LOW-POWER LOW-NOISE DC-COUPLED SENSOR AMPLIFIER IC

By

WEI ZHENG

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

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

WASHINGTON STATE UNIVERSITY
School of Electrical Engineering and Computer Science

August 2008

To the Faculty of Washington State University:

The members of the Committee appointed to examine the thesis of WEI ZHENG find it satisfactory and recommend that it be accepted.

_____

Chair

_____

_____

ACKNOWLEDGEMENT

I sincerely thank my advisor, Professor George La Rue, for his direction, advice and support throughout my master program.

I would also like to thank Professor David M. Rector in Department of Veterinary and Comparative Anatomy, Pharmacology, and Physiology for funding this research project.

LOW-POWER LOW-NOISE DC-COUPLED SENSOR AMPLIFIER IC

ABSTRACT

By Wei Zheng, M.S.
Washington State University
August 2008

Chair: George S. La Rue

A DC-coupled low-power low-noise sensor amplifier IC for recording neural activities in the brain of mice is designed, fabricated and measured. To acquire neural signals at different locations of the brain simultaneously through implanted electrode array, this multi-channel sensor amplifier chip has 16 independent channels. Each channel has a programmable gain ranging from 5 to 250, a DC offset cancellation range from - 0.3 V to + 0.3 V, and a bandwidth from DC to 7 KHz. With a power supply of +/- 1.5 V, the power dissipation is 1 mW per channel. The measured input referred noise integrated from 1 Hz to 7 KHz is 2.1 μV . Die area is 5x3 mm^2 in TSMC's 0.25μ CMOS process.

Another version of the sensor amplifier IC using chopping amplifier and a different DC offset cancellation scheme is also designed, simulated and laid out. Simulations show that the same functions (programmable gain, DC offset cancellation range, etc.) are achieved while noise, power dissipation and die area are significantly reduced. The simulated input referred noise integrated from 1 Hz to 7 KHz is 1.7 μV. With a supply voltage of +/- 1.2 V, power dissipation is reduced to 0.34 mW per channel. Die area is 2x3 mm^2 in TSMC's 0.25μ CMOS process.

TABLE OF CONTENTS

LIST OF PUBLICATIONS

1. Wei Zheng, George La Rue, ” Low-Power Low-Noise DC-Coupled Sensor Amplifier IC,” in Microelectronics and Electron Devices, 2008. WMED '08. 2008 IEEE Workshop on April 18

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

1. INTRODUCTION

All external signals that we may desire to detect, record and process are by nature analog and continuous, however, modern signal processes are mostly conducted in the digital domain. Hence, CMOS mixed-signal interface circuits are widely used in many areas to acquire analog signals and convert them into digital forms for further DSP processing. The functions that the interface circuits normally perform include amplification, filtering, sampling and analog-to-digital conversion.

Recent biomedical research needs many channels of signal acquisition. The interface circuits need to amplify the weak bioelectrical signals and to convert them into digital form for further DSP analysis. Typically a sensor channel consists of a front-end amplifier followed by an analog-to-digital converter. This research project is to develop a state-of-art 16-channel DC-coupled low-noise low-power high-accuracy sensor amplifier IC to amplify the neural signals from a mouse brain.

There is an increasing demand on neural signal recording systems that enable researchers to simultaneously record neural activities at many locations in the brain. Recently, the advent of implantable electrode arrays further creates the need for fully integrated multi-channel amplifier ICs. Small size is needed so that the amplifier circuits do not overburden the subject with too many cables and large and heavy circuit boards. In addition, noise can be coupled into the signal path through the cables, which severely

degrades the signal-to-noise ratio (SNR). Hence fully-integrated multi-channel sensor amplifier ICs are desirable.

The capability of CMOS integrated circuits to record and amplify neural signals has been demonstrated [1][2]. However, these circuits either have unacceptably high noise level or consume too much power for high volume integration. Noise and power dissipation of some designs [3][4] are low, but they lack programmable gain and use AC coupling, which filters out low frequency signals.

The fabricated sensor amplifier IC achieves a programmable gain from 5 to 250, bandwidth from DC to 7 KHz ,an input referred noise of 2.1 μV integrated from 1 Hz to 7 KHz at a channel gain of 250 and a power consumption of 1 mW per channel. The sensor amplifier IC has a DC offset cancellation range from – 0.3V to + 0.3V so there is no need for large off-chip capacitors to implement high-pass filters.

This sensor amplifier can be used in various biomedical applications where low-noise low-power high-accuracy and multi-channel sensing are required.

CHAPTER TWO

2. SENSOR AMPLIFIER IC

2.1 Design Challenges

As introduced in chapter 1, this sensor amplifier IC is used to record neural signals from the mice brains. This specific application set the requirements and challenges for the design.

2.1.1 DC Offset

Because of the potential difference between the electrode in the brain and the reference site on the skull, input signals to the amplifier channel have a DC offset with amplitude up to 0.3 V. Most existing sensors use AC coupling to remove the DC offset. The disadvantages are that some low frequency information will also be filtered out and that AC coupling requires large off-chip capacitors to implement high-pass filter with a low cutoff frequency. A DC-coupled offset cancellation scheme is proposed. Two 5-bit digital-to-analog converters (DACs) are connected to the positive input terminals of the second and third gain stage op-amps.

2.1.2 Programmable Gain

The neural signal to be recorded has widely varying amplitude; therefore programmable gains of 5, 20, 25, 62.5, 100 and 250 are provided. The variable gain is selected by closing different MOSFET switches in the op-amp's resistive feedback network so that the op-amp has different closed-loop gains.

To cover this wide gain range, three cascaded gain stages are used to avoid having feedback resistor values be too large. Since the DC offset is cancelled at the second and third stages, the gain of the first stage is limited to 2.5, so that the output signal of the first stage will exceed neither the output swing range of the first op-amp nor the input range of the second op-amp. The second and third stages are designed with programmable gains to achieve the desired total channel gain.

2.1.3 Power Consumption

Power consumption of the neural sensor amplifier needs to be low for two major reasons. It is desired not to have any wires that tether the mice since their behavior is modified. Thus the neural sensor IC is to be powered by a battery or remotely powered by inductive or some other means. Low power consumption is a necessity for reliable performance and long operating duration. In addition, circuits must dissipate little power to avoid generating excessive heat which can make the animal uncomfortable or harm the animal.

Thermal noise of resistors and transistors set a lower bound on power dissipation. Using large resistor values in the gain stages can achieve high output swing with low current and therefore low power dissipation, but require large die area and generate higher thermal noise. Resistor values in the range of several hundred kΩ are used as a compromise between noise and power dissipation. There is a trade-off between noise, power and transistor area.

Compared to a previous sensor amplifier chip, modifications are made to decrease power and die area. One bandgap reference is shared by all channels instead of one bandgap reference per channel. Two adjacent channels also share bias circuits.

2.1.4 Noise

To achieve high accuracy and high dynamic range, low input referred noise of the sensor amplifier IC is an important requirement.

Flicker noise and thermal noise are two major noise sources. The bandwidth of this sensor amplifier IC falls into the low frequency band, where the flicker noise dominates. At higher frequencies, thermal noise dominates.

Flicker noise is inversely proportional to the active area of the MOSFETs. Thermal noise can be reduced by burning more power or increasing the width over length ratio. The first gain stage op-amp is the major noise contributor of the total channel noise. Flicker noise is the main noise source within the first op-amp. Large MOSFETs are used in the first op-amp to reduce its flicker noise. Simulations show resistor thermal noise is a major noise source so resistors can not be made too large and the amplifier must provide enough current to achieve the required output swing, which affects power consumption. Therefore, tradeoffs are made among noise, power consumption and die area. The first op-amp has very large MOSFETs, relatively high power consumption, 0.4 mW and small resistors, 11.67 kHz and 17.5 kHz. The following stages can use smaller MOSFETs, have lower power consumption and larger resistors. However, since the gain of the first stage is only 2.5, the noise of the second stage amplifier must be kept relatively low so devices are smaller but still relatively large and power dissipation is only reduced to 0.2 mW.

Two off-chip capacitors are used to lower the noise contributed by the bandgap reference into the signal path. Therefore, input referred noise is not a function of DC offset as in a previous design where the noise increased significantly at large DC offsets.

2.1.5 Low Distortion

High SFDR is an important specification of the sensor amplifier. Many factors affect the linearity of the amplifier channel, such as op-amps, MOSFET switches, non-ideal resistors and capacitors. There are five op-amps in the channel, three in the gain stages, one in the Butterworth low-pass filter and one in the track-and-hold circuit. These op-amps should be designed with high open loop gain to ensure high linearity. Passive components, resistors and capacitors, are also non-ideal and generate distortion. Poly resistor and MIM capacitor are chosen for their high linearity. MOSFET switches in the feedback network of the gain stages to set programmable gains are nonlinear since the on-resistance of the switches are signal-related. Solutions for this are to design the transistors with relatively large W/L ratios or put the switches at the negative input terminal of the op-amps where the voltage is constant.

2.1.6 Crosstalk

The multi-channel sensor amplifier IC is designed to acquire signals from electrodes without one channel affecting another. A previous sensor amplifier chip suffers from a crosstalk problem due to improper ground bus layout. In this version, each channel has its own wide ground route and separate ground pads. The ground pads are wire-bonded to the package's ground plane.

2.2 Sensor Amplifier Design

2.2.1 Architecture

A 16-channel sensor amplifier IC is designed to acquire signals from different electrodes with low crosstalk among channels. The system-level architecture of the sensor amplifier IC is shown in Figure 1.

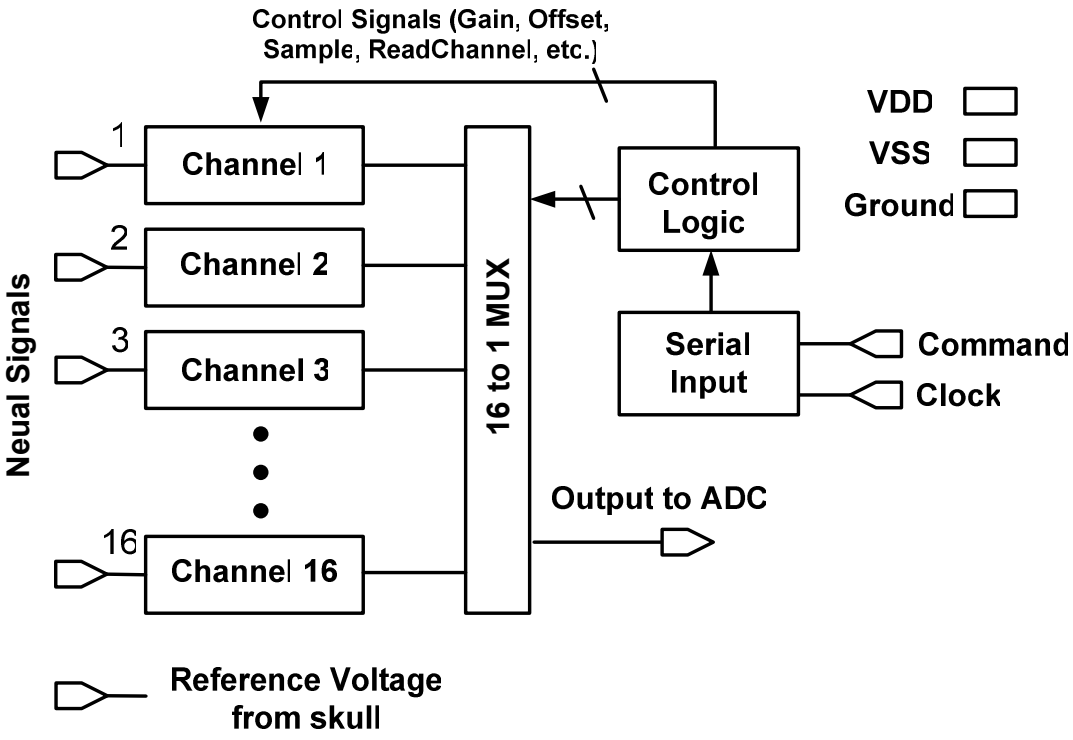

Figure 1 Sensor Amplifier IC Architecture

Sixteen channels amplify, filter and sample signals independently. The 16 channels' outputs are sent to the output through a 16-to-1 multiplexer. To control the sensor amplifier IC, a digital control circuit takes in commands serially, translates the commands into control signals to set up the operating mode of each channel. It also controls the 16-to-1 multiplexer to determine which channel to read.

The digital control circuit is designed based on a state machine. Commands are sent serially into a shift register of the digital control circuit. Figure 2 shows the command format. Each command begins with 3 starting bits "101", followed by the command type, data and channel address. When the highest 3 bits of the shift register are "101", the digital control circuit recognizes the 14 bits in the shift register as a command. The command then is transformed into control signals sent to each channel to set up its operation mode.

Start Bits (101)	Command Type	Data Field	Address Field

Figure 2 Command Format

To meet all the design requirements discussed in section 2.1, the amplifier channel is designed as shown in Figure 3. One amplifier channel consists of three op-amp gain stages, two 5-bit DACs, one 2nd-order low-pass Butterworth filter with a cutoff frequency of 7 kHz and one track-and-hold circuit.

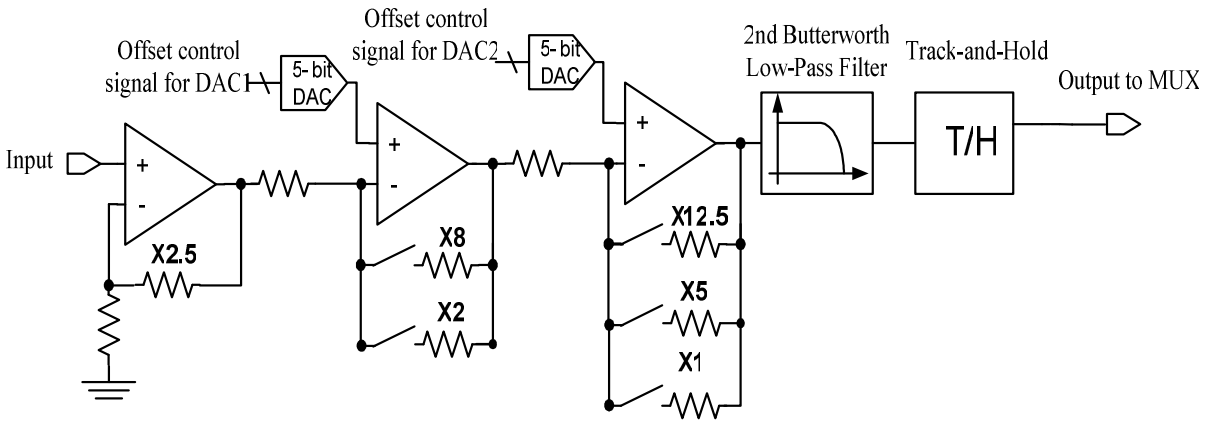

Figure 3 Amplifier Channel

2.2.2 Three Op-Amp Gain Stages

When designing the gain stages, the first consideration is to choose the type of amplifier. Because of the very high impedance of the electrode (>1 GΩ), many literatures have reported employment of the instrumentation amplifier for biomedical sensor application. The disadvantage of the instrumentation amplifier is that it requires more op-amps and therefore has a higher number of MOSFETs, dissipates more power requires more die area and generates more noise.

