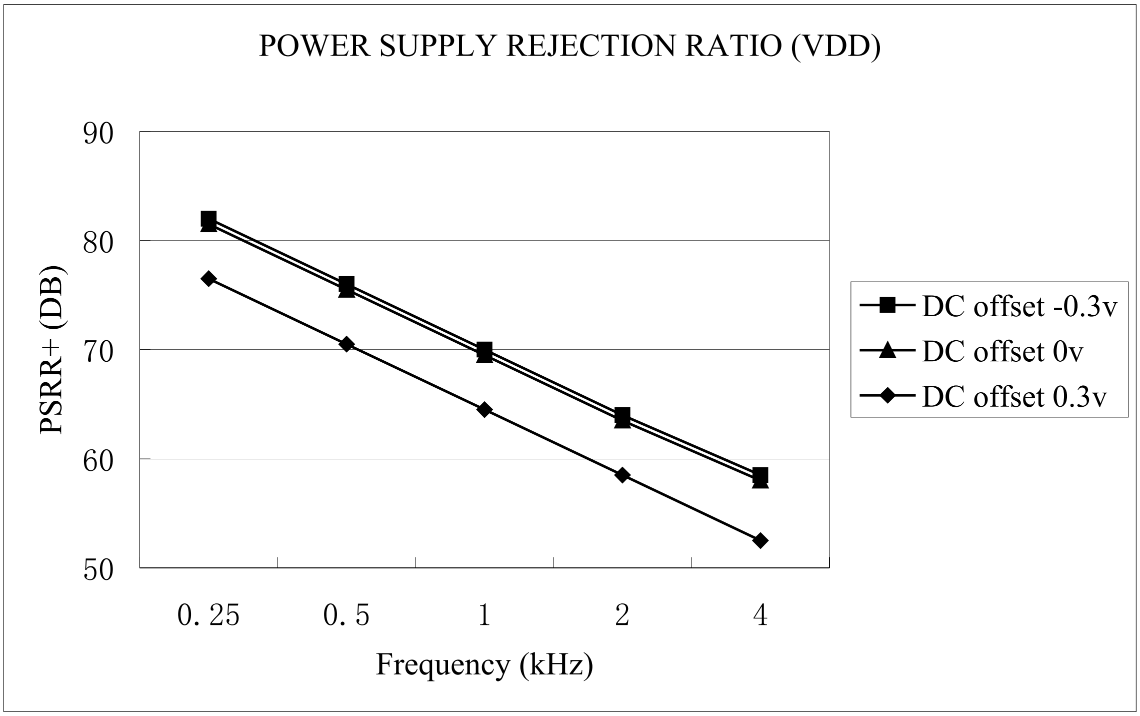


Figure 9 PSRR+ vs. frequency and DC offset

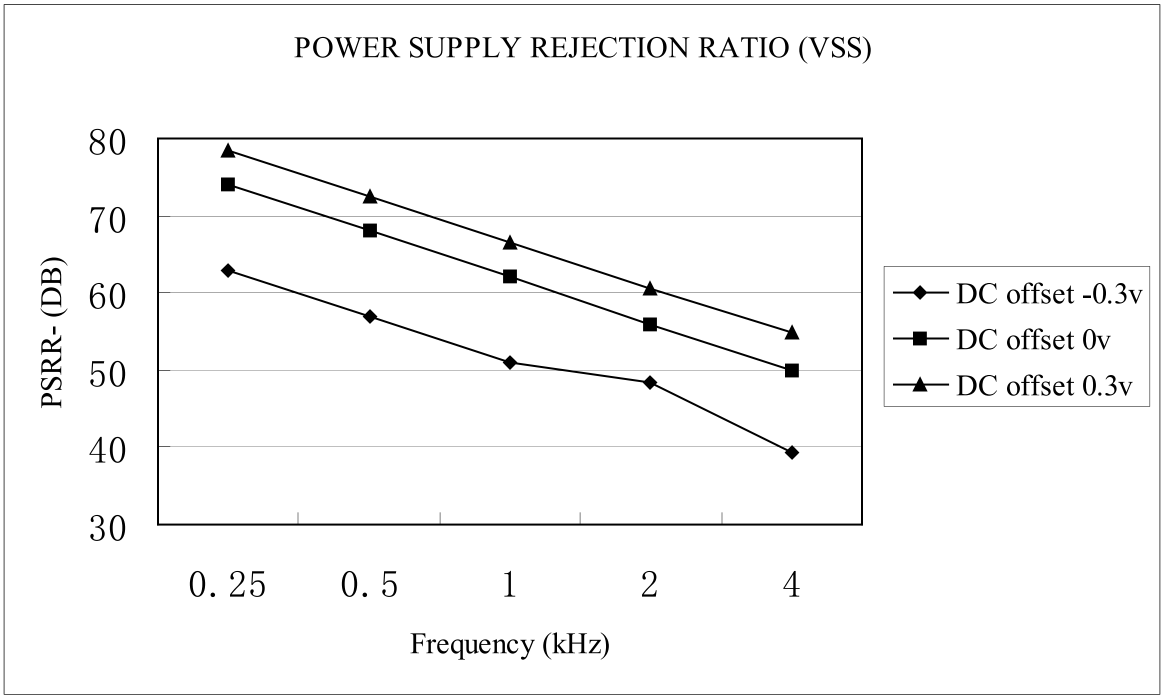


Figure 10 PSRR- vs. frequency and DC offset

2.3.4 Signal simulation

Figure 11 shows the output of the multiplexer with a 1 KHz sinusoidal input on channel 1 and a 3 KHz sinusoidal on channel 5. The multiplexer dwells 2 μs on each channel and the sampling is performed every 34 μs or at 30 KHz. The glitches in the output signal are due to charge injection and clock feed through of the FET switches in the multiplexer.

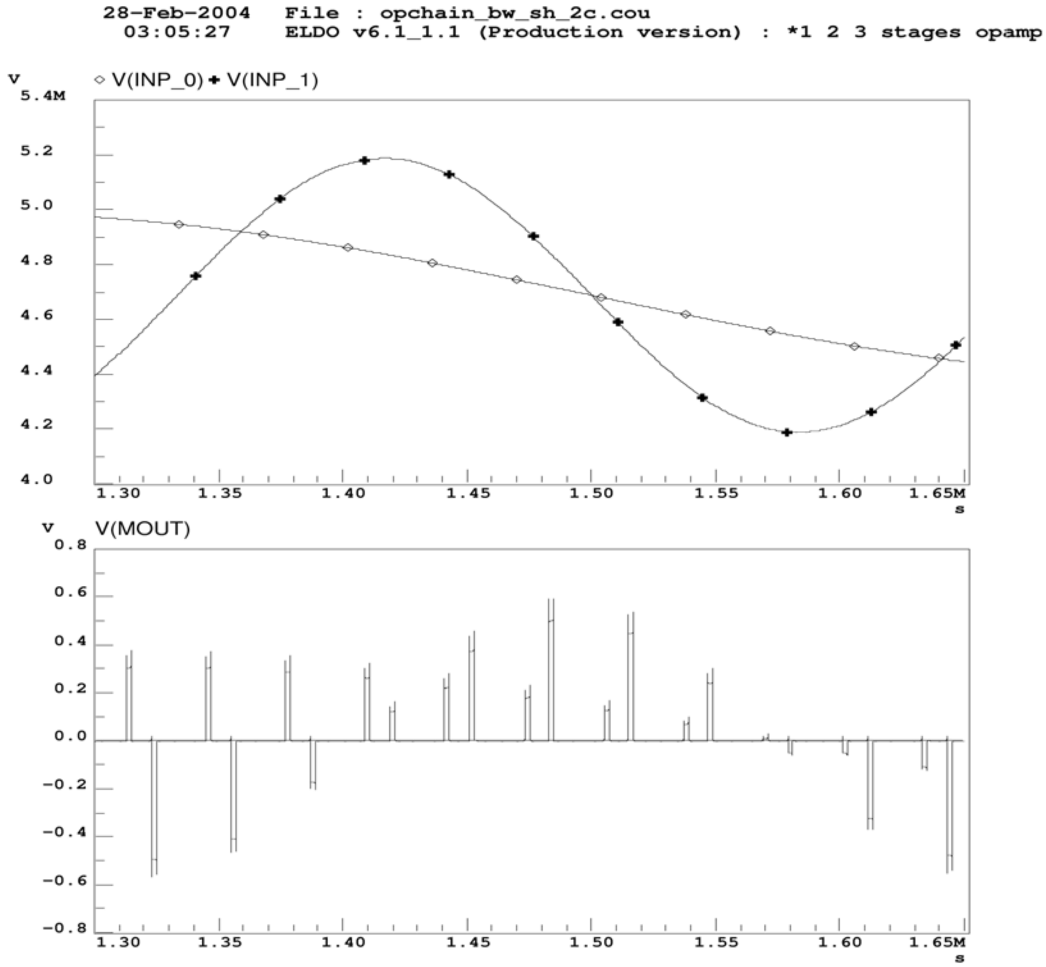


Figure 11 Signal simulation

2.4 Layout

The layout of the amplifier channel IC is shown in figure 12. 8 channels are laid out from top to bottom. Inside each channel, 1st, 2nd, 3rd gain stages, Butterworth filter and track and hold are laid out from left to right in order. The digital control circuit is in the right-hand bottom corner. The output buffer is in the right-hand top corner.

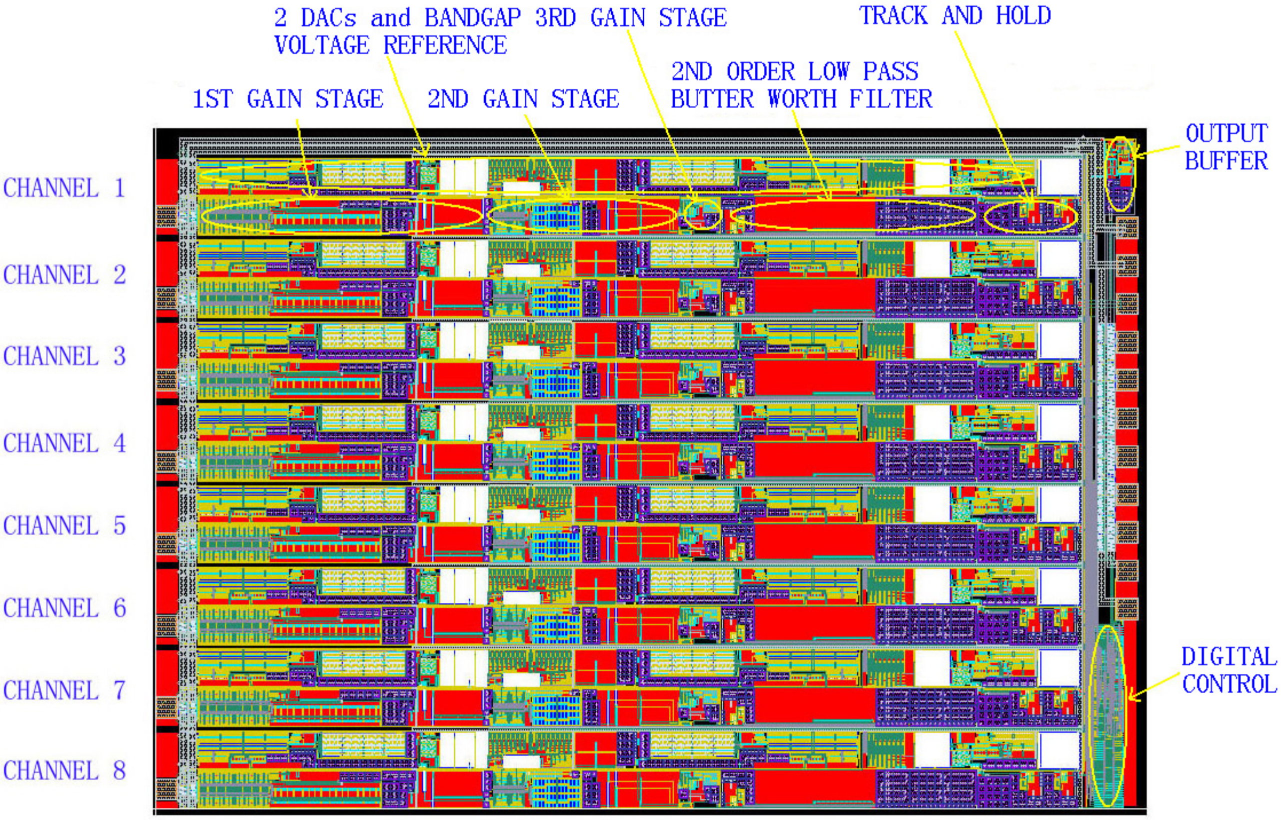


Figure 12 Amplifier channel IC layout

2.5 Measurement results

2.5.1 Measurement system

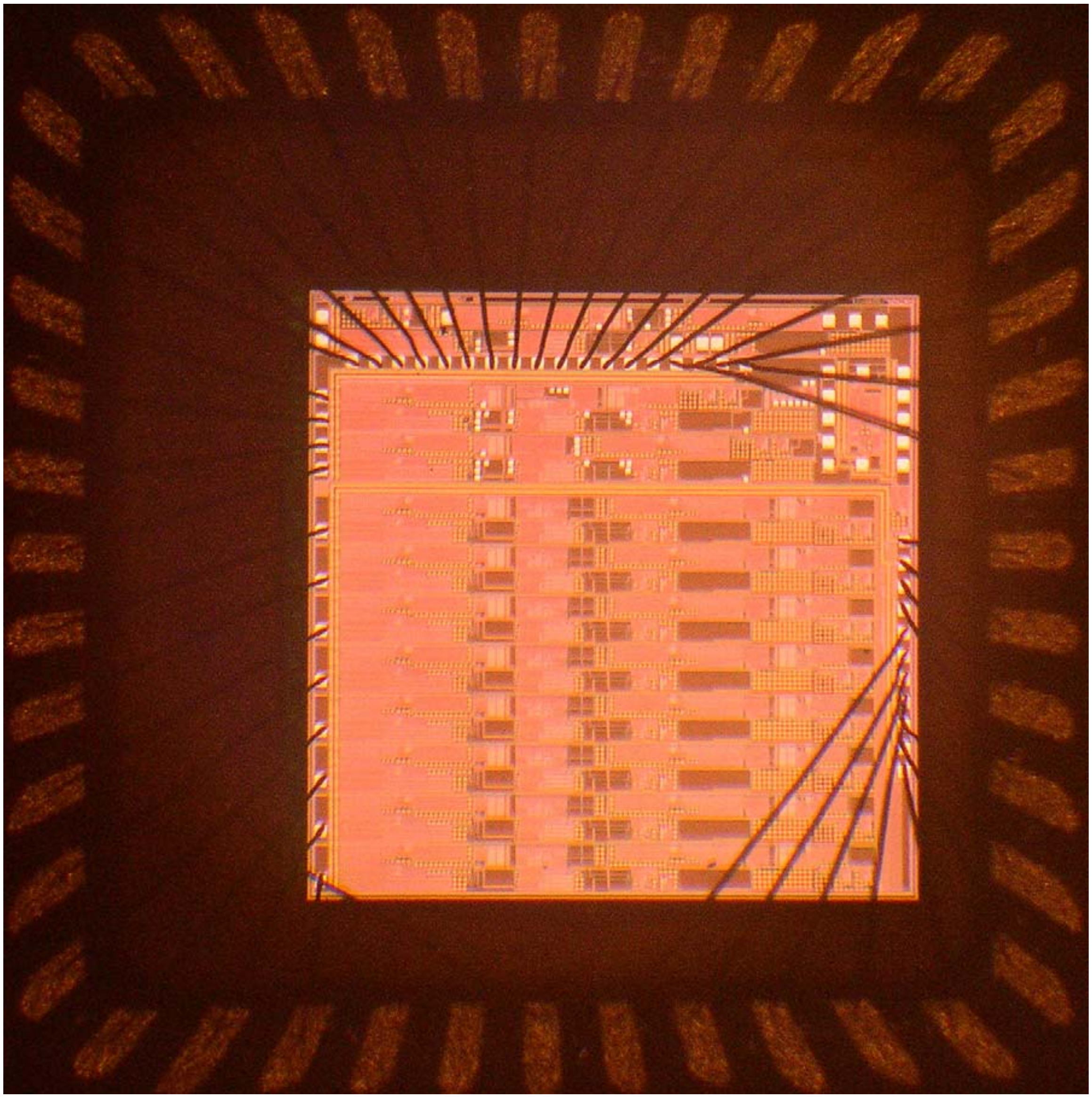


Figure 13 Die photograph of the amplifier channel chip

The amplifier channel chip was fabricated using the TSMC 0.25µ CMOS process. The chip is packaged in a 44 pin ceramic quad flat pack. Figure 13 shows the die photo. The measurement system includes a computer, which controls the multi-channel amplifier chip and data acquisition circuits. Board and amplifier channel IC and ADC circuit board.

The measurement system is shown in figure 14 and figure 15.

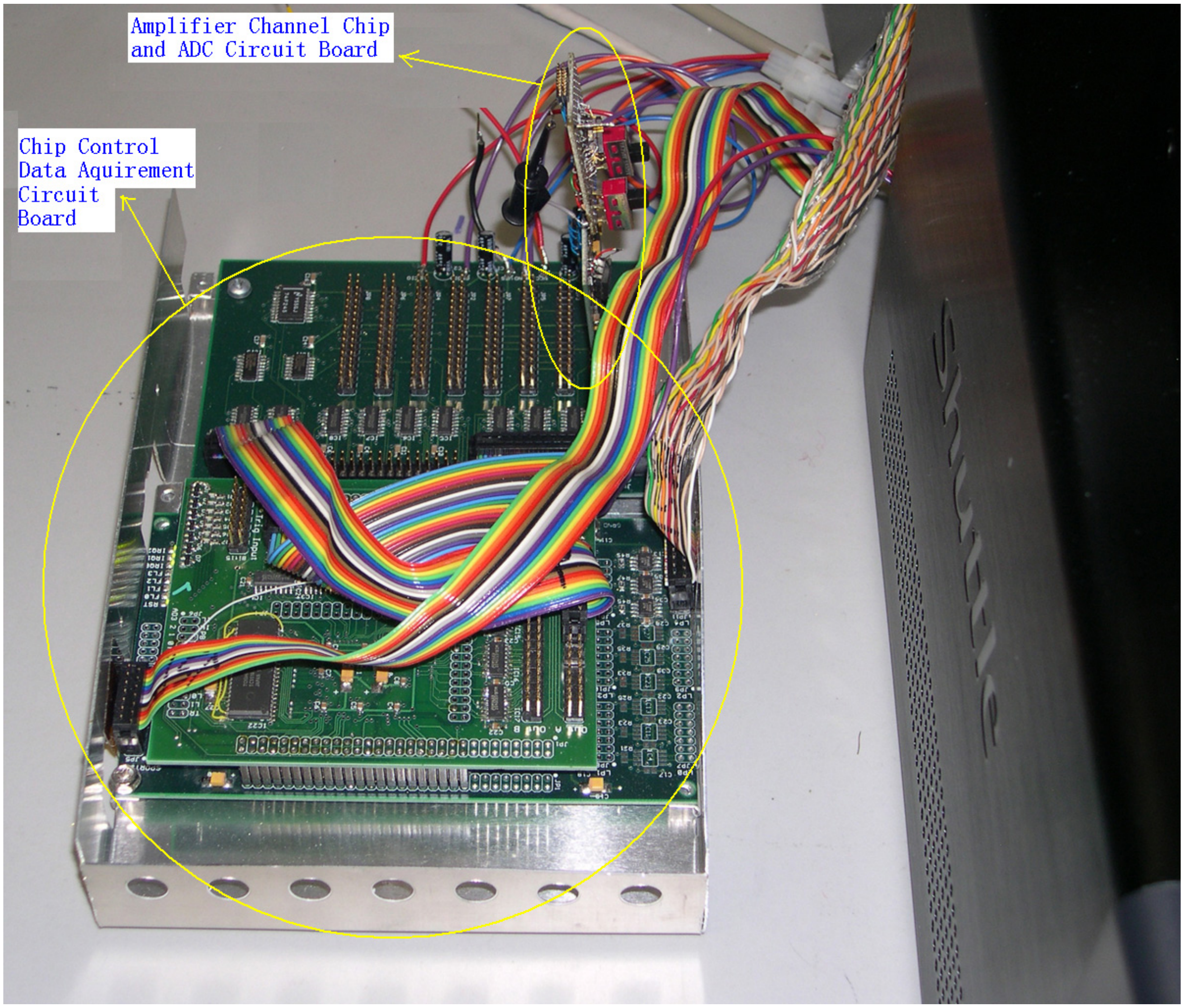


Figure 14 Measurement system

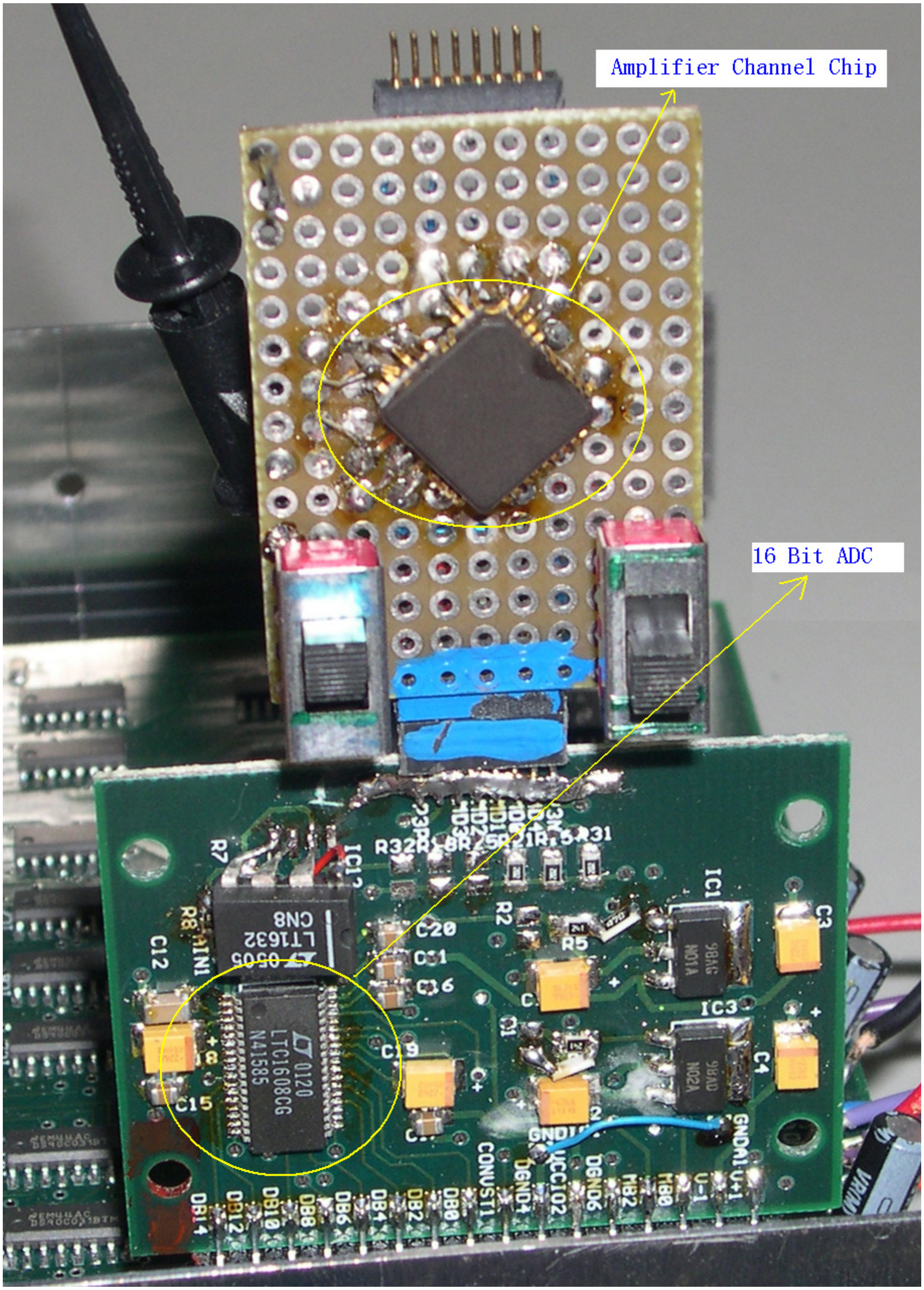


Figure 15 Amplifier channel chip and ADC circuit board

The packaged chip was mounted on a small PCB. The small PCB was connected to the ADC circuit board. A PCB, which can hold up to 8 ADC circuit boards, connects to a set of data acquisition and interface boards, which communicate to a computer. This measurement system enables computer control the chip, conversion of the analog signal to digital and storage of the digital signal into the computer for further analysis. Other equipment such as power supplies, millimeters, signal generators, a spectrum analyzer and oscilloscopes, are used in various measurements.

