MICRO–CONTROLLER COMMUNICATION MODULE

An investigation into digital communications,

with emphasis on simplicity and practicality.

by

Karl C. Hansen

Submitted in Partial Fulfillment of the Requirements

for the Degree of

Master of Science in Engineering

in the

Electrical Engineering

Program

Prof. Samuel J. Skarote 6/4/91

Advisor Date

Sally M. Hotchkiss June 14, 1991

Dean of the Graduate School Date

YOUNGSTOWN STATE UNIVERSITY

June, 1991

ABSTRACT

4-10-91

MICRO–CONTROLLER COMMUNICATION MODULE

An investigation into digital communications,

with emphasis on simplicity and practicality.

Karl C. Hansen

Master of Science in Engineering

Youngstown State University, 1991

Advances in digital electronics have brought about the use of microprocessors and digital techniques in diverse fields once thought to be the exclusive domain of analog. One such field is that of telephone systems. Telephone companies are presently at work designing circuitry for use in ISDN (Integrated Services Digital Network) systems. In such a system, there is a need for simple modules capable of digitizing input signals and reproducing output signals, possibly with multiple channels. This thesis investigates the design and implementation of a four channel, full-duplex, digital communication module. Theory and techniques involved in the areas of analog-to-digital conversion, signal processing, serial communication, and digital-to-analog conversion are discussed and compared. Design goals are laid out with emphasis on simplicity and practicality. The entire design phase from concept to implementation is detailed. The final design successfully implements a simple yet practical module capable of performing simultaneous bi-directional digital communication on four channels. Each channel has a bandwidth of approximately $3kHz$.

ACKNOWLEDGMENTS

I wish to acknowledge the assistance of my advisor Professor Skarote and Dr. Pansino for teaching the courses which gave birth to the ideas for this project. The communication aspects of the design stem from a course under Dr. Pansino, while the digital nature stems from a course under Professor Skarote.

I also wish to thank Jim Plescia and J. D. Brosnahan of Motorola, Inc., Bob DeBlasis of Toshiba, Inc., and Karen West of J. R. Thornberry for their assistance in obtaining literature, samples, and information. Thanks go to my employer, Rapid Design Service, Inc., for use of equipment after hours at work. I especially wish to thank the staff at Analog Devices for publishing their Analog–Digital Conversion Handbook, an excellent reference for anyone involved in this field. And finally, I must acknowledge my deep debt of gratitude to my wife, Lisa, without whose support, patience, and tolerance I would not have brought this project to fruition.

TABLE OF CONTENTS

PAGE

CHAPTER

LIST OF SYMBOLS

SYMBOL	DEFINITION	UNITS OR REFERENCE
A/D	Analog-to-Digital conversion	
bit	Binary digit (0 or 1)	
byte	Group of *bits*, usually eight	
D/A	Digital-to-Analog conversion	
dB	decibel, unit of logarithmic power ratio	none
DCGA	Digitally-Controlled Gain/Attenuation	
DSP	Digital Signal Processing	
DVM	Digital volt-meter	
EEPROM	Electrically-Erasable PROM	
EPROM	UV-Erasable PROM	
EVB	Evaluation board	
FIR	Finite impulse response filter	
Hz	Unit of frequency, Hertz	cycles/second
IC	Integrated circuit, 'chip'	
IIR	Infinite impulse response filter	
ISDN	Integrated Services Digital Network	
kbit	kilo- (one thousand) *bits*	
kbyte	kilo- (one thousand) *bytes*	
kHz	kilo- (one thousand) Hertz	cycles/second
MHz	Mega- (one million) Hertz	cycles/second
Mbit	Mega- (one million) *bits*	
μs	micro- (one millionth) second	

N_i	Nyquist interval	seconds/sample
N_r	Nyquist rate	samples/second
ns	nano- (one billionth) second	
ω	A variable representing frequency	Hz
ω_o	Cutoff frequency of filter	Hz
ω_x	Cutoff frequency of band-limited spectrum	Hz
PC	Personal Computer	
PLCC	Plastic Leaded Chip Carrier	
PROM	Programmable Read-Only Memory	
RAM	Random-Access Memory	
ROM	Read-Only Memory	
SAR	Successive-approximation register	
SDI	Sampled Difference Integrator	
UART	Universal Asynchronous Receiver/Transmitter	
UV	Ultra-Violet light	

LIST OF FIGURES

FIGURE PAGE

CHAPTER I

INTRODUCTION

Overview

The project designed uses Analog-to-Digital (A/D) sampling to digitize four communication channels, each with a bandwidth of $3kHz$. The digitized samples are time-division multiplexed onto a three-wire synchronous serial channel with a capacity in excess of $2MBit$ per second. Samples received on the channel are transmitted via a parallel bus to a four channel Digital-to-Analog (D/A) converter IC. Each of the D/A outputs is processed by a sampled difference integrator (SDI) module. The input channels and SDI output channels are filtered through $3kHz$ fourth-order Sallen-Key equal component low-pass active filters designed for a Bessel response. The project uses digitally controlled gain/attenuation on the input signals to broaden the range of signal strengths it can handle.

The project is separated into design areas as follows:

- Signal input and A/D conversion.
- Digital filtering and signal processing. (DSP)
- Time-division multiplexing and serial communications.
- D/A conversion.
- Sampled-difference integration. (SDI)
- Digitally-controlled gain/attenuation. (DCGA)

Each of the design areas is discussed with theory and techniques. Where a design area directly affects a component used in the final design, reasons for the choice of component are given. Also discussed are the choice of micro-controller, design problems and solutions, and potential areas for further development in the design.

Evolution of the project

Upon commencing studies at Youngstown State University, the author enrolled in two courses, E.E. 932 and E.E. 808R, which germinated the ideas for this thesis. Since both classes required a project, the author desired to expand upon the effort required for either class individually and come up with a project which could satisfy the requirements of both. This approach was also attractive to the author because his undergraduate degree in Computer Engineering gave him a solid foundation in the overall design of computer based systems from hardware to software.

Course requirements for E.E. 932 mandated that the project be digital in nature and have practical applications. Requirements for E.E. 808R mandated that the project focus on a form of communication and investigate the theory behind that form. This meant the project would by necessity involve a microprocessor, software, analog interfaces, and communication. It was also the author's desire to produce a complete system rather than focus on any aspect of a system.

The final result of that project was a module which, although marginally functional, was by no means refined. While intelligible communication was possible on all channels, the module was very sensitive to input signal levels and had large amounts of noise on all channels. Also, in the course of completing the original project, many areas of potential development had to be eliminated in the interests of obtaining a functional module. At the conclusion of the courses, it was the author's desire to pursue improving the module. After discussions with Professor Skarote and Dr. Pansino, the author decided to use this thesis for this opportunity. The additional refinements led to the module described in this thesis. Figure 1 shows a block diagram of the final design.[1] The schematic of the final design, a listing of the final version of the software, a photograph of the final breadboard, and a final list of parts for the project are contained in the appendices.

Selection of Design Criteria

Micro-controller

The MC68HC11 was chosen as the core around which the project would be designed. The author is fairly familiar with the MC68HC11 micro-controller from Motorola, and has for some time had the desire to attempt a design utilizing some of its features. Although many of its features were unused in the final design the versatility of the device was a large factor in choosing it. Additionally, the assembler language of the MC68HC11 micro-controller is fairly easy to learn and use.

This micro-controller has an eight channel, eight bit analog-to-digital (A/D) converter, whose pins also double as an input-only digital port. It has one bidirectional port and one input-only port which together can also be used as a multiplexed address/data bus in special modes of operation. A special purpose port has some

[1] See next page for Figure 1. Due to the fact that all figures are one full page in size, each figure follows the page on which it is first referenced.

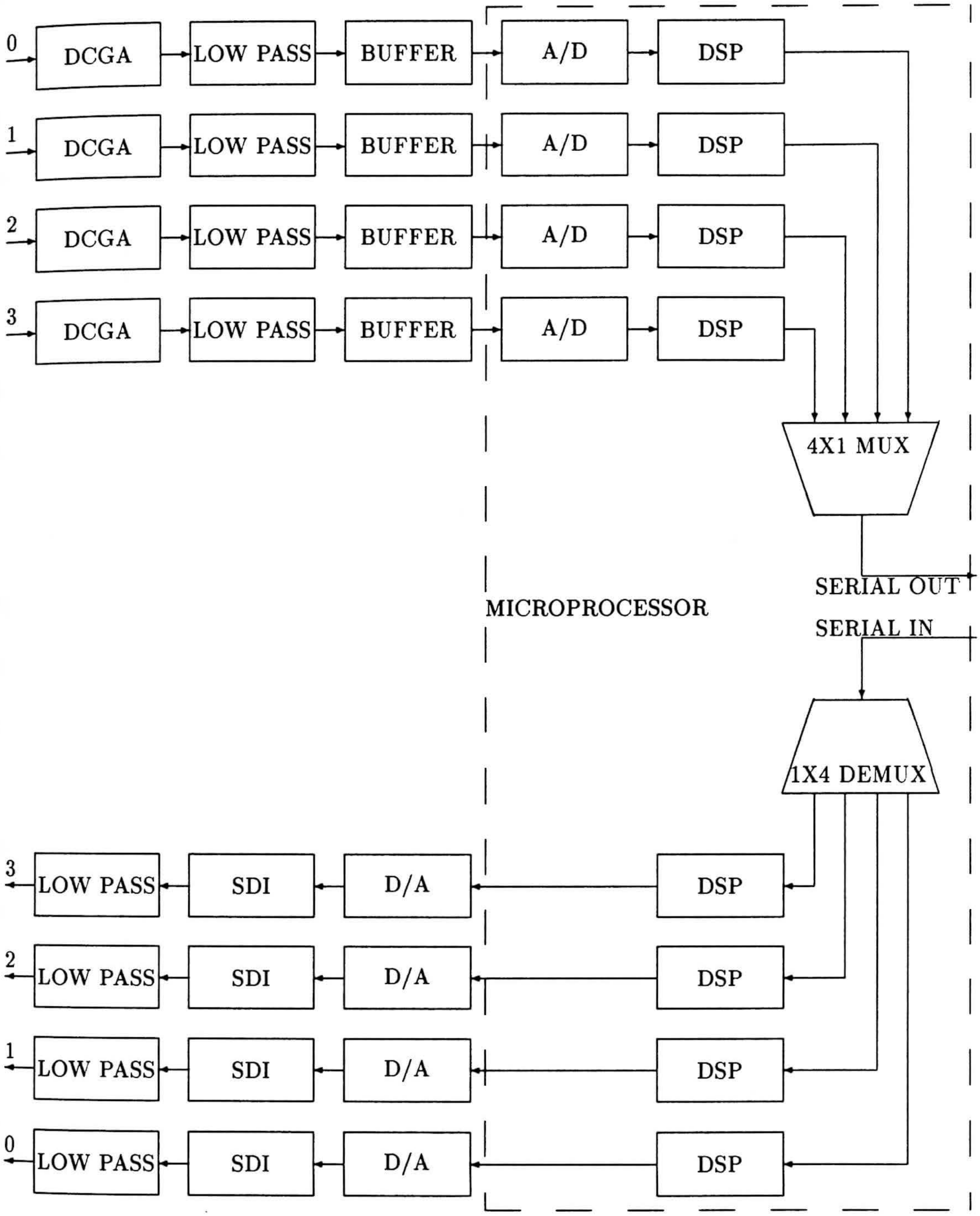


Figure 1. SYSTEM BLOCK DIAGRAM

output-only pins, some input-only pins and a bidirectional pin. These pins can be configured for input-capture functions[2] or output-compare functions.[3] The bidirectional pin can be used for pulse accumulation or pulse width measurement. Another five-bit port is used as either a bidirectional port or a pair of serial ports, one synchronous, the other asynchronous and functioning similarly to a UART.

Also on the chip are 512 bytes of EEPROM, 256 bytes of RAM, and up to 8 kbytes of ROM. Some versions of this chip have EPROM in place of the ROM, and a clear window on the chip so the EPROM can be erased and the code changed.

The MC68HC11 was originally released with a maximum speed of $8MHz$. Improvements in the design as well as shrinking geometries for integrated circuits in general have led to the availability of parts with a maximum speed of $16MHz$. The final design used a $16MHz$ part. Throughout this thesis, calculations are made based on the $8MHz$ part with comparison made to the results if a $16MHz$ part were used. Where the capabilities of the $16MHz$ MC68HC11 were required for the final design it is noted.

In the original module input signals connected directly to the MC68HC11 A/D port. This severely limited the capability of the module. Signals which were too strong were clipped by the conversion process, while signals which were considerably less than $5V$ peak-to-peak did not utilize the full digitizing range of the

[2] An input pin together with a dedicated 16-bit register which 'captures' the present count value of a free-running counter internal to the micro-controller upon detection of a user selectable negative or positive edge. (Pulse width measurement.)

[3] A dedicated 16-bit register together with an output pin configurable to go to a particular state (0 or 1) or to toggle when a free-running counter internal to the micro-controller matches the value stored in the register. (General-purpose timing signals.)

micro-controller. A digitally-controlled gain/attenuation (DCGA) stage was added to the revised module, between the input signal and the input filter. This allows the software to adjust input signal levels to maximize the use of the full conversion range of the micro-controller.

Channel capacity and bandwidth

The $3kHz$ bandwidth was selected due to the fact that a typical telephone line has the same bandwidth. This bandwidth gives intelligible conversation while at the same time allowing many more channels to be multiplexed onto the same lines than if the full audible spectrum of $20Hz$ to $20kHz$ were to be transmitted. 'Typical' telephony uses frequency-division multiplexing to combine multiple channels onto a single line with a large bandwidth; this module would use time-division multiplexing to combine digital samples from four channels onto a single serial channel with a large bandwidth. From the outset the author was aware that digitizing the signals would add error to them in both the input and reproduction phases. These effects are discussed later in this paper.

The selection of four channels was made because the MC68HC11 digitizes four samples in one conversion pass of its A/D converter. These samples can be taken from one channel or from four adjacent channels. This conversion pass lasts 128 bus cycles. For a MC68HC11 running off of an $8MHz$ oscillator, a bus cycle lasts $500ns$, and a conversion pass lasts $64\mu s$. This means the MC68HC11 could theoretically generate 62,500 samples in one second. Dividing these samples among four channels yields a potential sample rate per channel of 15,625 per second. This is well over the amount necessary for the bandwidth chosen. The theory to demonstrate this is discussed later in this report. With a $16MHz$ MC68HC11 the potential sample rate per channel doubles to 31,250 per second. Whether or not this rate is achieved depends on the choice of method of serial communication.