To ensure high linearity, high open loop gain is required for the op-amps. Folded-cascode op-amps provide high gain because of its high output impedance due to the cascode technique. Folded-cascode op-amps can also provide large input common mode range and large output swing provided that a wide-swing bias circuit is used. However, compared to the two-stage op-amp, folded-cascode op-amp requires more MOSFETs. Two-stage op-amps are selected because of their lower noise, smaller die area and high output range. Active load FETs and FETs in the second stage use longer channel

lengths to increase the output impedance, and therefore the open loop gain. With careful design, an open loop DC gain of 110 dB is achieved to ensure high linearity.

Once the two-stage op-amp is chosen, the next step is to decide what kind of input stage to use. Noise and input common mode range are two major factors to consider. The major noise source of CMOS op-amp is the flicker noise. Flicker noise is caused by carriers randomly being trapped by defects near the semiconductor surface. Since the majority carrier (hole) of the p-channel FET has less mobility and therefore less potential to be trapped, the p-channel FET has lower flicker noise than the n-channel FET with same active area. Thus, the op-amp in the first gain stage uses p-channel FETs for the input differential pair to minimize noise due to the smaller flicker noise. After the 2.5 fixed gain of the first gain stage, the DC offset is amplified to +/- 0.75 V, which puts a higher requirement on the input range of the second op-amp. Since in the TSMC's 0.25μ CMOS process, the n-channel FET has considerably smaller threshold voltage than the p-channel FET, n-channel FET input stages are used in the second and third op-amp to provide a larger input range.

The next consideration is how many gain stages are needed. Since the +/- 0.3 V DC offset is not cancelled and will be amplified by the first stage, the gain of the first stage is set to 2.5. To achieve a total channel gain as high as 250 in only two gain stages requires the feedback resistor of the second stage to be 100 times larger than its input resistor. With the input resistor of the second stage at 25 KΩ, the feedback resistor has to be 2.5 MΩ, which is unacceptably large. Therefore, to cover this wide programmable gain range, the total gain needs to be distributed in three gain stages.

The programmable gain is selected by closing different switches to connect different feedback resistors. Large resistors will allow op-amps to have large output swing without burning too much power, but they will cause more thermal noise. Tradeoffs are made between noise, power and die area when deciding resistor values. Since relatively large current is used in the first gain stage and thermal noise from the resistors in the first gain stage is referred to the input directly, resistors of the first op-amp gain stage are set to be 11.667 kΩ and 17.5 kΩ. Input resistor of the second op-amp gain stage is set to be 100 kΩ according to the first op-amp's driving ability. Similarly, the input resistor of the third gain stage is set to be 125 kΩ according to the second stage's driving ability.

For low noise design, large transistors are used. Flicker noise, which is inversely proportional to the active area of the transistors, dominates at low frequencies. At higher frequencies, thermal noise, which decreases with increasing current, dominates. So the first gain stage op-amp is designed with large input differential pair and active load as well as adequate current. Due to the low gain of the first stage, the op-amp of the second stage also uses large FETs for the input differential pair and active load, and consumes 0.2 mW. The op-amps in the third gain stage, the Butterworth filter and the track-and-hold circuit use smaller size transistors, since the noise contributed by them are divided by the gain from the first and second gain stages.

2.2.3 DACs and Bandgap Reference

The bandgap reference drives resistor-string DACs. Resistor-string DACs are chosen because of their simple circuit implementation. In TSMC 0.25μ CMOS process, poly resistor has high sheet resistance, good matching and high linearity, so it is chosen to implement the resistor-sting DAC.

Compared to a previous sensor amplifier, instead of one bandgap reference for each channel, one bandgap is shared by all 16 channels. As shown in Figure 4, because there are now 16 DACs in parallel, the parallel resistance is 16 times smaller than one individual DAC's resistance. To form a voltage divider to generate the same DAC voltage, the value of R1 and R2 are reduced from 65 KΩ in the previous design to 6.55 KΩ here.

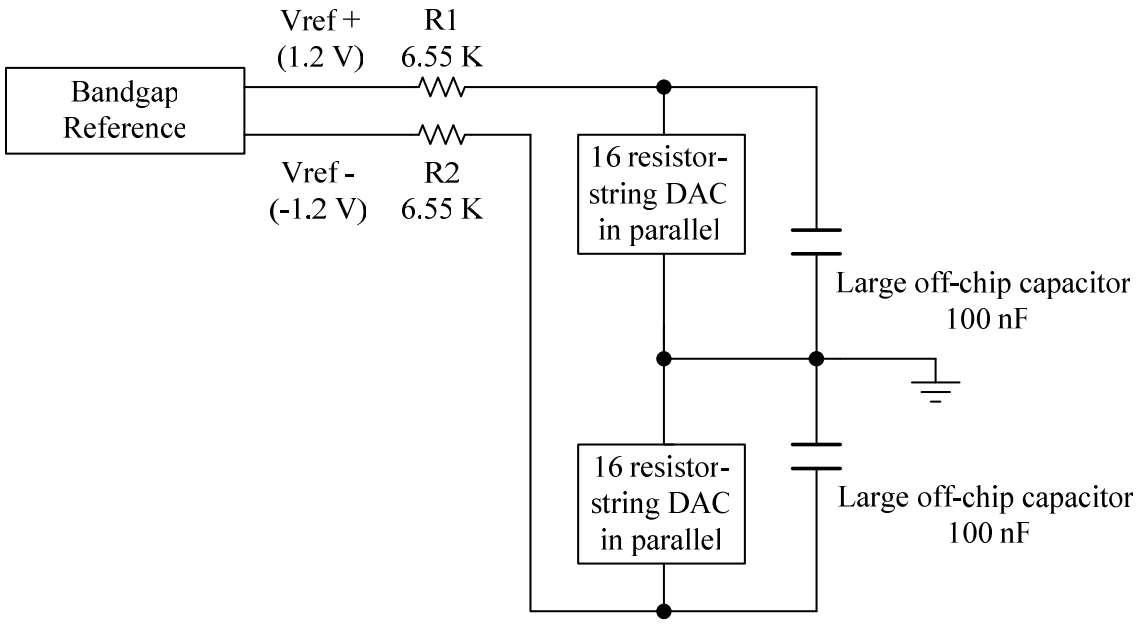

Figure 4 Bandgap reference and DACs

The noise from the bandgap reference and DACs are also coupled into the signal path, so the sizes of the MOSFETs in the bandgap reference are made almost as large as the MOSFETs in the first op-amp for lower flicker noise. The bandgap reference circuit has high current for lower thermal noise and to drive the resistor-string DACs. Large off-chip capacitors are used and as a result, both simulation and measurement of noise show that with different DC offset, the input referred noise is nearly constant.

2.2.4 Butterworth Low-Pass Filter

A second order Butterworth low-pass filter[5] with 7 KHz cutoff frequency filters out the wide-band noise and spurious signals by limiting the bandwidth. It also serves as the anti-aliasing filter following the gain stages and before the track-and-hold and ADC.

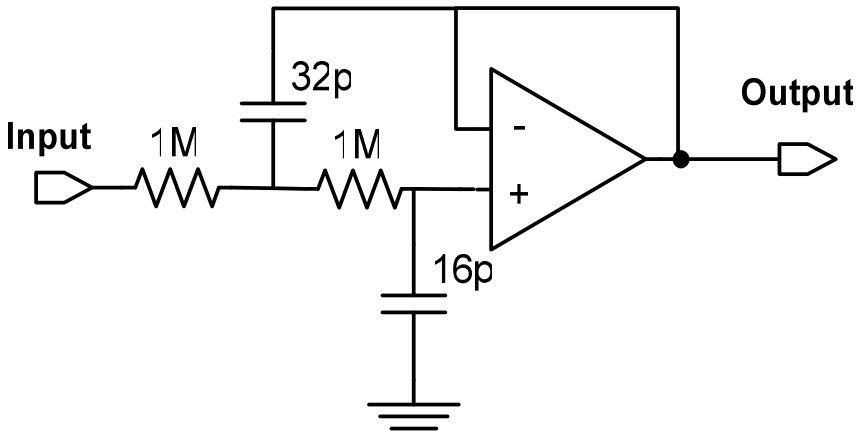

Figure 5 Second Order Low-Pass Butterworth Filter

The op-amp in the filter is designed with high open loop gain for high linearity. The sizes of capacitor and resistor were balanced to reduce the layout area. Large poly resistor values of 1 MΩ are used.

The large hold capacitor in the following track-and-hold circuit is simulated together with the op-amp in the filter to make sure there is enough phase margin, in case the large hold capacitor causes stability problems for the filter's op-amp.

2.2.5 Track-and-Hold Circuit

Figure 6 shows the structure of the track-and-hold circuit. To minimize charge injection and clock feedthrough, a dummy transmission gate, which is half the size of the transmission gate and is driven by slightly delayed clocks, is used.

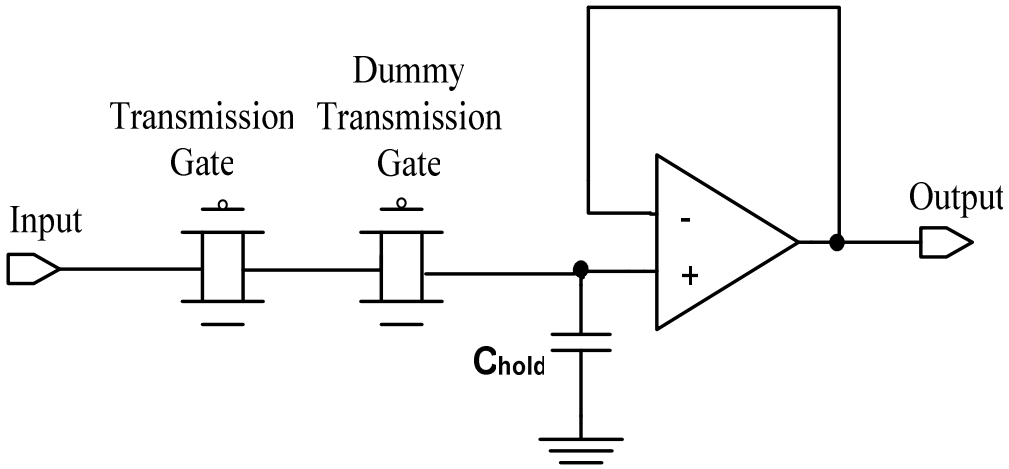

Figure 6 Track-and-Hold Circuit

2.3 Simulation

Specifications such as noise, power dissipation, DC offset cancellation range, power supply rejection ratio (PSRR), spurious free dynamic range (SFDR) are simulated. Table 1 summarizes the simulation results.

Table 1 Simulation Results Summary

Power Supply	+/- 1.5 V
Power Consumption	1 mW per channel
Input referred noise integrated fro 1 Hz to 7 kHz	3 μV (with no DC offset and a channel gain of 25
Output Swing	- 1 V ~ + 1 V
DC Offset Cancellation Range	- 0.3 V ~ + 0.3 V
Programmable Gain	5, 20, 25, 62.5, 100, 250

Power Supply Rejection Ratio (PSRR+)	-77 dB (@ 250 Hz with no input DC offset)
Power Supply Rejection Ratio (PSRR-)	-74 dB (@ 250 Hz with no input DC offset)

Input referred noise increases as the total channel gain decreases, because the noise from the third gain stage, the low-pass filter and the track-and-hold circuit are divided by a smaller gain when referred to the input. Table 2 summarizes the input referred noise of the channel with different gains and no input DC offset.

Table 2 Simulated input referred noise with different gains

Programmable Gain	Input Referred Noise (μV)
5	7.25
20	3.5
25	4.2
62.5	4
100	3.2
250	3

When the amplifier channel works with input DC offset cancellation, the channel noise

also increases since the bandgap reference and DACs inject noise into the signal path. Table 3 shows the simulated input referred noise with different DC offset cancellation voltage. Large off-chip capacitors are used between the reference voltage and ground to reduce the noise injected into the channel from the bandgap and DACs, therefore, as indicated by the simulation, noise remains relatively stable with different input DC offset voltage. Table 4 shows the input referred noise with various size of off-chip capacitors applied when the channel gain is set to 250 and there is no input DC offset. According to the simulations, the 100 nF capacitor is chosen to minimize the noise.

Table 3 Simulated input referred noise with different DC offset cancellation (with a total channel gain of 250)

Input DC Offset (V)	Input Referred Noise (μV)
-0.3	3.63
-0.2	3.45
-0.1	3.21
0	3
0.1	3.10
0.2	3.14
0.3	3.2

Table 4 Simulated input referred noise with different off-chip capacitors

Off-chip Capacitor (nF)	Input Referred Noise (μV)
100	3.2
30	3.55
20	3.65
10	3.8
1	4.1

Figure 7 and Figure 8 summarize power supply rejection ratio simulated with various DC offset and frequencies.