2.5.2 DC characterization

The power supply for the amplifier channel IC is 3 V. Power consumption for one amplifier channel was measured to be 1.2 mW. The amplifier channel chip can handle a +/- 0.3 V input DC signal and a +/- 0.1 V AC signal. The output signal has a +/- 0.8 V output range.

2.5.3 1st DAC INL and DNL

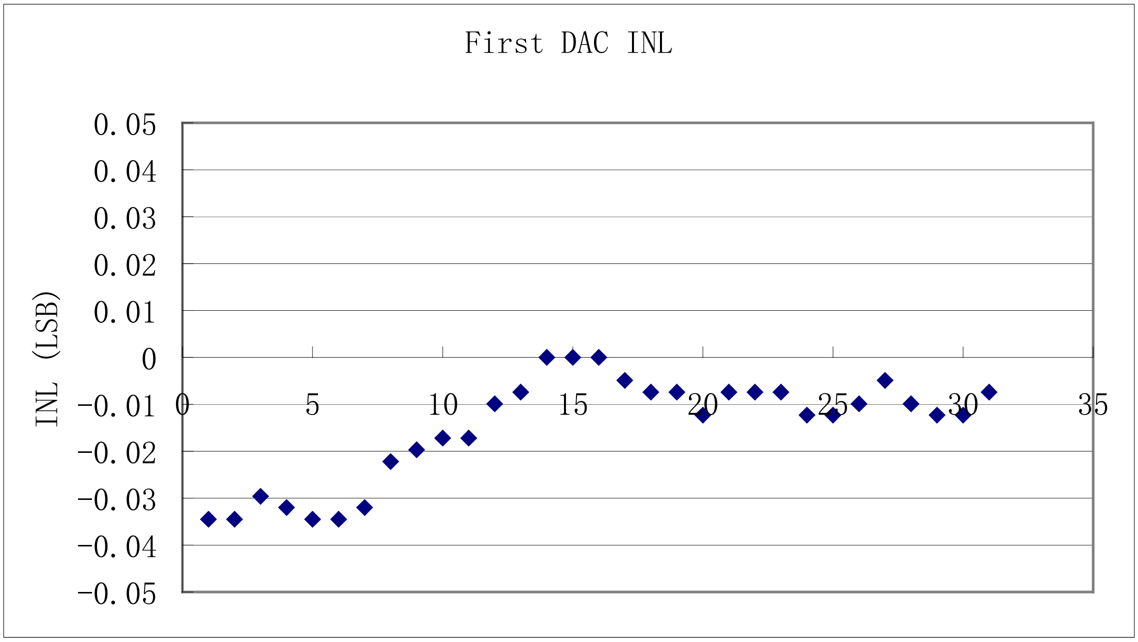


Figure 16 First DAC INL

The differential non-linearity (DNL) and integral non-linearity (INL) of first 5-bit DAC was measured. The result is shown in figure 16 and figure 17. The INL and DNL are very good.

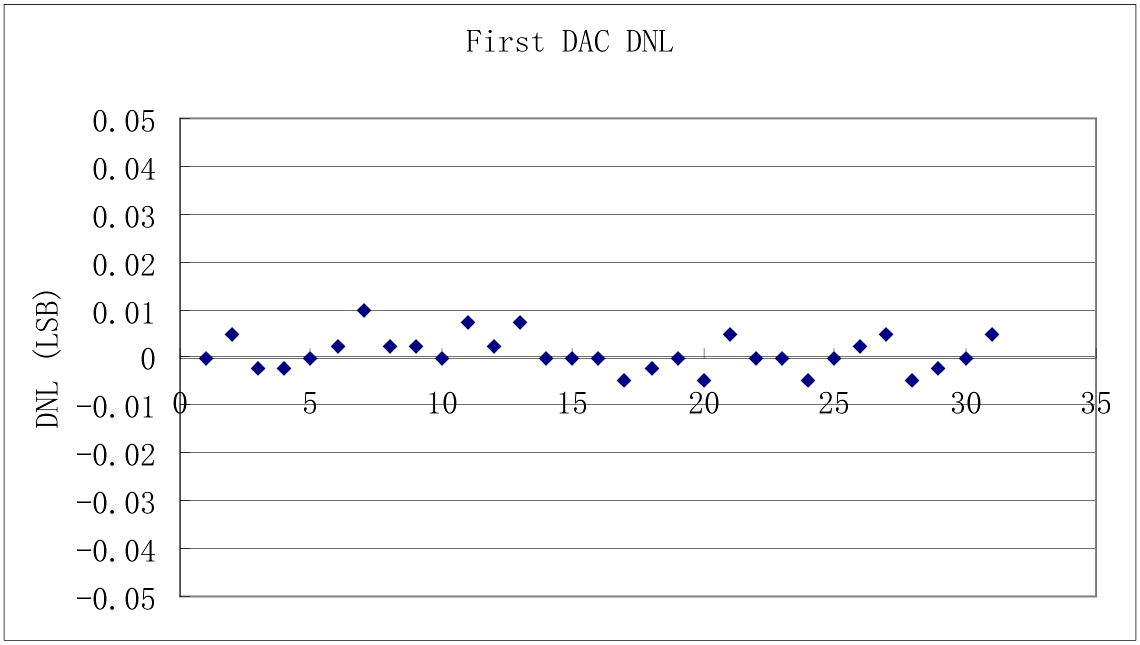


Figure 17 First DAC DNL

2.5.4 2nd DAC INL and DNL

The differential non-linearity (DNL) and integral non-linearity (INL) of second 5-bit DAC was measured. The result is shown in figure 18 and figure 19. The INL and DNL are also very good but not as good as the first DAC. The reason is that the second DAC has smaller resistor values, and matching is not as good.

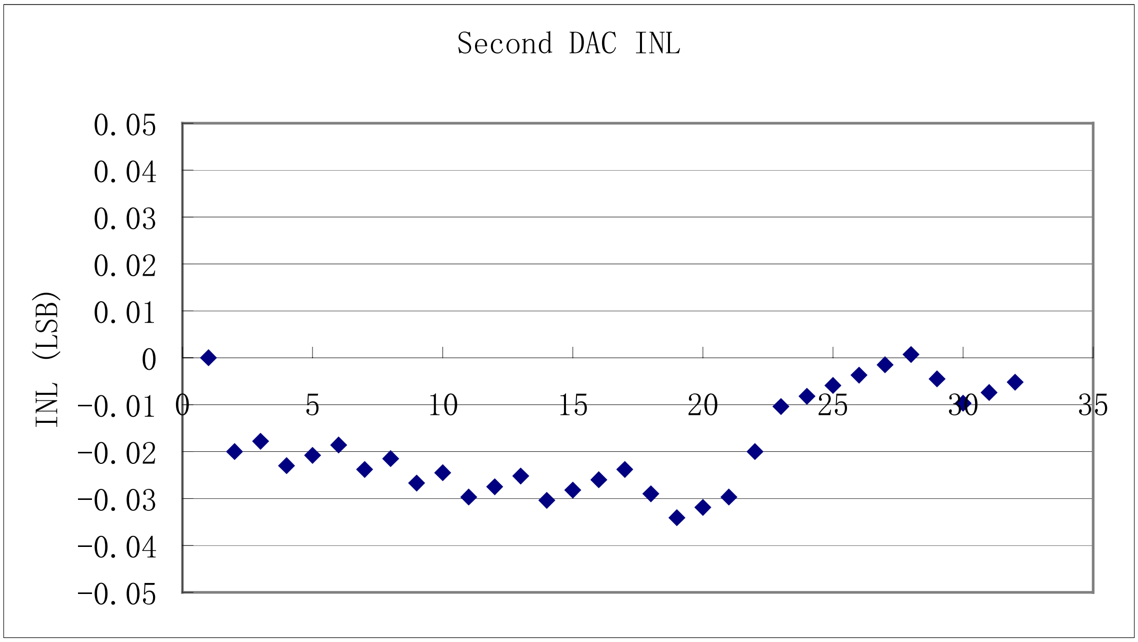


Figure 18 Second DAC INL

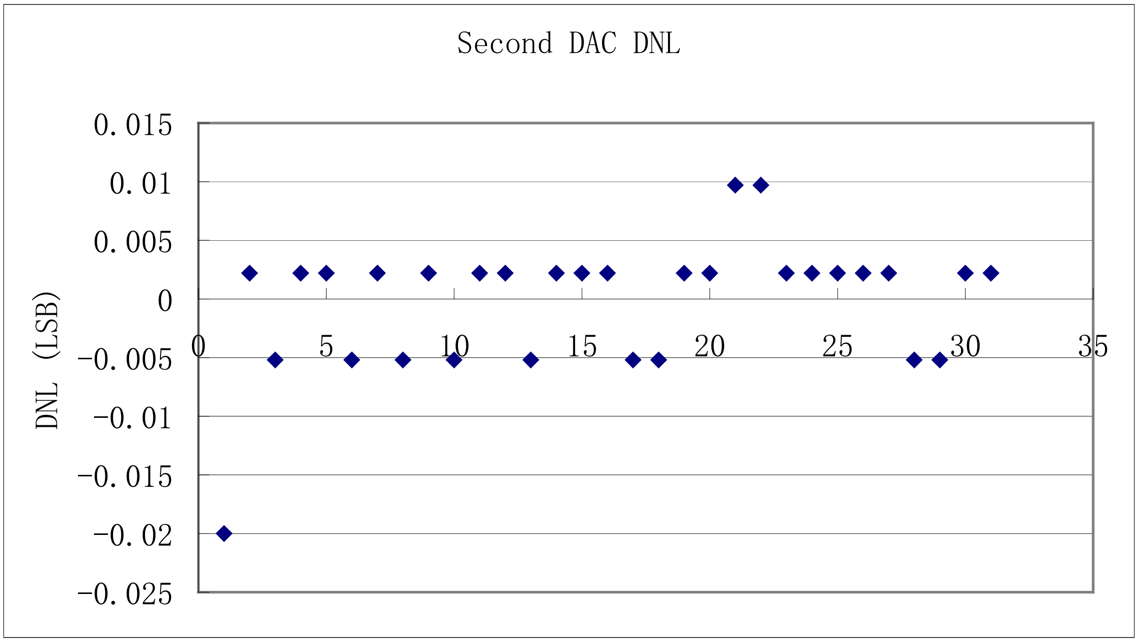


Figure 19 Second DAC DNL

2.5.5 Programmable gains

The programmable gain value is very close to the design values. This is due to the large resistor sizes used in the design, which help to make the resistors match very well.

Table 3 Programmable gain values

Ideal Gain Value	Measured Gain Value
12	11.986
30	30.013
48	47.947
100	99.992
120	120.025
250	249.564

2.5.6 Amplifier channel long term stability

The long term stability of the amplifier channel was measured. Results are shown in figure 20 and figure 21. The first DAC has 30 μV variations within 22 hours when referred back to the input. The second DAC has 22 μV variations within 22 hours when referred back to the input. The major variation affecting the long term stability is from the bandgap voltage reference used in the first and second resistor DACs. The bandgap voltage reference output voltage is related to temperature and power supply variation. Op-amps also have an affect on the long term stability and are also related to temperature and power supply variations.

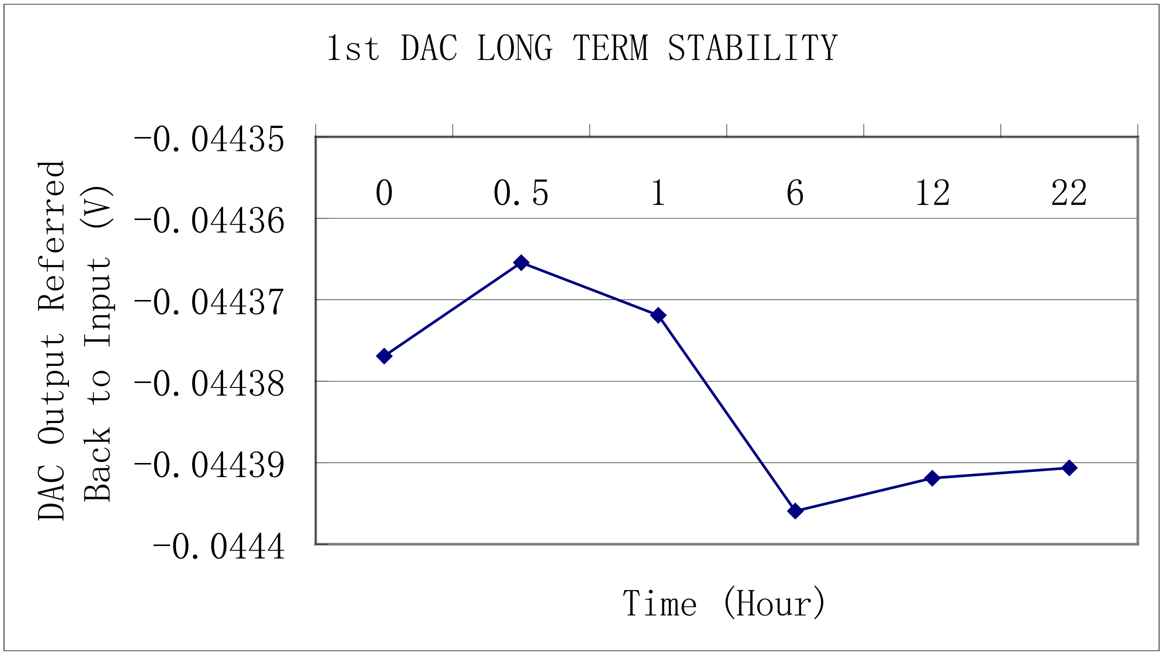


Figure 20 First DAC long term stability

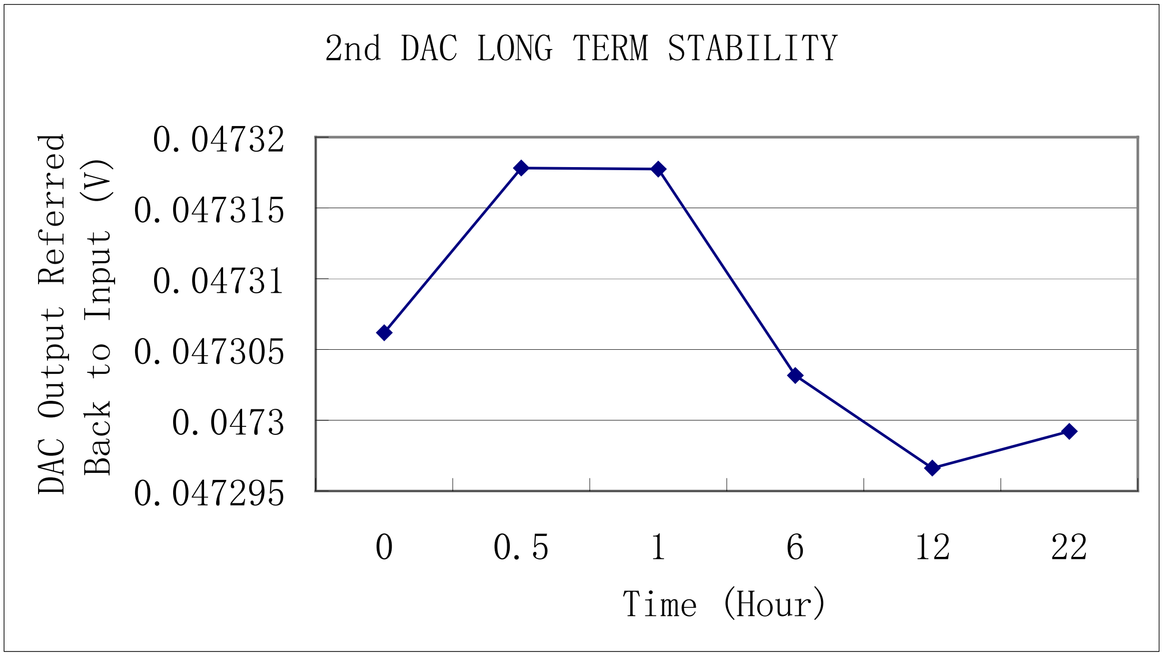


Figure 21 Second DAC long term stability

2.5.7 Amplifier channel noise

Noise at different DC offsets of the input signal were measured. Figure 22 shows a typical time domain waveform. Most of noise is within 8 μV. Table 4 shows the RMS noise value integrated from 1 Hz to 7 KHz at different DC offsets of the input signal. These noise measurements are higher than simulated results for two reasons. First, because the foundry noise model overestimates the noise significantly at low frequencies and low gate-source voltages compared to measured noise data, the flicker noise coefficient KF in SPICE was modified for simulations. Although it is a better estimate of KF, it is not that accurate and causes the measurement values to disagree somewhat with the simulation results. Second, because of finite PSRR, especially at higher frequencies, power supply noise is injected into the signal path and increases the noise.

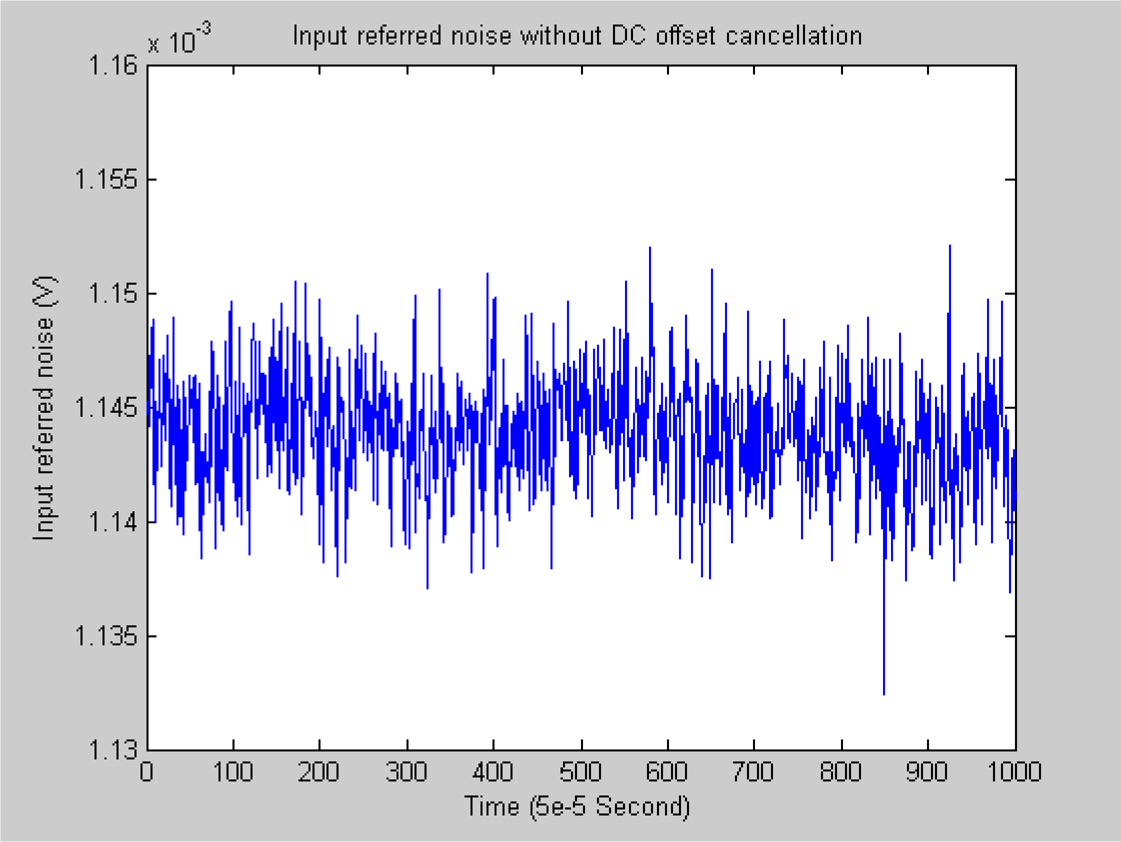


Figure 22 Input referred noise without DC offset cancellation

Table 4 DC offset vs. measurement noise (Channel gain is 250)

INPUT SIGNAL DC OFFSET (V)	0.3	0	-0.3
INPUT REFERED NOISE (μV)	4.57	2.50	5.31
Simulated INPUT REFERED NOISE (μV)	2.7	2.2	3.76

2.5.8 Power supply rejection ratio

Power supply rejection ratios were measured. Figure 23 and figure 24 show the measurement results. The trend of the PSRR measurement results matches simulation results. PSRR+ is lower than the simulation value by 4 db to 10 db for every frequency measured at different DC offsets. PSRR- is lower than the simulation value by 1 db to 6 db for every frequency measured at different DC offsets. Because of the process variation, the op-amps performance is different from the simulation. The compensation capacitor and resistor are the most important factors for PSRR. Resistor normally has big variation. This probably causes the measurement PSRR is lower than the simulation results. The measurement errors can reduce the PSRR also.