Serial communication method

The original version of this module used the synchronous serial channel of the MC68HC11 rather than the asynchronous channel. This choice was dictated by channel capacity. Use of an $8MHz$ oscillator gives capacities of $1Mbit$ per second for the synchronous channel and $125kbit$ per second for the asynchronous channel. The synchronous channel transmits eight bits per byte while the asynchronous channel transmits ten bits per byte giving capacities of $125kbytes$ and $12.5kbytes$ respectively per second. Use of a $16MHz$ part raises these channel capacities to $250kbytes$ and $25.0kbytes$ per second.

Theory (discussed later) shows each of the input channels must generate at least $6kbytes$ of data each second in order to have any possibility of reproducing the signal at the digital-to-analog (D/A) conversion stage. Based on this minimum data rate, simple arithmetic shows the asynchronous serial channel is incapable of handling sample data from four input channels when the MC68HC11 is operated at $8MHz$, while the synchronous serial channel can do so with ease. In fact, the synchronous channel has the capacity for up to ten times as many samples as the asynchronous channel, as demonstrated by the following set of equations. To simplify the math it will be assumed that a channel generates $6.25kbytes$ of data per second.

$$(Asynchronous\ channel\ capacity) \qquad \frac{12.5kbytes/second}{6.25kbytes/second/channel} = 2.0\ channels$$

$$(Synchronous\ channel\ capacity) \qquad \frac{125kbytes/second}{6.25kbytes/second/channel} = 20\ channels$$

$$(Synchronous\ to\ asynchronous\ ratio) \Rightarrow \frac{(Synchronous\ channel\ capacity)}{(Asynchronous\ channel\ capacity)} = 10$$

However, one drawback of using the synchronous channel is that the modules must share a common clock. This can be alleviated to a degree with the use of special hardware such as Manchester encoder/decoder chips which combine a clock and data into a single signal in the encoder and reproduce the clock and data at the decoder. Use of such hardware would more than double the cost of each module. Another drawback is that there is no easy way to enable multiple sources of data to transmit on the same bus. Use of the asynchronous channel would alleviate these problems, and with a $16MHz$ MC68HC11, the asynchronous channel appears to have *just* enough capacity. However, some form of identifying the first byte of the four byte packet is necessary. Adding an address bit to each byte increases the amount of data to the point that it will no longer fit on the asynchronous channel. Because of this, the author was forced to continue to use the synchronous channel in the revised module. Motorola has recently announced a chip called the MC68HC16 which runs at $16MHz$ and has double the capacity of a $16MHz$ MC68HC11. Use of a MC68HC16 would allow the design to use the asynchronous channel for communication. As of yet, the author has been unable to obtain samples for such an experiment.

A Motorola Evaluation Board (EVB) was used in the design of this module to simplify code and hardware testing. The EVB is essentially a single-board computer which emulates the single-chip mode of operation of the MC68HC11. A user can build a design using a 52-pin PLCC socket for the MC68HC11, and then connect the EVB to the socket with an adaptor cable. This allows software and hardware debug using a standard PC terminal. Because the author had only one EVB available, the decision was made to loop the serial transmit line back to the serial receive line. This allows demonstration of full-duplex serial communication without requiring a second communication module. Once the code had been debugged, the EVB was replaced with a single MC68HC11 with EEPROM for final tests.

Digital-to-analog conversion

Various methods of D/A conversion are available so no decision was made in the preliminary stages of setting the design criteria as to which method would be used. Several methods were investigated, and two different D/A converter IC's were tested. Discussion of various D/A methods occurs later in this report.

In the original module, the output of the AD7226 D/A converter directly fed a second-order Sallen-Key low-pass analog filter to reduce the step-noise that occurs when the D/A transitions from one output level to another. This method gave poor results because of the difficulty associated with filtering out the high-frequency content of the steps inherent in the D/A output. The final module introduces a stage which uses a sample-and-hold combined with a difference integrator to generate ramps rather than steps. The ramp is easier to filter than the step function, and the output has correspondingly lower audible noise. The final module also uses fourth-order low-pass filters rather than second- order low-pass filters.

Analog filtering

From the outset, it was known that both the input channels and the output channels would require low-pass filters. A $3kHz$ cutoff frequency was chosen due to the input channel bandwidth specification. It was not known whether the output channels would require higher-order filters than the input channels, so experiments were performed to determine what kind of filters would be both effective and easy to implement. The MC68HC11 A/D input channels can digitize signals ranging from zero to approximately five volts,[4] so the input filters would need to produce a signal in that range. The gain/attenuation stage added gives the MC68HC11 the ability to adjust the input over a range of $-48dB$ to $+48dB$.[5]

[4] The range is set by the voltages applied to the **Vrh** and **Vrl** pins. **Vrl** can range from -0.3 Volts up to **Vrh**. **Vrh** can range from **Vrl** up to approximately 6 Volts. For proper operation **Vrh** should be at least 2 Volts greater than **Vrl**.

[5] The attenuation range is a little overdone since it would, in theory, allow the module to handle signals measuring almost $1,000V$ peak-to-peak while the breakdown voltage of the chips used is on the order of $30V$, but the symmetrical design was fairly easy compared to a more realistic design.

CHAPTER II

DESIGN AREAS

Signal Input and A/D Conversion

Nyquist interval and Nyquist rate

Sampling a signal with a nonzero bandwidth requires familiarity with a concept known as the Nyquist criterion or Nyquist rate. In brief, this concept states that if a signal has a finite energy spectrum with a bandlimited Fourier transform in which $F(\omega) = 0$ for $\omega \geq \omega_x$, then that signal can be completely reconstructed if the sampling period is less than or equal to half the period of ω_x. This sampling period is known as the Nyquist interval, N_i. A related value, the Nyquist rate, N_r is the minimum sampling rate. If the signal has nonzero energy content above ω_x, sampling will *alias* the energy of the higher frequencies into the range $-\omega_x$ through ω_x. This effect is usually not desirable, but if the designer is aware of the effect, it can be intentionally used for interesting results.

If a signal is not bandlimited and the designer is only interested in a small portion of the lower frequencies in the signal, prefiltering the signal and increasing the sample rate above the required Nyquist rate for the maximum frequency of the portion of interest can enable one to process and reproduce the portion of interest from the original signal. In such a case the sampling period is usually much less than the Nyquist interval for the highest frequency of interest, and conversely the sampling rate is much greater than the Nyquist rate, sometimes as much as ten times greater. Filtering the signal through a low-pass filter with a steep rolloff rate reduces the effects of aliasing due to significant attenuation of the frequencies higher than the cutoff point.

The Nyquist interval, N_i, for a $3kHz$ bandwidth is easily calculated:

$$\begin{aligned} N_i &= 0.5(PERIOD(\omega_x)) \\ PERIOD(\omega_x) &= PERIOD(3kHz) \\ PERIOD(3kHz) &= \frac{1}{3000}\ sec \\ \Rightarrow N_i &= 0.5(333\ 1/3\ \mu s) \\ &= 166\ 2/3\ \mu s \end{aligned}$$

The Nyquist rate, N_r, is just the inverse of N_i or $6,000$ samples per second.

Aliasing effects

If we sample at the Nyquist rate, integer multiples of $6kHz$ will appear as a DC bias in our signal. It is easy to verify this because samples will occur at the exact same position on the signal. Integer multiples of $3kHz$ will appear as either DC bias or as a $3kHz$ signal. This is because even multiples of $3kHz$ correspond to the $6kHz$ case just described, while odd multiples of $3kHz$ have the same relative amplitude as the $3kHz$ signal at the sample times. Integer multiples of $1.5kHz$ will appear as DC bias, $1.5kHz$, or $3kHz$ signals. Integer multiples of $0.75kHz$ will appear as DC bias, $0.75kHz$, $1.5kHz$, $2.25kHz$, or $3kHz$. This progression continues as one divides the base frequency by higher and higher powers of two. From this it can be seen that the entire spectrum above $3kHz$ is folded back (*aliased*) into the range from DC to $3kHz$.

In this design, the input signals are not bandlimited. Even after filtering some energy will exist in the frequencies above the filter cutoff frequency. This requires that the input filter have a fairly steep rolloff after its cutoff frequency, and that the module sampling rate be greater than the Nyquist rate, the greater the better, to avoid having undesired signal energy map back into the desired band.

Analog filters

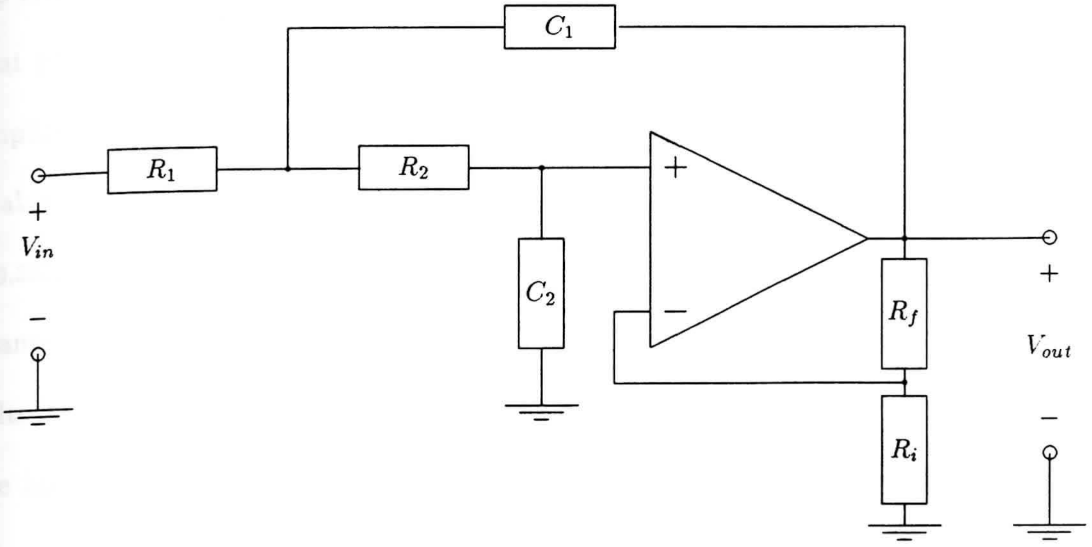
Several good sources of information exist in the area of active filter design. For this design the author utilized the texts *Applications and Design with Analog Integrated Circuits* and *Signals and Systems.*[6] For simplicity of design a Sallen-Key equal component filter was selected. With this filter the type of filter response is determined solely by the gain-setting resistors. The cutoff frequency is set by two identical resistors and two identical capacitors. Figure 2 shows a single-stage filter together with associated design equations. Another nice feature of the Sallen-Key filter is that only a single Op Amp is required for each second-order section.

A Bessel response was selected over the Butterworth and Chebyshev responses because it has a more linear phase characteristic (or spectra) over the passband. It also has no ripple in the passband. Since the goal of the passband was to have good reproduction, these two features of the Bessel filter made it more attractive.

It was felt that a first-order filter would not sufficiently attenuate the frequencies above $3kHz$ because it would have a rolloff rate of only $6dB$ per octave.[7] With a second-order filter this could be increased to $12dB$ per octave. With a fourth-

[6] See entries in the bibliography.

[7] Actually $20dB$ per decade.



$$R_1 = R_2 = R$$
$$C_1 = C_2 = C$$
$$\omega_o = \frac{K_{lp}}{\sqrt{R_1 \times R_2 \times C_1 \times C_2}} = \frac{K_{lp}}{R \times C}$$
$$A_o = 1 + \frac{R_f}{R_i}$$
$$\alpha = 3 - A_o$$

For a 2^{nd}-order Bessel response, $K_{lp} = 1$ and $\alpha = 1.73$.
For a 4^{th}-order Bessel response, cascade two 2^{nd}-order filters with $K_{lp} = 0.684$ and $\alpha = 1.916$ for the first stage, and with $K_{lp} = 0.479$ and $\alpha = 1.241$ for the second stage.

Figure 2. SALLEN-KEY EQUAL-COMPONENT FILTER

order filter this would increase to $24dB$ per octave. A second-order filter would require one Op Amp and six discrete components. A fourth-order filter would require two Op Amps and twelve discrete components. With a second-order $3kHz$ low-pass filter that produces a $5V$ peak-to-peak output from a 0 to $3kHz$ signal input with a given amplitude and a $6kHz$ signal input with the same amplitude would produce a $1.25V$ peak-to-peak output. Under the same conditions, a fourth-order filter would produce a $0.315V$ peak-to-peak output. For initial testing, the module was designed with two channels having second-order filters and two channels having fourth-order filters. Results indicated that the second-order filters did not give acceptable performance, and the final version of the project reflects this by the use of fourth-order filters in all cases.

The $8MHz$ MC68HC11 is capable of sampling each of the four input channels once every 64 μs, while the $16MHz$ MC68HC11 requires only 32 μs. From the value for N_i computed above, it is seen that this allows approximately 102 μs for additional processor overhead with an $8MHz$ chip and 134 μs with a $16MHz$ chip. It is also apparent that considering only the A/D converter limitations the absolute best that could ever be accomplished would be to sample at approximately $2.6(N_r)$ using an $8MHz$ chip.

Use of a $16MHz$ MC68HC11 increases this to approximately $5.2(N_r)$. If all of the processor overhead associated with transmitting the last four samples, receiving four samples and sending them to the D/A converter can be accomplished while four new samples are being taken, then this maximum can be achieved. If overhead pushes the software loop time beyond 64 μs for the slower chip or 32 μs for the faster chip, then the module will operate at less than maximum rate. In no case can the loop time be permitted to exceed N_i. If the loop time does exceed N_i, the ability to reconstruct the sampled signal is lost.

Using the synchronous serial channel in the preliminary design enabled the project to run at very near the theoretical sampling capacity. Using the asynchronous serial channel would reduce the sample rate but make communication among several modules much simpler. This was not possible due to the additional software required to handle the DCGA and SDI functions. The final module continues to use the synchronous serial channel.