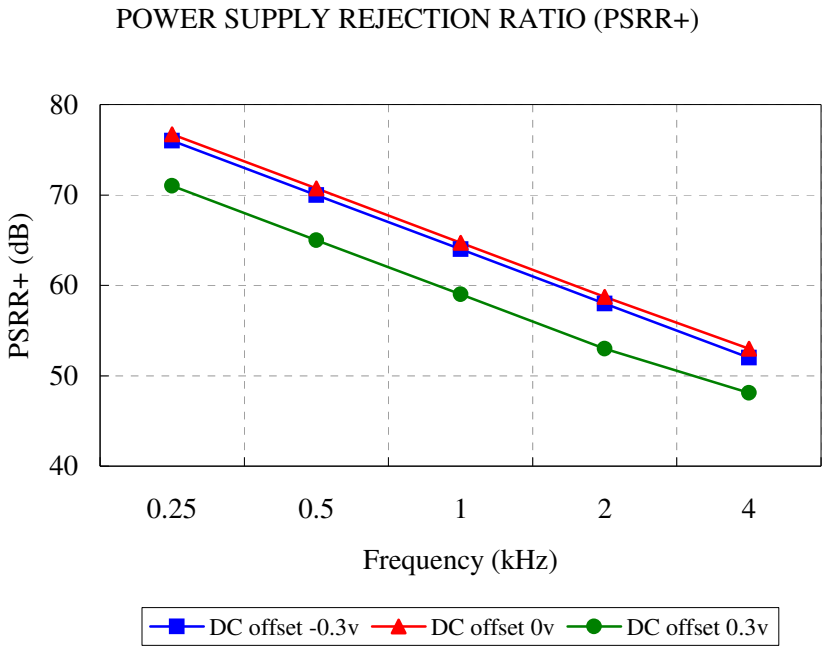

Figure 7 Power Supply Rejection Ratio (PSRR+)

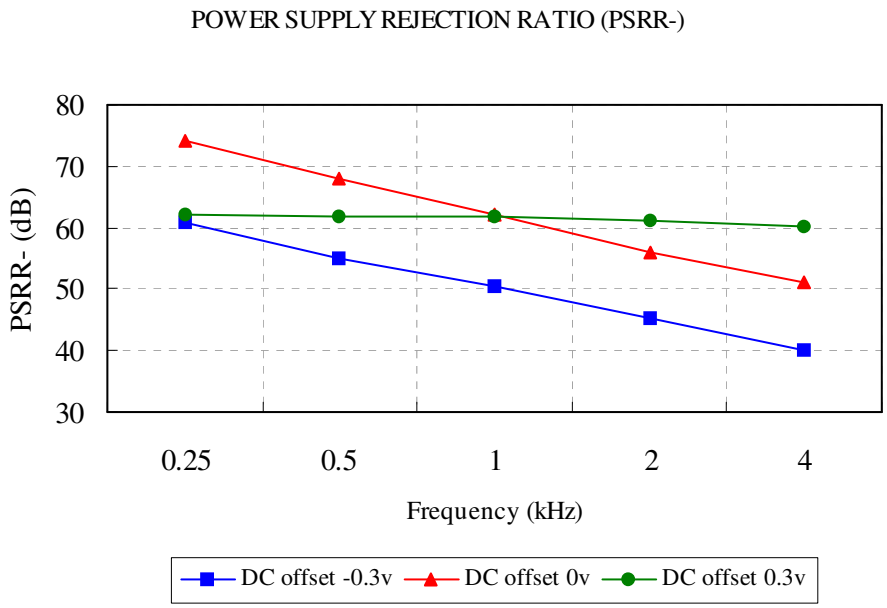

Figure 8 Power Supply Rejection Ratio (PSRR-)

2.4 Layout

For analog and mixed-signal circuits, attention must be paid to layout to ensure a working chip. Improper layout could run the risk of ruining a well designed chip.

The digital control circuit is laid out in using automatic place and route software. A good floorplan minimizes the length of wires in the digital circuits which may cause clock skew problems. Adding inverter buffers prevent clock skew from causing incorrect operation. A clock tree is also used to distribute clock in the digital circuit for equal propagation delay.

For analog circuits, bias circuits should be placed locally to avoid mismatch and ground loop. Instead of using the poly resistor cell provided in the library, customized poly resistors are used in the layout for two reasons. Firstly, the library cell only allows a

maximum resistance value of 4.9 KΩ. It is inconvenient and inaccurate to use many library cells in series to layout the several hundred kΩ resistor employed in the design. Secondly, with a customized poly resistor cells, larger dimensions can be used for better matching and linearity.

For low noise design, thermal noise from the resistive poly gate and the thermal noise due to the resistive substrate can become important. FETs with many fingers is commonly used for FETs with large W/L ratio for smaller bulk-source and bulk-drain capacitances and for laying out common-centroid structures. Analysis shows that in order to minimize the poly gate thermal noise, the number of fingers should be as large as possible. In the sensor amplifier layout, for example, one input differential transistor in the first gain stage op-amp has 216 figures and are laid out in the common-centroid structure.

A previous version of this sensor amplifier IC suffered from crosstalk problem due to a shared ground bus. In this version, a separate ground trace and ground pad are used for each channel. These ground pads are wire-bounded to the package's ground plane. Enough substrate contacts and N-well contacts should be added to avoid crosstalk between transistors. Substrate/N-well contacts and transistors should be placed as close to their supply rail as possible.

2.5 Measurement

2.5.1 Measurement system

The measurement system includes a computer, an FPGA board, a PCB board containing a pre-amplifier and a parallel 16-bit ADC, and the sensor amplifier chip.

Figure 9 shows a block diagram of the measurement system and the data flow. Figure 10 shows a photograph of the measurment setup.

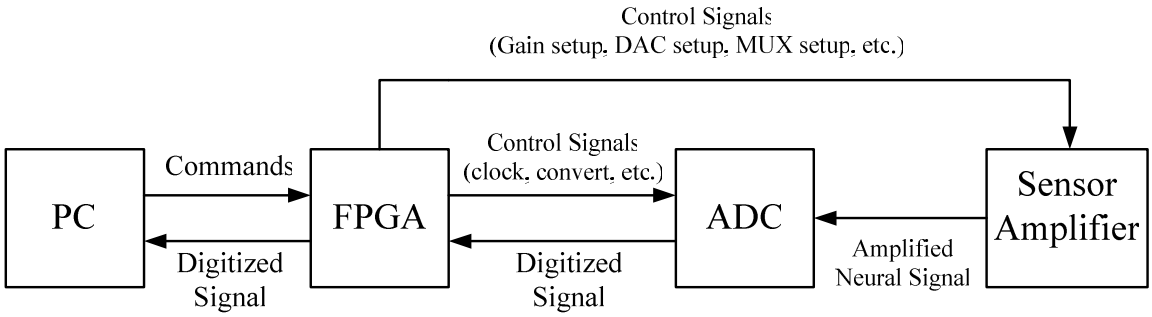

Figure 9 Measurement System

The computer program sends out commands that control the sensor amplifier and the ADC to the FPGA through the USB port. The FPGA takes in the commands, translates the commands into control signals and sends the signals to the sensor amplifier and the ADC. Measured data are read into the FPGA through its I/O, stored in the on-board block-RAM memory unit and read into the PC through the USB port. Since the output swing of the sensor amplifier is +/- 0.8 V and the reference voltage of the 16-bit ADC is 2.5 V, an op-amp with a gain of 3 is needed between the output of the sensor amplifier and the ADC to make use of the full range of the ADC.

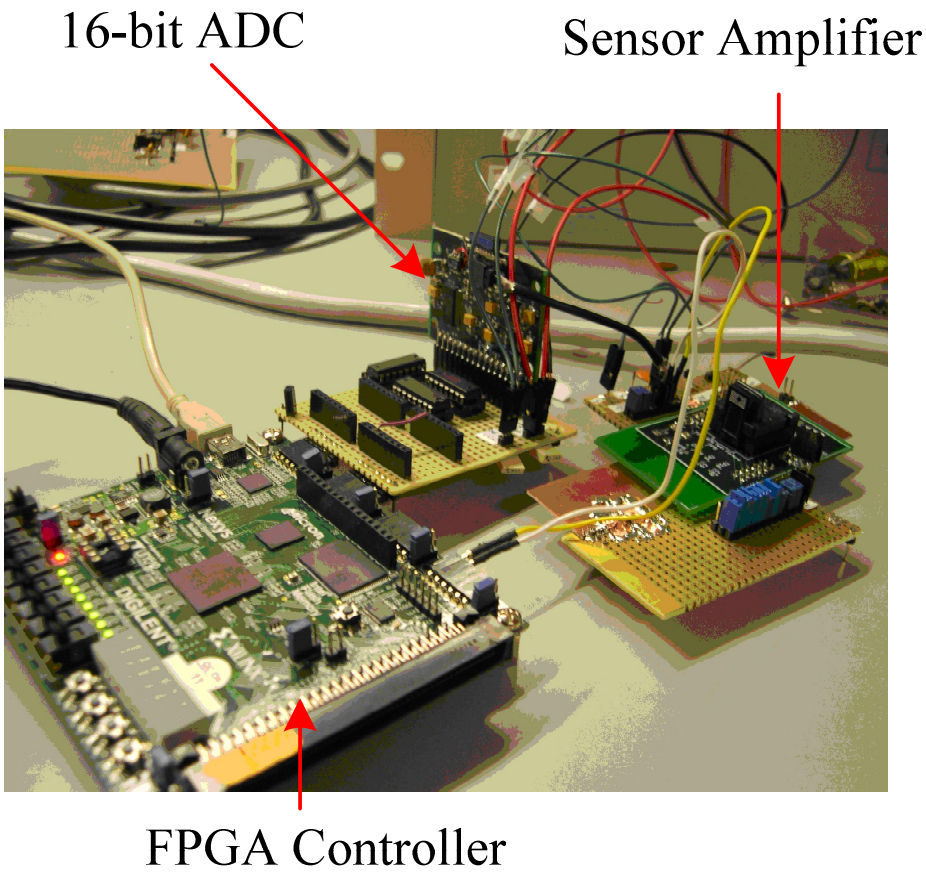

Figure 10 Measurement System

Figure 11 shows the PCB board containing the pre-amplifier and the 16-bit ADC. A long coaxial cable connecting the output of the sensor amplifier to the input of the amplifier on the PCB board causes the amplifier to oscillate. In the measurement system, coaxial cables should be used for minimal noise coupling. However, the length of the coaxial cable should be made as short as possible to minimize coupling and to avoid possible oscillation.

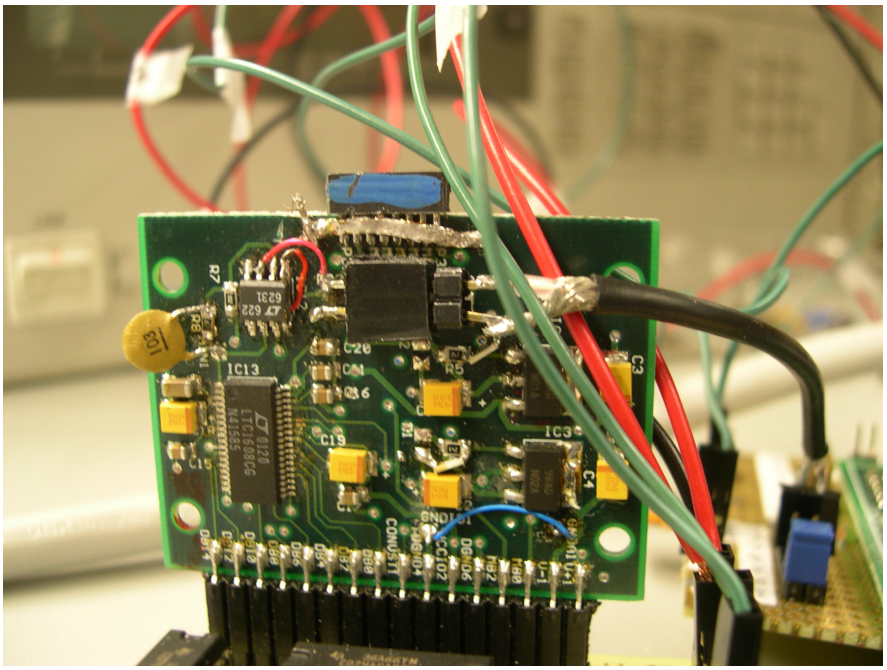

Figure 11 Pre-amplifier and ADC

Figure 12 shows the packaged sensor amplifier chip mounted in a socket.

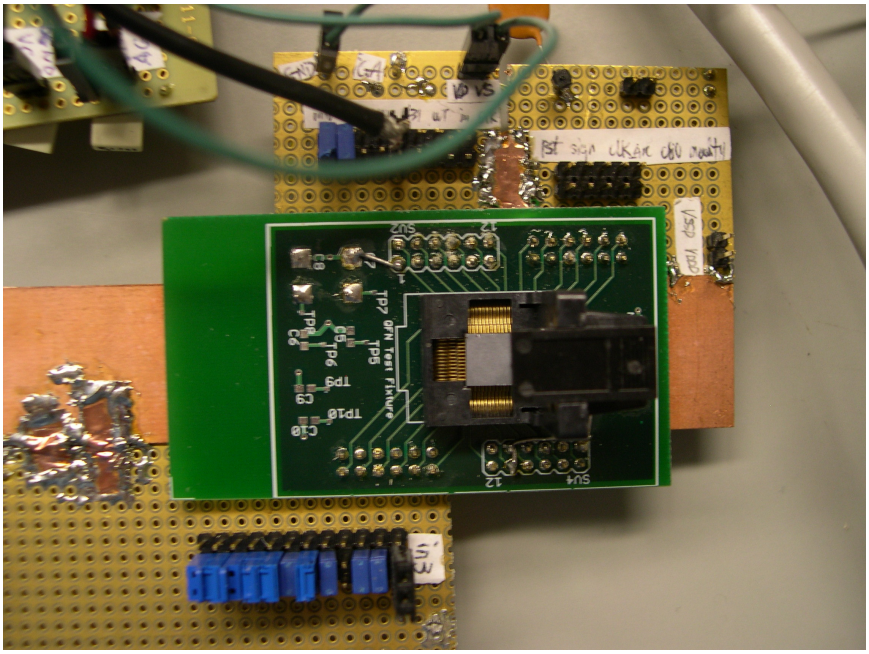

Figure 12 Sensor Amplifier Chip

2.5.2 Power Dissipation

With a power supply of +/- 1.5V, the total current drawn from the power supply by the sensor amplifier is 5.7 mA. The bandgap and DACs shared by all 16 channels consume 300 μA current. The power consumption is 1 mW for each amplifier channel.

2.5.3 Programmable Gain

Poly resistors are used because of their high sheet resistance and good matching. Custom poly resistor cells are made in the layout so that the poly resistor has larger length and width than the standard library cell for better matching. Table 5 shows that the measured programmable gain is very accurate.

Table 5 Programmable Gain

Ideal Gain	Measured Gain
5	5.07
20	19.93
25	25.02
62.5	61.38
100	100.05
250	250.1

2.5.4 Noise

Noise of the sensor amplifier IC working at different input DC offset cancellations and channel gains were measured. Table 6 shows the measured input referred noise with different channel gains. The measured value is smaller than the simulated value, because the noise model provided by TSMC overestimates the low frequency noise [10].

Table 6 Noise v.s. Programmable (channel gain is set to 250)

Programmable Gain	Input Referred Noise (uV)
5	6.29
20	3.15
25	3.8
62.5	3.43
100	2.7
250	2.1

The bandgap reference and the DACs inject more noise into the channel as the input DC offset voltage gets larger. Two off-chip capacitors (100 nF), are used between the reference voltages and the ground to lower the noise. Measurements show that with the off-chip capacitors employed, noise remains relatively constant. Table 7 shows the measured input referred noise with different DC offset cancellations using a channel gain of 250.

Table 7 Noise v.s. Input DC Offset (channel gain is set to 250)

Input DC Offse (V)	Measured Input Referred Noi (μV)	Simulated Input Referred Noi (μV)
- 0.3	2.31	3.63

- 0.2	2.23	3.46
- 0.1	2.18	3.22
0	2.1	3
0.1	2.14	3.16
0.2	2.20	3.14
0.3	2.23	3.2

2.5.5 Power Supply Rejection Ratio

Figure 13 and Figure 14 show the measured power supply rejection ratio, which agrees with the simulations. The deviation between measured PSRR and simulated PSRR is within 6 dB.