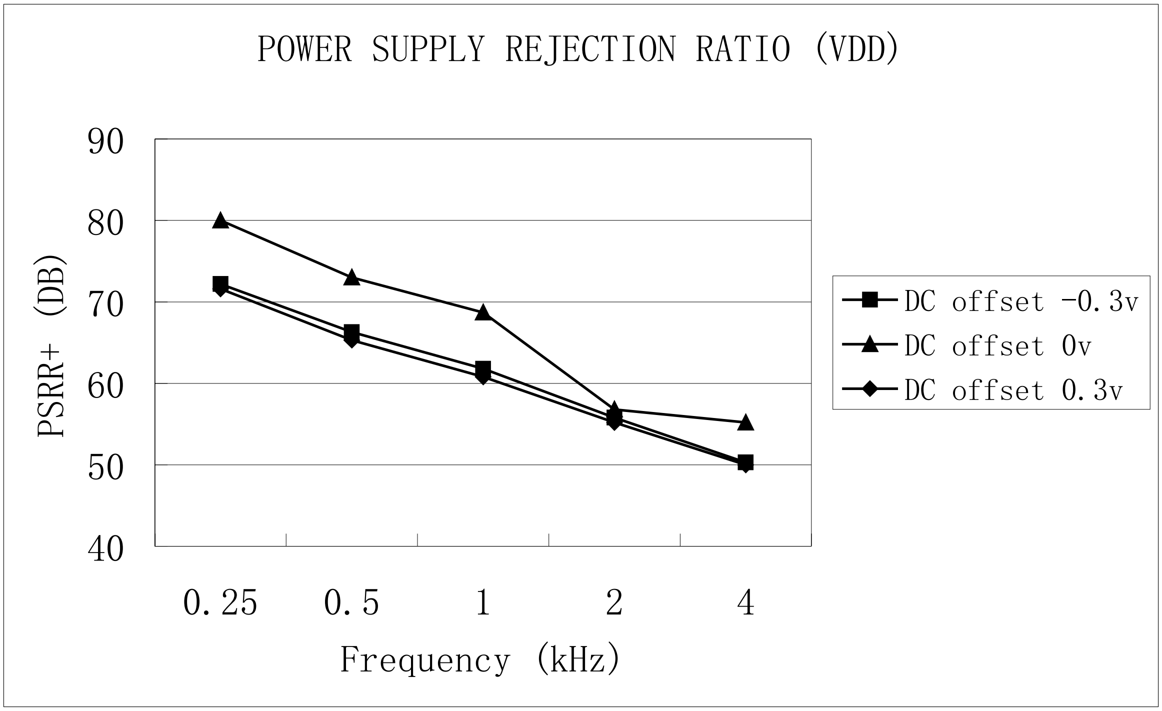


Figure 23 PSRR+ vs. frequency and DC offset of input

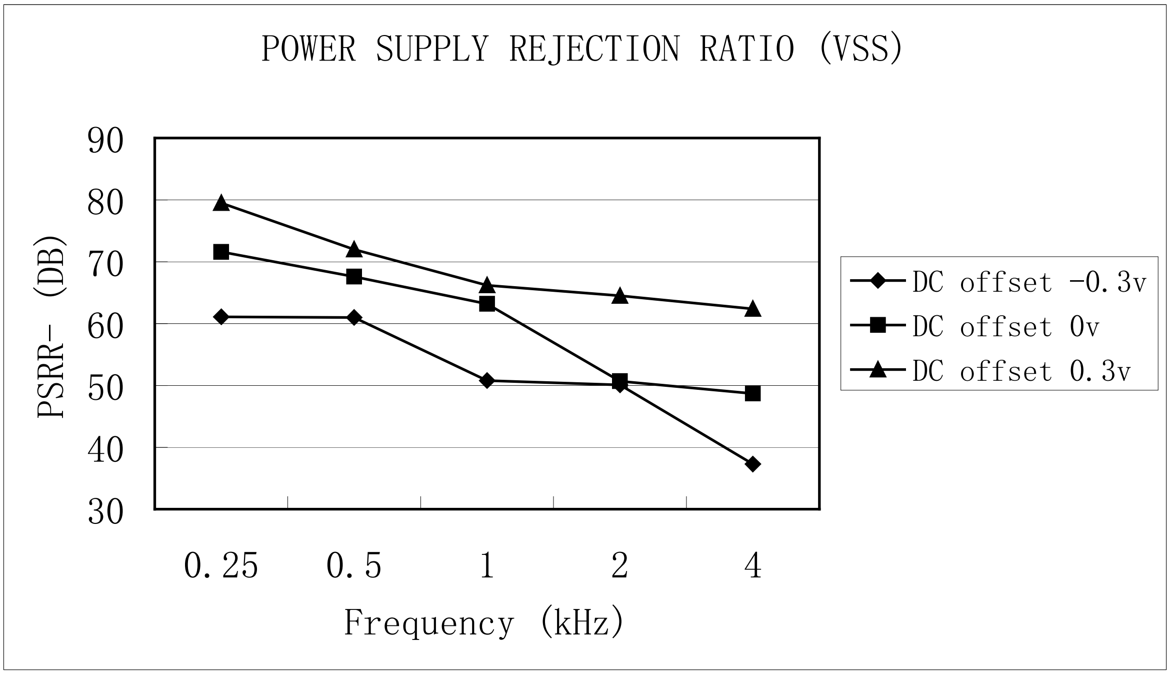


Figure 24 PSRR- vs. frequency and DC offset of input

2.5.9 SFDR

The SFDR is measured with a sine wave source, which has 88 db SFDR. The source is created by a sine wave generator, and then filtered by a notch filter. The SFDR sine wave source is showed in figure 25. The source drives a resistor divider to create a 1.5 mV amplitude sine wave, which also has an SFDR of 88db. This 1.5 mV amplitude 595 Hz sine wave is connected to the input of one channel of the IC. The output of the amplifier channel is measured with the ADC circuit board. Figure 26 shows that the amplifier channel has an SFDR of 85db. This matches the simulation result. Figure 27 shows the amplifier channel SFDR vs. frequency and DC offset. The SFDR trend is matched with the simulation result. The SFDR values are within 1 dB to 3 compared to the simulated results at different DC offsets of the input.

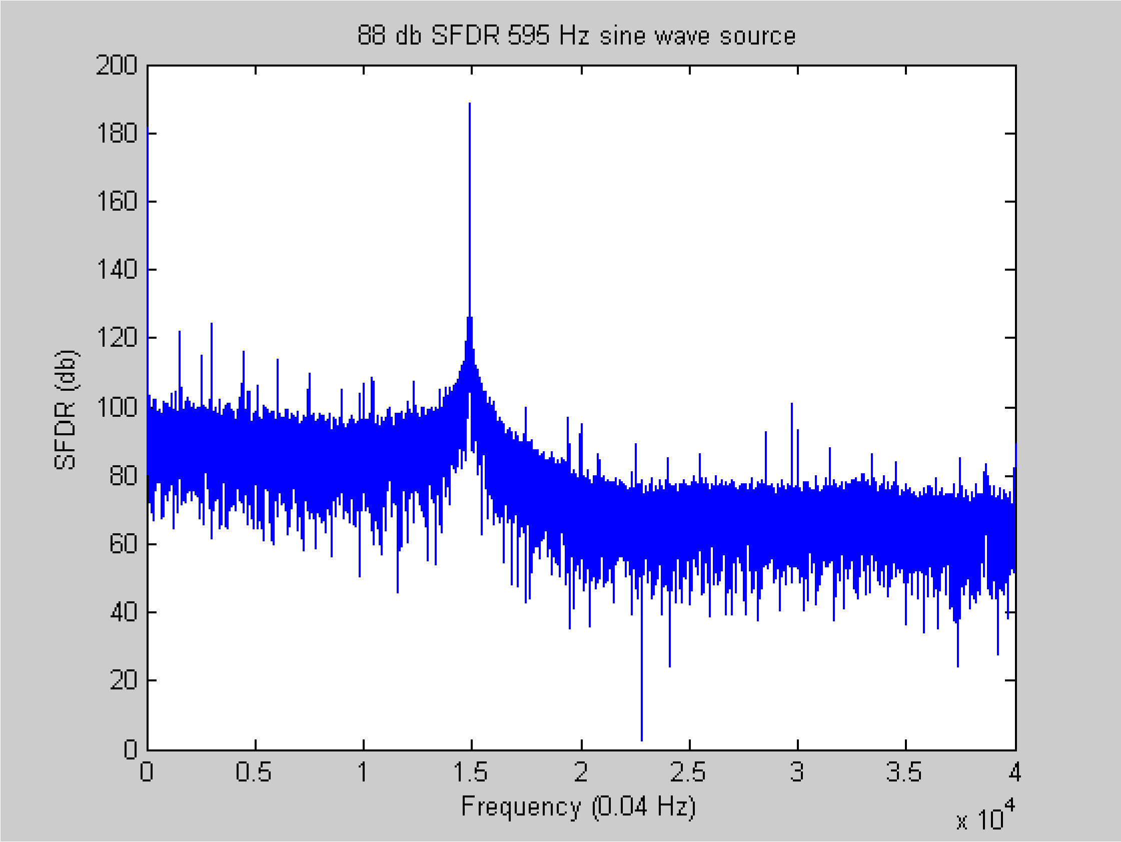


Figure 25 88 db SFDR 595 Hz sine wave source

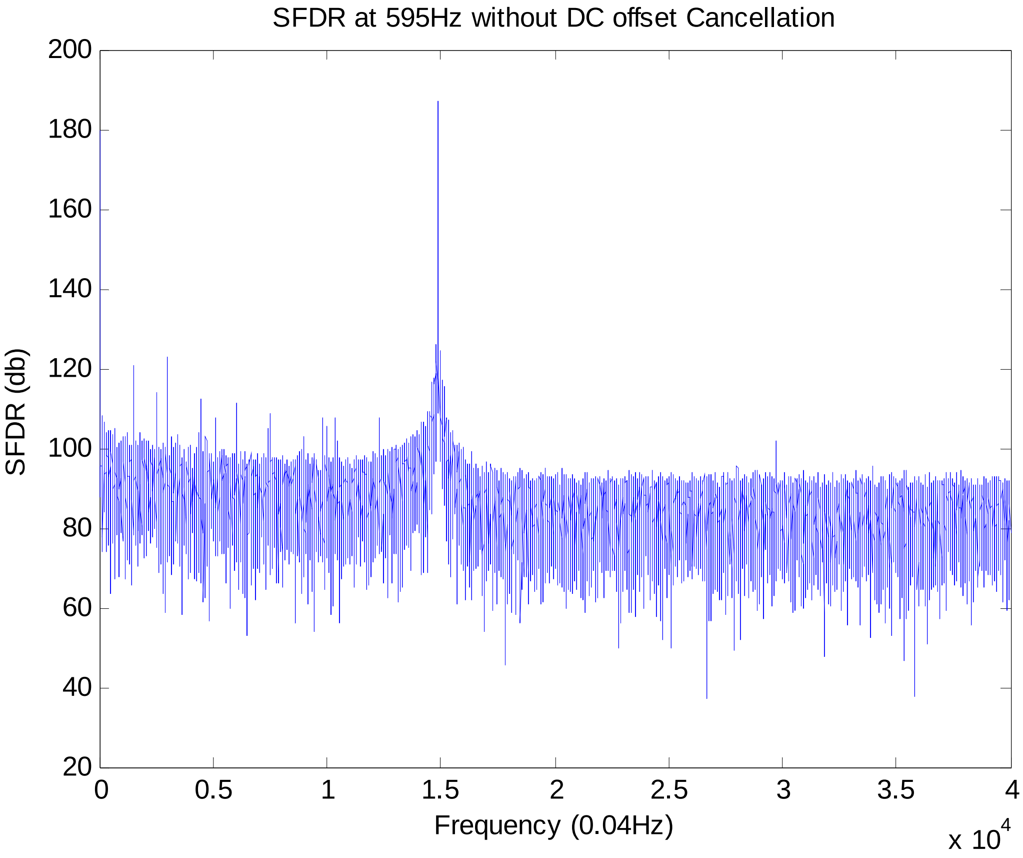


Figure 26 Channel output SFDR without DC offset at sine wave source of 595 Hz and 88 db SFDR

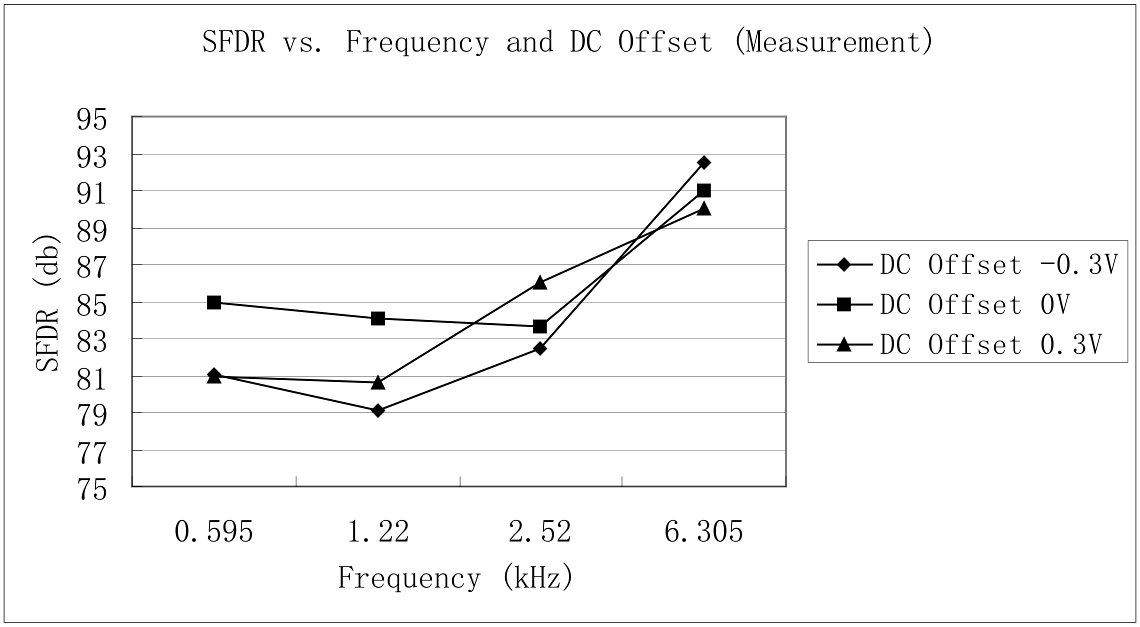


Figure 27 SFDR vs. frequency and input signal DC offset.

2.5.10 Amplifier channel output with 2.5 mV amplitude 500 Hz sine wave input

Figure 28 shows the amplifier channel output waveform, which is the top trace. The input signal (lower trace) is a sine wave with 4 mV amplitude and 500 Hz frequency. The input sine wave is noisey and the sine wave generator produces many spikes. The track and hold circuit is set to always be in track mode and the multiplexer is set to dwell on channel 0 only, so the output waveform is continuous. The sine wave input is amplified by a factor of 250 and the high-frequency noise and spikes are filtered out.

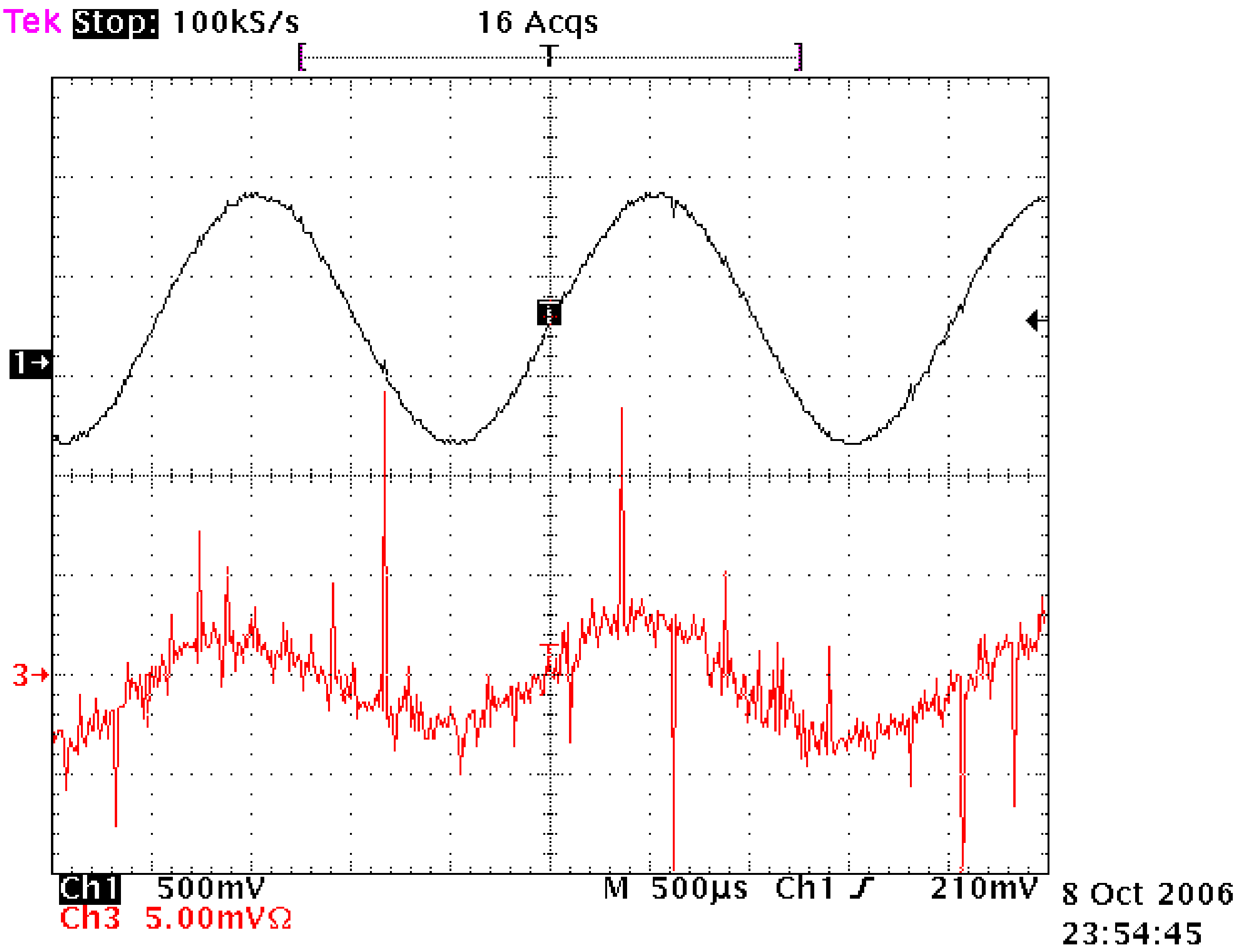


Figure 28 Amplifier channel output with 4 mV amplitude 500 Hz sine wave

2.5.11 Cross talk between channels

There is cross talk between channels due to the layout of ground connections. The resistance of the ground connections between channels is too low compared to the resistance between the ground connections of each channel and the external ground connection. This allows the signal to transfer from one channel to the other through the ground connections. The cross talk is 1:500 at 1 KHz with the maximum gain 250. If the input signal on one channel is 4 mV peak to peak, the output of an adjacent channel, with its input grounded, is 2 mV. Using wider ground metal connections and separating the grounds of each channel and routing them separately to the ground pad can correct this problem.

2.6 Conclusion

We have designed an 8-channel CMOS amplifier channel IC to provide superior performance and functionality for recording electrophysiological data. It meets nearly all of the design requirements. The performance of the amplifier channel IC is summarized in the Table 5.

Table 5 Performance summary of the amplifier channel IC

	Simulation	Measurement
CMOS Process	TSMC 0.25μ	TSMC 0.25μ
Die area	19 mm^2	19 mm^2
Power supply	3 V	3 V
Power dissipation	1.1 mW per channel	1.2 mW per channel
DC offset cancellation range	+0.3 V ~ -0.3 V DC input	+0.3 V ~ -0.3 V DC input
DACs INL/DNL	0 LSB	Within 0.04 LSB
Output swing	1.6 Vpp	1.6 Vpp
Programmable gain	12~250	11.986~249.564
DAC long term stability	N/A	30 μV variation within 22

		hours
Input referred noise at DC offset 0.3 V	2.7 µV	4.57 µV
Input referred noise at DC offset -0.3 V	3.76 µV	5.31 µV
Input referred noise at DC offset 0 V	2.2 µV	2.5 µV
PSRR+	Minimal 64.5 db at 1k Hz	Minimal 61 db at 1k Hz
PSRR-	Minimal 51 db at 1k Hz	Minimal 51 db at 1k Hz
SFDR	Minimal 83.5 db at signal bandwidth (1 Hz~7k Hz)	Minimal 79 db at signal bandwidth (1 Hz~7k Hz)

The amplifier channel IC has cross talk problem, which is 1:500 at 1 KHz with maximum gain 250. This problem is corrected in the 16-channel CMOS amplifier channel IC layout. The modified amplifier channel IC is ready for fabrication.

CHAPTER THREE

3. 16-BIT 500 KSPS ANALOG DIGITAL CONVERTER

3.1 ADC architecture

The 16-bit 500 kSps ADC converts the analog signal from the output of amplifier channel to the digital signal before being sent to a computer. A block diagram of the ADC is shown in figure 29. Two sets of capacitor arrays are used to reduce power by interleaving the sampling/charging phase and A/D conversion phase. Op-amp buffers charge one capacitor array to the input voltage while comparators are in the bit cycling mode on the other capacitor array. The time for the comparators to cycle through the bits and for the input buffer to charge the arrays has doubled to a full cycle time of 2 μs and reduces power dissipation by about half and without degrading ADC 16-bit resolution, 15-bit accuracy and conversion rates. The drawback to interleaving is that differences in the two-capacitor arrays can cause harmonic noise. However, there are 16-channels of data from the sensor IC so that one capacitor array can be used exclusively for even channels and the other array for odd channels to avoid this problem. The power dissipation of the op-amp buffers and the comparators are 2.3 mW and 3.6 mW respectively.

A fully differential architecture [11] was used for each capacitor array, which can eliminate common-mode noise as well as errors due to linear capacitor voltage dependence and handle peak-to-peak signals up to twice the power supply voltage with only a 3 dB increase in the noise floor. Each of the capacitor arrays is formed with binary-weighted MIM capacitors. To meet the minimal noise requirement, the total capacitor of the capacitor array is 164 pF, having noise of 0.2 LSB and power dissipation of 0.2 mW for charging and discharging. Because of process and geometry variations, matching of capacitors does not naturally show 16-bit precision. Capacitor calibration [10] [12] is used to overcome this limitation.