Analog to Digital conversion methods

Dual-slope conversion

There are three major methods of A/D conversion. They are dual-slope conversion, successive-approximation conversion, and flash conversion. With dual-slope conversion a capacitor is allowed to charge for a fixed time to sample the input. The input is then switched off and the capacitor is discharged through a constant current source. The length of time required to discharge the capacitor to zero charge is recorded, and the count is the digitized value. This method is fairly slow and is only accurate for a small number of bits, typically about four. Use of precision components and long (two to three second) sample times significantly improves the accuracy and the number of bits obtainable from a sample.

Successive-approximation conversion

Successive-approximation (SAR) conversion is a process whereby the input is sampled onto a capacitor connected to one side of a comparator. After sampling the input is switched off and various voltages are applied to the other side of the comparator. The output of the comparator determines what voltages will be tried next. The most efficient method determines the value of one bit for each 'pass' and generates an eight bit result after eight passes. This is the method used in the MC68HC11. Another version of SAR conversion uses a free-running counter that counts up or down depending on the output of the comparator. The counter controls the voltage applied to the comparator input as if it were a D/A converter hooked to the comparator. For slowly varying signals this type of SAR converter will generate samples much faster than the previous type. For rapidly varying signals the first type is usually faster. Both the dual-slope converters and SAR converters require a clock for proper operation. Sometimes this clock is generated internally by an on-chip oscillator.

Flash conversion

Flash conversion pregenerates a distinct voltage for each possible digital value. These voltages are each connected to one input of the same number of comparators. The other input of all of the converters is connected to the input signal. For a given input voltage V_x, the converters connected to a voltage lower than V_x will output a 'low' signal. The converters connected to a voltage greater than V_x will output a 'high' signal. The outputs of all of the comparators connect to a combinational circuit which converts them into the proper binary codes. For an eight-bit converter 256 voltages, 256 comparators, and a 256 × 8 decoder are required. For a twelve bit converter 4,096 voltages, 4,096 comparators, and a 4,096 × 12 decoder are required. As the number of output bits required increases, the circuit complexity increases geometrically; flash converters are extremely fast, but they are also expensive.

Half-flash conversion

Recently a circuit has been developed which is called a half-flash converter. In this circuit, flash converter techniques are used to convert the high-order half of the desired bits. The value obtained is used with a fast D/A to generate a voltage which is subtracted from the input signal. The resulting signal is passed through another flash converter to generate the low-order half of the desired bits. Although slower than a full-flash converter, this type of converter requires substantially less circuitry for a given number of bits than the full-flash converter. In an eight-bit design, for example, two sets of 16 voltages, 16 comparators, and a 16 × 4 decoder are required, for a total of 32 voltages, 32 comparators, and two 16 × 4 decoders. One set is used to generate the high-order bits and the other is used to generate the low-order bits. This compares very favorably to the 256 voltages, 256 comparators, and 256 × 8 decoder described above. A D/A converter and an Op Amp are required to subtract the first-stage voltage from the input, but still the savings in circuitry is great. As output bit-width increases, the savings increases even more. Flash-type converters generally do not require a clock.

Digital Filtering and Signal Processing (DSP)

A sampled signal can be represented by its z-transform. This is similar to the transformation from the t-domain to the s-domain performed by the Laplace and Fourier transforms. With analog signals, s-domain equations can be used to compute transfer functions and to analyze circuit response. One can also write an equation for a desired transfer function and derive the circuit which would have that transfer function. Methods such as state-variable analysis can be and are used in more complex control situations. With such a method, conversion from one state variable to another is performed by integration, or multiplication by s^{-1}.

The z-transform

Similar methods may be used in the z-domain. Transfer functions can be determined, circuits designed and analyzed, and state-variable control methods implemented. The primary difference is that in the z-domain multiplication by z^{-1} represents a delay of one sample period where in the s-domain multiplication by s^{-1} represents integration. Time delay is a *much* simpler operation than integration. This implies that if some method of converting an s-domain equation to a z-domain equation existed it would be fairly straightforward to implement analog-type filters in the z-domain. Various methods exist, such as the bilateral z-transform and the Impulse-invariant technique, and they are well covered in the references *Signals and Systems* and *Signals and Systems: Continuous and Discrete* in the bibliography.

IIR and FIR digital filters

Two types of digital filters are Infinite Impulse Response (IIR) and Finite Impulse Response (FIR). The names are derived from the output response of the filter to a unit impulse. An IIR filter has a response of infinite duration,[8] while a FIR has a response of finite duration. IIR filters are characterized by the output feeding back and summing with the input prior to entering the filter. FIR filters are characterized by no feedback paths. There are, however, some FIR filter designs which have feedback paths, but are designed to have finite response. IIR filters generally have steep rolloff rates, but are prone to instability due to the fact that the roundoff errors inherent in any digital system alter the locations of the filter's poles and zeroes. The feedback paths in an IIR filter act to multiply the effects of any roundoff. FIR filters require several more stages to accomplish the same rolloff rates as IIR filters, but the roundoff errors are not fed back into the system so FIR filters are more stable. FIR filters also have the feature that they all have linear phase characteristics or spectra.

Other DSP techniques

Other digital signal processing techniques were investigated, such as encryption and compression, but due to lack of sufficient processor time, they were not implemented. One simple method of 'encryption' is to Exclusive-OR a fixed value with each byte prior to transmission. Received data is Exclusive-ORed with the same value to recover the original data. This is a simple technique and offers some pro-

[8] In theory. Analog filters also have theoretical infinite response. In practice, though, the response after a certain length of time can be ignored.

tection to the data, but is also fairly easy to crack. A simple compression method is to transmit repeated data as a group together with a count indicating the number of times to repeat the group. For a repetitive signal this method can significantly reduce the amount of data transmitted.

Digital signal filtering is one-ended processing in that it can be performed at either the transmitting end or the receiving end with no difference in the final results. Encryption and compression are examples of two-ended processing because actions are required at both the transmitter and receiver. Where encryption, compression, or other two-ended processing occurs, the receiver must invert whatever the transmitter has done to recover the original data.

Design constraints

It was originally desired to perform FIR digital filtering in the module, but analysis indicated that there was not sufficient processor time to implement even a four-stage filter for each channel. Since such a filter would not have much effect, it was decided to eliminate all digital filtering from the module. When the $16MHz$ micro-controller was obtained, the possibility of digital filtering was again examined, but the overhead of handling the DCGA modules and their data prevented it.

Time-Division Multiplexing and Serial Communications

Time division multiplexing is the process whereby data is sampled on several channels at one rate, and the samples are transmitted serially over a channel with the capacity to transmit all of the samples from one sampling pass prior to completion of another sampling pass. If the data is transmitted in a fixed order, demultiplexing is accomplished by storing the data in sequential order to the output channels as it comes across the serial line. If the data is transmitted in a varying order, addressing data usually accompanies the sample data so that it can be routed properly at the receiver.

To improve the reliability of the serial channel, parity codes and error-correction codes such as Hamming-codes can also be transmitted with the data. Such codes are stripped out at the receiver and used to correct or eliminate erroneous data. These codes add overhead to the data transmitted and therefore increase the required bandwidth of the communication channel used.

For ease of use, an asynchronous channel is preferred because no clock is required to synchronize the data between modules. Synchronous channels, however, generally have higher bandwidth because bit uncertainties are eliminated by the common clock. In the MC68HC11, the asynchronous channel has a capacity of approximately 12,500 bytes per second, while the synchronous channel has a capacity of 125,000 bytes per second, nearly 10 times the capacity. The asynchronous channel can handle the data generated from only 2.16 input channels sampled at 6,000 samples per second. The synchronous channel can handle the data from 20.8 such input channels. Since the serial channel used had to be able to service four input channels, the synchronous channel was selected. In a multi-module design some method of synchronizing all of the modules to a common clock would be required.

In the MC68HC11, transmission of one byte on the synchronous serial line takes 8 μs, four bytes take at least 32 μs. The overhead associated with setting up to transmit and saving received data takes another 4 to 6 μs per byte, for an additional 16 to 24 μs. The total transmission overhead is then between 48 and 56 μs. From these figures it is apparent that transmitting the data from one conversion pass takes nearly as long as the conversion itself.

D/A Conversion

The MC68HC11 does not have D/A conversion capability. The pulse width modulated outputs can sometimes be used for a crude form of D/A conversion, but it is only effective for low bandwidth (i.e. $1kHz$ or less) signals. This required the use of some form of external D/A conversion. The author was unfamiliar with the D/A field and had no D/A converter IC's so the original efforts were directed toward conversion using discrete components.

Binary-weighted resistor ladder

If a bipolar supply is used, an Op Amp used as an inverting summer can be used for D/A conversion. The obvious method to use here is to connect each of the output bits to the inverting input of the Op Amp through a set of resistors whose values increase by powers of two. This set of resistors is often referred to as a binary-weighted resistor ladder and is shown in Figure 3. If the resistor connected to BIT 7 has the value $1k\Omega$, the next lower bit resistor would have the value $2k\Omega$, the next lower bit $4k\Omega$, then $8k\Omega$ etc. The lowest-order bit resistor in an eight bit system (BIT 0) with the values described above would then have the value $128k\Omega$.

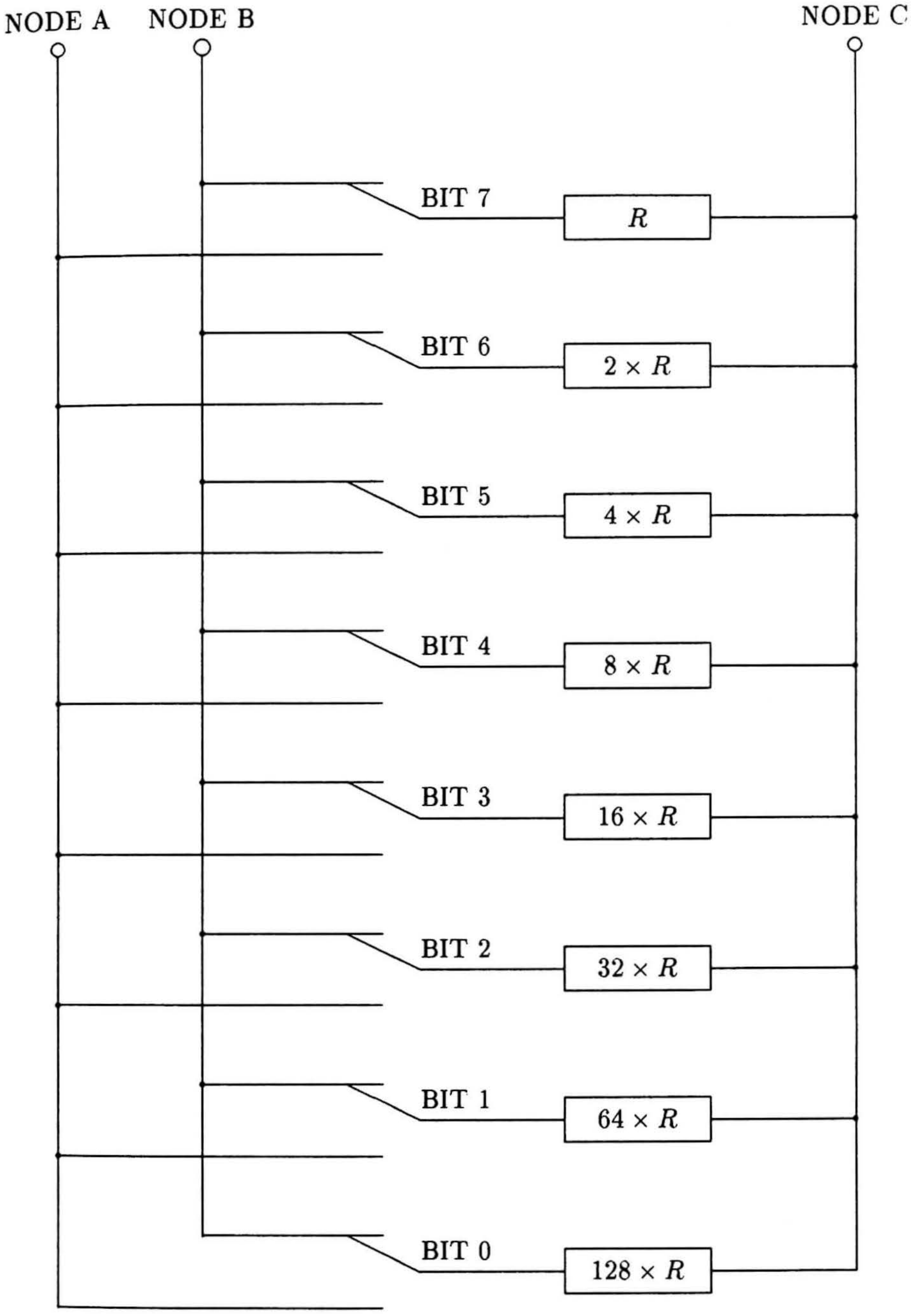


Figure 3. BINARY-WEIGHTED LADDER

The connection at each BIT position is switchable to either NODE A or NODE B. With NODE A connected to a logic-high voltage source, NODE B grounded, and NODE C connected to the virtual ground of the inverting Op Amp, it is fairly easy to see that the current flowing into NODE C is the sum of the currents from the ladder rungs which are connected to NODE A. Because of the binary weights of the resistors, the currents will also be binarily weighted, i.e., BIT 6 contributes half the current BIT 7 would if connected to NODE A, BIT 5 contributes a quarter, BIT 4 an eighth, and so on.

Another possible method of operation for the binary-weighted resistor ladder is to connect NODE A to the inverting input of the amplifier, NODE B to ground, and NODE C to the logic-high voltage source. In this method, assuming that make-before-break switches are used, there will always be current flowing through each rung of the ladder, and any parasitic effects of capacitance and inductance due to the ladder are eliminated. The basic operation of the circuit remains unaffected. Still yet another method of operation involves elimination of NODE A and NODE B and connecting the BIT nodes directly to the outputs of digital IC's. This method is electrically equivalent in function to the first method described. These methods are easily extended to the number of bits desired.

One difficulty with these methods is the need for a negative supply so the Op Amp can be run as an inverting summer. Another difficulty is obtaining resistors with the proper values. If exact powers of two are not used the output will exhibit non-linearity. With an eight-bit design, the lowest-order bit represents 1/256 of the maximum output. This is less than 0.4 percent of the maximum output. If resistors are used with a tolerance any greater than approximately 0.15 percent, the output errors due to tolerance variations in the highest-order bit will swamp the contributions to the output of the lowest-order bit. For each bit added to the converter, the allowable tolerance of the resistors used decreases by one-half.