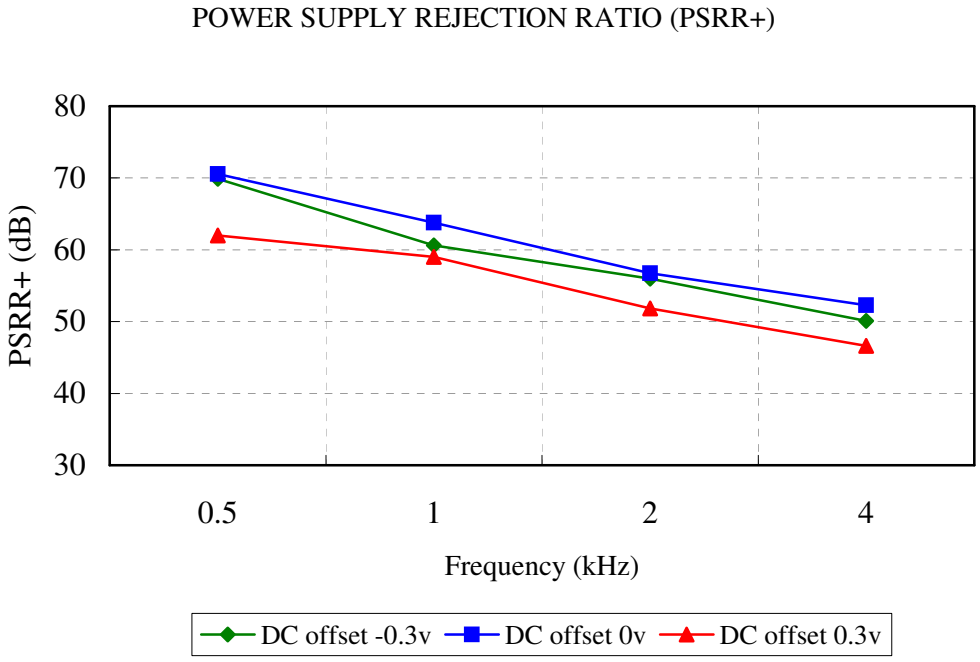

Figure 13 Power Supply Rejection Ratio (PSRR+)

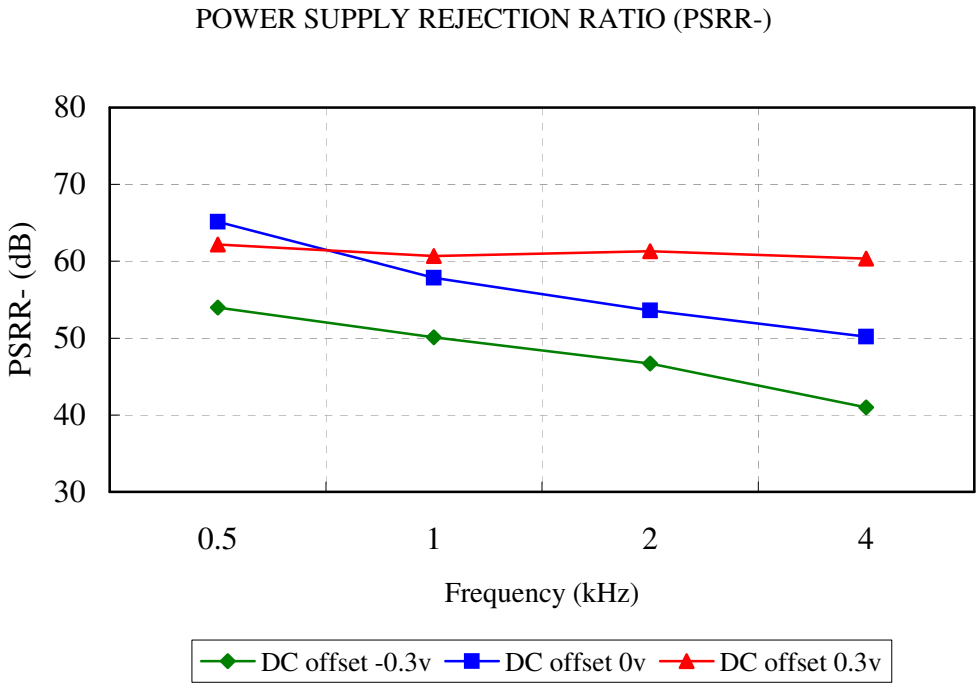

Figure 14 Power Supply Rejection Ratio (PSRR-)

2.5.6 Spurious Free Dynamic Range

To measure the spurious free dynamic range (SFDR) of the sensor amplifier IC, an HP 3325A Source/Function Generator is used to provide a sine wave signal as the input to the sensor amplifier. However, the HP3325A only has an SFDR of 76 dB by itself. Notch filters [6] are used to dampen the 2^{nd} and 3^{rd} harmonics of the HP3325A's output sine wave signal to provide a "spurious free" source to the sensor amplifier. Figure 15 shows that a source SFDR of 91dB is achieved by using the HP3325A and notch filters. Figure 16 shows the SFDR of the sensor amplifier is 81 dB with an input sine wave signal at 1.5 kHz.

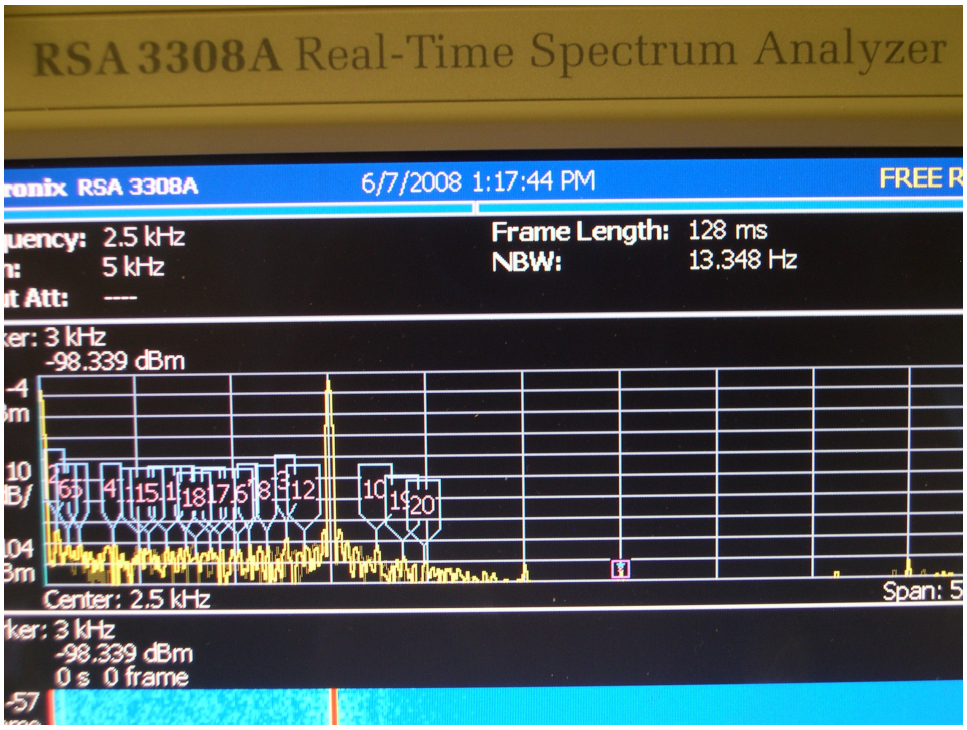

Figure 15 SFDR of the Source Generator with Notch filters

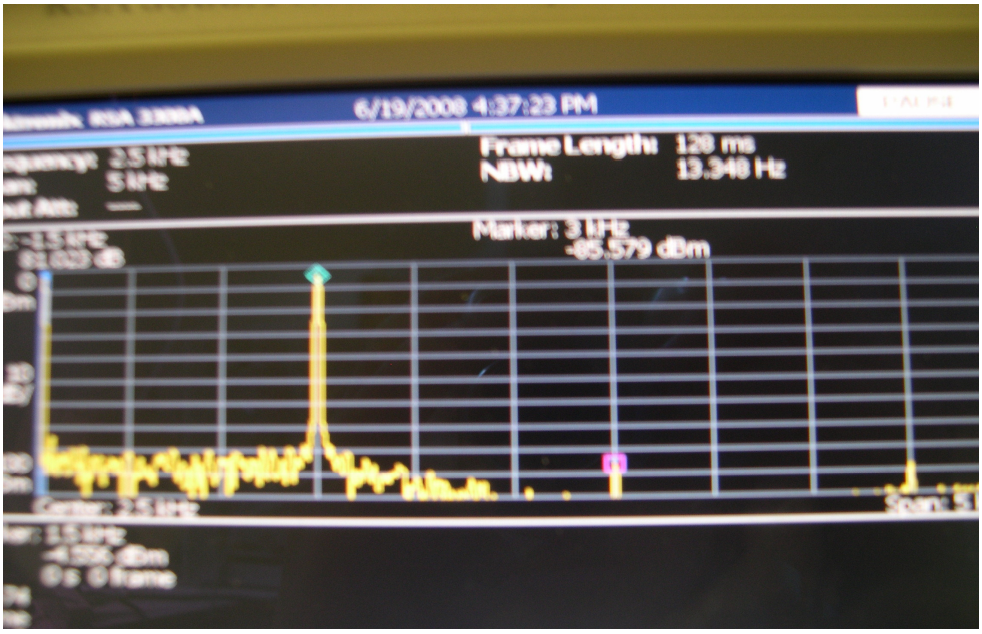

Figure 16 SFDR of the Sensor Amplifier IC

2.5.7 DAC Linearity

Differential Nonlinearity (DNL) and Integral Nonlinearity (INL) of the two 5-bit DACs were measured. Figure 17 plots the measured DNL of the first DAC. Figure 18 shows the INL of the first DAC. The measurements of the second DAC's INL and DNL are shown in Figure 19 and Figure 20. The measured INL and DNL are within 0.05 LSB. The linearity of the resistor-string DAC is very good because the poly resistors are very well matched.