The calibration circuits, which include comparator offset calibration, capacitor ratio error calibration, common mode input signal cancellation and ADC gain error calibration, are formed using finite state machines. A main logic control circuit connects these calibration circuits to capacitor arrays to do the calibrations in order. It finally stores the calibration results to registers and releases the capacitor arrays to the analog-to-digital conversion logic control circuit for A/D conversion. The analog-to-digital conversion logic control circuit, with power dissipation less than 0.1mW, was implemented with a finite state machine to control the capacitor switches and store the digital outputs.

Fine and coarse comparators are used to mitigate voltage stress effects on the gate of the FETs to reduce hysteresis due to traps in the gate oxide. Another benefit of using two comparators is to make the interleaving of the charging phase and A/D conversion phase possible.

During sampling mode, the lower capacitor array top plates are tied to the mid point of the power supplies (GND), while the upper capacitor top plates are tied to the analog input, and both of the capacitor array bottom plates are tied to the average voltage of the analog input and GND. After the voltage settles, the bottom plates are disconnected from the voltage source and the charge remains on the bottom plates during the remainder of the conversion. During bit cycling the bits are determined successively in order from

MSB to LSB by switching the binary-weighted capacitor top plates of the array to the reference voltages REF+ and REF-. The comparator determines which input is larger and the logic determines whether the switches remain set or not.

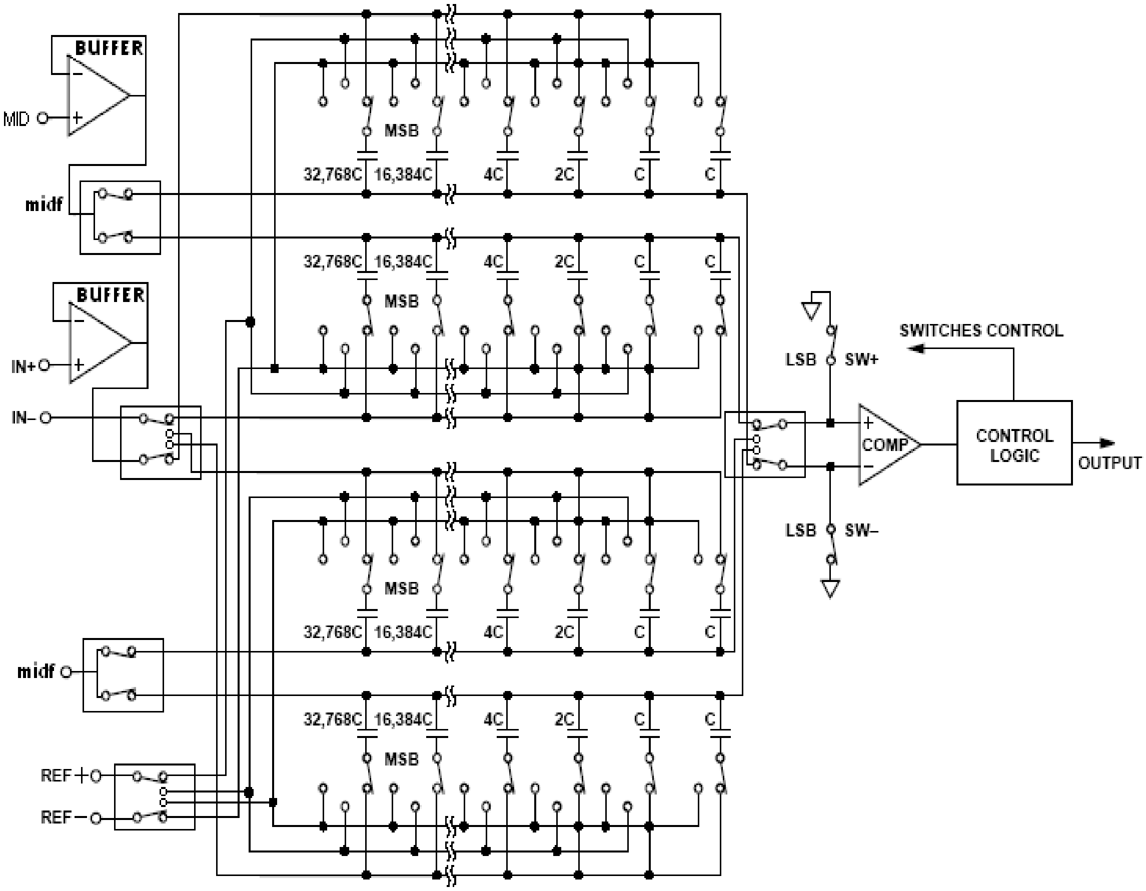


Figure 29 Analog digital converter

3.2 Capacitor correction and calibration

3.2.1 Capacitor ratio error calibration

The largest ADC conversion error is due to the mismatch of capacitors in the arrays. A self-calibration algorithm [13] is applied to remove this ratio error in the binary-weighted capacitor arrays. The basic idea is to measure the differential non-linearity in the ADC transfer function and trim the 10 largest capacitors in the capacitor array to remove the ratio errors. A block diagram of the trimming circuits are shown in figure 30. Multiple small binary-weighted capacitor arrays are used for trimming. Using this approach can bypass the use of an ALU for error code calculation and the requirement for a high-linearity binary-weighted capacitor array. The calibration procedure progresses in the order from LSB to MSB and has a drawback from geometric propagation of residual errors. This can be overcome by using larger value binary-weighted capacitor arrays to extend the trim range and obtain the desired calibration capacitor value.

To determine the capacitor ratio error precisely during self-calibration, a comparator with high resolution and low noise is required. The gain of our fine comparator is high enough for the self-calibration, but the noise is too large. This problem is overcome using two methods. First, a low gain low signal bandwidth fully differential op-amp and a two stage op-amp are added after the 2nd low gain fully differential op-amp of the fine comparator to form the calibration comparator, which is shown in Figure 35. The low gain low signal bandwidth fully differential op-amp and two stage op-amps are only enabled during calibration operation. Those op-amps supply additional gain and act as well as a low pass filter, which filters out the noise at high frequency. This reduces the total noise to 1/3 of its original value. Second, by repeating each measurement of capacitor ratio error 128 times and averaging the results, the approximate amount of improvement in signal-to-noise ratio (SNR) is proportional to the square root of the number of measurements or about 21dB.

The capacitance of the MIM capacitors varies with voltage. The linear coefficient has no effect due to the fully differential architecture of the ADC. Errors due to the quadratic voltage coefficient are less than half LSB so there is no need to correct for it.

The process to calibrate the capacitor ratio error is that the low 6 bit capacitors value are assumed to be correct and the 7th bit capacitor is compared to the summation of the lower 6 bit capacitors and an LSB dummy capacitor. The 7th bit capacitor is either added or subtracted for capacitor ratio error correction depending on the comparison result. To simplify the capacitor ratio error calibration, the 7th bit capacitor is designed smaller than it ideal so only addition is needed for capacitor ratio error correction. The rest of high bit capacitors are calibrated similarly, until the MSB capacitor is reached.

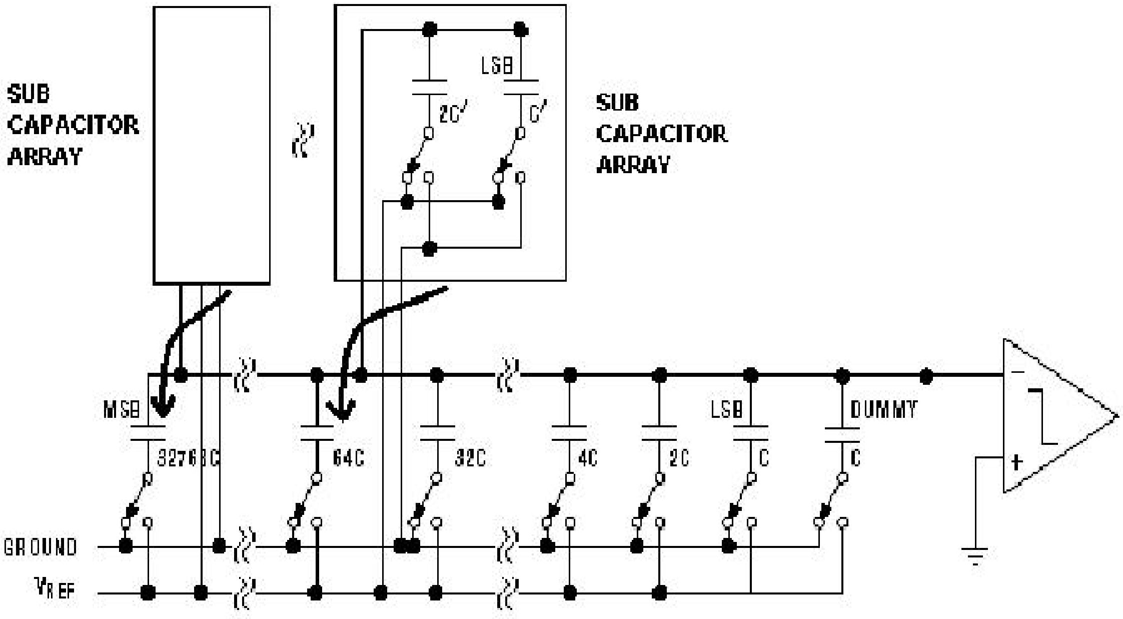


Figure 30 Capacitor ratio error calibration circuit

3.2.2 Common mode input signal cancellation

The common-mode (CM) signal applied to the input comparator can cause the comparator offset to vary and the ADC linearity will then be dependent on the CM signal. The fully differential architecture and input sampling scheme have infinite common mode rejection ratio (CMRR) in theory. Because of parasitic capacitance on the bottom plates of the capacitors, the common mode input signal remains on the parasitic capacitor after the sampling and limits the CMRR to about 40 dB. A comparator with high CMRR can solve this problem but device matching limits the CMRR to seldom exceeding 50 dB. Figure 31 shows the circuit [12], which was used to correct for errors because of CM signals. V_{com} is the average of the analog input voltage and GND. The idea behind the circuit is to use C_{cm} to sample the CM signal and then reverse the charge stored on it to cancel the CM signal charge, which is stored on the parasitic capacitance C_p. If C_{cm} is made equal to C_p, the CM signal can be corrected to zero. A self-calibration scheme is used to trim the binary-weighted capacitor C_{cm} value to C_p.

The process to trim the binary-weighted capacitor C_{cm} to the value of parasitic capacitor has several steps. First, both plates of the upper capacitor array are connected to the voltage reference and both plates of the lower capacitor array are connected to ground to do the sampling. Second, the bottom plates are disconnected from the voltage reference and ground and the upper capacitor array top plate is switched to ground. Third, the comparator figures out which bottom plate has the higher voltage and the comparison result is stored. Fourth, the MSB binary-weighted capacitor C_{cm} is connected to one of lower capacitor array bottom plates depending on the previous comparison result and the previous three steps are repeated. These two comparison results determine whether the

MSB binary-weighted capacitor Ccm should be kept or discarded. This process repeats until reaching the LSB binary weighted capacitor Ccm. Then the calibration is done. Because the parasitic capacitor on both bottom plates are roughly same, both binary-weighted capacitor Ccms are set to the same value. During A/D conversion, Ccm and Ccm' are applied to cancel the CM signal on the capacitor array bottom plates.

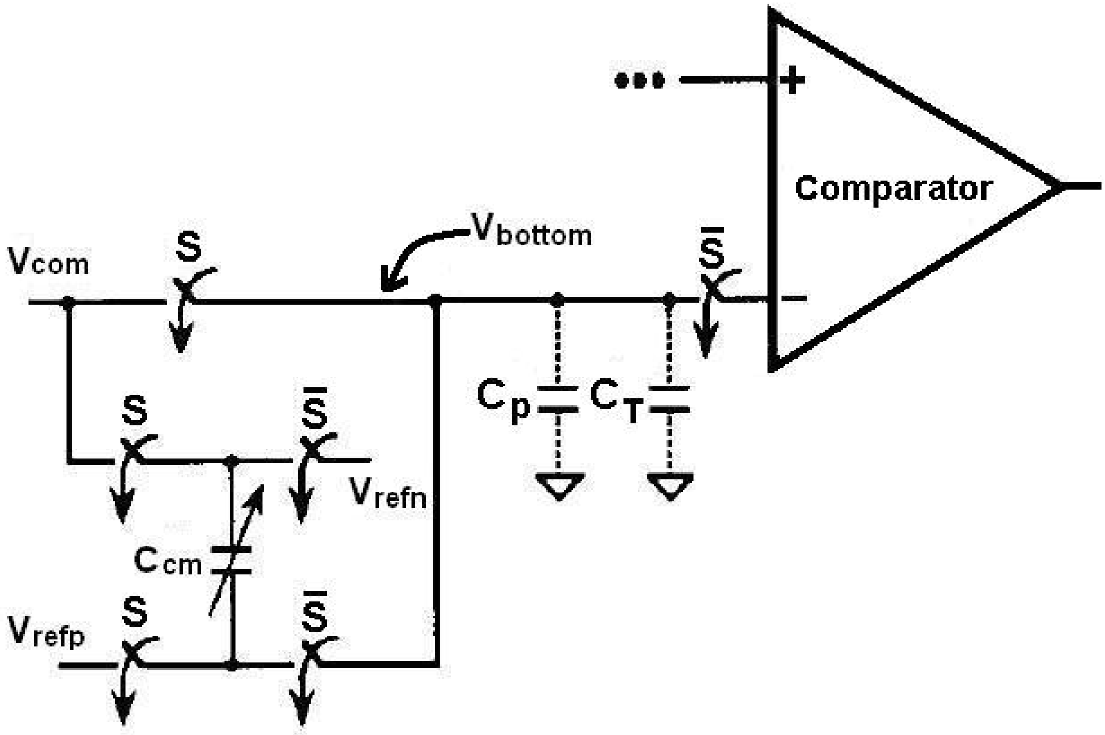


Figure 31 CM signal calibration circuit

3.2.3 Comparator offset calibration

The comparator offset can cause analog-to-digital conversion error if it is not cancelled. The major comparator offset source is from charge injection from the switches, which are used for connecting the capacitor arrays to the comparator and the comparator's offset cancellation, because of mismatch. Figure 32 shows how the calibration works. Two binary weighted capacitor arrays are connected to the bottom plates of capacitor arrays. Initially all charge on capacitors is discharged. Thereafter switches are opened. If the top switch injects more positive charge than the bottom switch, in other words, the comparator offset is positive, the comparator output will be high. The top binary weighted capacitor C$_{offset}$ is switched to V$_{refp}$ to cancel the additional charge. If comparator offset is negative, the binary weighted capacitor C$_{offset'}$ is switched to V$_{refp}$ for charge cancellation. The comparator's offset voltages are determined by the following equation. C$_{offset}$ or C$_{offset'}$ is applied during A/D conversion to cancel the comparator offset.

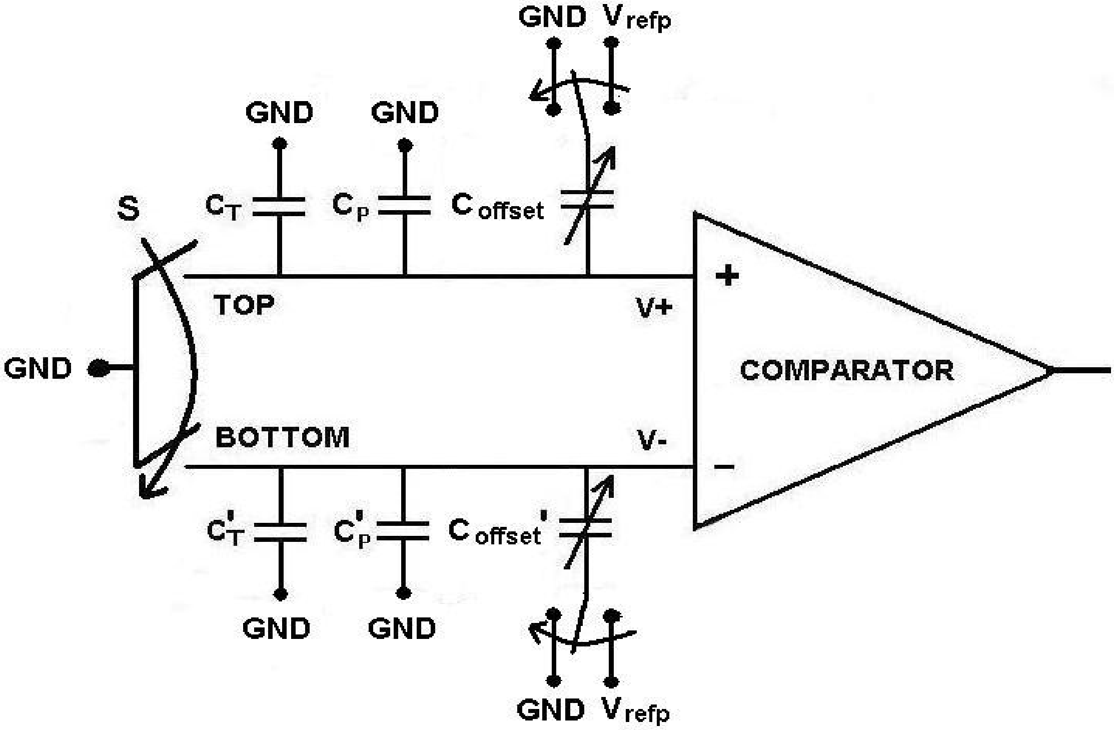


Figure 32 Comparator offset calibration circuit

$$V_{offset} = \left(\frac{V_{ref+} C_{offset}}{C_T + C_p + C_{offset}} \right) \tag{1}$$

$$V_{offset}' = \left(\frac{V_{ref+} C_{offset}'}{C_T' + C_p' + C_{offset}'} \right) \tag{2}$$

3.2.4 ADC gain error calibration

With the above capacitor ratio error calibration, the total capacitance of the upper capacitor array and lower capacitor array normally is not equal. This difference can cause a gain error in the ADC. Figure 33 shows the voltage of the capacitor array bottom plates during the analog-to-digital conversion. The upper bottom-plate voltage V+ and lower bottom-plate voltage V- are given in the equations 7 and 8.

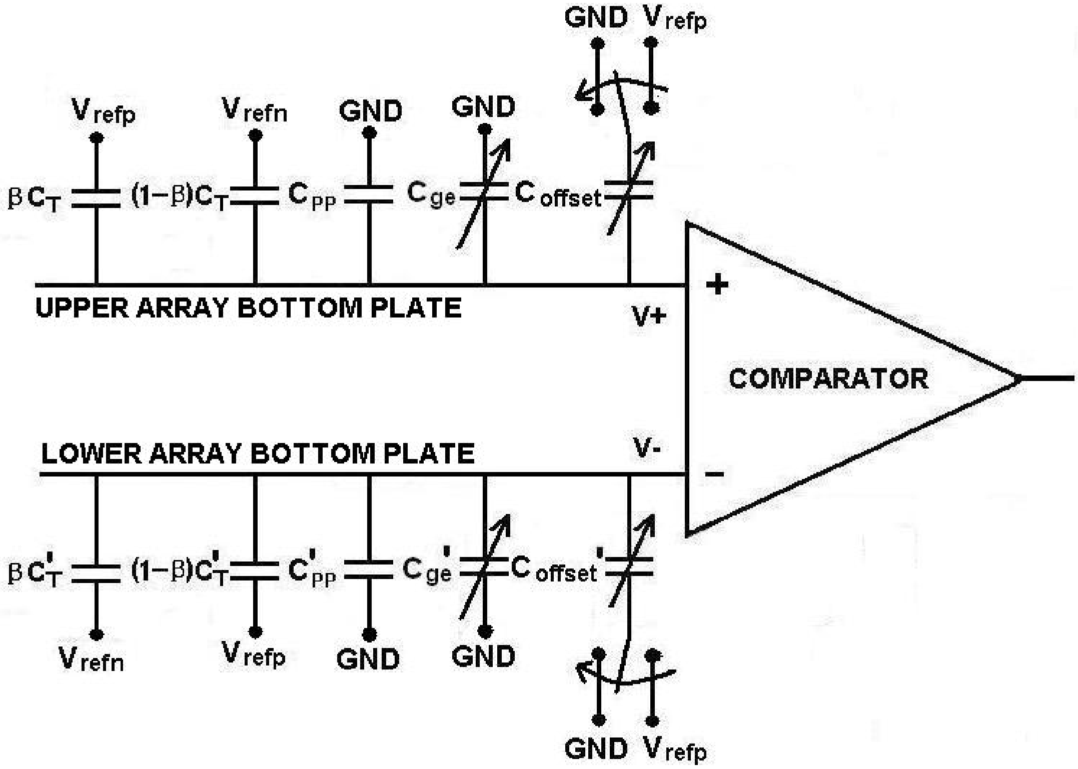


Figure 33 ADC gain error calibration circuit

$$C_{pp} = C_p - C_{cm} \tag{3}$$

$$C_{pp}{}' = C_p{}' - C_{cm}{}' \tag{4}$$

$$\alpha = \frac{C_T}{C_T + C_{pp} + C_{ge} + C_{offset}} \tag{5}$$

$$\alpha^{'} = \frac{C_T^{'}}{C_T^{'} + C_{pp}^{'} + C_{ge}^{'} + C_{offset}} \tag{6}$$

$$V_+ = \alpha\left(\beta V_{refp} + (1-\beta)V_{refn} - \frac{V_{in}}{2}\right) + \alpha\frac{V_{refp}C_{offset}}{C_T} + \alpha\frac{V_{in}\left(C_{pp} + C_{ge}\right)}{2C_T} \tag{7}$$

$$V_- = \alpha'\left(\beta V_{refn} + (1-\beta)V_{refp} + \frac{V_{in}}{2}\right) + \alpha'\frac{V_{in}\left(C_{pp}' + C_{ge}'\right)}{2C_T'} \tag{8}$$

C_T and C_T’ are the sum of all capacitances in the upper and lower capacitor arrays respectively. The gain error is caused by the third factor in V_+ and second factor in V-. If the parasitic capacitance C_{pp} and C_{pp}’ on the bottom plates of the capacitor arrays are zero, then the gain error is also zero. But a finite parasitic capacitance causes the voltages V+ and V- to depend on array capacitance values and results in a gain error A_{GE} approximately equal to $\alpha\frac{V_{in}\left(C_{pp} + C_{ge}\right)}{2C_T} - \alpha'\frac{V_{in}\left(C_{pp}' + C_{ge}'\right)}{2C_T'}$. However, if the first factor and the second factor in the equation can be made equal, the gain error can be eliminated. There are two approaches to calibrate the gain error, either by adjusting the upper or lower capacitor array values C_T and C_T’ or by adjusting the upper or lower binary weighted capacitor array C_{ge} or C_{ge}’ to make them equal. The latter method was selected to implement the calibration because of its simple implementation. The algorithm works as follows: First both upper and lower capacitor array plates are connected to the voltage reference and then the top plates are connected to ground and bottom plates are disconnected from the voltage reference. The reference voltage is stored on the bottom plate’s parasitic capacitance. The charge evenly distributes on the bottom plates. The comparator compares these two voltages on the bottom plates and the result is stored. Second, either binary weighted capacitor array C_{ge} or C_{ge}’ MSB capacitor is connected to its bottom plate depending on the previous result. The MSB capacitor is kept if the comparator gives the same result as the previous one. Otherwise it is discarded. This process is repeated till the LSB capacitor. After calibration, the capacitors C_{ge} or C_{ge}’ is applied during A/D conversions to eliminate the gain error.