If a NORTON[9] amplifier is used at the summing junction, a bipolar supply is not required. Since the inputs to the NORTON amplifier, when properly biased, are fixed at approximately one diode-drop above ground, the binary-weighted ladder can be connected to the non-inverting input. To improve accuracy, the IC which supplies current to the bits in the ladder should have a diode inserted in series with its ground lead. This will fix the output-low voltage of the IC at approximately the same voltage as the NORTON amplifier inputs.

With this design, bits which are high supply binary weighted currents to the non-inverting input of the amplifier. Bits which are low act as open-circuits due to the fact that the voltage at both ends of the resistor connected to that bit are approximately equal. Biasing a NORTON amplifier is very easy, and the negative feedback resistor is set in the same manner as with the Op Amp. With the highest-order bit high and other bits low, the feedback resistor is set to give an output voltage equal to one-half the desired maximum.[10] One should remember that NORTON amp outputs are fairly high-impedance (kilo-Ohm range), so the amplifier may need to be buffered with a low-impedance output buffer to give the desired output current.

[9] A NORTON amplifier is a current-differencing amplifier. It subtracts the current into its inverting input from the current into its non-inverting input. With negative feedback an output voltage is produced which is proportional to this difference.

[10] In both Op Amp and NORTON-amp circuits one must always be aware that most amplifiers are incapable of driving rail-to-rail. Proper design requires that the amplifier outputs never be driven into saturation.

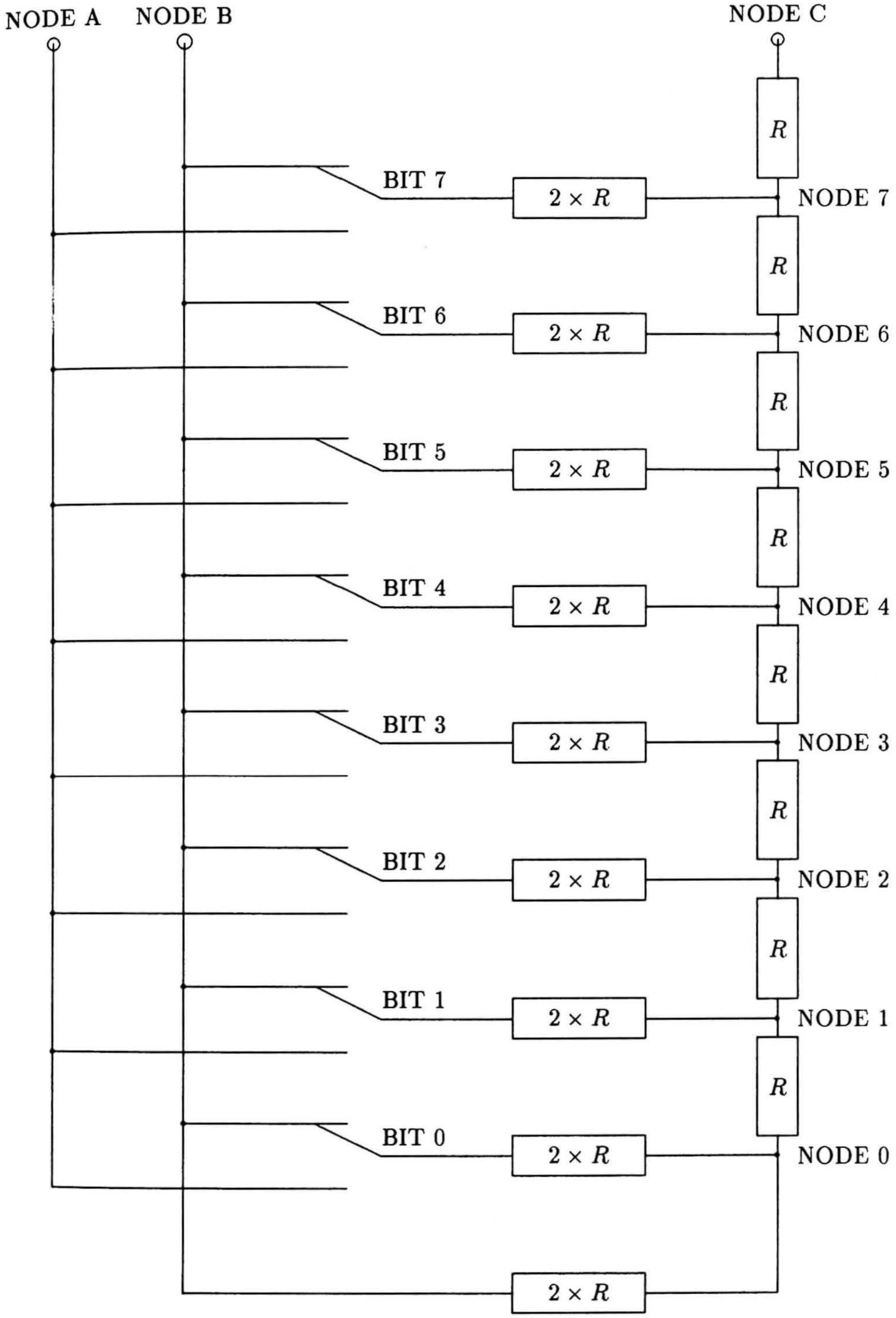


Figure 4. THE R–$2R$ LADDER

R–$2R$ resistor ladder

Figure 4 shows another ladder structure called an R–$2R$ ladder. The name comes from the fact that all of the rungs of the ladder have the value $2 \times R$ while all of the resistors in the leg of the ladder have the value R. For analysis assume that NODE A and NODE B are grounded, and NODE C is connected to a logic-high voltage source. At NODE 0 it is obvious that half of the current flowing into it from the leg of the ladder will flow out the rung connected to BIT 0, and the other half will flow out the other rung connected to NODE 0 because both rungs have the same resistance and both are connected to ground.

Moving on to NODE 1, the effective resistance towards the bottom of the ladder is $2R$ in parallel with $2R$, for a value of R, plus the leg resistance of R, for a total of $2R$. This effective resistance of $2R$ is in parallel with the rung resistance of $2R$, so the same condition exists as existed at NODE 0, and again, half of the current flowing into NODE 1 will exit via the rung to BIT 1, while the other half continues down the ladder towards BIT 0.

The similarity of the upper portion of the ladder leads one to suspect that, as is born out by detailed analysis, the same condition exists at each rung. Since each rung carries half of the current available at that level on the ladder, these currents have binary weighting as in the binary-weighted ladder previously discussed.

A major advantage to the R–$2R$ ladder is that it is easy to obtain resistors that are different by a factor of 2. Also, in integrated circuit manufacturing, it is not easy to obtain exact values of resistance from one batch to another, but it is easy to ratio resistors in a single batch. Since the ratio of values in the R–$2R$ ladder is the critical parameter, it lends itself well to manufacturing processes.

Current mode versus voltage mode

The R–$2R$ ladder can also be used in diverse ways to achieve D/A conversion, just as the binary-weighted ladder. All of the methods fall into one of two categories: voltage mode operation, or current mode operation. The distinction is that in current mode, the $2R$ rungs of the ladder have constant current flowing through them, while in voltage mode, this current is either switched on and off, or it changes magnitude or direction of flow.

The designs described above are two examples of voltage mode operation with a binary-weighted ladder, and one example of current mode operation. An R–$2R$ ladder could easily replace the binary-weighted ladder with no change in operation. It would not be as obvious how the circuits worked, but they would function the same.

For current mode operation, a fixed reference voltage is connected to NODE C at the top of the ladder. Each of the $2R$ rungs is connected to a switch controlled by the corresponding input bit. The rung is connected to one common node, NODE B, when its controlling bit is low, and to another common node, NODE A, when its controlling bit is high. The switches on the ladder rungs are of the make-before-break type so that current flow is uninterrupted during switching.

In a typical converter, NODE B is connected to analog ground, and NODE A is connected to the virtual ground of an Op Amp connected as an inverting summer.

The maximum output voltage is determined by the feedback gain resistor used on the Op Amp. If the feedback resistor has the same value as the R legs in the R–$2R$ ladder, the maximum output voltage is equal to the reference voltage in magnitude, *but of opposite polarity.* In this design, the feedback is common to NODE A.

Several D/A converter IC's which operate in current mode have a built-in feedback resistor to NODE A with its value matching the leg resistances. Although the precise value of the feedback resistor may vary widely from IC to IC, the on-chip matching of values will typically be within 0.1 percent. The AD7528[11] is such an IC, with its R values varying from $2k\Omega$ to $8k\Omega$. The AD7528 also is a dual D/A converter with both converters sharing a common analog ground. This feature is used to advantage in the digitally controlled gain/attenuation module described later.

In a more complex system, NODE B could also connect to the virtual ground of another Op Amp and the system could provide differential D/A conversion. Both of these systems would require bipolar power supplies.

Current mode of operation for the R–$2R$ ladder requires that both NODE A and NODE B be at the same potential. This 'minor' requirement implies that *an Op Amp cannot be used in a noninverting configuration in current mode.* If NODE B is connected to ground through a series diode, NODE A can be connected directly to the noninverting input of a NORTON amplifier and the ladder will still operate in current mode. With a NORTON amplifier as with a standard Op Amp, the negative feedback resistor determines the maximum output voltage. If it has the same value as

[11] Data sheets for all IC's mentioned in this report are readily obtained from the manufacturer data books listed in the bibliography.

the R legs in the R–$2R$ ladder, the maximum output voltage is equal to the reference voltage and, in this case, is of the same polarity. This is an attractive feature when signal inversion must be avoided.

For this configuration, the feedback is not common to NODE A so the built-in feedback resistor in IC's such as the AD7528 cannot be used. This means that if an IC is used, the internal value of R would have to be determined and matched with an external feedback resistor to achieve precise D/A conversion, yet due to the fact that resistor values are affected slightly by operating temperature, the match in values would only hold for a small temperature range. A simple solution to this problem would be for the chip designers to provide a built-in feedback resistor that could be used for either amplifier configuration. This would give close matching with the R–$2R$ legs and also thermal tracking between the ladder and the feedback resistor.

A different method of voltage mode operation is also possible. In this method, NODE A is connected to the fixed reference voltage. NODE C at the top of the ladder which in the previous discussions has been connected to the reference voltage, is instead connected to the noninverting input of an Op Amp. This high-impedance connection acts like an open circuit to the rest of the ladder. A high bit now switches its $2R$ rung to the reference voltage. A low bit switches its rung to ground. If the Op Amp is connected as a noninverting voltage follower, the output will range from ground to the reference voltage.[12] It is easy to see in this configuration that current is not constant in the $2R$ rungs of the ladder. It varies in magnitude and also changes direction when a given rung bit switches from low to high or vice-versa.

[12] Actually the output will range from ground to 255/256 times the reference.

Current mode operation of an R–$2R$ ladder is advantageous because with constant currents flowing in all parts of the ladder, parasitic reactive elements inherent in all of the ladder components have no effect. The frequency response of the converter will be determined solely by the response of the amplifier used. In voltage mode operation, the changing currents and voltages in the ladder cause the parasitic reactive elements to come into play. Because of this, current mode D/A conversion is inherently faster in operation than voltage mode.

Design constraints

In this project one of the design goals was simplicity. For this reason, the author wished to avoid the use of bipolar supplies. Since it was desired to convert $5V$ peak-to-peak signals, the Op Amps used for filtering needed to have a supply that was greater than $5V$. To avoid having to introduce a third supply, it was decided to use the NORTON-amp designs.

In the early stages of the design the author requested samples of several different D/A converter IC's from various manufacturers. In each case the author was informed there would be a five to eight week lead time on delivery. Because of this, a discrete-component D/A was first attempted. Four 74HC373 IC's were used to latch the MC68HC11 D/A output bus. The outputs of two of these IC's were connected to binary-weighted ladders with different starting resistance values. The outputs of the other two IC's were connected to R–$2R$ ladders with different base values for R. In all cases the ladder outputs were connected to noninverting NORTON amplifiers. The ground leads of the 74HC373's were connected through a common series diode to ground. This allowed comparison of the effects of the different impedances of the ladders. All of the circuits functioned well, but the resistors used had a tolerance of one percent, so the accuracy in the low-order bits was severely affected.

Later into the design phase of the project several samples of the AD7528 were received. The AD7528 is a dual eight-bit D/A converter so the project was reworked to use two of them rather than the four 74HC373's and the four discrete-component ladders. The analog-ground leads for the AD7528's were connected to the same series diode to ground. The same NORTON-amps were used on the outputs.

Using a DVM, it was possible to measure the impedance of the built-in feedback resistor and match it fairly closely, but this external resistor would not track across temperature. Because of this, and because the AD7528 can be operated in the modified voltage mode described above, it was decided to change operation to the modified voltage mode. Although this sacrifices some conversion speed, the feedback resistor is eliminated along with the process of determining its proper value, and the series diode to ground is no longer needed. Since the conversion rate in the new configuration was still several orders of magnitude faster than required, and the new design had a lower component count, the trade-off was considered advantageous. The new configuration was tested and worked well. The code for addressing the AD7528's was shorter than that required for the 74HC373's so the effective sampling rate increased slightly.

The author then received two samples each of the AD7225 and the AD7226. These IC's are quad eight-bit D/A converters, operating in voltage mode, with built-in noninverting voltage-follower Op Amp outputs. The AD7225 has four more pins than the AD7226, but allows separate reference voltages for each D/A, and has a common *load-dac* strobe that allows simultaneous changes on the D/A outputs. The AD7226 has a common reference voltage input, and writes to a D/A channel update the corresponding output immediately.

The internal circuitry of the AD7226 was virtually identical to the pair of AD7528's together with the four voltage-follower Op Amps which were being used in the D/A section in the project. By incorporating the AD7226 into the design, the voltage-follower Op Amps were eliminated, and the two AD7528's were replaced with one AD7226. Programming for the AD7226 was even simpler, so the loop time dropped even more, to less than the A/D sample time on the MC68HC11. This meant that the software was now waiting on the A/D converter.