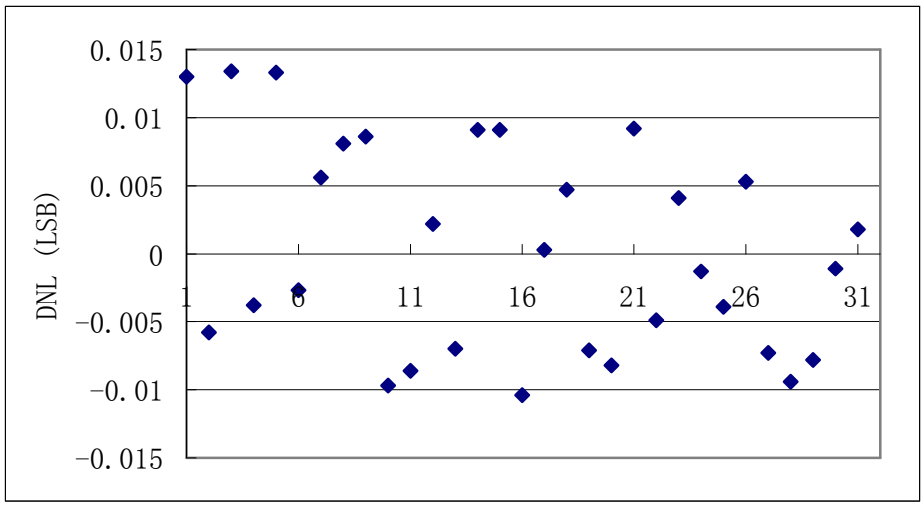

Figure 17 First DAC DNL

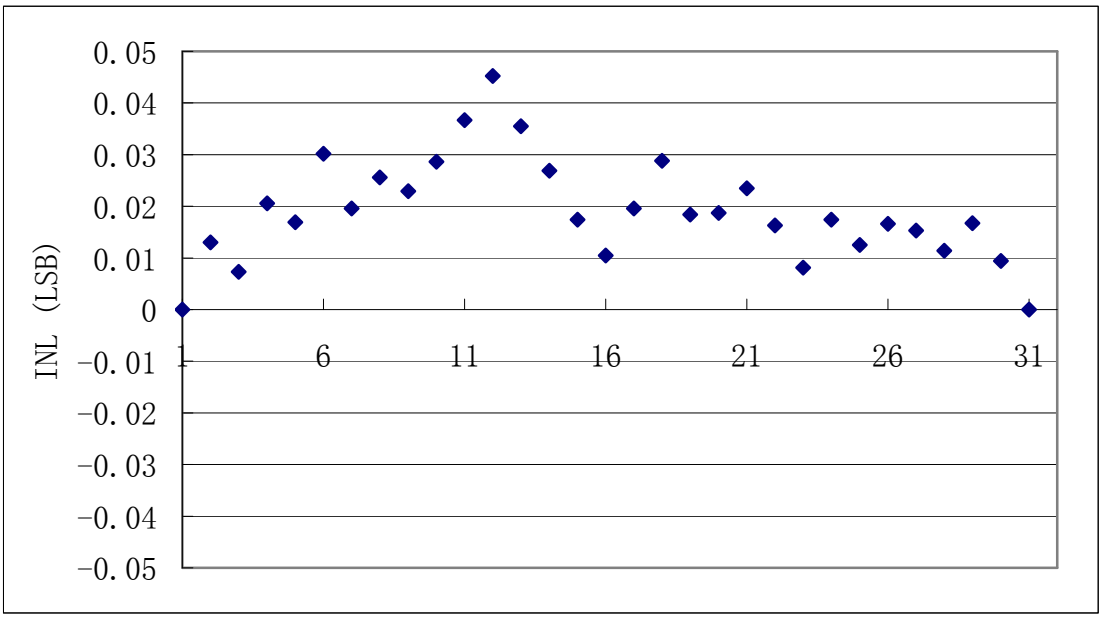

Figure 18 First DAC INL

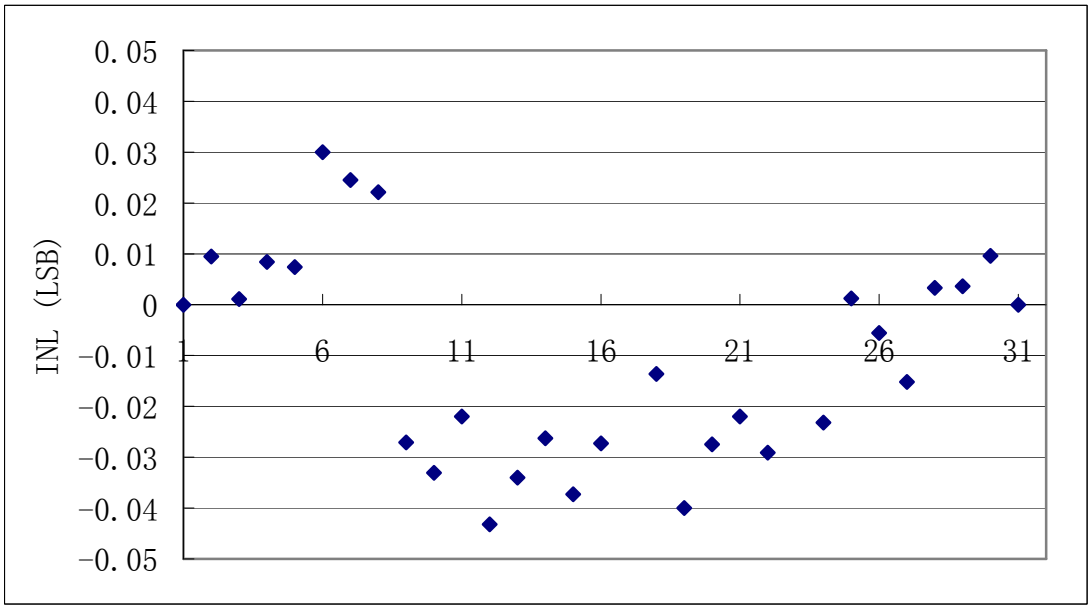

Figure 19 Second DAC INL

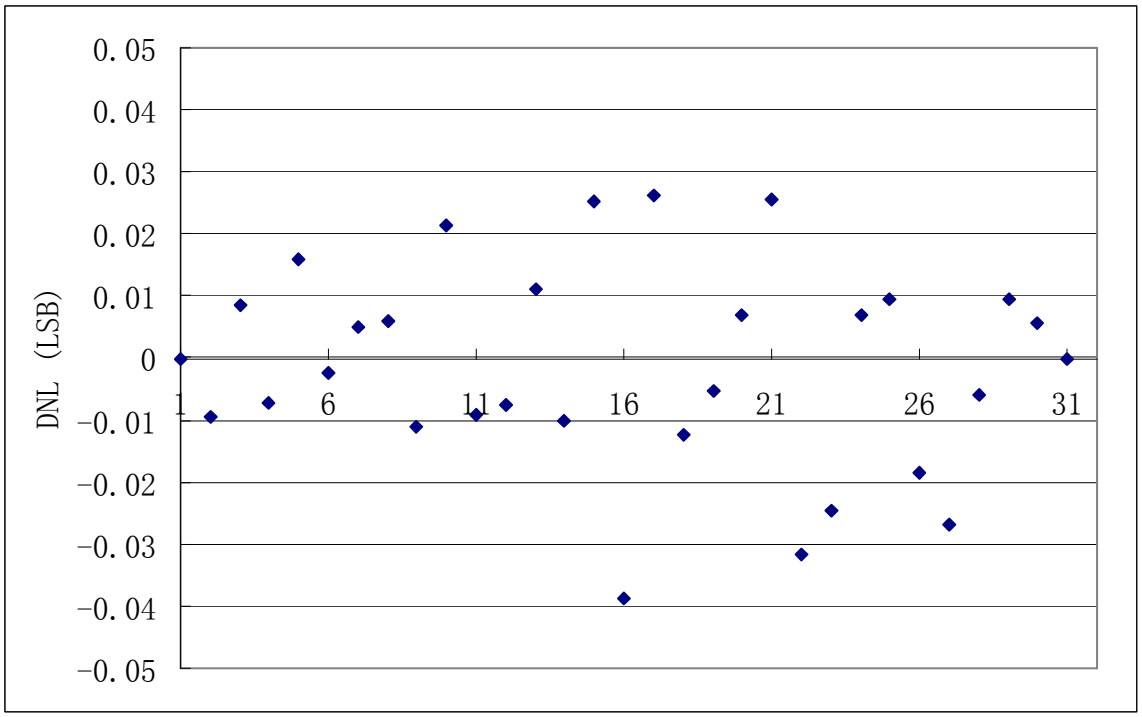

Figure 20 Second DAC DNL

2.5.8 DAC Long Term Stability

Figure 21 and Figure 22 shows the long term stability of the two 5-bit DACs. The DAC's voltages are slightly time-varying because of temperature variations. The bandgap reference that drives the DAC has small but non-zero temperature coefficients. Heat was applied to the IC and the DAC's variation correlated well reference voltage variation,

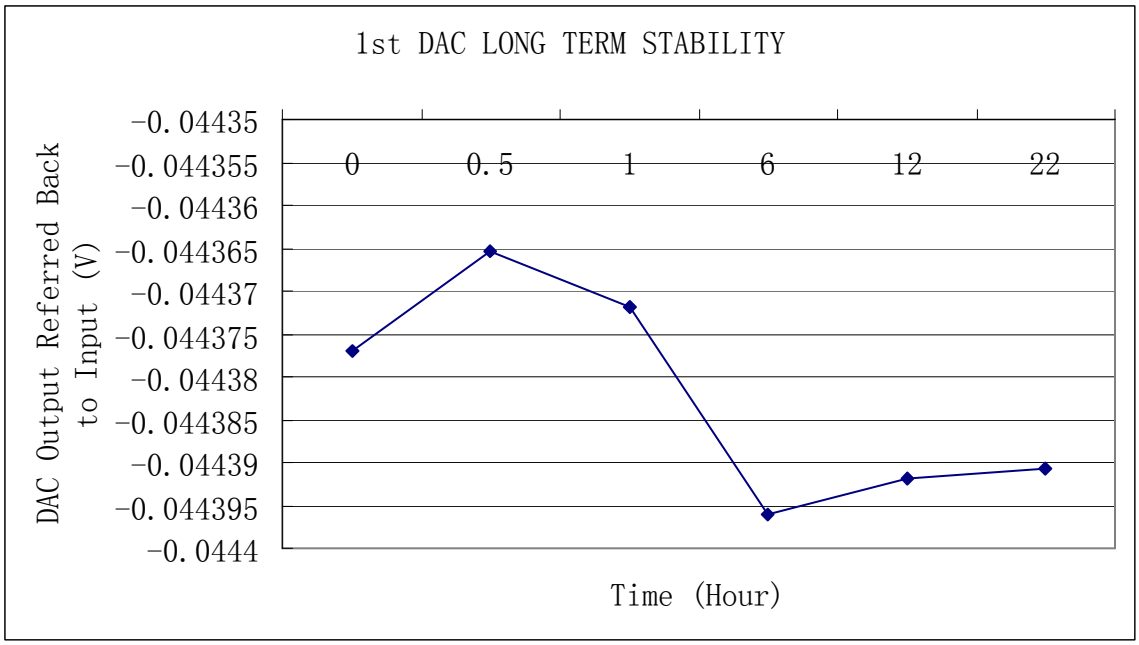

Figure 21 First DAC long term stability

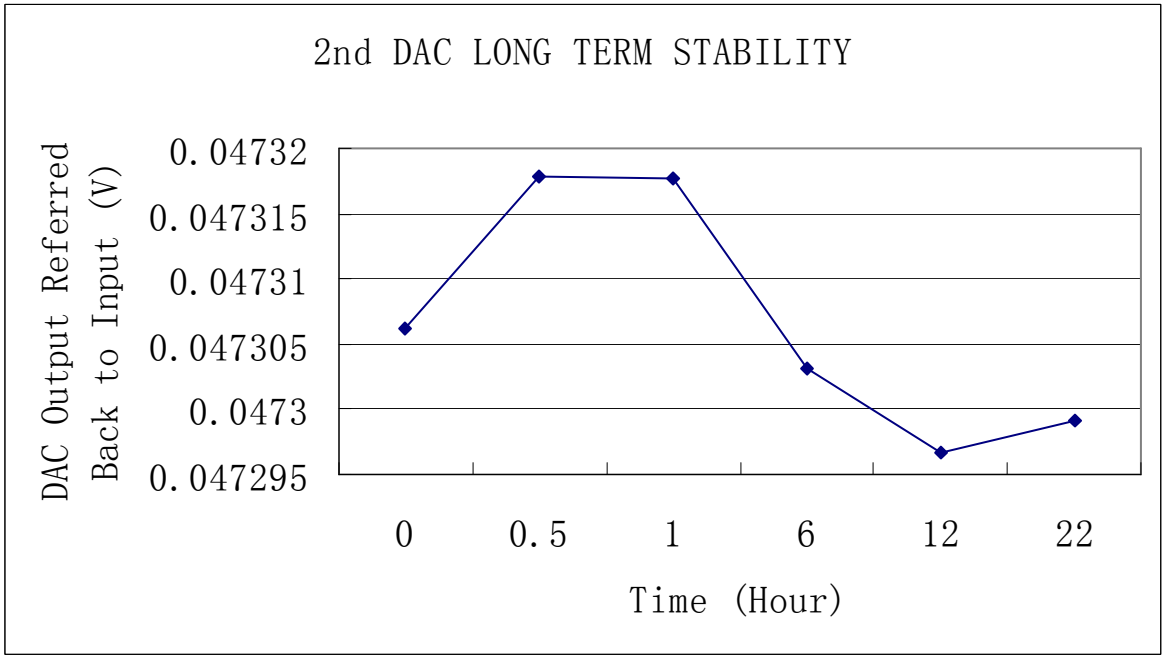

Figure 22 Second DAC long term stability

2.5.8 Crosstalk between channels

The previous version sensor amplifier suffers from crosstalk problem because of the layout of shared ground between channels. For this sensor amplifier IC, separate ground pads are used for each channel, which greatly reduces the crosstalk between channels. To measure the crosstalk, a 5 mV peak-to-peak signal is applied to the input of one channel whose gain is set to 250. The inputs of the other channels are all grounded. Gains of the other channels are also set to 250. Measurements show a crosstalk less than -90 dB for each channel.

2.6 Conclusion

A 16-channel low-power low-noise high-accuracy DC-coupled sensor amplifier IC is designed, fabricated and measured. Measured specifications indicate the sensor amplifier meets the requirements for a high precision neural signal recording system. The performance specifications of the sensor amplifier IC are summarized in Table 7.

Table 8 Performance summary of the amplifier channel IC

	Simulation	Measurement
Process	TSMC 0.25µ	TSMC 0.25µ
Die area	15 mm^2	15 mm^2
Power supply	+/- 1.5 V	+/- 1.5 V
Power dissipation	1 mW per channel	1 mW per channel
Signal Bandwidth	DC ~ 7 kHz	DC ~ 7 kHz

DC offset cancellati range	+0.3 V ~ -0.3 V	+0.3 V ~ -0.3 V
Output swing	+1 V ~ -1 V	+1 V ~ -1 V
Programmable gai	5~250	5.07~250.1
Input referred noise v a channel gain of 250 a an input DC offset 0.	3.2 μV	2.23 μV
Input referred noise v a channel gain of 250 a an input DC offset -0.3	3.63 μV	2.31 μV
Input referred noise v a channel gain of 250 a an input DC offset 0 V	3 μV	2.1 μV
PSRR+	> 59 db at 500 Hz	> 59 db at 1k Hz
PSRR-	> 51 db at 1k Hz	> 50 db at 1k Hz
SFDR	> 80 dB over the bandwi	81 dB @ 1.5 kHz

CHAPTER THREE

3. IMPROVED SENSOR AMPLIFIER IC

3.1 Introduction

A modified version of the sensor amplifier IC is investigated, simulated and laid out. The same functions (programmable gain, DC offset cancellation, etc.) are achieved while noise, power dissipation and die area are significantly reduced. A chopping amplifier is employed to reduce flicker noise. A DC offset cancellation scheme controlling the bulk voltages of the input differential pair of the op-amp is proposed. The simulated input referred noise integrated from 1 Hz to 7 KHz is reduced to 1.7 μV at a channel gain of 250. The supply voltage is lowered to +/- 1.2 V to save power without sacrificing output swing. The power dissipation is reduced to 0.34 mW per channel. Since the supply voltage is lowered, 2.5 V MOSFETs are employed instead of the 3.3 V MOSFETs used in the previous designs. Because 2.5 V MOSFETs have thinner gate oxide, their flicker noise is considerable smaller than the flicker noise of the 3.3 V MOSFETs with the same size. The DC offset cancellation range is from -0.3 V to + 0.3 V. The programmable gain is in the range from 5 to 250.

3.2 Chopping Amplifier

As IC design technology keeps pursuing lower power consumption and lower power supply voltage, op-amp's input dc offset voltage and noise start to limit the dynamic range and resolution of the circuits. Some techniques have been developed to reduce the dc offset and the low-frequency noise of the op-amp. There are two main categories:

autozeroing, which is a sampling technique and chopper stabilization, which is a modulation technique.

The idea of autozeroing is to sample the noise and offset and then subtract them from the distorted signal. The idea of chopper stabilization is to use an input chopper to modulate the signal to higher frequencies (the odd harmonics of the chopping frequency) where there is less flicker noise. The modulated signal is amplified and then demodulated back to baseband by the output chopper. The dc offset and flicker noise are up-converted to higher frequencies (the odd harmonics of the chopping frequency) by the output chopper. A low-pass filter finally removes the out-of-band offset and noise. Figure 23 demonstrates the principle of a chopper stabilization amplifier.

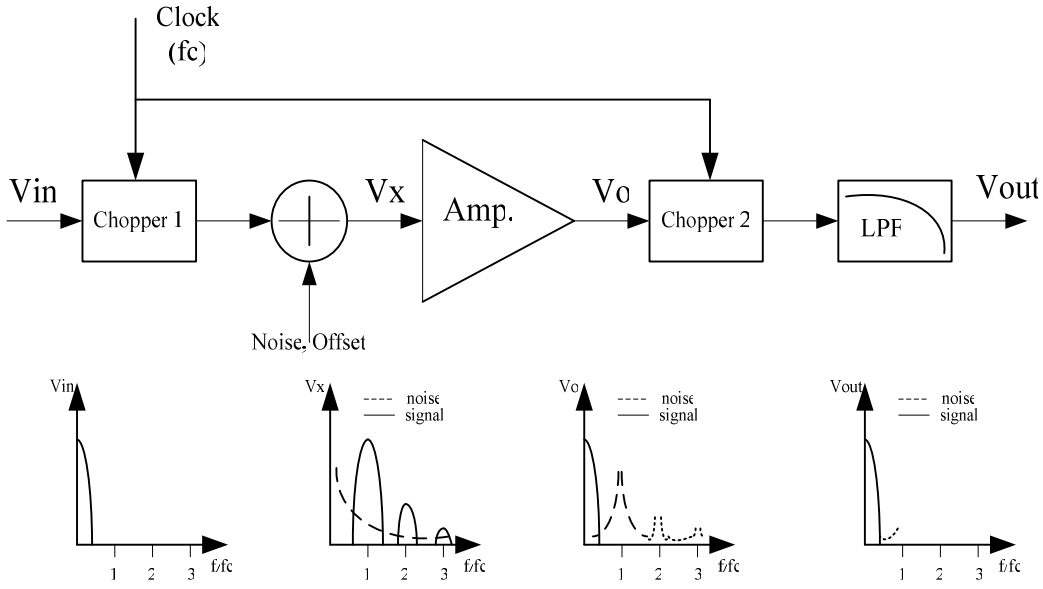

Figure 23 Principle of Chopper Stabilization

The sensor amplifier IC proposed here employs chopper stabilization to reduce flicker

noise. Because of the inherent sampling property of autozeroing, thermal noise is folded back down to the baseband [7]. Therefore, chopper stabilization amplifier has lower thermal noise than an amplifier using the autozeroing technique.