3.3 Comparator

3.3.1 Hysteresis

16-bit ADC accuracy will be affected if the comparator hysteresis is not well controlled. A dual-comparator approach [12] is used to mitigate the hysteresis. The first comparator or coarse comparator has a low gain fully differential op-amp with a p-channel MOS differential pair followed by a latched comparator. The coarse comparator is used for the highest 8 MSB conversions, when the voltages at the inputs to the comparators are large. P-channel FETs have less hysteresis [14] when the gate is stressed with large voltages than n-channel FETs. Another purpose of the low gain op-amp is to isolate the latch comparator switching noise from the capacitor arrays and cancel the comparator offset. The second or fine comparator is used for converting the last 8-bits, when voltages are small and therefore hysteresis is small. Hysteresis in the coarse comparator may still cause an error at its 8th bit comparison due to previous large voltage stresses. The fine comparator, which never experiences large voltage stress, is used to re-compare the 8th bit to correct any coarse comparator errors. Below is a table, which shows how the correction works.

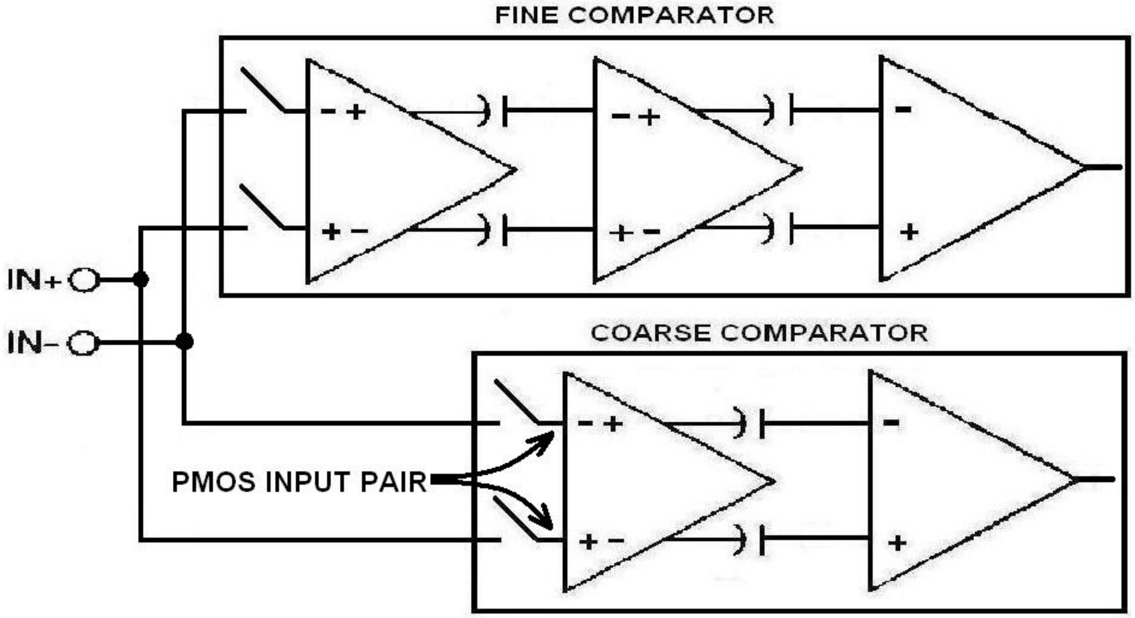


Figure 34 Dual comparator

Table 6 Algorithms for correcting hysteresis error

Fine Comparator Decision for D	**Fine Comparator Decision for D+1**	**Final Result**
0	1	D
0	0	D+1
1	1	D-1
1	0	----

The coarse comparator provides the first 8 bits A/D conversion result D. The fine comparator re-compares the 8 bits A/D conversion result D and D+1[12]. The final result is determined by the above error correction table. This technique not only corrects the hysteresis problems but also corrects the error caused by different offsets between coarse and fine comparators.

The fine comparator is carefully designed with high gain, high speed, and low noise for comparing very small signals.

3.3.2 Fine comparator

The fine comparator is composed of two low-gain fully differential op-amps [11] followed by a latched comparator [15] as well as another low-gain low signal bandwidth fully differential op-amp and a two stage op-amp. The latched comparator and low gain low signal bandwidth op-amp and two stage op-amp share the first two low gain fully differential op-amps. The first one with two low gain op-amps forms the comparator for the A/D conversion. The second one with two low gain op-amps forms the comparator for calibration only. The latched comparator is disabled during calibration. The low gain low signal bandwidth op-amp and two stage op-amp are disabled during A/D conversion. The figure 35 shows the block diagram of the fine comparator. Figure 36 shows the schematic of the low-gain fully differential op-amps.

The first two op-amps have low gain of 30 dB each, they offer high speed and provide sufficient amplification for the latched comparator to resolve the small voltage differences at the input. This approach can distinguish a small signal in a short time with less power than other approaches.

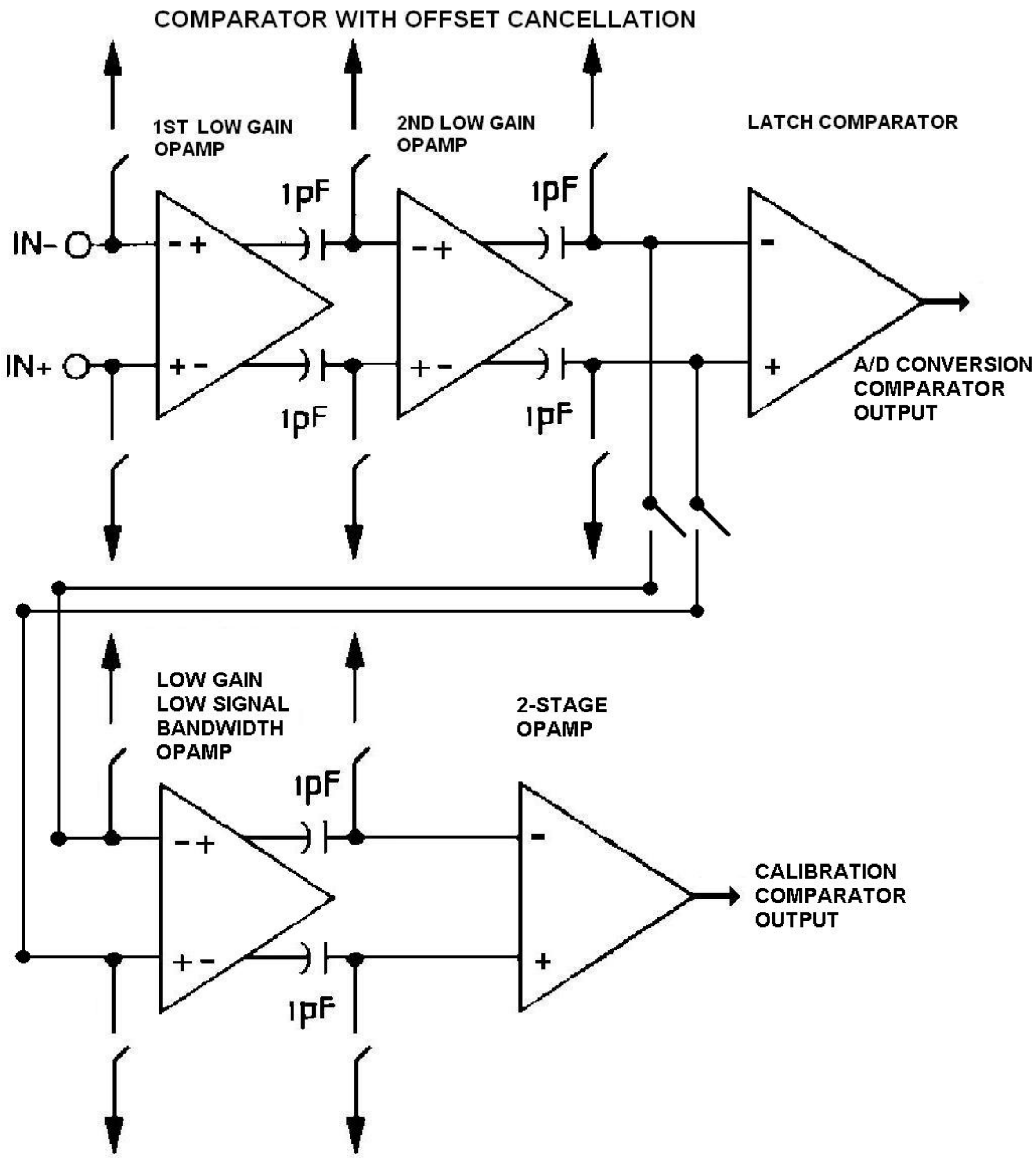

COMPARATOR WITH OFFSET CANCELLATION
1ST LOW GAIN OPAMP
2ND LOW GAIN OPAMP
LATCH COMPARATOR
1pF
1pF
1pF
1pF
IN-
IN+
- +
+ -
- +
+ -
-
+
A/D CONVERSION COMPARATOR OUTPUT
LOW GAIN LOW SIGNAL BANDWIDTH OPAMP
2-STAGE OPAMP
1pF
1pF
- +
+ -
-
+
CALIBRATION COMPARATOR OUTPUT

Figure 35 Schematic of fine comparator

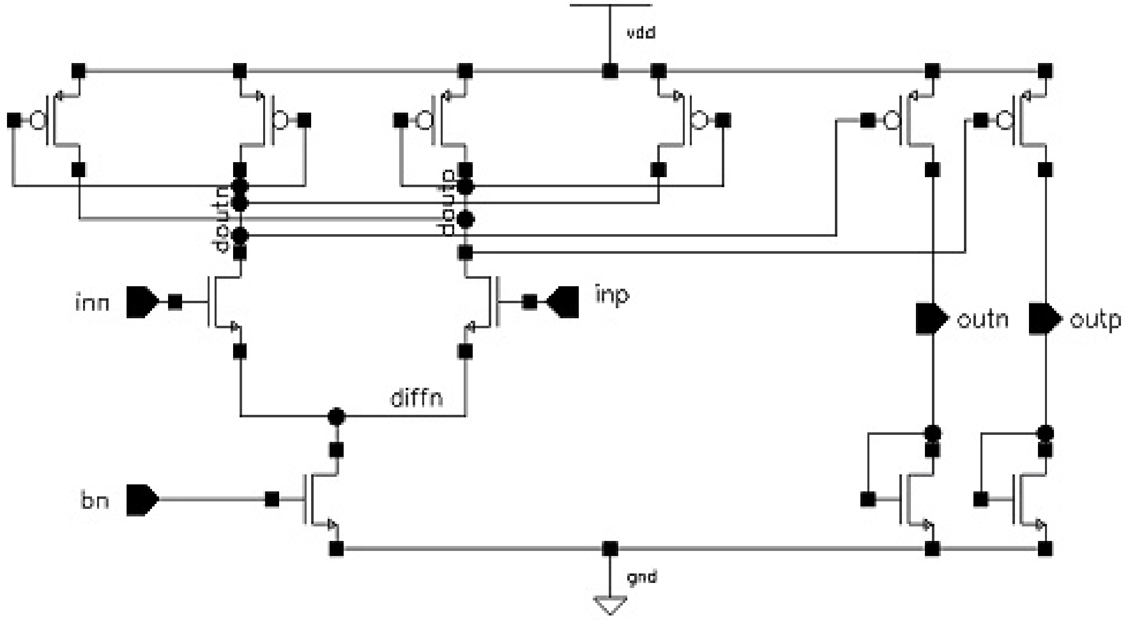


Figure 36 Low gain fully differential op-amp

3.3.3 Comparator offset cancellation

An offset cancellation technique [11] was used to eliminate comparator offset. The first low gain op-amp is designed with high current and large W/L ratio to lower the overall thermal noise. Because of the offset cancellation technique, flicker noise at low frequencies is reduced greatly [16] and is not a major contributor to noise. The noise contributions of the circuits following the first op-amp are small due to the 30 dB gain of the first stage. The latch comparator uses positive feedback to achieve high speed and high gain at low current levels. The overall gain of the comparator is 110 dB, which exceeds 16-bit resolution. It can distinguish a ¼ LSB signal within 100ns with noise of 0.29 LSB.

3.4 Simulation results

3.4.1 Noise

Figure 37 shows the comparator output noise versus frequency. The noise is very small at low frequencies because the offset cancellation technique removes low frequency noise. The integrated output noise referred back to the input is 7 μV or about 0.29 LSB. Together with the capacitor noise 0.2LSB, total noise is 0.41 LSB, which is equivalent to 15 effective bits [17].

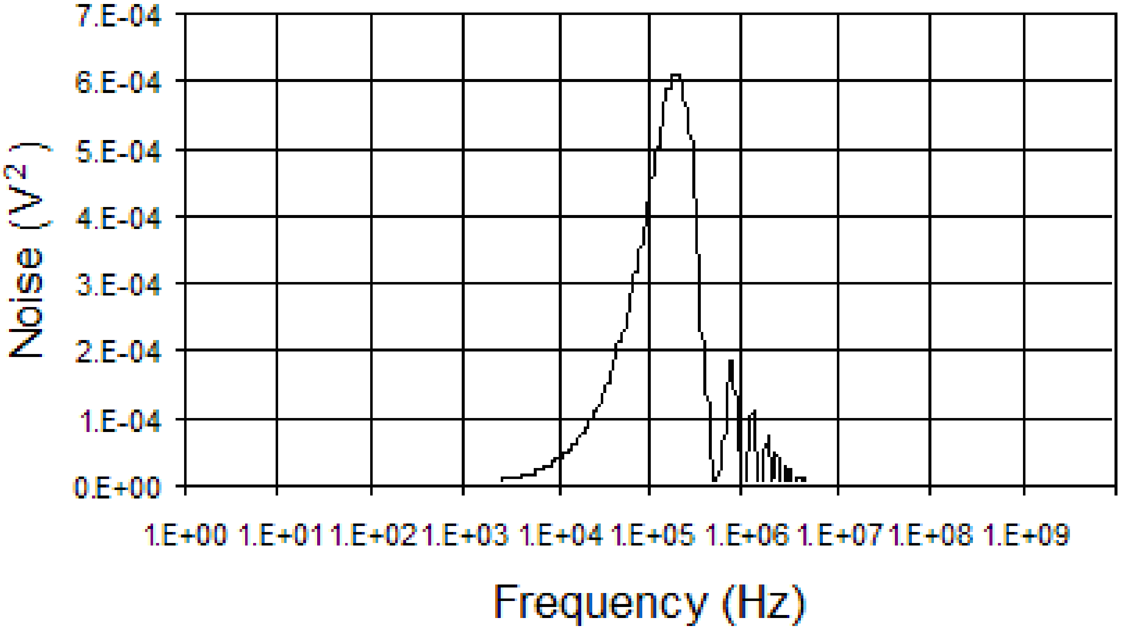


Figure 37 Comparator output noise versus frequency

3.4.2 Calibration simulation

To verify the calibration circuit, several modifications to the switches and capacitors are introduced to the simulation. Comparator reset switches and switches for connecting the capacitor array and comparator are set to 4% mismatch for comparator offset calibration simulation. The 10 MSB capacitors in the capacitor array are set a random value between 0.995 and 0.997 of its ideal value randomly for capacitor ratio error simulation. The bottom plates are connected with 2 parasitic capacitors, which are 1 pF and 1.05 pF respectively for CM input signal cancellation and ADC gain error calibration simulation. Figure 38 shows the first set of capacitor array's upper and lower capacitor array bottom plates voltage waveform during the calibration simulation. The red line is the upper capacitor array bottom plate voltage and the blue line is the lower capacitor array bottom plate voltage for the 4 different calibrations. These 4 calibrations are comparator offset calibration, upper capacitor array capacitor ratio error calibration, lower capacitor array MSB capacitor ratio error calibration, common mode input signal cancellation and ADC gain error calibration. The calibration comparator compares two bottom plate voltages to determine which of for the binary weighted capacitor array's trim capacitors should be kept.

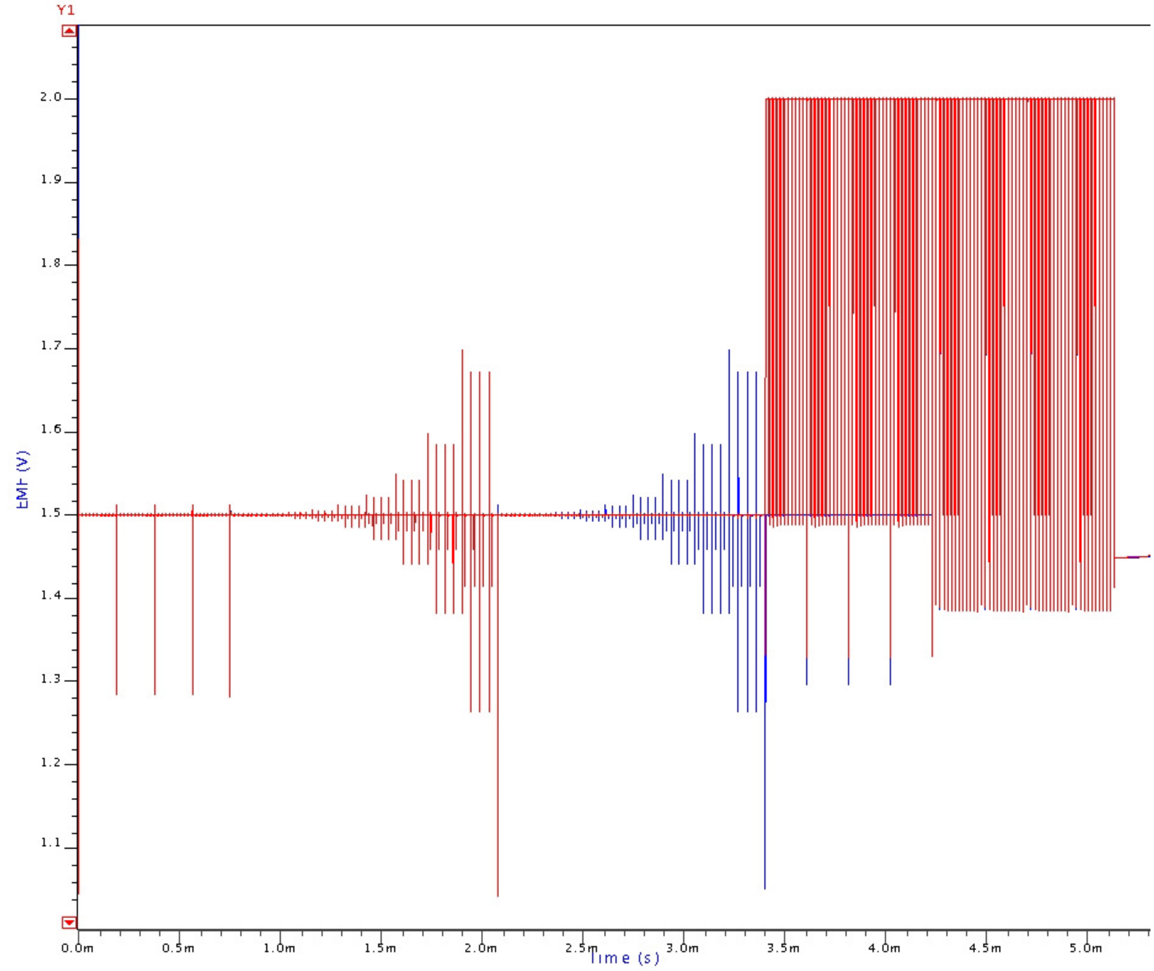


Figure 38 The first set capacitor array calibration process

Figure 39 shows the zoomed-in comparator offset calibration, which shows a 1 LSB capacitor of offset capacitor array being applied for comparator offset cancellation. This result matches the calculated result due to intentional switch mismatches used for comparator offset calibration verification.