Signal reproduction noise

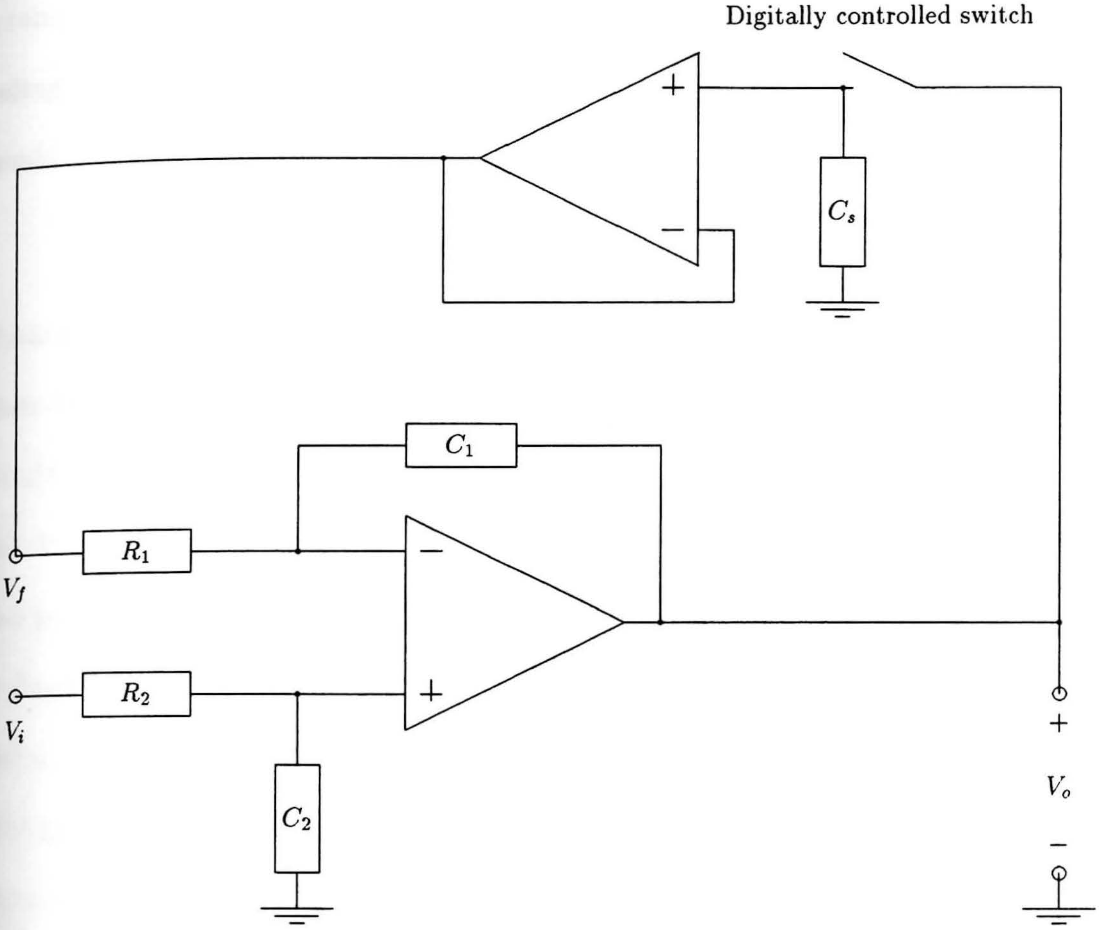
Any time that a signal is produced through D/A conversion there is switching noise added to the signal. Part of this noise is due to the fact that the D/A converter is only capable of producing a fixed number of outputs. Any transition from one level to another occurs as a step-function, with its accompanying odd-harmonic noise. The base frequency of this noise will be the rate at which new values are sent to the D/A converter. In this project, this is the same as the sampling rate.

Another part of this noise is due to a phenomenon known as 'glitch' noise. Glitch noise occurs when some bits in the D/A converter affect the output faster than others. If, for example, the highest-order bit in an eight-bit D/A converter affects the output slightly faster than the other bits, whenever the D/A value changes from binary 01111111 to 10000000, the output will momentarily respond as if the D/A value was 11111111. For a transition in the opposite direction, the output will momentarily respond as if the D/A value was 00000000. This spike is more prevalent in voltage mode converters due to the parasitics mentioned. In current mode converters, all the switches transition simultaneously and current flow is unchanged, so glitch noise is reduced considerably or eliminated.

Because of the noise inherent in the D/A conversion process it was decided to test both second-order and fourth-order Sallen-Key low-pass filters identical to those used on the input stages for filtering of the D/A outputs. Testing with a $3kHz$ sine wave showed that a second-order filter appeared to provide acceptable response. Audio tests revealed, however, that what appeared to be insignificant on an oscilloscope was easily detected by the ear. Even a fourth-order filter still left much of the step noise from the D/A conversion. Moving to a $16MHz$ MC68HC11 doubled the sample rate and the frequency of the D/A conversions, but the noise was still evident.

Sampled-Difference Integration (SDI)

The search for a method of eliminating more noise without using higher-order filters led to the design of the sampled-difference integration module, shown in Figure 5. Because the primary culprit of the noise in the reproduced signal was the step-function associated with each update of the D/A converter, elimination of that step-function was the emphasis of the search.



$$R_1 = R_2 = R$$
$$C_1 = C_2 = C$$
$$C_3 = 4.7\mu F(typical)$$
$$V_o = \frac{1}{R \times C}\int_t (V_i - V_f)dt$$
$$\omega_o = \frac{\sqrt{2}-1}{R \times C}$$

D/A convertor output is connected to V_i.
Digitally controlled switch samples V_o at four times the update rate of the D/A.
The integrator acts as a low-pass filter. Set its cutoff using the last equation above.
V_o is connected to the input of the signal reproduction low-pass filter.

Figure 5. SAMPLED DIFFERENCE INTEGRATOR

The solution which was considered first was to integrate the step to generate a ramp. This would work fine if the D/A converter output both positive and negative voltages, but the one chosen only outputs positive voltages, and simple integration would lead only to amplifier saturation, and quickly.

The next solution considered the use of two D/A channels for each output channel and to subtract one from the other using a difference amplifier which would then drive an integrator. The D/A channel connected to the positive amplifier input would be updated with the present sample value for a given output channel, while the D/A channel connected to the negative amplifier input would be updated with the previous sample value for the same channel. This solution would work, since the voltage output by the difference amplifier would be equivalent to $V_t - V_{t-1}$, and would swing both positive and negative depending upon whether the present sample value was greater or lesser than the previous sample value, but it doubled the number of D/A converters needed, and *substantially* increased the software loop time due to the added code required to maintain the data and handle the new converters.

Also considered was the use of a difference amplifier feeding an integrator with the negative input of the amplifier connected to the output of the integrator. Analysis of this configuration indicated that it had the exact same transfer function as a simple low-pass *R–C* single-stage network. During the analysis of this circuit, however, the author found that the two separate stages[13] could be combined into one circuit with proper choice of components. This circuit is the first stage of the final SDI module shown in Figure 5.

[13] The difference amplifier followed by the integrator.

To modify the response of the circuit, it was decided to add a sample-and-hold function to the output of the difference integrator, and feed the output of the sample-and-hold to the negative input of the difference integrator. With this modification, the circuit began to give desired results. With the digitally-controlled switch closed, the sampling capacitor C_s follows the output of the difference integrator and the circuit responds as if the sample-and-hold did not exist. When the switch is opened, assuming the voltage droop on C_s is essentially zero between samples, the output of the module is a ramp, *whose slope is determined solely by the difference between the present D/A output voltage and the sampled voltage stored on C_s*. The circuit elements are chosen so that the output voltage would be able to ramp from zero volts to the supply voltage in one D/A update period. This allows the circuit to respond to the worst-case step from the D/A converter.

Operating the sample-and-hold at four times the D/A update frequency gives four distinct slopes for each update period. Since the discontinuities associated with the original step-function and the slope changes are a major contributor towards the noise on the output channel, the circuit has effectively multiplied the frequency of that noise by four, which moves it two octaves deeper into the cutoff range of the following fourth-order filters. Those two octaves translate into $12dB$ additional attenuation. The four-times update rate is easily achieved by sampling every output channel each time the D/A value is updated for one channel. This was achieved in this project through the method described next.

Pin 7 of the MC68HC11 micro-controller is set up in software to generate a pulse every time a write occurs to port c of the controller.[14] This signal is gated through one of the gates in U10[15] together with bit 5 of port d which is used as a select line for the AD7226.[16] The resulting signal, $\overline{W7226}$, is used to enable the sample-and-hold stage on every output channel each time one of the output channels is updated.

Also contributing to improved performance is the fact that just changing from step functions to ramps decreases the noise due to higher order harmonics of the signal. Figure 6, generated using the PSPICE simulation program, shows the signal strengths in the frequency spectrum for a $4kHz$ square wave and a $4kHz$ triangle wave for $2kHz$ to $30kHz$.[17] The vertical axis is logarithmic. It is evident from the figure that there is considerably less signal strength in the higher-order harmonics of the triangle wave than in those of the square wave. At the fundamental frequency of $4kHz$ the triangle wave has slightly lower strength. At $12kHz$ the triangle wave is nearly $10dB$ down from the square wave. At higher frequencies, the attenuation is even greater. The combined results of converting the DAC output to a ramp rather than a step, and operating the SDI sampling switch at four times the channel sampling rate greatly reduce the noise in the reproduced signal.

[14] Port c is used as a parallel data bus for all of the chips in the design which require parallel data, in this case the AD7226 and the four AD7528 chips.

[15] 74HC00, a quad NAND-gate chip.

[16] When port d bit 5 is high, writes to port c send data to the AD7226 chip. When port d bit 5 is low, writes to port c send data to the AD7528 chips.

[17] The $4kHz$ frequency was chosen because it is slightly above the designed-for maximum of $3kHz$ and because the module runs fast enough that it can sample and reproduce a signal of that frequency easily.

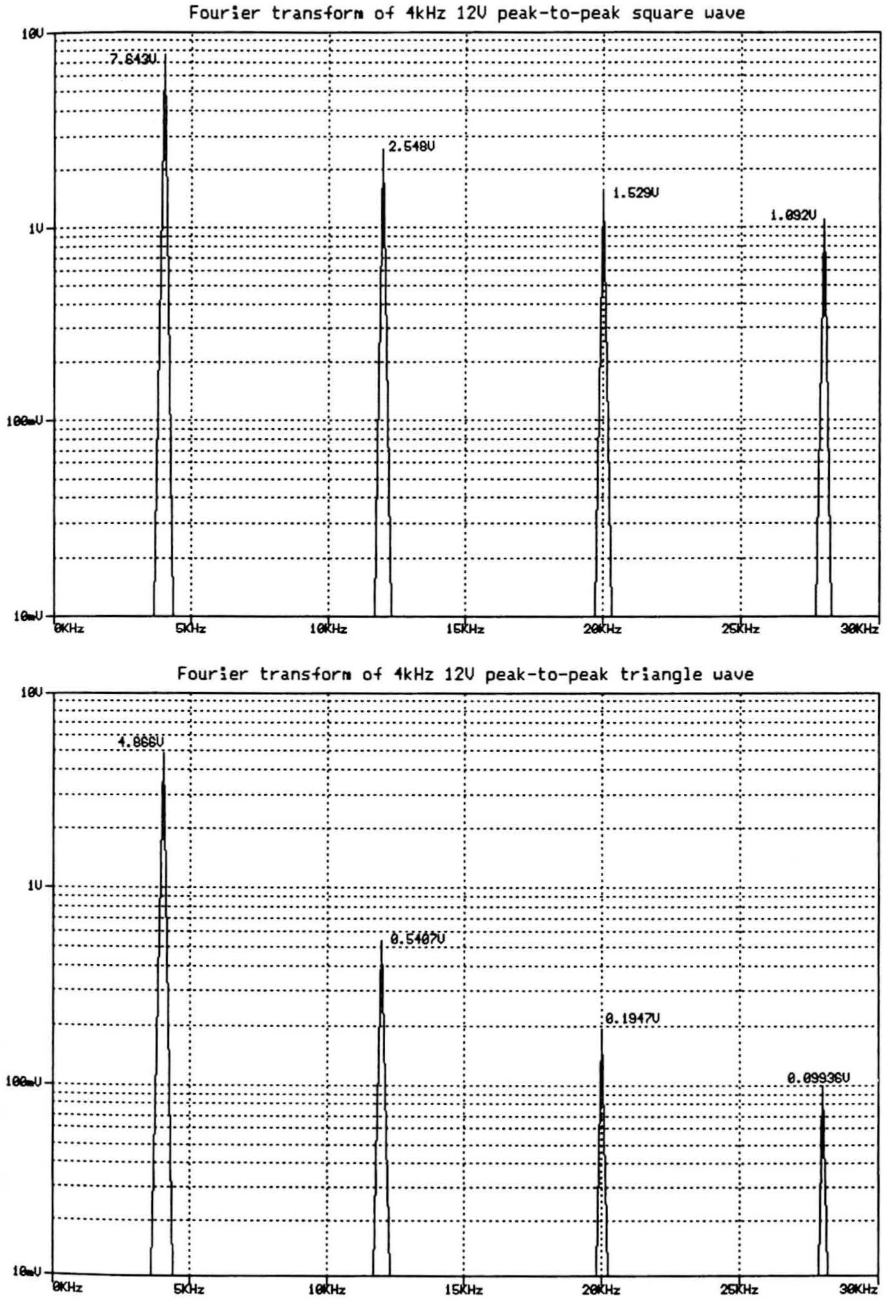


Figure 6. MAGNITUDE SPECTRA FOR 4kHz WAVES

Digitally-Controlled Gain/Attenuation (DCGA)