3.3 Amplifier Channel Design

3.3.1 Channel Architecture

Figure 24 shows the block diagram of the amplifier channel. Each channel has two op-amp gain stages, one 9-bit DAC to remove the DC offset within the first gain stage and one 2nd-order low-pass Butterworth filter. The gain of the first gain stage is increased to 10. Programmable gain is realized by closing MOSFET switches in the resistive feedback network of the second gain stage op-amp. Chopper stabilization technique is employed in the first gain stage to reduce flicker noise. The track and hold circuit is removed. The requirement of simultaneous sampling was removed since the application software that conducts neural signal processing can take the sampling time difference between channels into account. A little bit of power and a large amount of area can be saved because the hold capacitor, which needed to be large to reduce kT/C noise, is no longer required. The signal is routed by the multiplexer directly to the ADC.

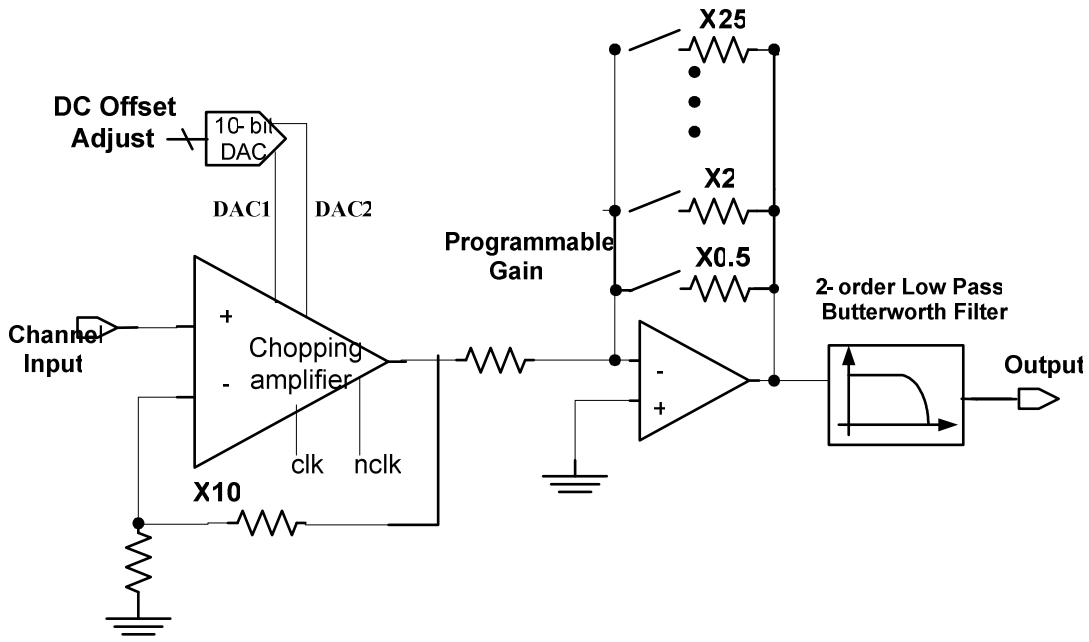

Figure 24 Block diagram of the amplifier channel

3.3.2 First Gain Stage and DC Offset Cancellation

Two major modifications made to the first gain stage are: DC offset cancellation and utilization of the chopper stabilization technique. These two modifications significantly improve the performance of this version of the sensor amplifier IC. In this section, the DC offset cancellation scheme is introduced. Utilization of the chopper stabilization technique will be discussed in section 3.3.3.

To cancel the input DC offset voltage, a 9-bit resistor-string DAC is used to control the bulk voltages of the input differential pair. The body biases on the p-channel input differential pair transistors are individually controlled to cause a threshold difference that cancels any DC offset voltage up to +/- 0.3 V. Since FET threshold voltages are a non-linear function of body bias, the resistor values in the DAC are chosen to linearize the DC offset cancellation. Since the DAC is connected to the body, thermal noise of the

resistor string has less effect on the noise than if the DAC were connected directly to the inputs and somewhat larger resistor values can therefore be used, reducing DAC power dissipation. External capacitors on the reference voltage nodes are still required to lower the noise.

Figure 25 shows the structure of the resistor-string DAC. A 9-bit DAC requires 512 resistors and switches. The DAC proposed here is a two-stage DAC. It significantly reduces the number of resistors and switches needed. Another benefit of canceling the offset at the first stage is that the gain of the first stage can be made larger and then the number of gain stages could be reduced from 3 to 2, resulting in reduced power consumption and die area.

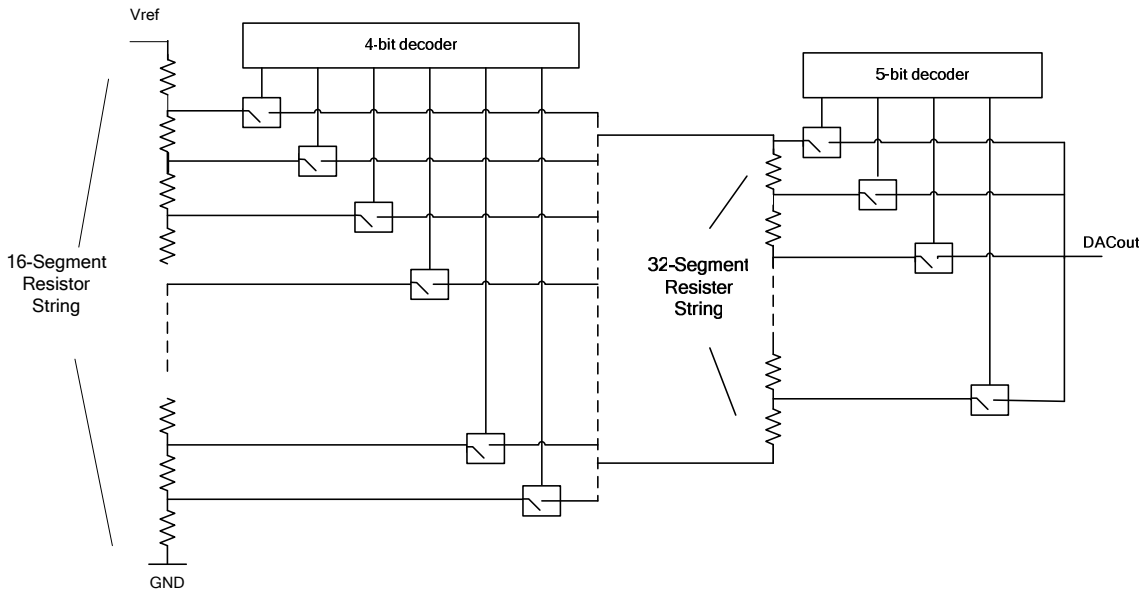

Figure 25 Structure of the 9-bit two-stage resistor-string DAC

Since the DC offset cancellation is accomplished within the first gain stage, the gain of the first stage can now be set to 10, larger than the fixed gain of 2.5 in the previous design. This will make the first gain stage the major noise contributor of the total channel noise. Therefore, reducing first stage's noise, such as utilizing chopper stabilization, is more important in reducing the channel's total noise. Smaller devices and lower currents can be used in the second gain stage and the low pass filter without severely degrading the channel noise. With the DC offset cancelled in the first stage and the gain of the first stage set to 10, to cover the wide programmable gain range, the maximum gain needed from the second stage now is 25. 100 KΩ input resistor is chosen for the second gain stage according to the first stage's driving ability. Then the maximum feedback resistor needed for the second gain stage is 2.5 MΩ, whose noise contribution in the total channel noise is still very small.

3.3.4 First Gain Stage and Chopper Stabilization

A chopping amplifier is employed in the first gain stage op-amp to further reduce flicker noise without resorting to unacceptably large transistors. Figure 26 shows a schematic of the chopper amplifier used in the first stage amplifier.

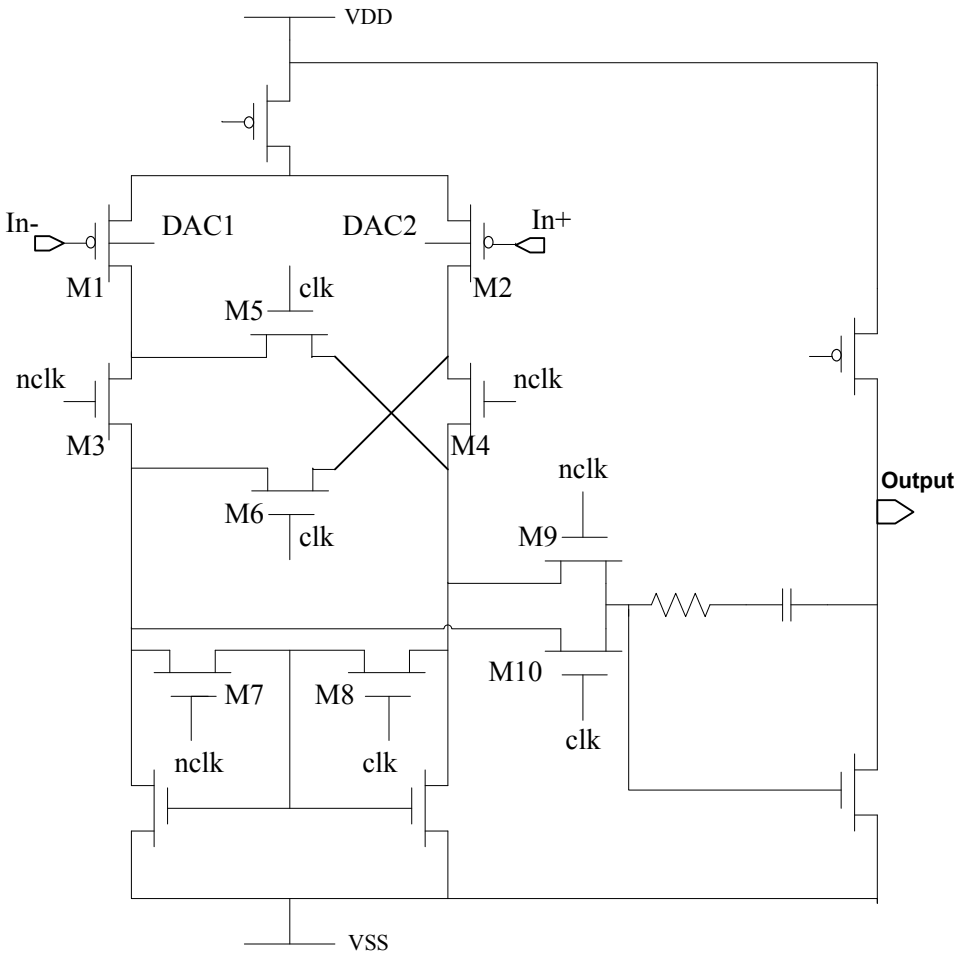

Figure 26 Chopper Op-amp in the first gain stage

Unlike the conventional chopper amplifier which chops at the differential inputs and differential outputs, it chops at the drains of the input pair and has a single-ended output. Because the input is DC coupled and has a DC offset, chopping at the input generates an unacceptably high average dc current. Only extremely small input currents are allowed in this application to measure neurophysiological signals from the brain. Therefore the modulator chopper, implemented by M3 -M6, is placed at the drains of the input pair, while the demodulator (M9 - M10) is placed at the active loads. M7–M8 provides a single-ended output. The limitation of this topology is that flicker noise from only the

active load transistors is reduced. Flicker noise of the input differential transistors is reduced by making them larger. The size of the active load transistors are reduced by 60%.

An advantage of this topology is that by demodulating at low impedance nodes the effect of spikes on the output signal is reduced and there is no need to compensate for the phase difference between the modulator and demodulator. For a conventional chopper amplifier, there is a non-negligible phase delay between the modulator and the demodulator introduced by the op-amp. To ensure the largest dc gain from the chopper amplifier, the phase difference between the modulator and demodulator clocks need to match exactly with this delay [8].

The chopping frequency should be at least twice the highest signal frequency to avoid signal aliasing. The bandwidth of the op-amp should be ten times larger than the chopping frequency to allow enough settling time and so that the power spectral density of the white noise remains unchanged [8]. Considering these factors, a chopping frequency of 30 kHz is used.

3.3 Simulations

3.3.1 Periodic Steady State Analysis

Because the chopping amplifier has no DC operating point, SpectreRF simulations are required to analysis the noise property of the chopping amplifier [9]. Periodic Steady State (PSS) analysis linearizes the circuit around its periodic operating point. They are carried out based on the operating point calculated from the PSS analysis.

3.3.2 DC Offset Cancellation and Programmable Gain

The amplifier channel was designed in the TSMC 0.25 μm process with power supply voltages of +/- 1.2 V. Simulations shows a DC offset cancellation range of +/- 0.3 V and a programmable gain from 5 to 250. Figure 27 shows a 1 kHz sine wave input signal with a DC offset of 0.3 V and amplitude of 10 mV in the solid curve. The dashed curve is the simulated output signal with a gain of 50 and shows that the DC offset has been removed by controlling the threshold voltages of the input transistor pair.