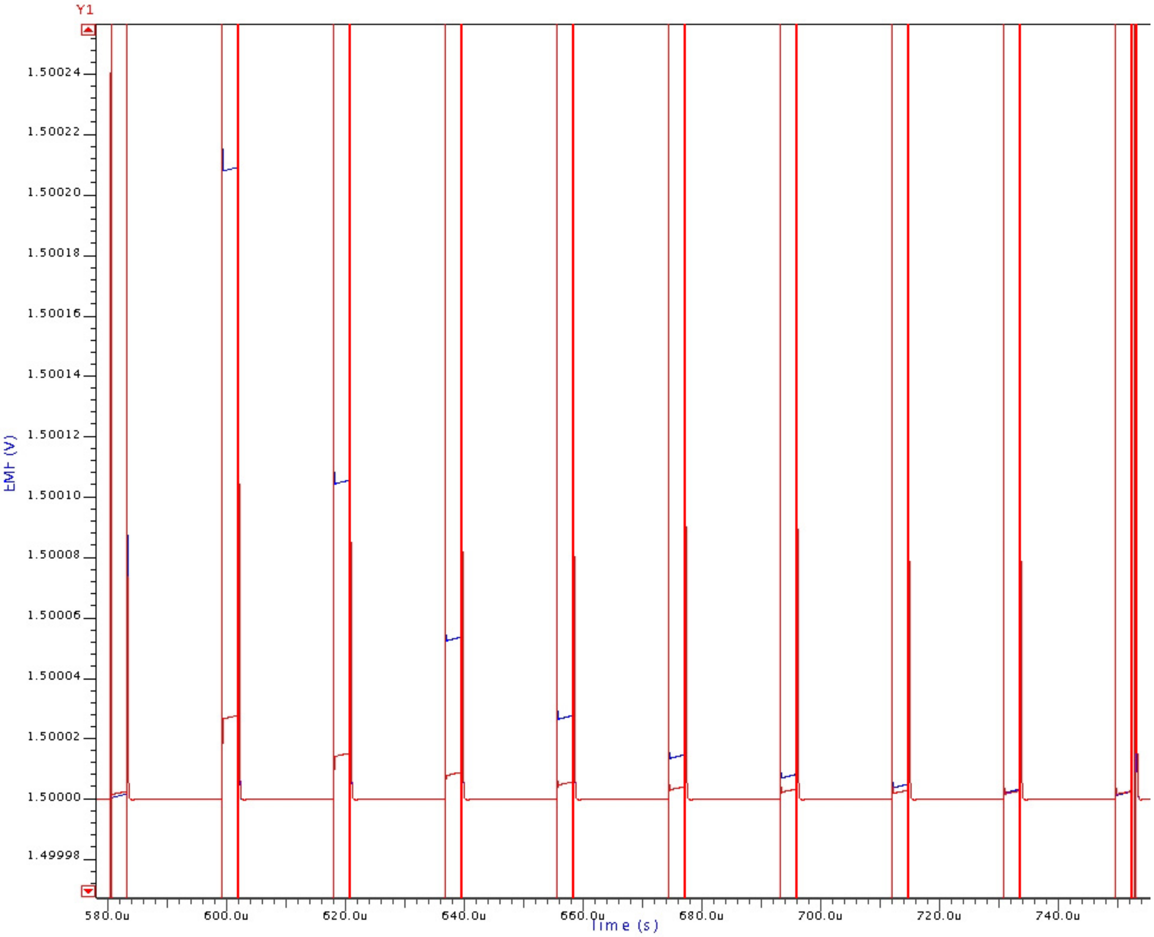


Figure 39 Comparator offset calibration

Figure 40 shows the zoomed-in lower capacitor array MSB capacitor ratio error calibration. The simulation result is 01011011000. This means the 2nd MSB, 4th MSB, 5th MSB 7th MSB and 8th MSB capacitors in capacitor ratio error calibration array for lower array MSB capacitor are applied. The simulation results show that the ratio error between adjacent capacitors is well calibrated within a quarter LSB.

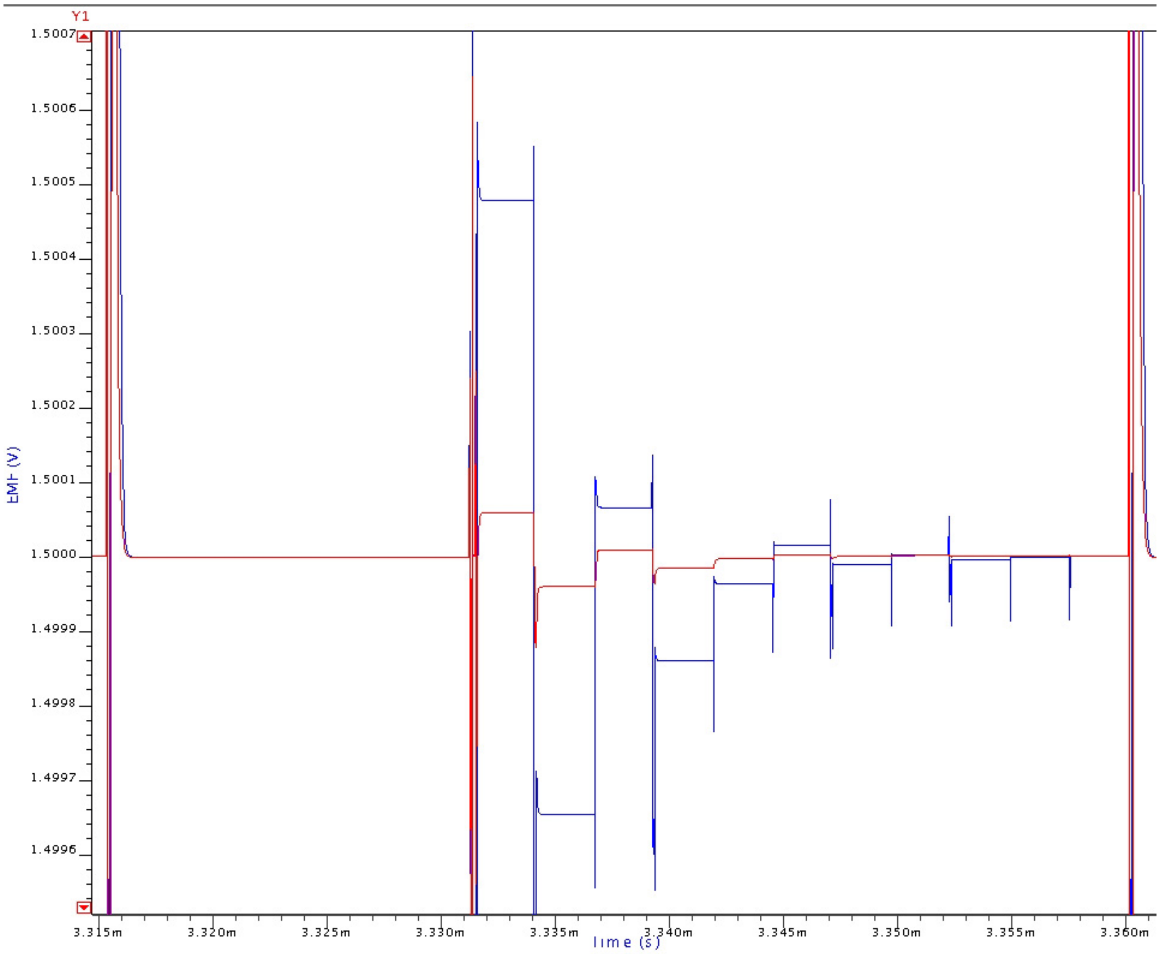


Figure 40 Lower capacitor array MSB capacitor ratio error calibration

Figure 41 shows the zoomed-in common mode input signal cancellation. The simulation result is 100001001100. This means the parasitic capacitor on capacitor array bottom plates is 2.655 pF, which includes 1 pF that we created for common mode input signal cancellation verification and parasitic from switches and metal wires. This matches the calculated result.

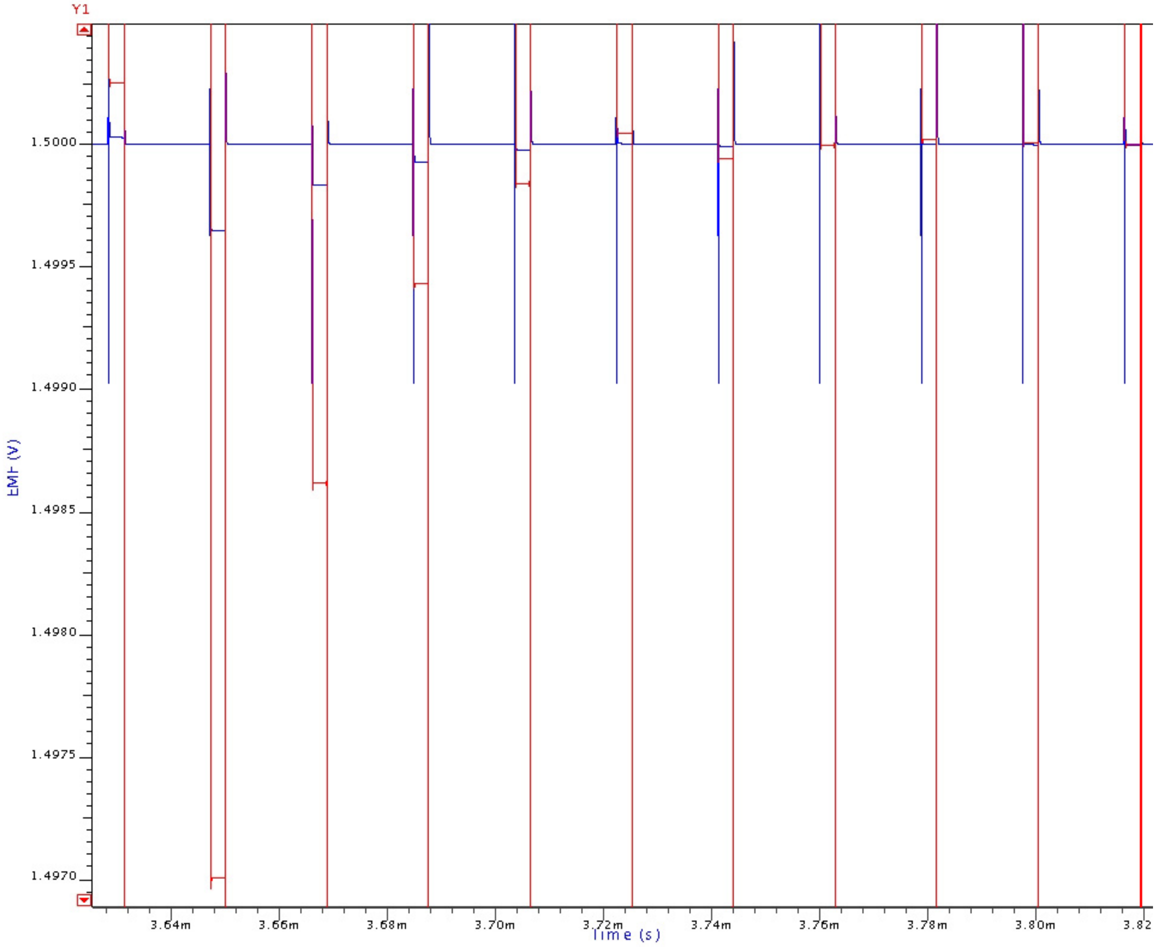


Figure 41 Common mode input signal cancellation

Figure 42 shows the zoomed-in ADC gain error calibration. The simulation result is 100100. This means that a 22.5 fF capacitor is added to correct the ADC gain error. This result also matches calculations.

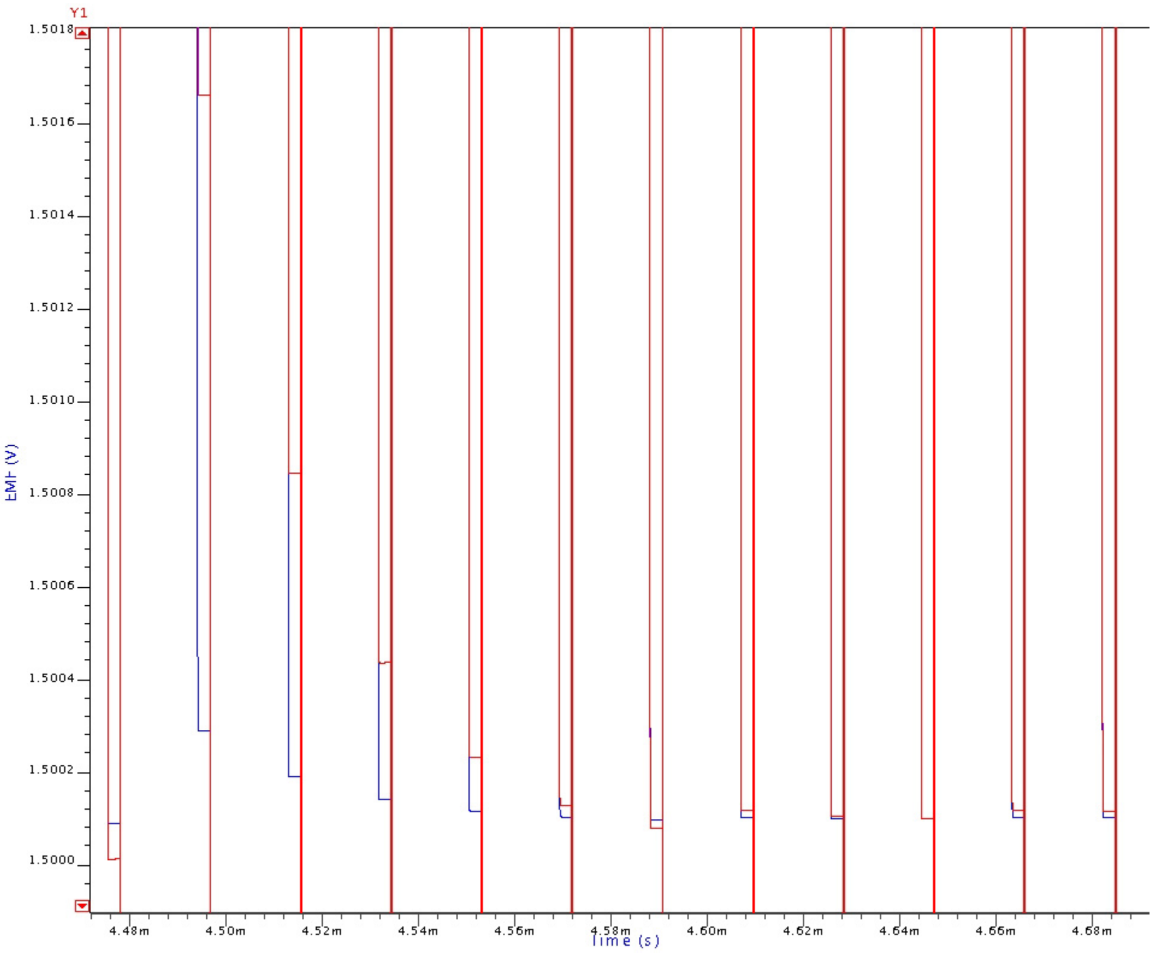


Figure 42 ADC gain error calibration

3.4.3 Analog-to-digital conversion

Figure 43 shows the ADC simulation results with an input analog signal of +0.9 V. If the bottom plate's voltage of lower capacitor array, which is green line, is higher than the bottom plate's voltage of upper capacitor array, which is red line, the current bit capacitor switch will be kept on. Otherwise will be kept off. The pink line is the output of coarse comparator. The blue line is the output of fine comparator. Those outputs signals determine the capacitor switches on or off. The voltages on the bottom plates converge to ½ the power supply voltage. The 16-bit conversion result is 1111001100110010.

Comparing the ideal result 1111001100110001, there is 1 bit error due to the capacitor ratio error calibration. The capacitor ratio error calibration has 0.1 LSB calibration accuracy for each bit capacitor. The whole ADC has 1 LSB calibration accuracy for capacitor ratio error. This is the reason why there is 1 bit conversion error. The first conversion bit some times need big current to charge MSB capacitor. Double clock cycles are assigned to it for saving power. The hysteresis error correction is done in 10th, 11th and 12th clock cycles. This technique also corrects the error due to different offset between coarse and fine comparators.

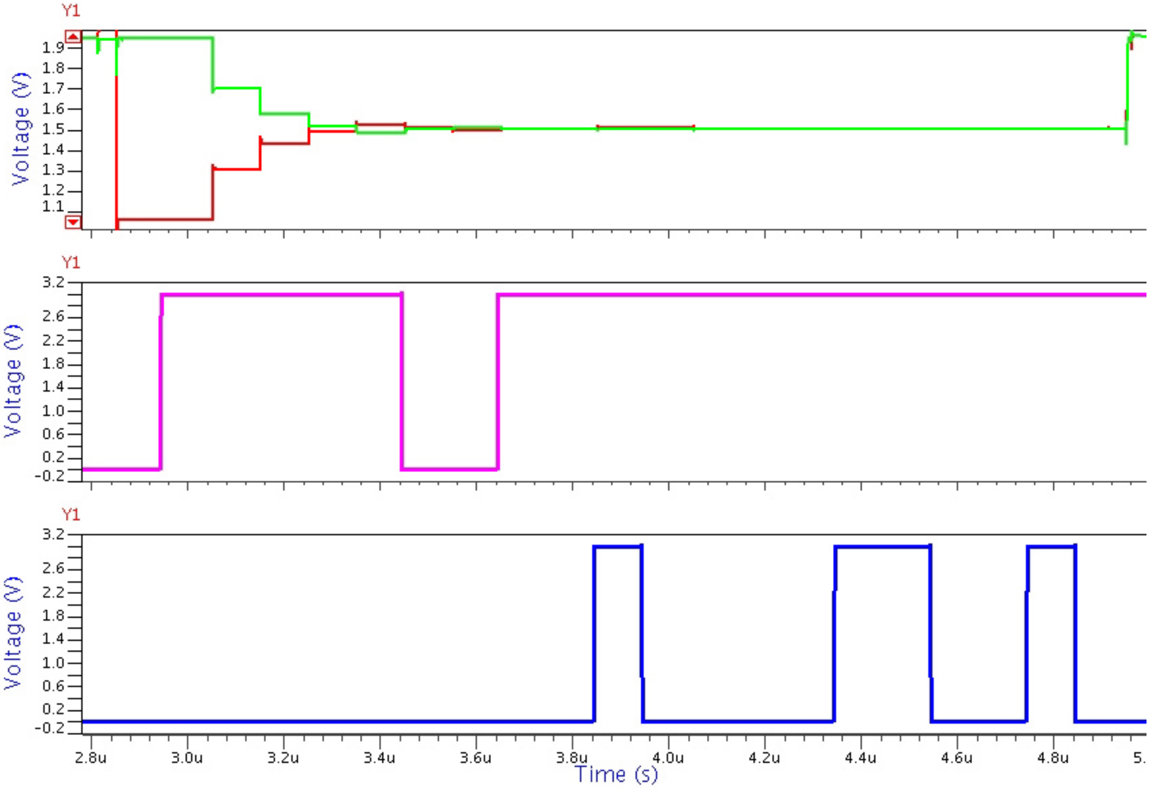


Figure 43 Comparator outputs and bottom plates voltage with +0.9 V applied

3.5 Layout

Figure 44 shows the ADC layout. The calibration circuit is on the left side. The analog to digital conversion control circuit is on the left side top corner. In the center of the die, there are two sets of capacitor array. The calibration capacitor arrays are on the right side of the die. The total die area is 7 mm^2.

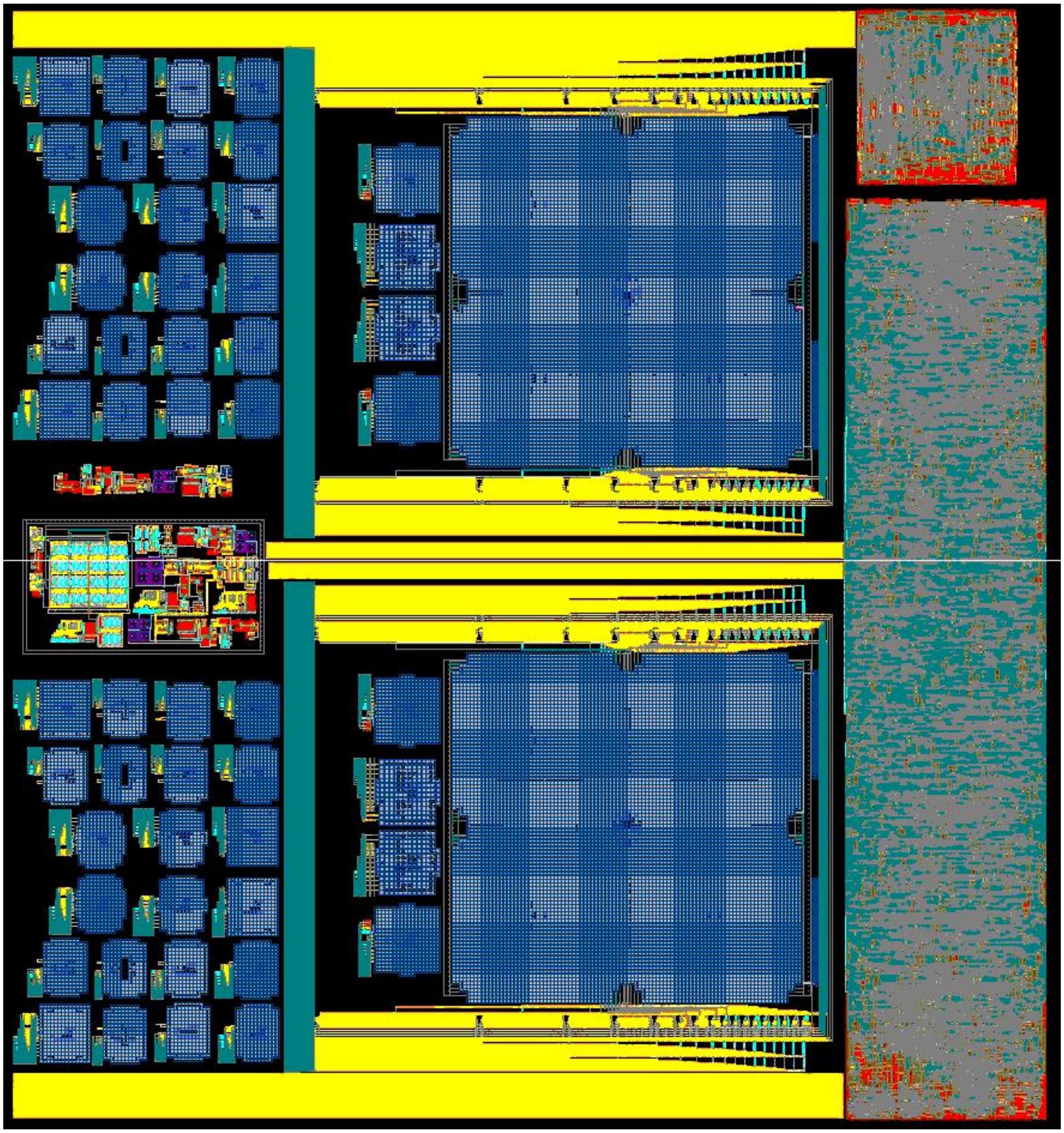


Figure 44 ADC layout

3.6 Fine comparator testing

The fine comparator schematic and die photo are shown below in Figure 45 and Figure 46. It was fabricated to verify functionality of comparing 2 small voltage signals. Simulations show it can resolve 15 μV. There are 5 signals to control the comparator input connection switch, 3 comparators reset switches and latch comparator latch signal. Test results show the comparator can distinguish a 60 μV voltage signal, which is two LSBs of the ADC. This may be due to noise of testing setup. A probe station is used for this comparator testing. It is hard to decouple the power bus noise with this probe station. The input signal, which is generated by a multi-meter, has certain noise too. Environment noise can also couple into the wires for connecting the multi-meter and the comparator input. All of those noises can lower the comparator resolution.