Hardware design

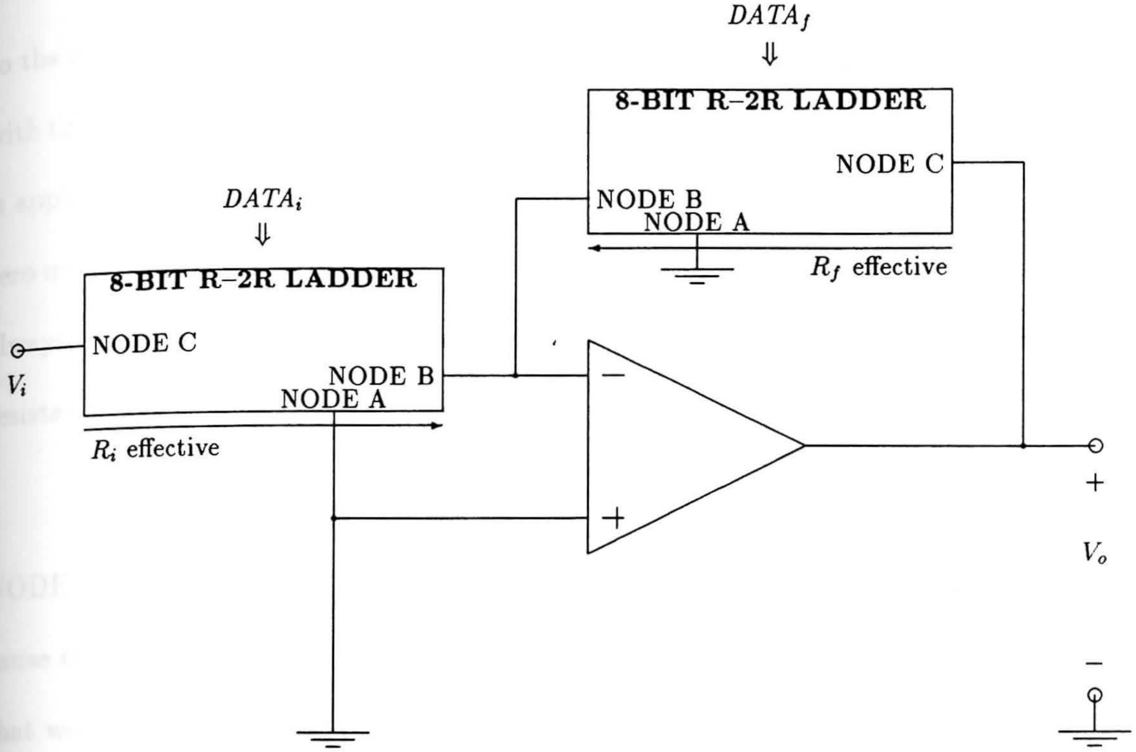
The last addition to the project affected the input channels. Figure 7 shows the diagram of a single digitally-controlled gain/attenuation stage. The circuit uses an AD7528 and an Op Amp connected as an inverting amplifier. One half of the AD7528 is used as R_i, and the other half is used as R_f. The nodes labeled NODE A in Figure 7 and Figure 4 correspond to the OUTa and OUTb pins on the AD7528. The nodes labeled NODE B correspond to the common AGND pin. The nodes labeled NODE C correspond to the REFa and REFb pins. This circuit takes advantage of the fact that the reference voltages for the AD7528 can be AC signals of up to $100kHz$.

Examination of one of the converters reveals that for a data value of zero, all of the $2R$ rungs of the ladder are connected to NODE B.[18] In this state, maximum current flows to NODE B, and no current flows to NODE A.[19] As the data value increases towards its maximum of 255[20] the current flow to NODE B decreases and the current flow to NODE A increases by exactly the same amount. When the value of 255 is reached, maximum current flows to NODE A, and minimum current flows to NODE B. A little thought reveals that the effective resistance from NODE C to NODE A decreases as the data value increases, and the effective resistance from NODE C to NODE B increases as the data value increases.

[18] Figure 4 shows the details of the R–$2R$ ladders.

[19] Disregarding any leakage current in the device.

[20] Any eight-bit data can have a value from 0 to $2^8 - 1$ or 255.



$$\frac{V_o}{V_i} = \frac{R_f}{R_i}$$

$$R_f = R_{max} \times (1 + DATA_f)$$

$$R_i = R_{max} \times (1 + DATA_i)$$

$$\Rightarrow \frac{V_o}{V_i} = \frac{1 + DATA_f}{1 + DATA_i}$$

See Figure 4 for internal structure of the R–$2R$ ladders.
AD7528 is a dual D/A with NODE B common to both sections and called AGND.
Gain is set by the ratio of the D/A control data of the two ladders.

Figure 7. DIGITALLY-CONTROLLED GAIN/ATTENUATION

The current flowing to NODE B when a data value of 255 is applied is equal to the amount of current controlled by the BIT 0 rung of the ladder. It is impossible with this design to reduce the current flow to NODE B to zero when a nonzero voltage is applied to NODE C, but it is possible to reduce the current flow to NODE A to zero under the same conditions. This fact is critical to the DCGA function. One path always has some finite resistive value, while the other has a potential state of infinite resistance.

When considering the use of the D/A converters, both the NODE C—NODE A (CA) path and the NODE C—NODE B (CB) path were examined. Because the CA path allows the selection of infinite resistance, it was rejected because that would make possible the selection of a feedback resistance of infinite value, effectively turning the input amplifier into a comparator which is not very useful for voice communications. Use of the CB path makes it impossible to select either zero resistance or infinite resistance. Because of the extra $2R$ rung connected to NODE B, the resistance values along CB can vary from R to $256R$. Connecting the two halves of the AD7528 as shown allow software to select a gain from 1/256 to 256/1.

Software control

Software keeps track of the peak values obtained for each of the input channels. Approximately 100 times each second, the peaks are checked to verify that they are in the range of 192 to 255 decimal.[21] Channels whose peaks fall below the limit have their gain increased. Channels which are being clipped at the 255 value

[21] These values were selected as $0.75 \times V_{max}$ (192) and signal clipped (255), based on the availability of values from 0 to 255 in an eight bit system.

have their gain decreased. Each time the gain is adjusted for a given channel, the peak stored for that channel is reset to zero. The adjustment rate was chosen to be fairly frequent but to have minimal impact on the signal sampling.

To simplify the adjustment method for the DCGA, it was decided to store a single value as the 'gain' setting for a given channel. This value is a single byte, from 0 to 255. A setting of 0 represents minimum gain or maximum attenuation. A setting of 255 represents maximum gain or minimum attenuation. Using this value as $DATA_f$ and the complement[22] value as $DATA_i$, a situation occurs where as the gain for a channel increases, the effective value of R_f in its DCGA stage increases and the effective value of R_i decreases. There are 256 distinct gain settings with this method. The final equation from Figure 7 indicates that the gain is determined by the equation $(1 + DATA_f)/(1 + DATA_i)$. Since $DATA_i = 255 - DATA_f$, this equation becomes $(1 + DATA_f)/(256 - DATA_f)$ with $DATA_f$ ranging from 0 to 255.

[22] The complement is obtained by subtracting the value from 255. In an eight bit system, the sum of a number and its complement is always 255.

CHAPTER III

MICRO-CONTROLLERS

A micro-controller is a microprocessor with dedicated ROM, RAM, I/O ports, and communication capabilities on a single IC. Often there are additional capabilities such as asynchronous timers, pulse width modulation, A/D—D/A conversion, EPROM/EEPROM, and more. They are available from a variety of manufacturers: Motorola, TI, Harris, Intel, Hitachi, and others.

The micro-controllers from Motorola are based around the 6800 and 68000 families of microprocessors. Those from TI are based around their TMS7000 series. The RTX series of micro-controllers from Harris are unique in that their assembly language is the language FORTH, and they are stack oriented. Intel makes the popular 8051 and similar chips. Hitachi has the 64HD180, based around the ever-popular Z80 microprocessor. Within any given family, there is usually a wide range of options and even processor speeds from which to select.

Each of these families is radically distinct from the others, yet they all fall into the general description given above. All of these companies manufacture eight bit micro-controllers. Most of them manufacture sixteen bit controllers. The author is only aware of one thirty-two bit family, that being the 68332 from Motorola, designed especially for General Motors.

Micro-controllers are used in extremely diverse areas, from the factory floor to the automotive engine compartment, from hot, dry desert labs to the polar regions. The nonvolatile memory included in many of them allows storage of data even if power is lost. The extremely low power consumption of some of the CMOS parts allows their use in battery powered equipment that is placed on site and serviced sometimes as seldom as months apart. Virtually all of the manufacturers are willing to work with volume customers to create special configurations, which they then can market to the rest of their customers who may not have the financial means available to ask for special purpose designs. Because of this, new members of the micro-controller families are constantly being introduced.

This author has extensively used a member of TI's TMS7000 series, and the MC68HC05 and MC68HC11 from Motorola, designed around the 6800. These two families are radically different in that the 7000 series is register oriented with a large number of registers that can also be accessed as RAM and be used as index registers or accumulators. The 6800 family has fixed accumulators and index registers and storage RAM. The MC68HC05 has one accumulator and one index register, while the MC68HC11 has two of each. The 7000 series and the MC68HC05 series are similar in that the index register contains a eight bit value, usually combined with a sixteen bit offset contained in the instruction. In the MC68HC11 the index register contains a sixteen bit value, which is combined with an eight bit value contained in the instruction. The 7000 series is available with faster clocks, but the 6800 family generally is packaged with desirable peripheral options. The MC68HC05 can generally execute a program written to perform a given time-critical task slightly faster than the MC68HC11 can, but with fewer registers available it is harder to code the MC68HC05 efficiently. Cost considerations are also a factor. The 7000 series, being older technology is generally cheaper than the MC68HC05 or the MC68HC11. The MC68HC11 is presently not quite twice as expensive as the MC68HC05.

The goal of the preceding paragraph was to illustrate to the reader that in choosing a micro-controller one must weigh the options and capabilities carefully against the requirements of the project. Trade-offs between performance and cost must be considered. Peripheral package options must be compared with the needs of the design. A new micro-controller, although initially much more expensive than older ones, may over the course of the design cycle reduce in price to where it is a better choice, but one must always remember that in the world of high volume production, a few pennies can make or break a product.

CHAPTER IV

SELECTED DESIGN PROBLEMS AND SOLUTIONS

Throughout the course of this project various problems were encountered. Some were solved by the correction of errors, others were inherent in the design choices made. In all cases, solutions were attempted with the goals of functionality and simplicity. Some of the problems are presented here, together with steps taken to resolve them.

- During the first functional tests of the software, it was discovered that the system was having problems with signals at a higher frequency than about $2.5kHz$.

  — Observing the input to the A/D converter with an oscilloscope showed that the MC68HC11 was receiving a good signal. Observing the output of the D/A converter showed that the signal was bad prior to filtering. The wiring checked out fine. This led to questioning the sampling rate. Theoretical calculations were double-checked and indicated there should be no problem obtaining the proper rate. Examination of the software revealed that, rather than processing the previous A/D pass results while a new pass is occurring, the main loop was starting a pass, waiting until it was complete, then processing the results. The loop time for just the original version of the software was over 110 μs. This was in addition to the 64 μs required for the A/D conversion. The total loop time was in excess of 175 μs, in excess of N_i necessary for reproduction of a $3kHz$ signal. *I had accidentally proved the Nyquist theorem.* Reworking the software so it worked as originally intended solved the problem, bringing the loop time down to approximately 110 μs, well within N_i.

- Observed significant errors in outputs of NORTON-amp D/A converters during DC testing.

  — The finite voltage at the NORTON-amp inputs had not been considered. Adding a diode to the ground lines of IC's as described previously eliminated most of the error. There was still a potential difference of approximately $0.08V$ between the NORTON-amp inputs and the top of the diode. Adding a voltage-follower whose output drove the top of the diode and whose noninverting input was connected to the input of the NORTON-amp was considered and rejected as overkill.

- First attempt at multi-channel communications failed miserably. The only channel with the proper signal on it was channel zero.

  — Double checking the wiring revealed that the A/D inputs were wired incorrectly. While all of the other input ports on the MC68HC11 were ordered sequentially on adjacent pins as 0, 1, 2, 3, 4, 5, 6, 7, the A/D port was interleaved as 0, 4, 1, 5, 2, 6, 3, 7. Checking the outputs revealed that, indeed, the channel two input was showing up as the channel one output. Corrected wiring and all four channels operated properly.

- Input signals were being clipped by the filters.

  — The author had neglected to ac-couple the input signal and to bias the filter inputs. This caused clipping of the negative half of input waveforms. Proper ac-coupling and biasing solved the problem.

- After replacing AD7528's with the AD7226 and modifying the software to access the AD7226, all of the output channels were garbled. On the oscilloscope, with an input applied to one channel, it appeared as if the signal was somehow coupled to the other channels. Every channel had a nearly constant dc voltage interrupted at less than the sampling rate with what looked like segments of the input waveform.

  — Operator Error(!!)

  In rewriting the software to access the AD7226 and tweaking it to obtain slightly faster loop times, a command to increment the address to the next location was omitted. This meant that the address was changing only three times through the loop. Data which should have been written to channels sequenced as 0, 1, 2, 3, 0, 1, 2, 3, ... was instead being written to channels sequenced as 0, 1, 2, 2, 3, 0, 1, 1, 2, 3, 0, 0, 1, 2, 3, 3, ... and as luck would have it, channels 1, 2, and 3 all had no signal on them and were biased to the same dc voltage. The information from the one channel with a signal on it was being distributed to all of the other channels.

CHAPTER V

CONCLUSIONS

The project demonstrates the feasibility of the design. Although unable to implement DSP as desired, the module could, with little modification, be used as the basic building block of a four-line digital telephone system. In such a system the EVB would be replaced by a single MC68HC11 operating in single-chip mode.

The project also effectively demonstrates how the correct micro-controller with proper support circuitry can greatly simplify a design. The estimated total cost for the parts used, in volume, is under fifty dollars.

Areas For Further Development

The final module exhibited a phenomenon referred to as surging, caused by too rapid response of the DCGA module. Some investigation of closed-loop response in a digitally controlled system is in order for further refinement of this module. Also, the attenuation range of the DCGA should be reworked to cover a more realistic signal range. Another area for investigation is making the module adaptable to use its full capacity on only the channels actually being used for communication. This would enable a single high-quality channel to be transmitted if no other channels are being used.

Several D/A converter IC's have the property of allowing an ac signal to be applied to the reference input. Some of them allow the reference voltage to have positive and negative polarity. These IC's have the property of being able to provide either digitally controlled gain or attenuation as evidenced by the DCGA section, or two-quadrant or four-quadrant multiplication. The AD7528 is one such IC. The reference voltage supplied to this IC can be a signal with frequency content of up to approximately $100kHz$. This capability could be used to create a digitally modulated four-channel transmitter using the MC68HC11 micro-controller. Another possible area of development is the use of fiber-optics for the transmission media.