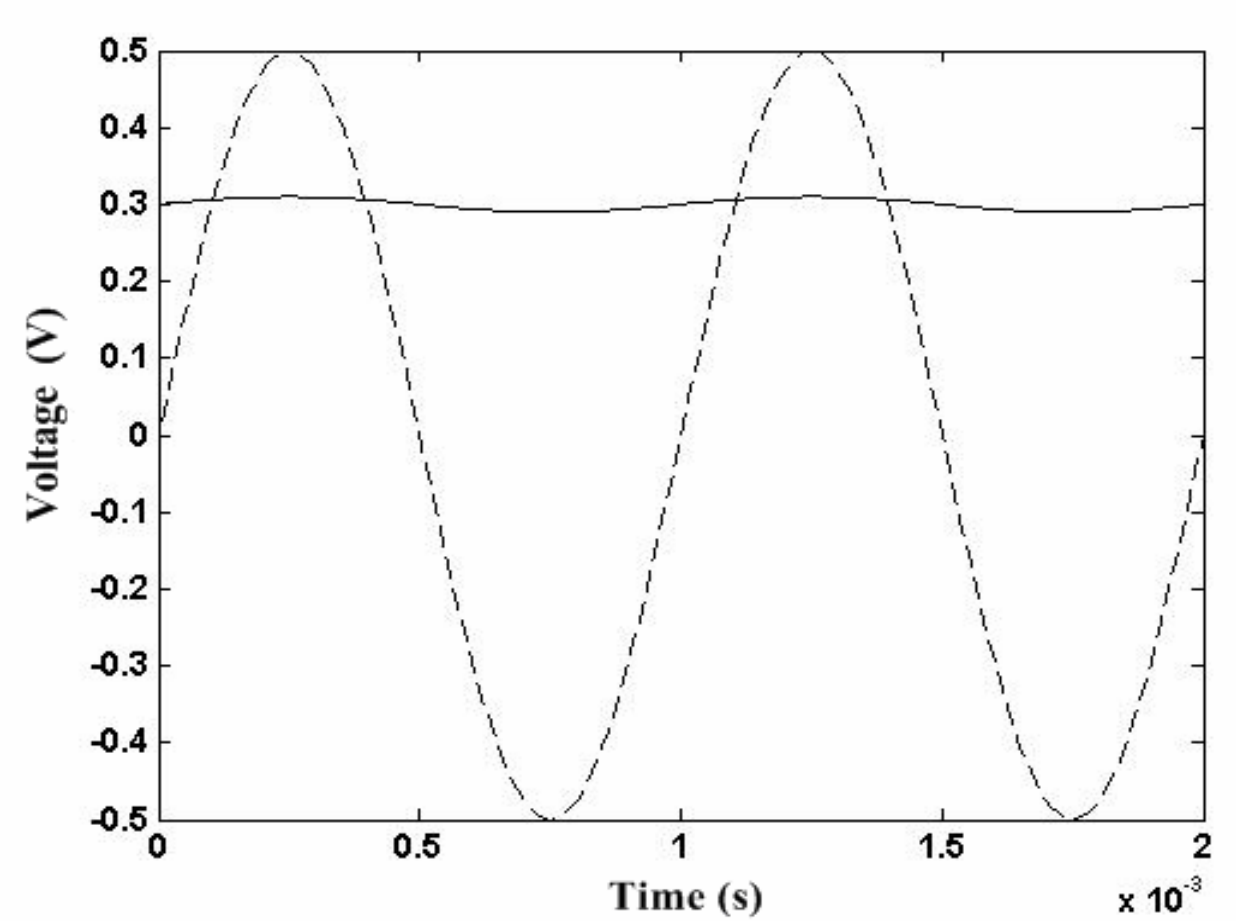

Figure 27 Simulation result of DC offset cancellation and Gain

3.3.3 Noise

Simulation shows an input referred noise integrated from 1 Hz to 7 KHz of 1.7 μV with a gain of 250. Figure 28 shows the input referred noise spectrum.

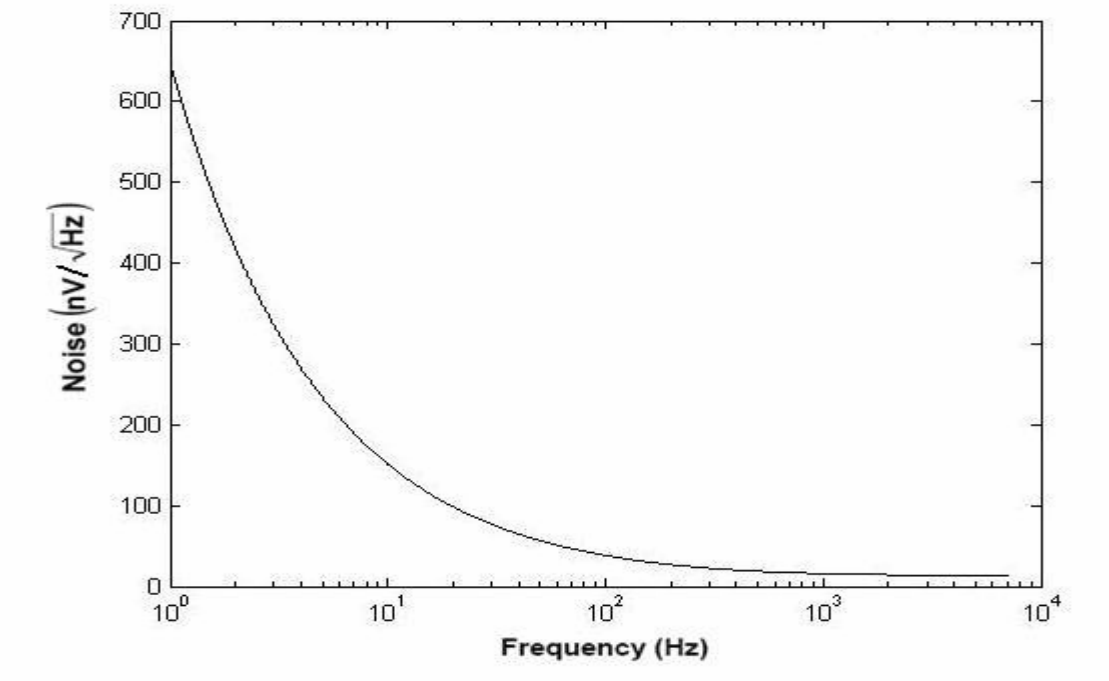

Figure 28 Frequency spectrum of the input referred noise of the sensor amplifier with no DC offset and a channel gain of 250

Because the noise model provided by the foundry significantly overestimates the noise as discussed in Chapter 2, the actual measured noise should be smaller than the simulated noise. Table 8 and Table 9 summarize the simulated noise of the sensor amplifier working with different gains and different input DC offset cancellation.

3.3.4 PSRR

Figure 29 and Figure 30 summarize the simulated power supply rejection ratios of the sensor amplifier working with different input DC offset. The simulated PSRR of the modified sensor amplifier is about 5 dB lower than the simulated PSRR of the previous version.

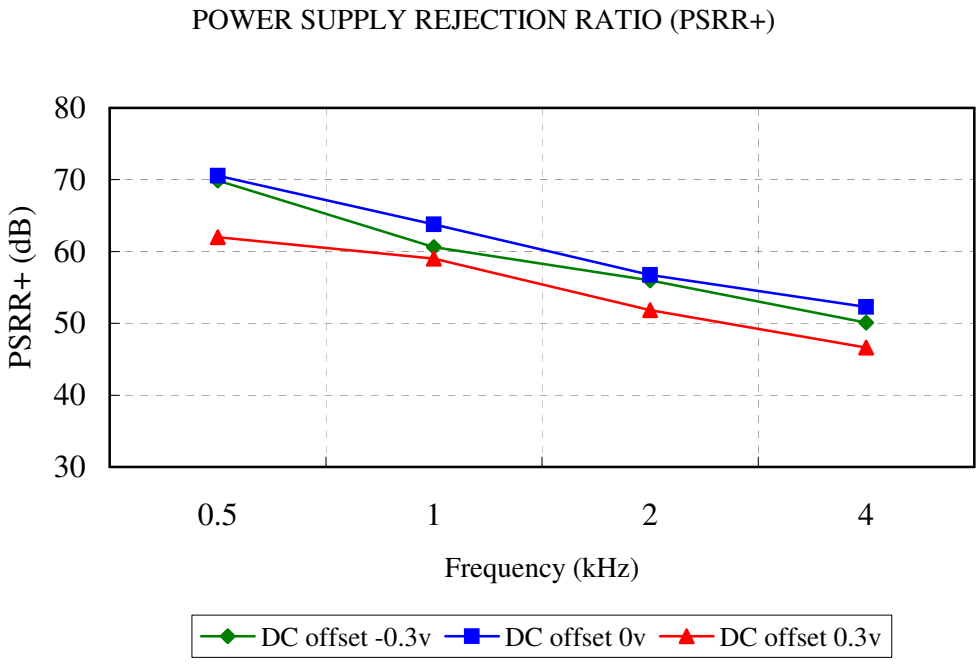

Figure 29 Power Supply Rejection Ratio (PSRR+)

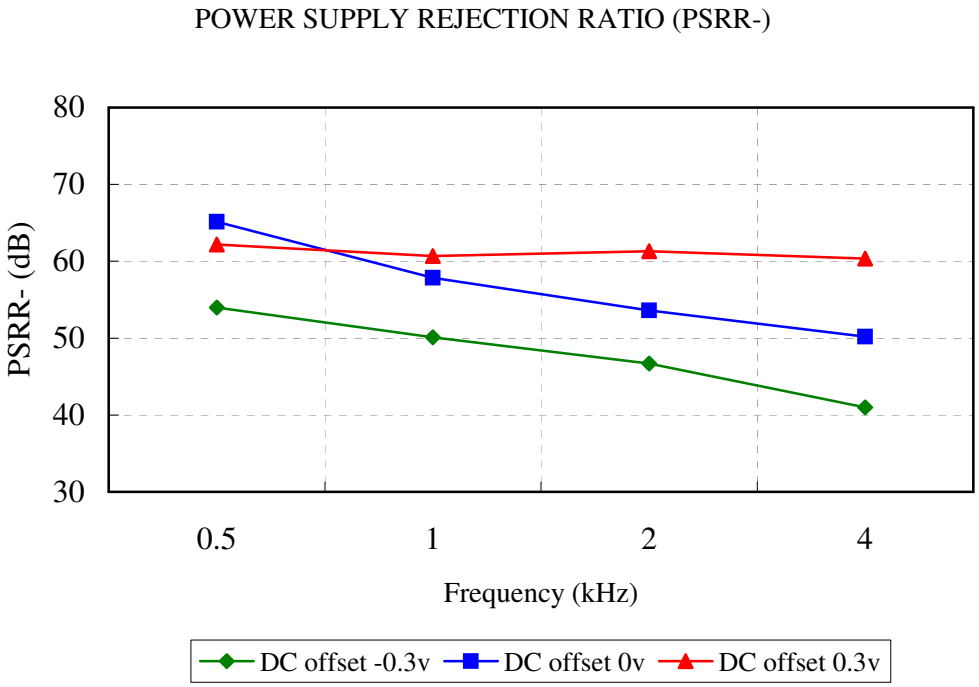

Figure 30 Power Supply Rejection Ratio (PSRR-)

CHAPTER FOUR

4. CONCLUSION

A 16-channel sensor amplifier IC is designed and fabricated in TSMC's 0.25μ CMOS process. Measurement results show that it has a power dissipation of 1 mW per channel, programmable gains from 5 to 250, an input referred noise of 2.1 μV with no DC offset and a channel gain of 250. PSRR+ is 59 dB at 1 kHz, and PSRR- is 50 dB at 1 kHz. The input DC offset up to +/- 0.3V is cancelled with two 5-bits DACs controlling the positive input nodes of the second and third gain stages op-amps. The crosstalk among channels is less than -90 dB.

Another 16-channel sensor amplifier IC is designed, simulated and laid out in TSMC's 0.25μ CMOS process. The input DC offset is cancelled by controlling the bulk voltages of the input differential pair of the first gain stage op-amp. A chopping amplifier is employed in the first gain stage op-amp to reduce its flicker noise. Power supply voltage is lowered to +/- 1.2 V, therefore the 3.3 V MOSFETs are replaced by the 2.5 V MOSFETs, which further decreases the channel's noise. Since the input DC offset voltage is cancelled in the first gain stage, the gain of the first stage can be set to 10. Therefore only two gain stages are needed to cover the programmable gain range, resulting in further reduction in power dissipation, noise and die area. Simulations show the power dissipation is reduced to 0.34 mW per channel. The input referred noise of the channel working with no DC offset and a channel gain of 250 is only 1.7μV. The die area shrinks to 2x3 mm^2.

This sensor amplifier IC can be used in many biomedical applications where low-power low-noise high-accuracy sensor system is required. Because of this sensor amplifier is DC coupled, it's especially suitable for applications where very low frequency signal is desirable.

BIBLIOGRAPHY

References

1. 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.
2. M. S. J. Steyaert, W. M. C. Sansen, C. Zhongyuan, "A micropower low-noise monolithic instrumentation amplifier for medical purposes", IEEE J. Solid-State Circuits, Vol. SC-22, pp. 1163 – 1168, Dec. 1987.
3. M. Dagtekinn, W. Liu, R. Bashirullah, "A multichannel chopper modulated neural recording system", in Proc. IEEE EMBS Int. Conf., pp. 757 – 760, 2001.
4. R. R. Harrison, C. Charles, "A Low-Power Low-Noise CMOS Amplifier for Neural Recording Applications", IEEE J. of Solid-State Circuits, Vol. 38, no. 6, pp. 958 – 965, June 2003.
5. "Analog Filter Design Demystified", Maxim
6. "High Q Notch Filter", National Semiconductor
7. Christian C. Enz, Gabor C. Temes, "Circuit Techniques for Reducing the Effects of Op-amp Imperfections: Autozeroing, Correlated Double Sampling, and Chopper Stabilization", PROCEEDINGS OF THE IEEE, Vol. 84, no. 11, pp. 1584 – 1614, November 1996.
8. C. C. Enz, E. A. Vittoz, R. Krummenacher, "A CMOS Chopper Amplifier", IEEE J. of Solid-State Circuits, Vol. SC-22, no. 3, June 1987.