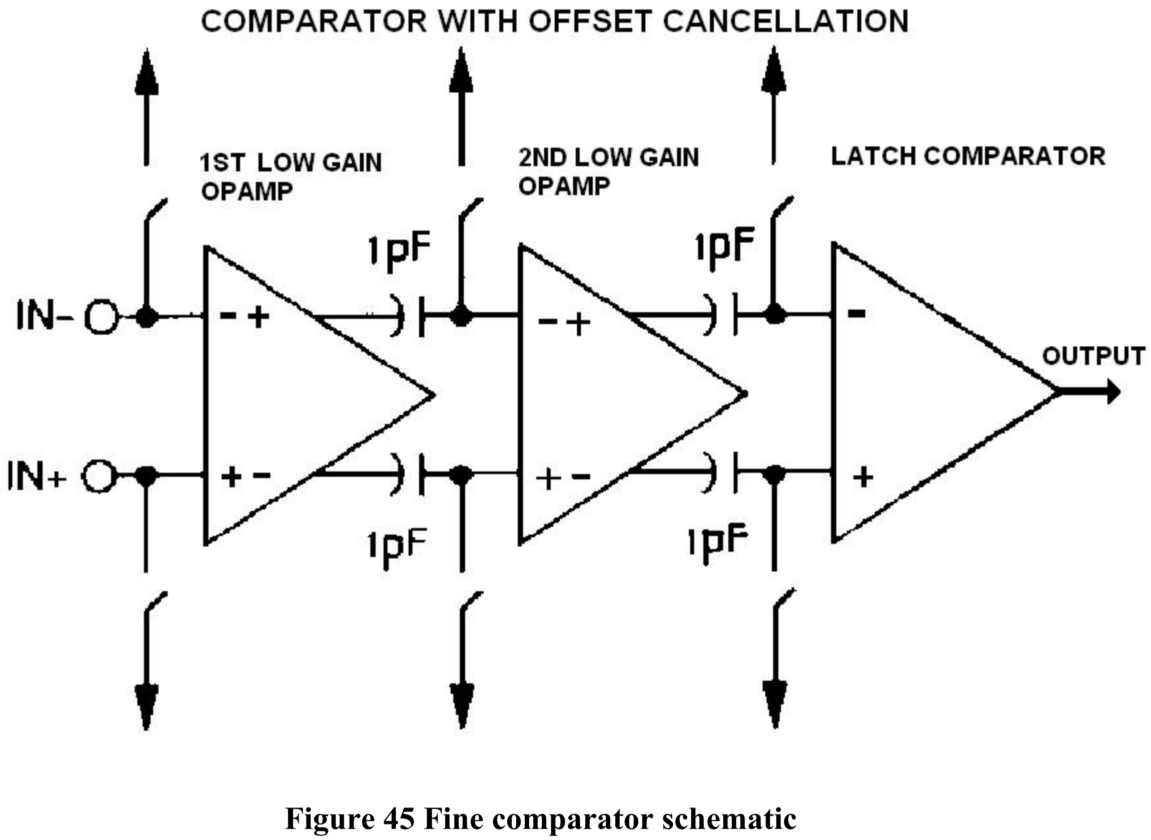


Figure 45 Fine comparator schematic

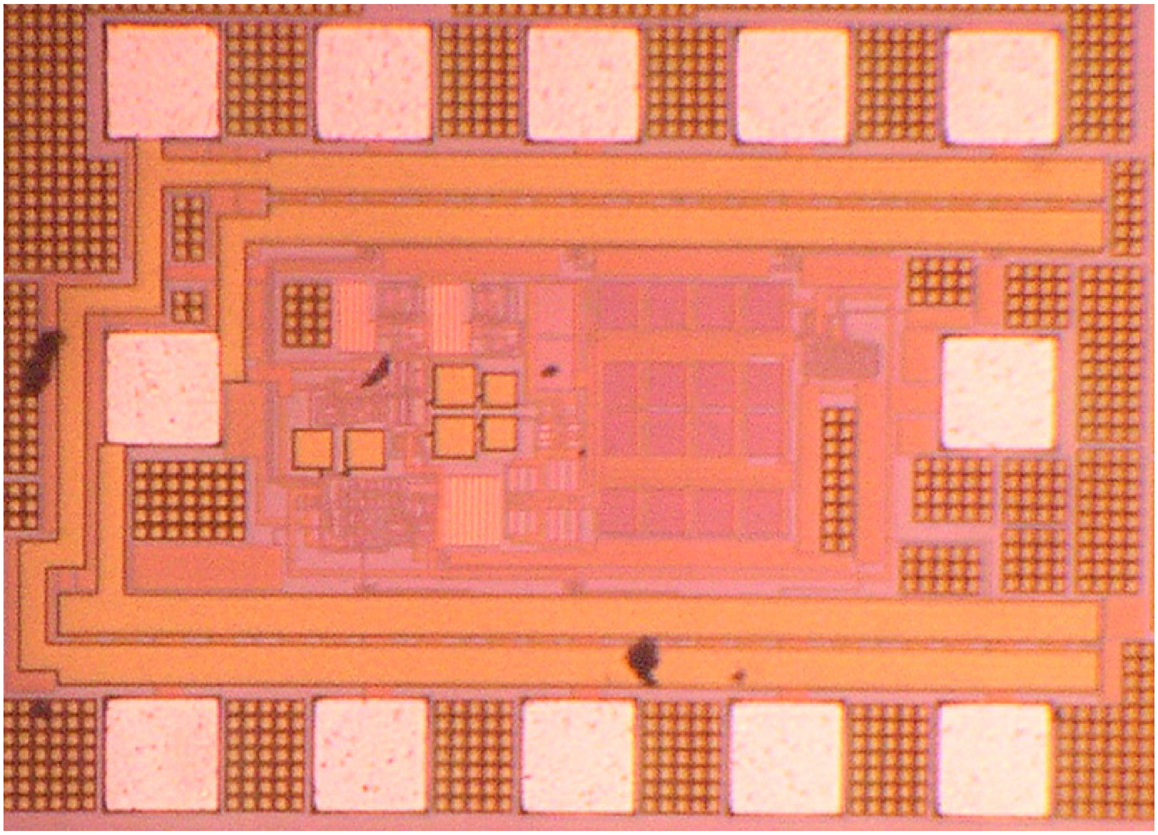


Figure 46 Fine comparator die photo

3.7 Conclusion

A 6.2 mW 16-bit 500 KSps charge redistribution self-calibrating successive approximation analog-to-digital converter (ADC) is designed. It has an input range of 2 V, a resolution of 16 bits and operates with +/- 1.5 V supplies. Simulations show a signal-to-noise ratio of 90 dB for an effective accuracy of 15 bits in TSMC's 0.25µ CMOS process. A novel interleaving architecture and an improved comparator design contribute to reducing power dissipation while maintaining accuracy and speed. The ADC is designed to digitize the amplified neurophysiological signals from the companion 16-channel amplifier channel IC.

CHAPTER FOUR

4. CONCLUSION

An 8-channel amplifier channel IC was designed, fabricated and tested using TSMC's 0.25µ CMOS process. Results show power dissipation of 1.2 mW per channel, programmable gains from 12 to 250, and 79 db SFDR up to 7 KHz for different input DC offsets. The integrated noise from 1 Hz to 7 KHz is 2.5 µV with 0 V DC offset input. The +/- 0.3 V dc input offset of each channel is cancelled with two 5-bits DACs controlling the second and third gain stages op-amp positive input node. PSRR+ at 1 KHz is 61 db. The PSRR- at 1 KHz is 51 db. The 8 analog signals are sampled simultaneously and multiplexed to one output. The amplifier channel IC has a cross talk problem, which is 1:500 at 1 KHz with maximum gain 250. This problem is corrected in the 16-channel CMOS amplifier channel IC layout. The modified amplifier channel IC is ready for fabrication.

The design of 16-bit 500 KSps ADC was completed using TSMC's 0.25µ CMOS process. The simulation results show that a 6.2 mW 500 KSps ADC with 16-bit resolution is achieved. It has a 2 V input range. The overall ADC noise is less than 0.41 LSB, which is equivalent to 15-bit accuracy. A novel interleaving architecture, a novel gain error calibration and an improved comparator design contribute to reducing the power while maintaining the accuracy and speed. This state-of-the-art ADC was designed along with a low-power 16-channel amplifier channel IC for remote sensing of neurophysiological signals in small rodents.

The whole system is designed for the Department of Veterinary and Comparative Anatomy, Pharmacology, and Physiology at Washington State University to acquire small animals' neural signals for studying sleep. Considering its power consumption, speed and accuracy, this is a state-of-the-art sensor IC. This sensor IC also has potential for applications in the clinical and animal research area for high-fidelity low-power low-noise high-accuracy sensor systems.

BIBLIOGRAPHY

References

1. K. Sasaki, T. Ono, H. Nishino, M. Fukuda, and K.I. Muramot,."A method for long-term artifact-free recording of single unit activity in freely moving animals," J. Neuroscience Methods, vol. 7, pp.43-47, 1983
2. Brokaw, A.P. "A simple three-terminal IC bandgap reference" Solid-State Circuits, IEEE Journal of Volume 9, Issue 6, Dec 1974 Page(s):388 - 393
3. T.B. Tang, et al., "Toward a miniature wireless integrated multisensor microsystem for industrial and biomedical applications," IEEE Sensors J., vol. 2, pp. 628-635, Dec. 2002.
4. I. Obeid, J.C. Morizio, K.A. Moxon, M.A.L. Nicolelis, and P.D. Wolf, "Two multichannel integrated circuits for neural recording and signal processing," IEEE Trans. Of Biomed. Engin. vol. 50, pp. 225-258, Feb. 2003.
5. T. Akin, K. Najafi, and R. M. Bradley, A wireless implantable multichannel digital neural recording system for a micromachined sieve electrode," IEEE J. of Solid-State Circuits, vol. 33, Jan. 1998.
6. K.D. Wise, "Wireless implantable microsystems: coming Breakthroughs in health care," Symposium on VLSI Circuits Digest of Technical Papers, pp. 106-109, 2002
7. A. Budak, "Passive and active network analysis and synthesis,", Boston: Houghton Mifflin, 1974, pp. 323-338.

8. Sone, K. Nishida, Y. and Nakadai, N. A 10-b 100-Msample/s pipelined subranging BiCMOS ADC, IEEE Journal of Solid-State Circuits, Vol. 28, no. 12, pp. 1180-1186, December 1993.
9. J.L. McCreary et al., “All-MOS charge redistribution A/D conversion techniques ---Part I,” IEEE J. Solid-State Circuits, Vol. SC-10, pp. 371-379, Dec. 1975.
10. H.S. Lee, D.A.Hodges, and P.R.Gray, “A self-calibration 15-bit CMOS A/D converter,” IEEE J. Solid-State Circuits, Vol. SC-19, pp. 813-819, Dec. 1984.
11. R.K. Hester et al., “Fully differential ADC with rail-to-rail common-mode range and nonlinear capacitor compensation,” IEEE J. Solid-State Circuits, vol. 25 no. 1, pp. 173-183, Feb. 1990.
12. Tan, K.-S.; Kiriaki, S.; de Wit, M.; Fattaruso, J.W.; Tsay, C.-Y.; Matthews, W.E.; Hester, R.K.; **“**Error correction techniques for high-performance differential A/D converters,” IEEE J. Solid-State Circuits, Vol. 25 , no. 6 , pp. 1318 – 1327, Dec. 1990.
13. K. S. Tan, “On board self-calibration of analog-to-digital and digital-to-analog converters,” U.S. Patent 4 399 426, Aug. 16, 1983.
14. Tewksbury, T.L.; Lee, H.-S.; Miller, G.A.; "The effects of oxide traps on the large-signal transient response of analog MOS circuits," IEEE J. Solid-State Circuits, Vol. 24 , no. 2 , pp. 542 – 544, April 1989.
15. Yin, G.M.; Eynde, F.O.; Sansen, W.; ”A latch comparator with 8-b resolution,**”** IEEE J. Solid-State Circuits, Vol. 27 , no. 2, pp. 208 – 211, Feb. 1992.
16. Kansy, R.J.; "Response of a correlated double sampling circuit to 1/f noise," IEEE J. Solid-State Circuits, Vol. 15, no. 3, pp. 373 - 375, June 1980.

17. Philip W. Yee; “Noise considerations in high-accuracy A/D converters” IEEE J. Solid-State Circuits, Vol, SC-21, No. 6, Dec 1986.

APPENDIX

A. THE VERILOG CODES FOR COMPARATOR OFFSET CALIBRATION

```
// fsm machine for main cap array switch
    always@(CurrentState or start or zero or oftend or times128 )
    begin
        case(CurrentState)


        S0:
            begin
            conngnd<=0;oftsample<=0;cmpsw<=0;png<=0;
            storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=1;
            setone<=0;setcap<=0;fend<=0;
            cmprst[3:0]<=4'b1111;
            latch<=0;cvalue[6:0]<=7'b0000000;
            if(start)  NextState <=    S1;
            else NextState <=    S0;
            end


        S1:
            begin
            conngnd<=1;oftsample<=1;cmpsw<=0;png<=1;
            storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=1;
            setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b1111;
    latch<=1;cvalue[6:0]<=7'b0001010;
    NextState <=    S2;
    end

S2:
    begin
    conngnd<=1;oftsample<=1;cmpsw<=0;png<=1;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=1;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b1111;
    latch<=0;cvalue[6:0]<=7'b0000000;
    if(zero)  NextState <=    S3;
    else      NextState <=    S2;
    end
S3:
    begin
    conngnd<=1;oftsample<=1;cmpsw<=0;png<=1;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=1;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b1101;
    latch<=1;cvalue[6:0]<=7'b1100100;
    NextState <=    S4;
```

```
      end


S4:
      begin
      conngnd<=1;oftsample<=1;cmpsw<=0;png<=1;
      storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
      setone<=0;setcap<=0;fend<=0;
      cmprst[3:0]<=4'b1101;
      latch<=0;cvalue[6:0]<=7'b0000000;
      if(zero)  NextState <=    S5;
      else      NextState <=    S4;
      end
S5:
      begin
      conngnd<=1;oftsample<=1;cmpsw<=0;png<=1;
      storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
      setone<=0;setcap<=0;fend<=0;
      cmprst[3:0]<=4'b1001;
      latch<=1;cvalue[6:0]<=7'b0101000;
      NextState <=    S6;
      end
S6:
      begin
```

```
    conngnd<=1;oftsample<=1;cmpsw<=0;png<=1;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b1001;
    latch<=0;cvalue[6:0]<=7'b0000000;
    if(zero)  NextState <=    S7;
    else      NextState <=    S6;
    end
S7:
    begin
    conngnd<=1;oftsample<=1;cmpsw<=0;png<=1;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b0001;
    latch<=1;cvalue[6:0]<=7'b0000101;
    NextState <=    S8;
    end
S8:
    begin
    conngnd<=1;oftsample<=1;cmpsw<=0;png<=1;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b0001;
    latch<=0;cvalue[6:0]<=7'b0000000;
    if(zero)  NextState <=    S9;
    else       NextState <=    S8;
    end
S9:
    begin
    conngnd<=1;oftsample<=0;cmpsw<=0;png<=1;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b0000;
    latch<=1;cvalue[6:0]<=7'b0011001;
    NextState <=    S10;
    end


S10:
    begin
    conngnd<=1;oftsample<=0;cmpsw<=1;png<=1;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b0000;
    latch<=0;cvalue[6:0]<=7'b0000000;
    if(zero)  NextState <=    S11;
    else      NextState <=   S10;
    end
S11:
    begin
    conngnd<=1;oftsample<=0;cmpsw<=1;png<=1;
    storep0<=1;storepn<=0;add<=0;shift<=0;setc1zero<=0;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b0000;
    latch<=0;cvalue[6:0]<=7'b0000000;
    NextState <=   S12;
    end

S12:
    begin
    conngnd<=1;oftsample<=0;cmpsw<=0;png<=1;
    storep0<=0;storepn<=0;add<=0;shift<=1;setc1zero<=0;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b1111;
    latch<=0;cvalue[6:0]<=7'b0000000;
    NextState <=   S13;
    end
```

```
S13:
    begin
    conngnd<=1;oftsample<=1;cmpsw<=0;png<=1;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
    setone<=1;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b1111;
    latch<=1;cvalue[6:0]<=7'b0001010;
    NextState <=    S14;
    end

S14:
    begin
    conngnd<=1;oftsample<=1;cmpsw<=0;png<=1;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b1111;
    latch<=0;cvalue[6:0]<=7'b0000000;
    if(zero)  NextState <=    S15;
    else      NextState <=    S14;
    end
S15:
    begin
    conngnd<=1;oftsample<=1;cmpsw<=0;png<=1;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b1101;
    latch<=1;cvalue[6:0]<=7'b1100100;
    NextState <=    S16;
    end

S16:
    begin
    conngnd<=1;oftsample<=1;cmpsw<=0;png<=1;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b1101;
    latch<=0;cvalue[6:0]<=7'b0000000;
    if(zero)  NextState <=    S17;
    else      NextState <=    S16;
    end
S17:
    begin
    conngnd<=1;oftsample<=1;cmpsw<=0;png<=1;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b1001;
    latch<=1;cvalue[6:0]<=7'b0101000;
    NextState <=    S18;
    end
S18:
    begin
    conngnd<=1;oftsample<=1;cmpsw<=0;png<=1;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b1001;
    latch<=0;cvalue[6:0]<=7'b0000000;
    if(zero)  NextState <=    S19;
    else      NextState <=    S18;
    end
S19:
    begin
    conngnd<=1;oftsample<=1;cmpsw<=0;png<=1;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b0001;
    latch<=1;cvalue[6:0]<=7'b0000101;
    NextState <=    S20;
    end
S20:
    begin
    conngnd<=1;oftsample<=1;cmpsw<=0;png<=1;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b0001;
    latch<=0;cvalue[6:0]<=7'b0000000;
    if(zero)  NextState <=    S21;
    else      NextState <=    S20;
    end
S21:
    begin
    conngnd<=1;oftsample<=0;cmpsw<=0;png<=1;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b0000;
    latch<=1;cvalue[6:0]<=7'b0011001;
    NextState <=    S22;
    end

S22:
    begin
    conngnd<=1;oftsample<=0;cmpsw<=1;png<=1;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b0000;
    latch<=0;cvalue[6:0]<=7'b0000000;
    if(zero)  NextState <=    S23;
    else      NextState <=    S22;
    end

S23:
    begin
    conngnd<=1;oftsample<=0;cmpsw<=1;png<=1;
    storep0<=0;storepn<=1;add<=0;shift<=0;setc1zero<=0;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b0000;
    latch<=0;cvalue[6:0]<=7'b0000000;
    NextState <=    S24;
    end

S24:
    begin
    conngnd<=1;oftsample<=0;cmpsw<=0;png<=1;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
    setone<=0;setcap<=1;fend<=0;
    cmprst[3:0]<=4'b0000;
    latch<=0;cvalue[6:0]<=7'b0000000;
    if    (oftend) NextState <=    S25;
    else NextState <=    S12;
    end

S25:
    begin
    conngnd<=1;oftsample<=0;cmpsw<=0;png<=0;
    storep0<=0;storepn<=0;add<=1;shift<=0;setc1zero<=0;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b1111;
    latch<=0;cvalue[6:0]<=7'b0000000;
    NextState <=    S26;
    end

S26:
    begin
    conngnd<=1;oftsample<=0;cmpsw<=0;png<=0;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=1;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b1111;
    latch<=0;cvalue[6:0]<=7'b0000000;
    if    (!times128)  NextState <=    S1;
```

```
    else NextState <=    S27;
    end
S27:
    begin
    conngnd<=1;oftsample<=0;cmpsw<=0;png<=0;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=1;
    setone<=0;setcap<=0;fend<=1;
    cmprst[3:0]<=4'b1111;
    latch<=0;cvalue[6:0]<=7'b0000000;
    NextState <=    S0;
    end

default:
    begin
    conngnd<=0;oftsample<=0;cmpsw<=0;png<=0;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=1;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b1111;
    latch<=0;cvalue[6:0]<=7'b0000000;
    NextState <=    S0;
    end

endcase
```

```
    end


// reset the fsm machine synchronous
    always@(posedge clock)
    begin
        if(reset)
            begin
            CurrentState <=    S0;
            end
        else
            begin
            CurrentState <=    NextState;
            end
    end
```