APPENDIX A

PHOTOGRAPH OF FINAL BREADBOARD

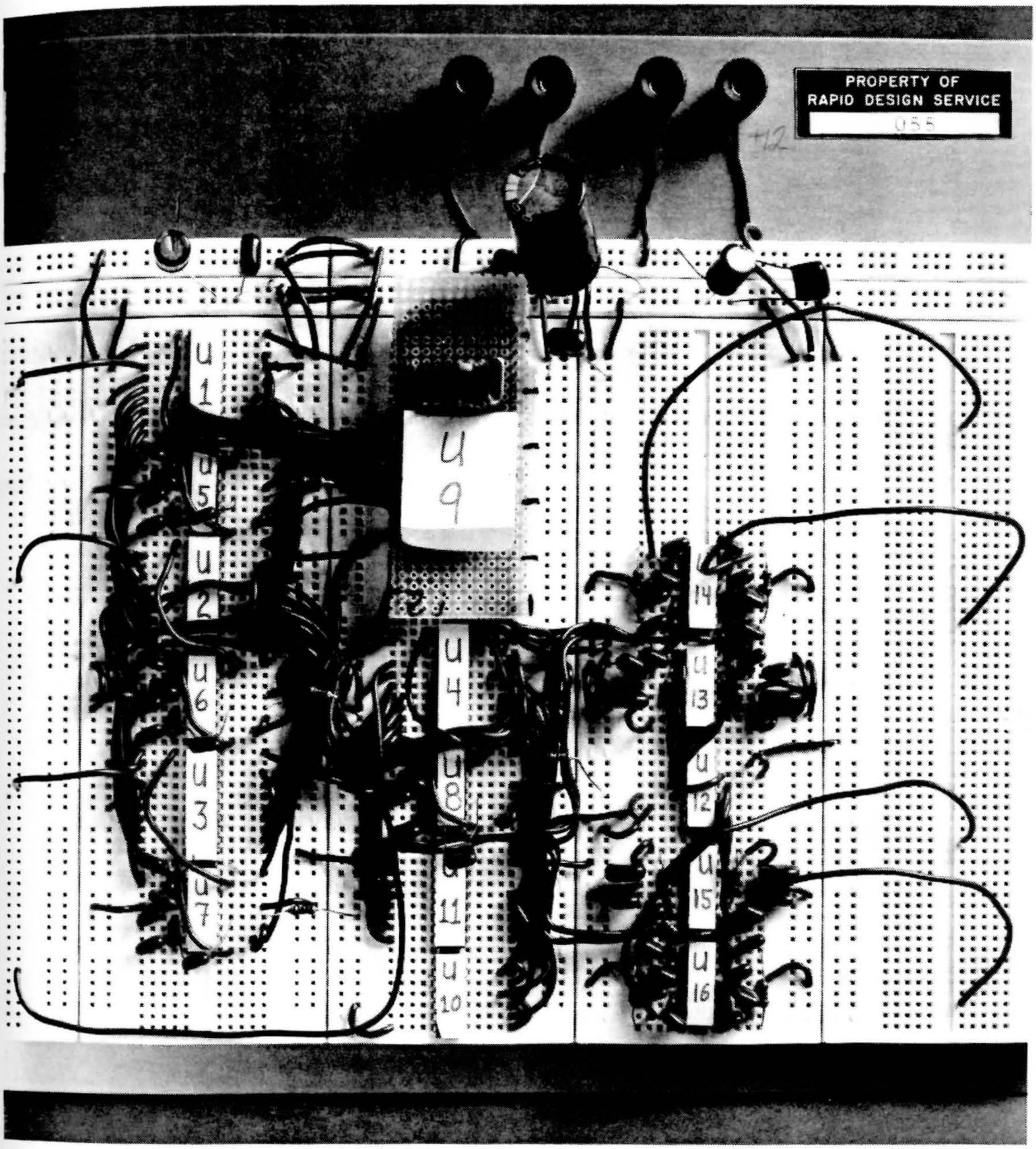


Figure 8. PHOTOGRAPH OF FINAL BREADBOARD

APPENDIX B

SCHEMATIC OF FINAL DESIGN

NOTE: Due to the size of the final schematic, it is presented here first on one page in nearly unreadable format so the reader can see where the various pieces that follow fit in relation to the complete circuit, then in pieces on successive pages at various levels of magnification so the reader can see the signal names, pin numbers, part values, etc.