9. Ken Kundert, “Simulating Switched-Capacitor Filters in SpectreRF”, Designer’s Guide Consulting, Inc. Version 6c, 28 July, 2006.
10. TSMC 0.25UM MIXED SIGNAL 2P5M or 1P5M+ SALICIDE 2.5V/3.3V NOISE MODEL

APPENDIX

Verilog Code for the digital control circuit of the sensor amplifier

```
`timescale 1 ns/ 10 ps

module cntlfsm
    (rdloop,strobeon,reset,data,addr,clock,st,cmdbits,dacf,dacs,gain,loc
k,rdloop,dacf_modi,dacs_modi,offset,strobe,clk);

    input  reset,clock,st,strobeon;
    input  [1:0]  cmdbits;
    input   [4:0]   data;
    output dacf,dacs,gain,lock,rdloop,clk;
    output  [2:0] offset;
    output  [6:0] strobe;
    output  [4:0] dacf_modi,dacs_modi;
    input [3:0] addr;
    reg clkadc,clk;
    reg [3:0] Mreg;
    reg  [2:0] offset;
    reg  [6:0] strobe,temp;
    reg  [4:0] dacf_modi,dacs_modi;

// [MSB:LSB]
    parameter [4:0]
        idle =     5'b00000,
        start =       5'b00001,
        cmd1tell =        5'b00011,
        cmddacf =     5'b00010,
        cmd3 =     5'b00110,
        cmd4 =     5'b00111,
        cmd5 =     5'b00101,
        cmd6 =     5'b00100,
        cmd7 =     5'b01100,
        cmd8 =     5'b01101,
        cmd9 =     5'b01111,
        cmd10 =       5'b01110,
        cmdend =      5'b01010,
        cmdread =     5'b01011,
        cmdgain =     5'b01001,
        cmdread1 =        5'b01000,
        cmddacs =     5'b11000,
        cmddacs1 =        5'b11100,
        cmddacf1 =        5'b11110,
        cmdgain1 =        5'b11111;

    parameter [1:0]
        casedacf = 2'b11,
        casedacs = 2'b00,
        casegain = 2'b10,
        caseread = 2'b01;

    reg [4:0]  CurrentState, NextState;
```

```
    reg dacf,dacs,gain,lock,rdloop;

// fsm machine

    always@(CurrentState  or st or cmdbits)
    begin
        case(CurrentState)

        idle:
            begin
            dacf<=0; dacs<=0; gain<=0; lock<=0; rdloop<=0;
            if(st) NextState <=  start;
            else   NextState <=  idle;
            end

        start:
            begin
            dacf<=0; dacs<=0; gain<=0; lock<=1; rdloop<=0;
            NextState <=  cmd1tell;
            end

        cmd1tell:
            begin
            dacf<=0; dacs<=0; gain<=0; lock<=0; rdloop<=0;
            case(cmdbits)
            caseread:  NextState <=  cmdread;
            casedacf:  NextState <=  cmddacf;
            casedacs:  NextState <=  cmddacs;
            casegain:  NextState <=  cmdgain;
            endcase

            end
        cmdread:
            begin
                dacf<=0; dacs<=0; gain<=0; lock<=0; rdloop<=1;
                NextState <=  cmdread1;
            end
        cmdread1:
            begin
                            dacf<=0; dacs<=0; gain<=0; lock<=0; rdloop<=0;
                NextState <=  cmd3;
            end
        cmddacf:
            begin
                dacf<=1; dacs<=0; gain<=0; lock<=0; rdloop<=0;
                NextState <=  cmddacf1;
            end
        cmddacf1:
            begin
                dacf<=1; dacs<=0; gain<=0; lock<=0; rdloop<=0;
                NextState <=  cmd3;
            end
        cmddacs:
            begin
                dacf<=0; dacs<=1; gain<=0; lock<=0; rdloop<=0;
                NextState <=  cmddacs1;
            end
```

```
        cmddacs1:
            begin
                dacf<=0; dacs<=1; gain<=0; lock<=0; rdloop<=0;
                NextState <=  cmd3;
            end
        cmdgain:
            begin
                            if (!data[4]&&!data[3])
                                begin
                    dacf<=0; dacs<=0; gain<=1; lock<=0; rdloop<=0;
                    NextState <=  cmdgain1;
                       end
                            else if (data[4]&&!data[3])
                                begin
                                dacf<=0; dacs<=0; gain<=1; lock<=0;
rdloop<=0;
                                Mreg=data[2:0];

                                case(Mreg)
                                3'b000:
                                begin
                                dacf_modi<=5'b00001;
dacs_modi<=5'b00000;
                                end
                                3'b001:
                                begin
                                dacf_modi<=5'b00010;
dacs_modi<=5'b00000;
                                end
                                3'b010:
                                begin
                                dacf_modi<=5'b00100;
dacs_modi<=5'b00000;
                                end
                                3'b011:
                                begin
                                dacf_modi<=5'b01000;
dacs_modi<=5'b00000;
                                end
                                3'b100:
                                begin
                                dacf_modi<=5'b10000;
dacs_modi<=5'b00000;
                                end
                                3'b101:
                                begin
                                dacf_modi<=5'b00000;
dacs_modi<=5'b00001;
                                end
                                3'b110:
                                begin
                                dacf_modi<=5'b00000;
dacs_modi<=5'b00010;
                                end
                                3'b111:
                                begin
                                dacf_modi<=5'b00000;
```

```
dacs_modi<=5'b00100;
                                end
                                default:
                                begin
                                dacf_modi<=5'b00000;
dacs_modi<=5'b00000;
                                end

                                endcase

                                NextState<= cmdgain1;
                                end

                               else if (!data[4]&&data[3])
                                begin
                                dacf<=0; dacs<=0; gain<=1; lock<=0;
rdloop<=0;
                                strobe={data[2:0],addr[3:0]};
                                NextState<= cmdgain1;
                                end

                               else //   when (data[4]&&data[3])
                                begin
                                dacf<=0; dacs<=0; gain<=1; lock<=0;
rdloop<=0;
                                offset=data[2:0];
                                NextState<= cmdgain1;
                                end
                    end
        cmdgain1:
            begin
                dacf<=0; dacs<=0; gain<=1; lock<=0; rdloop<=0;
                NextState <=  cmd3;
            end

        cmd3:
            begin
                NextState <=  cmd4;
                dacf<=0; dacs<=0; gain<=0; lock<=0; rdloop<=0;
            end
        cmd4:
            begin
                NextState <=  cmd5;
                dacf<=0; dacs<=0; gain<=0; lock<=0; rdloop<=0;
            end

        cmd5:
            begin
                NextState <=  cmd6;
                dacf<=0; dacs<=0; gain<=0; lock<=0; rdloop<=0;
            end
        cmd6:
            begin
                NextState <=  cmd7;
                dacf<=0; dacs<=0; gain<=0; lock<=0; rdloop<=0;
            end
        cmd7:
```

```
            begin
                NextState <=  cmd8;
                dacf<=0; dacs<=0; gain<=0; lock<=0; rdloop<=0;
            end
        cmd8:
            begin
                NextState <=  cmd9;
                dacf<=0; dacs<=0; gain<=0; lock<=0; rdloop<=0;
            end
        cmd9:
            begin
                NextState <=  cmd10;
                dacf<=0; dacs<=0; gain<=0; lock<=0; rdloop<=0;
            end
        cmd10:
            begin
                NextState <=  cmdend;
                dacf<=0; dacs<=0; gain<=0; lock<=0; rdloop<=0;
            end

        cmdend:
            begin
                NextState <=  idle;
                dacf<=0; dacs<=0; gain<=0; lock<=0; rdloop<=0;
            end

        default:
            begin
            NextState <= idle;
            dacf<=0; dacs<=0; gain<=0; lock<=0; rdloop<=0;
            end

        endcase
    end

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

////////////// ADC strobe
        always@(reset)
        begin
        temp<=7'b0000000;
        end

        always@(posedge clock)
        begin
        if(temp<(strobe>>1))
```

```
        begin
        clkadc<=1'b1;temp<=temp+1;
        end
        else if((temp>=(strobe>>1))&&(temp<(strobe-1)))
        begin
        clkadc<=1'b0;temp<=temp+1;
        end
        else
        begin
        clkadc<=1'b0;temp<=7'b0000000;
        end
        end

        always@(posedge clock)
        begin
        if(strobeon)
        clk<=clkadc;
        else
        clk<=clock;
        end

        always@(negedge clock)
        begin
        if(strobeon)
        clk<=clk;
        else
        clk<=clock;
        end


endmodule

module shiftreg
(reset,clock,in,out0,out1,out2,out3,out4,out5,out6,out7,out8,out9,out10
,out11,out12,out13);

    input  reset,clock,in;
    output
    out0,out1,out2,out3,out4,out5,out6,out7,out8,out9,out10,out11,out12,
out13;
    reg
    out0,out1,out2,out3,out4,out5,out6,out7,out8,out9,out10,out11,out12,
out13;

    always@(posedge clock)
    begin
        if(reset)
            begin
            out13<=0;
            out12<=0;
            out11<=0;
            out10<=0;
            out9<=0;
            out8<=0;
            out7<=0;
            out6<=0;
            out5=0;
```

```
            out4<=0;
            out3<=0;
            out2<=0;
            out1<=0;
            out0<=0;
            end
        else
            begin
            out13<=out12;
            out12<=out11;
            out11<=out10;
            out10<=out9;
            out9<=out8;
            out8<=out7;
            out7<=out6;
            out6<=out5;
            out5=out4;
            out4<=out3;
            out3<=out2;
            out2<=out1;
            out1<=out0;
            out0<=in;
            end
    end

endmodule



module senscntl
(rdloop,sample,strobeon,reset,clock,in,modify,data,dacf,dacs,gain,clkch
,readch,dacf_modi,dacs_modi,chan0,addr,strobe,offset,clk);

    input  reset,clock,in,modify,strobeon;
       output  [6:0]  strobe;
    output dacf,dacs,gain,chan0,clk,sample,rdloop;
    output [4:0] data,dacf_modi,dacs_modi;
    output [15:0] clkch,readch;
    output  [3:0]  addr;
    output  [2:0]  offset;

    reg    sample;
    wire   dacf,dacs,gain,chan0,rdloop;
    wire    [6:0]  strobe;
    reg [1:0]  cmdbits;
    reg [4:0]  data,datard,counter;
    wire    [15:0]  readch;
    reg [3:0]  addr;
       wire   [15:0]  clkch;
    wire   [13:0] regout;
    wire   lock,rdmode,st;
    wire   [4:0]  maxaddr;
    reg   [3:0]   rdaddr;
    wire   [4:0]   dacf_modi,dacs_modi;
       wire   [2:0]   offset;
```

```
    shiftreg ishreg
(.reset(reset),.clock(clock),.in(in),.out0(regout[0]),.out1(regout[1]),
.out2(regout[2]),.out3(regout[3]),.out4(regout[4]),.out5(regout[5]),.ou
t6(regout[6]),.out7(regout[7]),.out8(regout[8]),.out9(regout[9]),.out10
(regout[10]),.out11(regout[11]),.out12(regout[12]),.out13(regout[13]));
    cntlfsm    icfsm
(.strobeon(strobeon),.clk(clk),.reset(reset),.data(data),.addr(addr),.c
lock(clock),.st(st),.cmdbits(cmdbits),.dacf(dacf),.dacs(dacs),.gain(gai
n),.lock(lock),.rdloop(rdloop),.dacf_modi(dacf_modi),.dacs_modi(dacs_mo
di),.offset(offset),.strobe(strobe));



    always@(posedge lock)
    begin
        cmdbits<=regout[10:9]; data<=regout[8:4]; addr<=regout[3:0];
    end

    always@(posedge rdloop)
    begin
        datard<=data;
    end

// counter to replace the readfsm
       always@(posedge clk)
       begin
       if (counter<(maxaddr+1'b1))
       counter<=counter+1;
       else
       counter<=0;
       end

       always@(posedge rdloop)
       begin
       counter<=0;
       end

       always@(posedge clk)
       begin
        if (!rdmode)
          begin
           if (counter>0)
            begin
            sample<=0;
            if(counter>1)
            rdaddr<=(counter-1);
            else
            rdaddr<=0;
            end
           else
            begin
            rdaddr<=0;
            sample=1;
            end
          end
        else
         begin
```

```
        rdaddr<=maxaddr[3:0];
        if(rdloop)
        sample<=1;
        else
        sample<=!sample;
        end
      end

      // modify
      always@(posedge modify)
      begin
      if(rdaddr<offset)
      addr={1'b0,rdaddr}+5'b10000-offset;
      else
      addr=rdaddr-offset;
      end

    assign maxaddr = {1'b0,datard[3:0]};
    assign rdmode = datard[4];
    assign st=regout[13]&&!regout[12]&&regout[11];

    assign clkch[0]=!addr[3]&&!addr[2]&&!addr[1]&&!addr[0];
    assign clkch[1]=!addr[3]&&!addr[2]&&!addr[1]&&addr[0];
    assign clkch[2]=!addr[3]&&!addr[2]&&addr[1]&&!addr[0];
    assign clkch[3]=!addr[3]&&!addr[2]&&addr[1]&&addr[0];
    assign clkch[4]=!addr[3]&&addr[2]&&!addr[1]&&!addr[0];
    assign clkch[5]=!addr[3]&&addr[2]&&!addr[1]&&addr[0];
    assign clkch[6]=!addr[3]&&addr[2]&&addr[1]&&!addr[0];
    assign clkch[7]=!addr[3]&&addr[2]&&addr[1]&&addr[0];
    assign clkch[8]=addr[3]&&!addr[2]&&!addr[1]&&!addr[0];
    assign clkch[9]=addr[3]&&!addr[2]&&!addr[1]&&addr[0];
    assign clkch[10]=addr[3]&&!addr[2]&&addr[1]&&!addr[0];
    assign clkch[11]=addr[3]&&!addr[2]&&addr[1]&&addr[0];
    assign clkch[12]=addr[3]&&addr[2]&&!addr[1]&&!addr[0];
    assign clkch[13]=addr[3]&&addr[2]&&!addr[1]&&addr[0];
    assign clkch[14]=addr[3]&&addr[2]&&addr[1]&&!addr[0];
    assign clkch[15]=addr[3]&&addr[2]&&addr[1]&&addr[0];

    assign readch[0]=!rdaddr[3]&&!rdaddr[2]&&!rdaddr[1]&&!rdaddr[0];
    assign readch[1]=!rdaddr[3]&&!rdaddr[2]&&!rdaddr[1]&&rdaddr[0];
    assign readch[2]=!rdaddr[3]&&!rdaddr[2]&&rdaddr[1]&&!rdaddr[0];
    assign readch[3]=!rdaddr[3]&&!rdaddr[2]&&rdaddr[1]&&rdaddr[0];
    assign readch[4]=!rdaddr[3]&&rdaddr[2]&&!rdaddr[1]&&!rdaddr[0];
    assign readch[5]=!rdaddr[3]&&rdaddr[2]&&!rdaddr[1]&&rdaddr[0];
    assign readch[6]=!rdaddr[3]&&rdaddr[2]&&rdaddr[1]&&!rdaddr[0];
    assign readch[7]=!rdaddr[3]&&rdaddr[2]&&rdaddr[1]&&rdaddr[0];
    assign readch[8]=rdaddr[3]&&!rdaddr[2]&&!rdaddr[1]&&!rdaddr[0];
    assign readch[9]=rdaddr[3]&&!rdaddr[2]&&!rdaddr[1]&&rdaddr[0];
    assign readch[10]=rdaddr[3]&&!rdaddr[2]&&rdaddr[1]&&!rdaddr[0];
    assign readch[11]=rdaddr[3]&&!rdaddr[2]&&rdaddr[1]&&rdaddr[0];
    assign readch[12]=rdaddr[3]&&rdaddr[2]&&!rdaddr[1]&&!rdaddr[0];
    assign readch[13]=rdaddr[3]&&rdaddr[2]&&!rdaddr[1]&&rdaddr[0];
    assign readch[14]=rdaddr[3]&&rdaddr[2]&&rdaddr[1]&&!rdaddr[0];
    assign readch[15]=rdaddr[3]&&rdaddr[2]&&rdaddr[1]&&rdaddr[0];
    assign chan0=readch[0];

endmodule
```