B. THE VERILOG CODES FOR CAPACITOR RATIO ERROR CALIBRATION

```
    always@(CurrentState or start or zero or back)
    begin
        case(CurrentState)

        S0:
            begin
            capend<=0; setzero<=1; latch<=1;
            setone[10:0] <=11'b00000000000;
    setcmp[10:0] <=11'b00000000000;

    if(start)  NextState <=    S1;

    else       NextState <=    S0;

    end


S1:

    begin

    capend<=0; setzero<=0; latch<=0;

    setone[10:0] <=11'b00000000001;

    setcmp[10:0] <=11'b00000000000;

    if (zero) NextState <=    S2;

    else NextState <=    S1;

    end


S2:

    begin

    capend<=0; setzero<=0; latch<=1;

    setone[10:0] <=11'b00000000000;

    setcmp[10:0] <=11'b00000000001;

    NextState <=    S3;

    end


S3:
    begin
    capend<=0; setzero<=0; latch<=0;
    setone[10:0] <=11'b00000000010;
    setcmp[10:0] <=11'b00000000000;
    if (zero) NextState <=    S4;
    else NextState <=    S3;
    end

S4:
    begin
    capend<=0; setzero<=0; latch<=1;
    setone[10:0] <=11'b00000000000;
    setcmp[10:0] <=11'b00000000010;
    if(back[10]) NextState <=    S25;
    else NextState <=    S5;
    end

S5:
    begin
    capend<=0; setzero<=0; latch<=0;
    setone[10:0] <=11'b00000000100;
    setcmp[10:0] <=11'b00000000000;
    if (zero) NextState <=    S6;
    else NextState <=    S5;

    end


S6:

    begin

    capend<=0; setzero<=0; latch<=1;

    setone[10:0] <=11'b00000000000;

    setcmp[10:0] <=11'b00000000100;

    if(back[9]) NextState <=    S25;

    else NextState <=    S7;

    end


S7:

    begin

    capend<=0; setzero<=0; latch<=0;

    setone[10:0] <=11'b00000001000;

    setcmp[10:0] <=11'b00000000000;

    if (zero) NextState <=    S8;

    else NextState <=    S7;

    end


S8:

    begin
    capend<=0; setzero<=0; latch<=1;

    setone[10:0] <=11'b00000000000;

    setcmp[10:0] <=11'b00000001000;

    if(back[8]) NextState <=    S25;

    else NextState <=    S9;

    end


S9:

    begin

    capend<=0; setzero<=0; latch<=0;

    setone[10:0] <=11'b00000010000;

    setcmp[10:0] <=11'b00000000000;

    if (zero) NextState <=    S10;

    else NextState <=    S9;


    end


S10:

    begin

    capend<=0; setzero<=0; latch<=1;

    setone[10:0] <=11'b00000000000;

    setcmp[10:0] <=11'b00000010000;

    if(back[7]) NextState <=    S25;
    else NextState <=    S11;
    end

S11:
    begin
    capend<=0; setzero<=0; latch<=0;
    setone[10:0] <=11'b00000100000;
    setcmp[10:0] <=11'b00000000000;
    if (zero) NextState <=    S12;
    else NextState <=    S11;
    end

S12:
    begin
    capend<=0; setzero<=0; latch<=1;
    setone[10:0] <=11'b00000000000;
    setcmp[10:0] <=11'b00000100000;
    if(back[6]) NextState <=    S25;
    else NextState <=    S13;
    end

S13:
    begin
    capend<=0; setzero<=0; latch<=0;
    setone[10:0] <=11'b00001000000;
    setcmp[10:0] <=11'b00000000000;
    if (zero) NextState <=    S14;
    else NextState <=    S13;
    end

S14:
    begin
    capend<=0; setzero<=0; latch<=1;
    setone[10:0] <=11'b00000000000;
    setcmp[10:0] <=11'b00001000000;
        if(back[5]) NextState <=    S25;
        else NextState <=    S15;
    end

S15:
    begin
    capend<=0; setzero<=0; latch<=0;
    setone[10:0] <=11'b00010000000;
    setcmp[10:0] <=11'b00000000000;
    if (zero) NextState <=    S16;
    else NextState <=    S15;
    end

S16:
    begin
    capend<=0; setzero<=0; latch<=1;
    setone[10:0] <=11'b00000000000;
    setcmp[10:0] <=11'b00010000000;
    if(back[4]) NextState <=    S25;
    else NextState <=    S17;
    end

S17:
    begin
    capend<=0; setzero<=0; latch<=0;
    setone[10:0] <=11'b00100000000;
    setcmp[10:0] <=11'b00000000000;
    if (zero) NextState <=    S18;
    else NextState <=    S17;
    end

S18:
    begin
    capend<=0; setzero<=0; latch<=1;
    setone[10:0] <=11'b00000000000;
    setcmp[10:0] <=11'b00100000000;
    if(back[3]) NextState <=    S25;
    else NextState <=    S19;
    end

S19:
    begin
    capend<=0; setzero<=0; latch<=0;
    setone[10:0] <=11'b01000000000;
    setcmp[10:0] <=11'b00000000000;
    if (zero) NextState <=    S20;
    else NextState <=    S19;
    end

S20:
    begin
    capend<=0; setzero<=0; latch<=1;
    setone[10:0] <=11'b00000000000;
    setcmp[10:0] <=11'b01000000000;
    if(back[2]) NextState <=    S25;
    else NextState <=    S21;
    end
```

```
S21:
    begin
    capend<=0; setzero<=0; latch<=0;
    setone[10:0] <=11'b10000000000;
    setcmp[10:0] <=11'b00000000000;
    if (zero) NextState <=    S22;
    else NextState <=    S21;
    end

S22:
    begin
    capend<=0; setzero<=0; latch<=0;
    setone[10:0] <=11'b00000000000;
    setcmp[10:0] <=11'b10000000000;
    NextState <=    S25;
    end

S25:
    begin
    capend<=1; setzero<=0; latch<=0;
    setone[10:0] <=11'b00000000000;
    setcmp[10:0] <=11'b00000000000;
```

```
            NextState <=    S0;
            end


        default:
            begin
            capend<=0; setzero<=1; latch<=1;
            setone[10:0] <=11'b00000000000;
            setcmp[10:0] <=11'b00000000000;
            NextState <= S0;
            end

        endcase
    end

// reset the fsm machine synchronous
    always@(negedge clock)
    begin
        if(reset)
            begin
            CurrentState <=    S0;
            cmpoutdelay <= 0;
            end
        else
            begin
            CurrentState <=    NextState;
            cmpoutdelay <= cmpout;
            end
    end
```

C. THE VERILOG CODES FOR CM INPUT SIGNAL CANCELLATION

```
// fsm machine for main cap array switch
    always@(CurrentState or start or zero or cmeend or times16)
    begin
        case(CurrentState)

        S0:
            begin
            spl_connvrefp_gnd<=0;cmss=0;cmpsw<=0;
            setone<=0;setzero<=1;setcap<=0;
            shift<=0;cmprst[3:0]<=4'b1111;
            latch<=0;cvalue[6:0]<=7'b0000000;
            pv<=0;pg<=0;ng<=0;
            add<=0;fend<=0;setinitcapposi<=1;

            if(start)  NextState <=    S1;
            else NextState <=    S0;
            end

        S1:
            begin
            spl_connvrefp_gnd<=1;cmss=1;cmpsw<=0;
            setone<=1;setzero<=0;setcap<=0;
            shift<=0;cmprst[3:0]<=4'b1111;
            latch<=1;cvalue[6:0]<=7'b0001010;
            pv<=1;pg<=0;ng<=1;
            add<=0;fend<=0;setinitcapposi<=0;

            NextState <=    S2;
            end

        S2:
            begin
            spl_connvrefp_gnd<=1;cmss=1;cmpsw<=0;
            setone<=0;setzero<=0;setcap<=0;
            shift<=0;cmprst[3:0]<=4'b1111;
            latch<=0;cvalue[6:0]<=7'b0000000;
            pv<=1;pg<=0;ng<=1;
            add<=0;fend<=0;setinitcapposi<=0;
            if (zero) NextState <=    S3;
            else NextState <=    S2;
            end

        S3:
            begin
            spl_connvrefp_gnd<=1;cmss=1;cmpsw<=0;
            setone<=0;setzero<=0;setcap<=0;
            shift<=0;cmprst[3:0]<=4'b1101;
            latch<=1;cvalue[6:0]<=7'b1100100;
            pv<=1;pg<=0;ng<=1;
            add<=0;fend<=0;setinitcapposi<=0;

            NextState <=    S4;
            end

        S4:
            begin
            spl_connvrefp_gnd<=1;cmss=1;cmpsw<=0;
            setone<=0;setzero<=0;setcap<=0;
            shift<=0;cmprst[3:0]<=4'b1101;
            latch<=0;cvalue[6:0]<=7'b0000000;
            pv<=1;pg<=0;ng<=1;
            add<=0;fend<=0;setinitcapposi<=0;

            if (zero) NextState <=    S5;
            else NextState <=    S4;
            end

        S5:
            begin
            spl_connvrefp_gnd<=1;cmss=1;cmpsw<=0;
            setone<=0;setzero<=0;setcap<=0;
            shift<=0;cmprst[3:0]<=4'b1001;
            latch<=1;cvalue[6:0]<=7'b0101000;
            pv<=1;pg<=0;ng<=1;
            add<=0;fend<=0;setinitcapposi<=0;

            NextState <=    S6;
            end

        S6:
            begin
            spl_connvrefp_gnd<=1;cmss=1;cmpsw<=0;
            setone<=0;setzero<=0;setcap<=0;
            shift<=0;cmprst[3:0]<=4'b1001;
```

```
        latch<=0;cvalue[6:0]<=7'b0000000;
        pv<=1;pg<=0;ng<=1;
        add<=0;fend<=0;setinitcapposi<=0;

        if (zero) NextState <=    S7;
        else NextState <=    S6;
        end

    S7:
        begin
        spl_connvrefp_gnd<=1;cmss=0;cmpsw<=0;
        setone<=0;setzero<=0;setcap<=0;
        shift<=0;cmprst[3:0]<=4'b0001;
        latch<=1;cvalue[6:0]<=7'b0000101;
        pv<=1;pg<=0;ng<=1;
        add<=0;fend<=0;setinitcapposi<=0;

        NextState <=    S8;
        end

    S8:
        begin
        spl_connvrefp_gnd<=1;cmss=0;cmpsw<=0;
        setone<=0;setzero<=0;setcap<=0;

        shift<=0;cmprst[3:0]<=4'b0001;

        latch<=0;cvalue[6:0]<=7'b0000000;

        pv<=1;pg<=0;ng<=1;

        add<=0;fend<=0;setinitcapposi<=0;


        if (zero) NextState <=    S9;

        else NextState <=    S8;

        end


S9:

        begin

        spl_connvrefp_gnd<=0;cmss=0;cmpsw<=0;

        setone<=0;setzero<=0;setcap<=0;

        shift<=0;cmprst[3:0]<=4'b0000;

        latch<=0;cvalue[6:0]<=7'b0000000;

        pv<=1;pg<=0;ng<=1;

        add<=0;fend<=0;setinitcapposi<=0;


        NextState <=    S10;

        end


S10:
        begin
        spl_connvrefp_gnd<=0;cmss=0;cmpsw<=1;
        setone<=0;setzero<=0;setcap<=0;
        shift<=0;cmprst[3:0]<=4'b0000;
        latch<=1;cvalue[6:0]<=7'b0011001;
        pv<=0;pg<=1;ng<=1;
        add<=0;fend<=0;setinitcapposi<=0;

        NextState <=    S11;
        end

    S11:
        begin
        spl_connvrefp_gnd<=0;cmss=0;cmpsw<=1;
        setone<=0;setzero<=0;setcap<=0;
        shift<=0;cmprst[3:0]<=4'b0000;
        latch<=0;cvalue[6:0]<=7'b0000000;
        pv<=0;pg<=1;ng<=1;
        add<=0;fend<=0;setinitcapposi<=0;

        if (zero) NextState <=    S12;
        else NextState <=    S11;
        end
```

```
S12:
    begin
    spl_connvrefp_gnd<=0;cmss=0;cmpsw<=1;
    setone<=0;setzero<=0;setcap<=1;
    shift<=0;cmprst[3:0]<=4'b0000;
    latch<=0;cvalue[6:0]<=7'b0000000;
    pv<=0;pg<=1;ng<=1;
    add<=0;fend<=0;setinitcapposi<=0;

    NextState <=    S13;
    end

S13:
    begin
    spl_connvrefp_gnd<=0;cmss=0;cmpsw<=1;
    setone<=0;setzero<=0;setcap<=0;
    shift<=1;cmprst[3:0]<=4'b0000;
    latch<=0;cvalue[6:0]<=7'b0000000;
    pv<=0;pg<=1;ng<=1;
    add<=0;fend<=0;setinitcapposi<=0;
```

```
        if (cmeend) NextState <=    S14;

        else NextState <=    S1;

        end


S14:

        begin

        spl_connvrefp_gnd<=0;cmss=0;cmpsw<=0;

        setone<=0;setzero<=0;setcap<=0;

        shift<=0;cmprst[3:0]<=4'b1111;

        latch<=0;cvalue[6:0]<=7'b0000000;

        pv<=0;pg<=1;ng<=1;

        add<=1;fend<=0;setinitcapposi<=0;


        NextState <=    S15;

        end


S15:

        begin

        spl_connvrefp_gnd<=0;cmss=0;cmpsw<=0;

        setone<=0;setzero<=1;setcap<=0;

        shift<=0;cmprst[3:0]<=4'b1111;

        latch<=0;cvalue[6:0]<=7'b0000000;

        pv<=0;pg<=0;ng<=0;
        add<=0;fend<=0;setinitcapposi<=1;


        if(times16)         NextState <=    S16;
        else                NextState <=    S1;
        end
    S16:
        begin
        spl_connvrefp_gnd<=0;cmss=0;cmpsw<=0;
        setone<=0;setzero<=0;setcap<=0;
        shift<=0;cmprst[3:0]<=4'b1111;
        latch<=0;cvalue[6:0]<=7'b0000000;
        pv<=0;pg<=0;ng<=0;
        add<=0;fend<=1;setinitcapposi<=0;


        NextState <=    S0;
        end


    default:
        begin
        spl_connvrefp_gnd<=0;cmss=0;cmpsw<=0;
        setone<=0;setzero<=1;setcap<=0;
        shift<=0;cmprst[3:0]<=4'b1111;
        latch<=0;cvalue[6:0]<=7'b0000000;
```

```
            pv<=0;pg<=0;ng<=0;
            add<=0;fend<=0;setinitcapposi<=1;
            NextState <= S0;
            end

        endcase
    end

// reset the fsm machine synchronous
    always@(posedge clock)
    begin
        if(reset)
            begin
            CurrentState <=    S0;
            end
        else
            begin
            CurrentState <=    NextState;
            end
    end
```

D. THE VERILOG CODES FOR COMPARATOR GAIN ERROR CALIBRATION

```
// fsm machine for main cap array switch
```

```
always@(CurrentState or start or zero or cgeend or times128 )
begin
    case(CurrentState)

    S0:
        begin
        connvrefp<=0;cgesample<=0;cmpsw<=0;pnv<=0;png<=0;cmss<=0;
        storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=1;
        setone<=0;setcap<=0;fend<=0;
        cmprst[3:0]<=4'b1111;
        latch<=0;cvalue[6:0]<=7'b0000000;
        if(start)  NextState <=    S1;
        else NextState <=    S0;
        end

    S1:
        begin
        connvrefp<=1;cgesample<=1;cmpsw<=0;pnv<=1;png<=0;cmss<=1;
        storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=1;
        setone<=0;setcap<=0;fend<=0;
        cmprst[3:0]<=4'b1111;
        latch<=1;cvalue[6:0]<=7'b0001010;
        NextState <=    S2;
```

```
      end


S2:
      begin
      connvrefp<=1;cgesample<=1;cmpsw<=0;pnv<=1;png<=0;cmss<=1;
      storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=1;
      setone<=0;setcap<=0;fend<=0;
      cmprst[3:0]<=4'b1111;
      latch<=0;cvalue[6:0]<=7'b0000000;
      if(zero)  NextState <=    S3;
      else        NextState <=    S2;
      end
S3:
      begin
      connvrefp<=1;cgesample<=1;cmpsw<=0;pnv<=1;png<=0;cmss<=1;
      storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=1;
      setone<=0;setcap<=0;fend<=0;
      cmprst[3:0]<=4'b1101;
      latch<=1;cvalue[6:0]<=7'b1100100;
      NextState <=    S4;
      end


S4:
    begin
    connvrefp<=1;cgesample<=1;cmpsw<=0;pnv<=1;png<=0;cmss<=1;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b1101;
    latch<=0;cvalue[6:0]<=7'b0000000;
    if(zero)  NextState <=    S5;
    else      NextState <=    S4;
    end
S5:
    begin
    connvrefp<=1;cgesample<=1;cmpsw<=0;pnv<=1;png<=0;cmss<=1;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b1001;
    latch<=1;cvalue[6:0]<=7'b0101000;
    NextState <=    S6;
    end
S6:
    begin
    connvrefp<=1;cgesample<=1;cmpsw<=0;pnv<=1;png<=0;cmss<=1;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
    setone<=0;setcap<=0;fend<=0;
```

```
        cmprst[3:0]<=4'b1001;
        latch<=0;cvalue[6:0]<=7'b0000000;
        if(zero)  NextState <=    S7;
        else      NextState <=    S6;
        end
S7:
        begin
        connvrefp<=1;cgesample<=1;cmpsw<=0;pnv<=1;png<=0;cmss<=0;
        storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
        setone<=0;setcap<=0;fend<=0;
        cmprst[3:0]<=4'b0001;
        latch<=1;cvalue[6:0]<=7'b0000101;
        NextState <=    S8;
        end
S8:
        begin
        connvrefp<=1;cgesample<=1;cmpsw<=0;pnv<=1;png<=0;cmss<=0;
        storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
        setone<=0;setcap<=0;fend<=0;
        cmprst[3:0]<=4'b0001;
        latch<=0;cvalue[6:0]<=7'b0000000;
        if(zero)  NextState <=    S9;
        else      NextState <=    S8;
        end
S9:
        begin
        connvrefp<=1;cgesample<=0;cmpsw<=0;pnv<=1;png<=0;cmss<=0;
        storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
        setone<=0;setcap<=0;fend<=0;
        cmprst[3:0]<=4'b0000;
        latch<=1;cvalue[6:0]<=7'b0011001;
        NextState <=    S10;
        end


S10:
        begin
        connvrefp<=1;cgesample<=0;cmpsw<=1;pnv<=0;png<=1;cmss<=0;
        storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
        setone<=0;setcap<=0;fend<=0;
        cmprst[3:0]<=4'b0000;
        latch<=0;cvalue[6:0]<=7'b0000000;
        if(zero)  NextState <=    S11;
        else      NextState <=    S10;
        end
S11:
        begin
        connvrefp<=1;cgesample<=0;cmpsw<=1;pnv<=0;png<=1;cmss<=0;
        storep0<=1;storepn<=0;add<=0;shift<=0;setc1zero<=0;
        setone<=0;setcap<=0;fend<=0;
        cmprst[3:0]<=4'b0000;
        latch<=0;cvalue[6:0]<=7'b0000000;
        NextState <=    S12;
        end

S12:
        begin
        connvrefp<=1;cgesample<=0;cmpsw<=0;pnv<=0;png<=0;cmss<=0;
        storep0<=0;storepn<=0;add<=0;shift<=1;setc1zero<=0;
        setone<=0;setcap<=0;fend<=0;
        cmprst[3:0]<=4'b1111;
        latch<=0;cvalue[6:0]<=7'b0000000;
        NextState <=    S13;
        end


S13:
        begin
        connvrefp<=1;cgesample<=1;cmpsw<=0;pnv<=1;png<=0;cmss<=1;
        storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
        setone<=1;setcap<=0;fend<=0;
        cmprst[3:0]<=4'b1111;
        latch<=1;cvalue[6:0]<=7'b0001010;
        NextState <=    S14;
        end

S14:
        begin
        connvrefp<=1;cgesample<=1;cmpsw<=0;pnv<=1;png<=0;cmss<=1;
        storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
        setone<=0;setcap<=0;fend<=0;
        cmprst[3:0]<=4'b1111;
        latch<=0;cvalue[6:0]<=7'b0000000;
        if(zero)  NextState <=    S15;
        else      NextState <=    S14;
        end
S15:
        begin
        connvrefp<=1;cgesample<=1;cmpsw<=0;pnv<=1;png<=0;cmss<=1;
        storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
        setone<=0;setcap<=0;fend<=0;
        cmprst[3:0]<=4'b1101;
```

```
        latch<=1;cvalue[6:0]<=7'b1100100;
        NextState <=    S16;
        end

S16:
        begin
        connvrefp<=1;cgesample<=1;cmpsw<=0;pnv<=1;png<=0;cmss<=1;
        storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
        setone<=0;setcap<=0;fend<=0;
        cmprst[3:0]<=4'b1101;
        latch<=0;cvalue[6:0]<=7'b0000000;
        if(zero)  NextState <=    S17;
        else       NextState <=    S16;
        end
S17:
        begin
        connvrefp<=1;cgesample<=1;cmpsw<=0;pnv<=1;png<=0;cmss<=1;
        storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
        setone<=0;setcap<=0;fend<=0;
        cmprst[3:0]<=4'b1001;
        latch<=1;cvalue[6:0]<=7'b0101000;
        NextState <=    S18;
        end
```

```
S18:
    begin
    connvrefp<=1;cgesample<=1;cmpsw<=0;pnv<=1;png<=0;cmss<=1;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b1001;
    latch<=0;cvalue[6:0]<=7'b0000000;
    if(zero)  NextState <=    S19;
    else      NextState <=    S18;
    end
S19:
    begin
    connvrefp<=1;cgesample<=1;cmpsw<=0;pnv<=1;png<=0;cmss<=0;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b0001;
    latch<=1;cvalue[6:0]<=7'b0000101;
    NextState <=    S20;
    end
S20:
    begin
    connvrefp<=1;cgesample<=1;cmpsw<=0;pnv<=1;png<=0;cmss<=0;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b0001;
    latch<=0;cvalue[6:0]<=7'b0000000;
    if(zero)  NextState <=    S21;
    else      NextState <=    S20;
    end
S21:
    begin
    connvrefp<=1;cgesample<=0;cmpsw<=0;pnv<=1;png<=0;cmss<=0;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b0000;
    latch<=1;cvalue[6:0]<=7'b0011001;
    NextState <=    S22;
    end

S22:
    begin
    connvrefp<=1;cgesample<=0;cmpsw<=1;pnv<=0;png<=1;cmss<=0;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b0000;
    latch<=0;cvalue[6:0]<=7'b0000000;
    if(zero)  NextState <=    S23;
    else      NextState <=    S22;
    end

S23:
    begin
    connvrefp<=1;cgesample<=0;cmpsw<=1;pnv<=0;png<=1;cmss<=0;
    storep0<=0;storepn<=1;add<=0;shift<=0;setc1zero<=0;
    setone<=0;setcap<=0;fend<=0;
    cmprst[3:0]<=4'b0000;
    latch<=0;cvalue[6:0]<=7'b0000000;
    NextState <=    S24;
    end

S24:
    begin
    connvrefp<=1;cgesample<=0;cmpsw<=0;pnv<=0;png<=0;cmss<=0;
    storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=0;
    setone<=0;setcap<=1;fend<=0;
    cmprst[3:0]<=4'b0000;
    latch<=0;cvalue[6:0]<=7'b0000000;
    if    (cgeend)NextState <=    S25;
    else NextState <=    S12;
```

```
        end

S25:
        begin
        connvrefp<=1;cgesample<=0;cmpsw<=0;pnv<=0;png<=0;cmss<=0;
        storep0<=0;storepn<=0;add<=1;shift<=0;setc1zero<=0;
        setone<=0;setcap<=0;fend<=0;
        cmprst[3:0]<=4'b1111;
        latch<=0;cvalue[6:0]<=7'b0000000;
        NextState <=    S26;
        end

S26:
        begin
        connvrefp<=1;cgesample<=0;cmpsw<=0;pnv<=0;png<=0;cmss<=0;
        storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=1;
        setone<=0;setcap<=0;fend<=0;
        cmprst[3:0]<=4'b1111;
        latch<=0;cvalue[6:0]<=7'b0000000;
        if    (!times128)  NextState <=    S1;
        else NextState <=    S27;
        end
S27:
```

```
            begin
            connvrefp<=1;cgesample<=0;cmpsw<=0;pnv<=0;png<=0;cmss<=0;
            storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=1;
            setone<=0;setcap<=0;fend<=1;
            cmprst[3:0]<=4'b1111;
            latch<=0;cvalue[6:0]<=7'b0000000;
            NextState <=    S0;
            end

        default:
            begin
            connvrefp<=0;cgesample<=0;cmpsw<=0;pnv<=0;png<=0;cmss<=0;
            storep0<=0;storepn<=0;add<=0;shift<=0;setc1zero<=1;
            setone<=0;setcap<=0;fend<=0;
            cmprst[3:0]<=4'b1111;
            latch<=0;cvalue[6:0]<=7'b0000000;
            NextState <=    S0;
            end

        endcase
    end

// reset the fsm machine synchronous
```

```
always@(posedge clock)
begin
    if(reset)
        begin
        CurrentState <=    S0;
        end
    else
        begin
        CurrentState <=    NextState;
        end
end
```