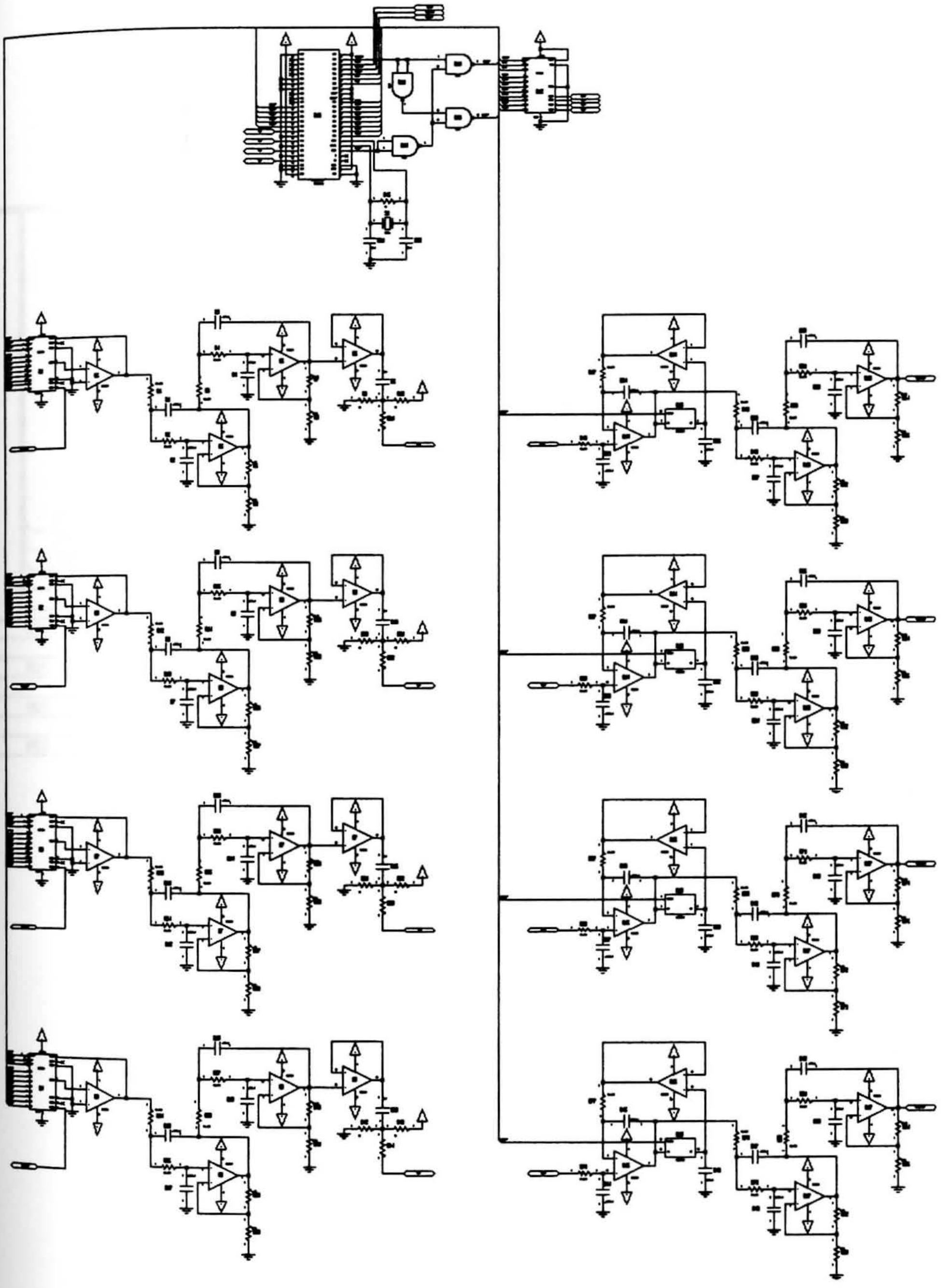


Figure 9. COMPLETE SCHEMATIC

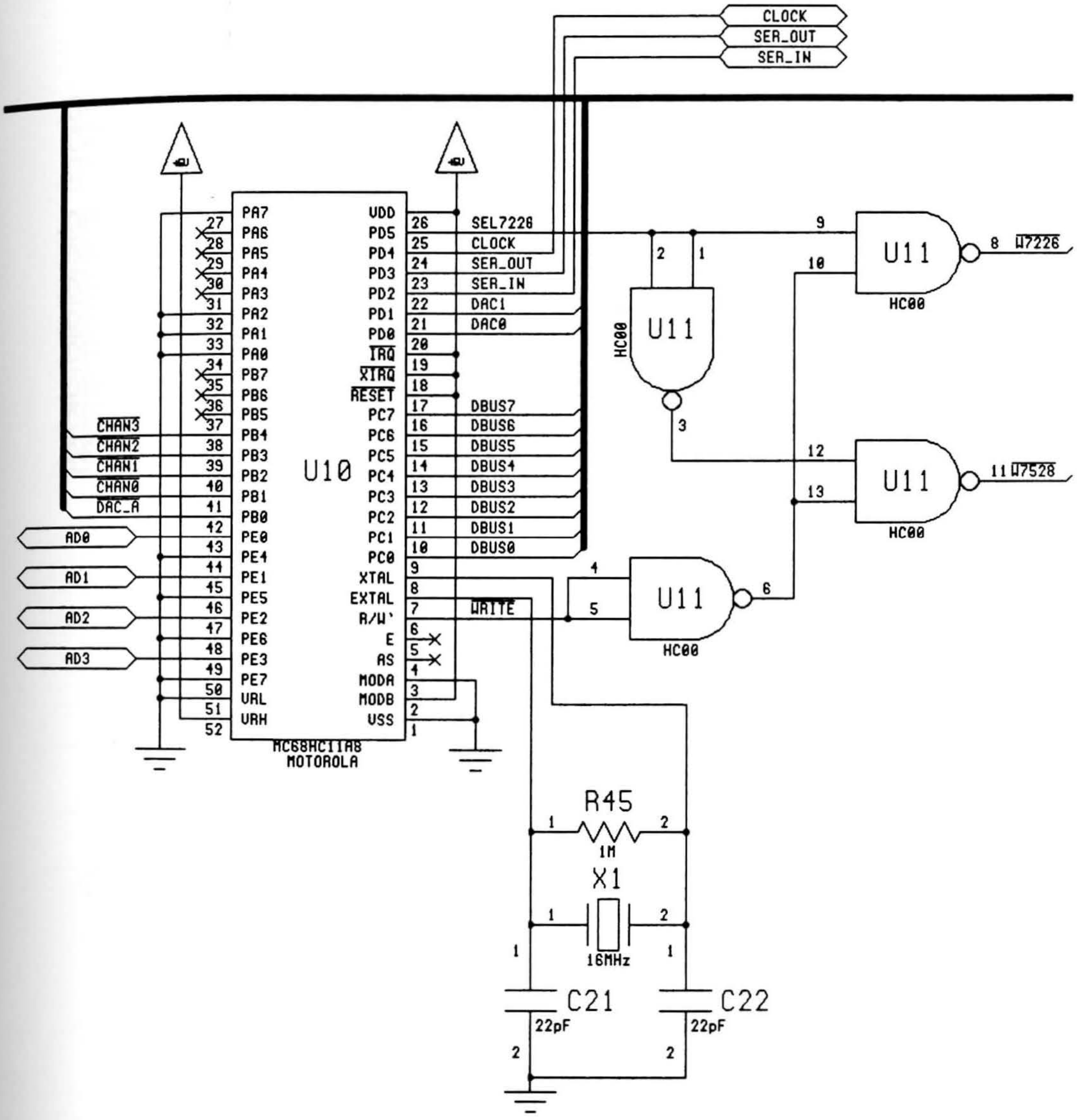


Figure 10. MICRO-CONTROLLER AND SUPPORT CIRCUITRY

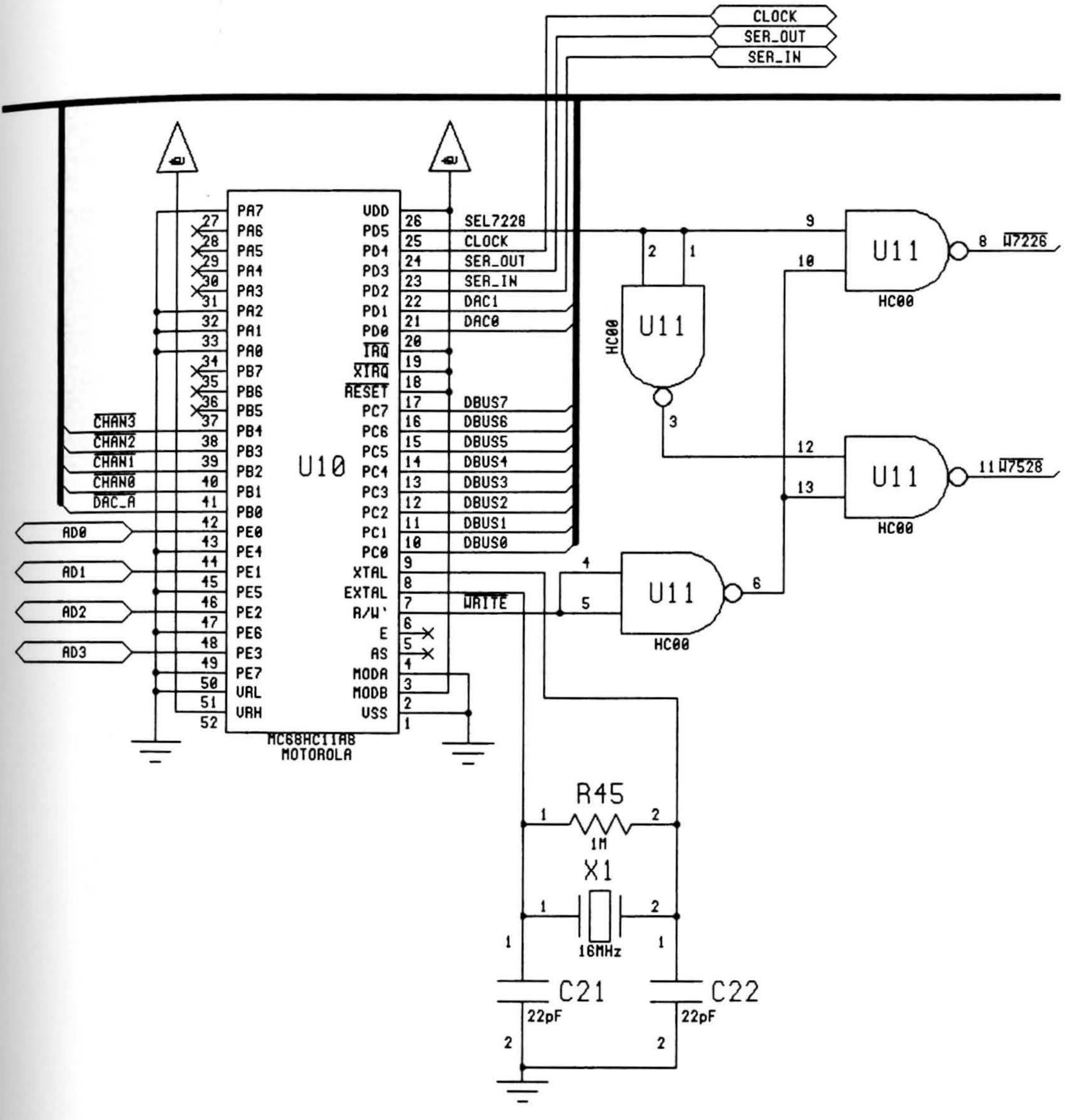


Figure 10. MICRO-CONTROLLER AND SUPPORT CIRCUITRY

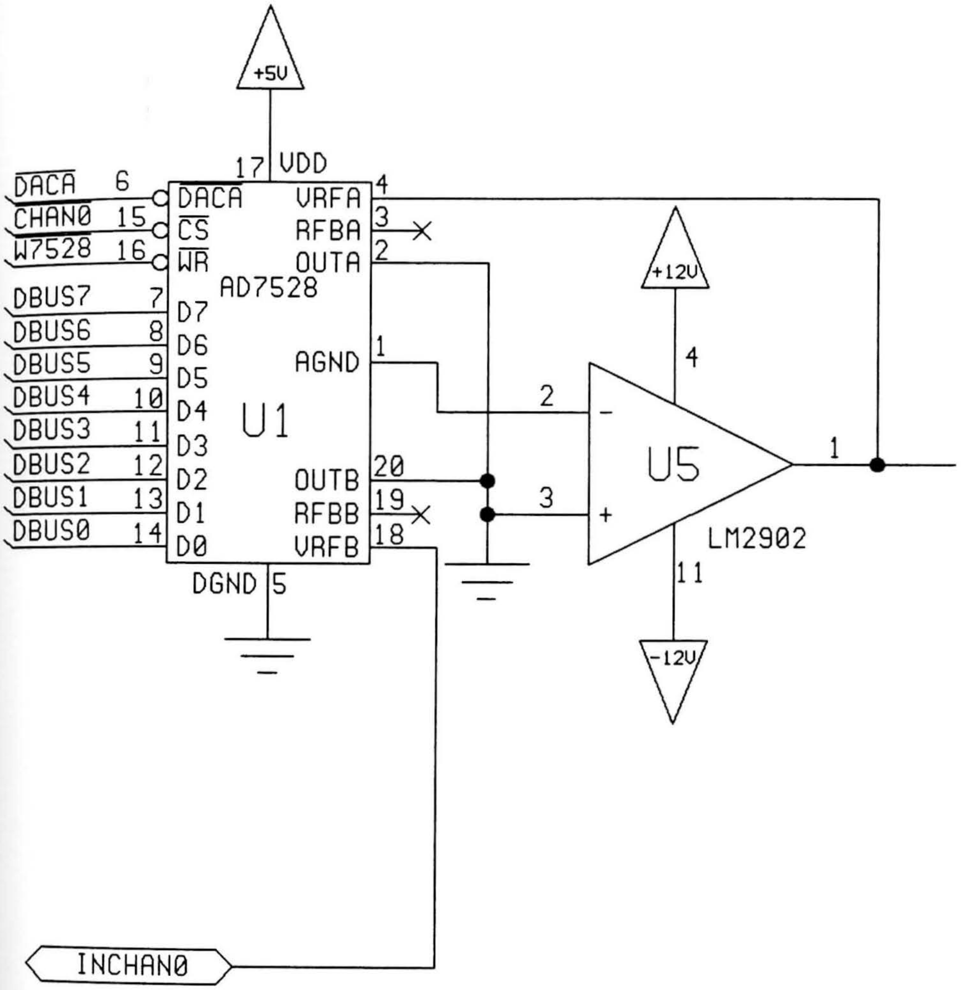


Figure 11. DCGA FOR INPUT CHANNEL 1

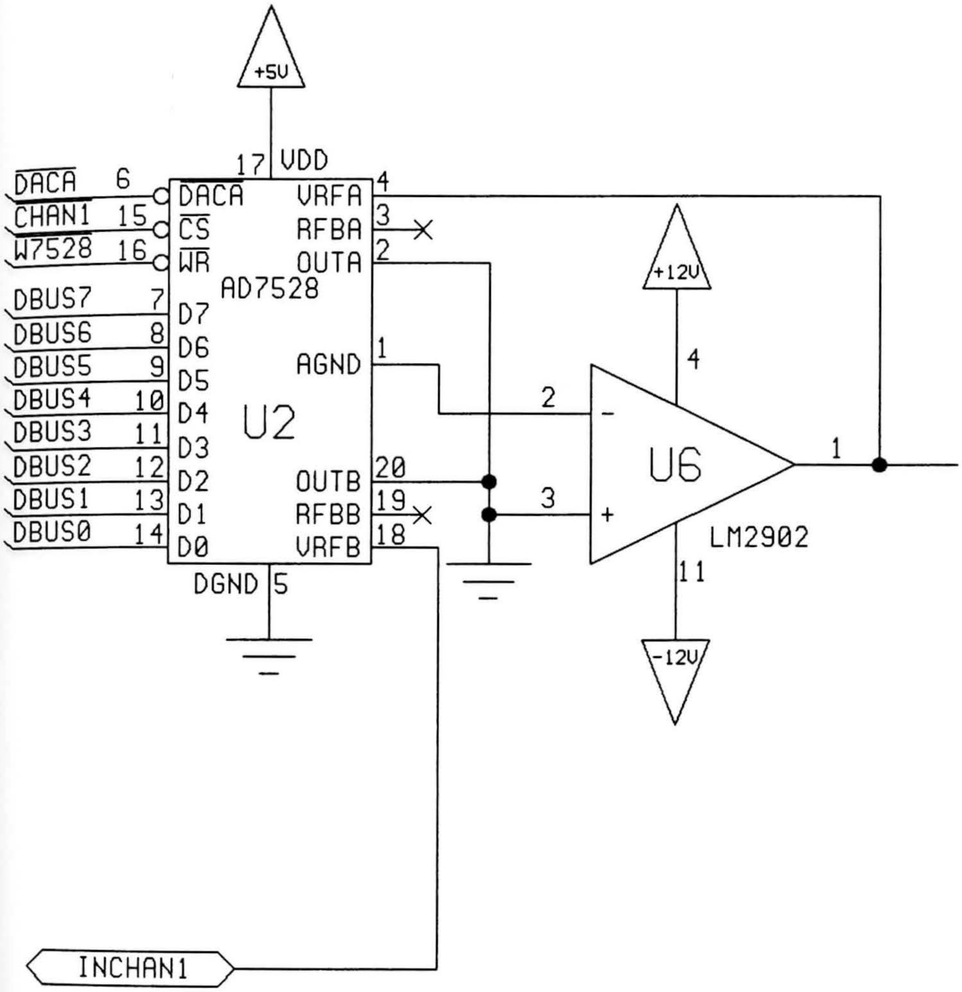


Figure 12. DCGA FOR INPUT CHANNEL 2

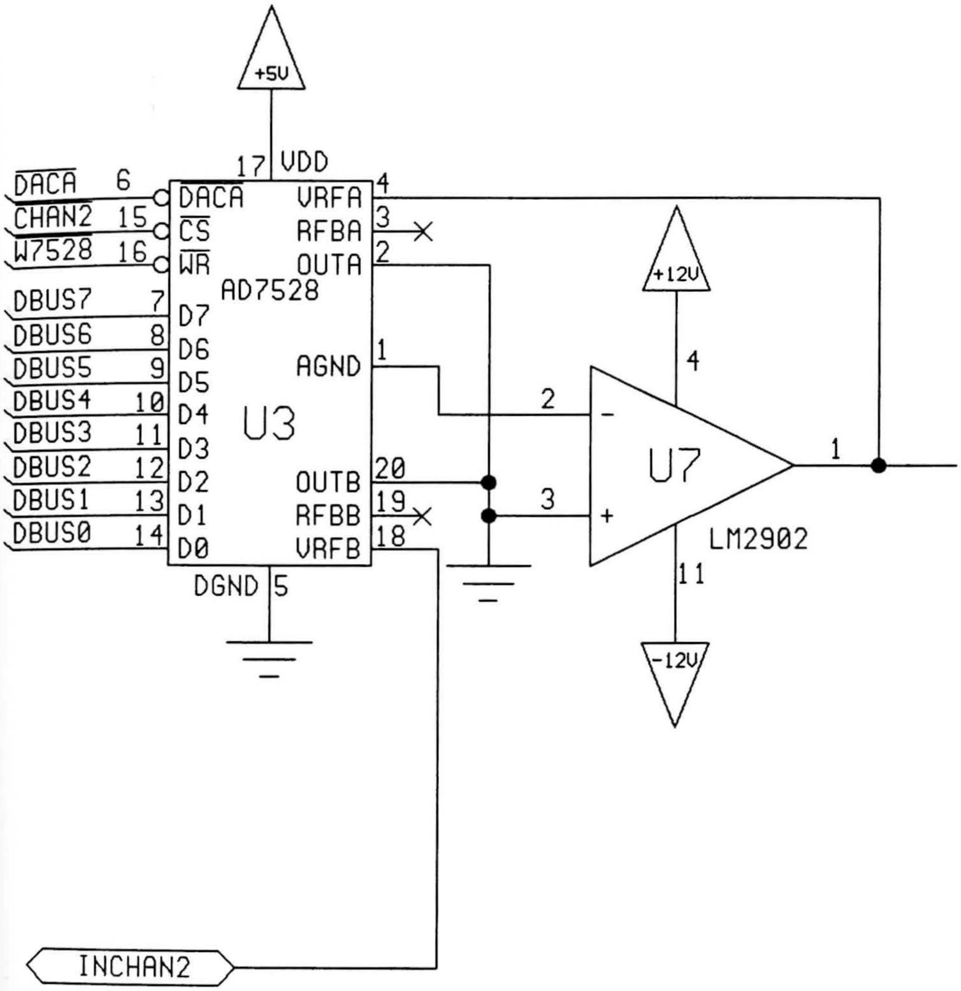


Figure 13. DCGA FOR INPUT CHANNEL 3

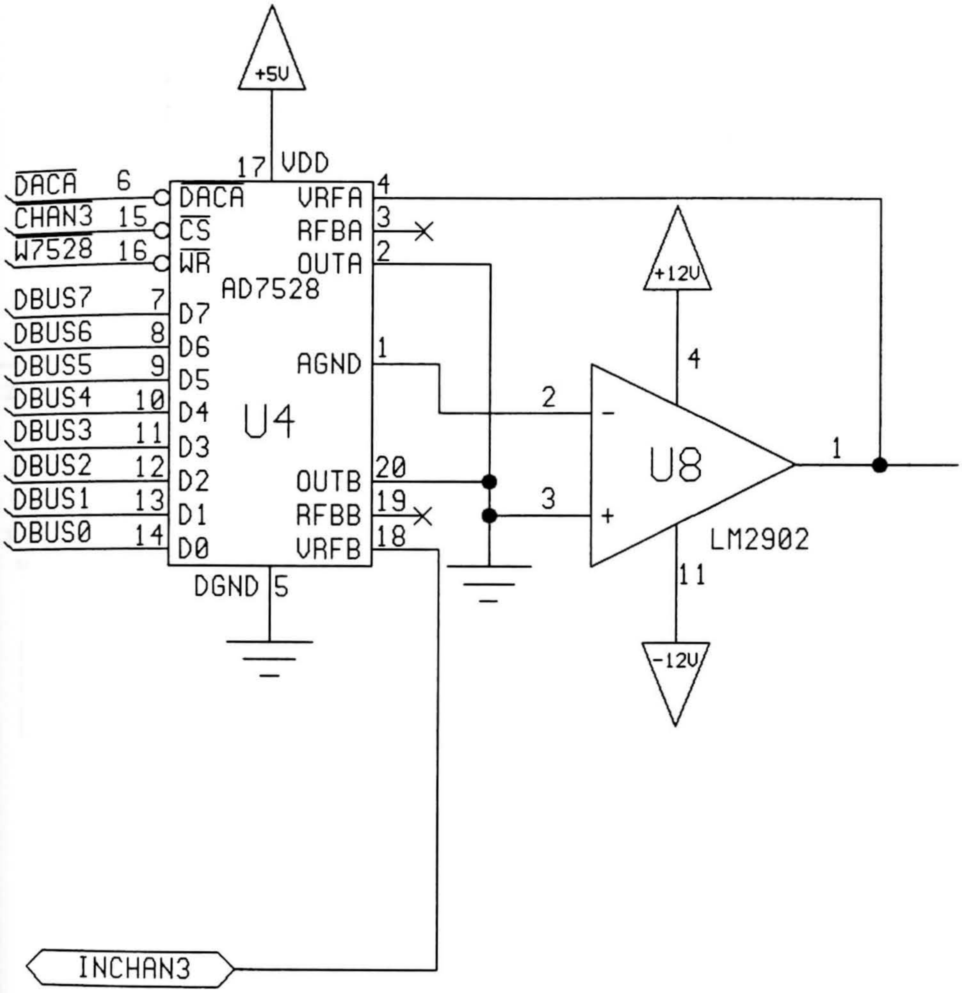


Figure 14. DCGA FOR INPUT CHANNEL 4

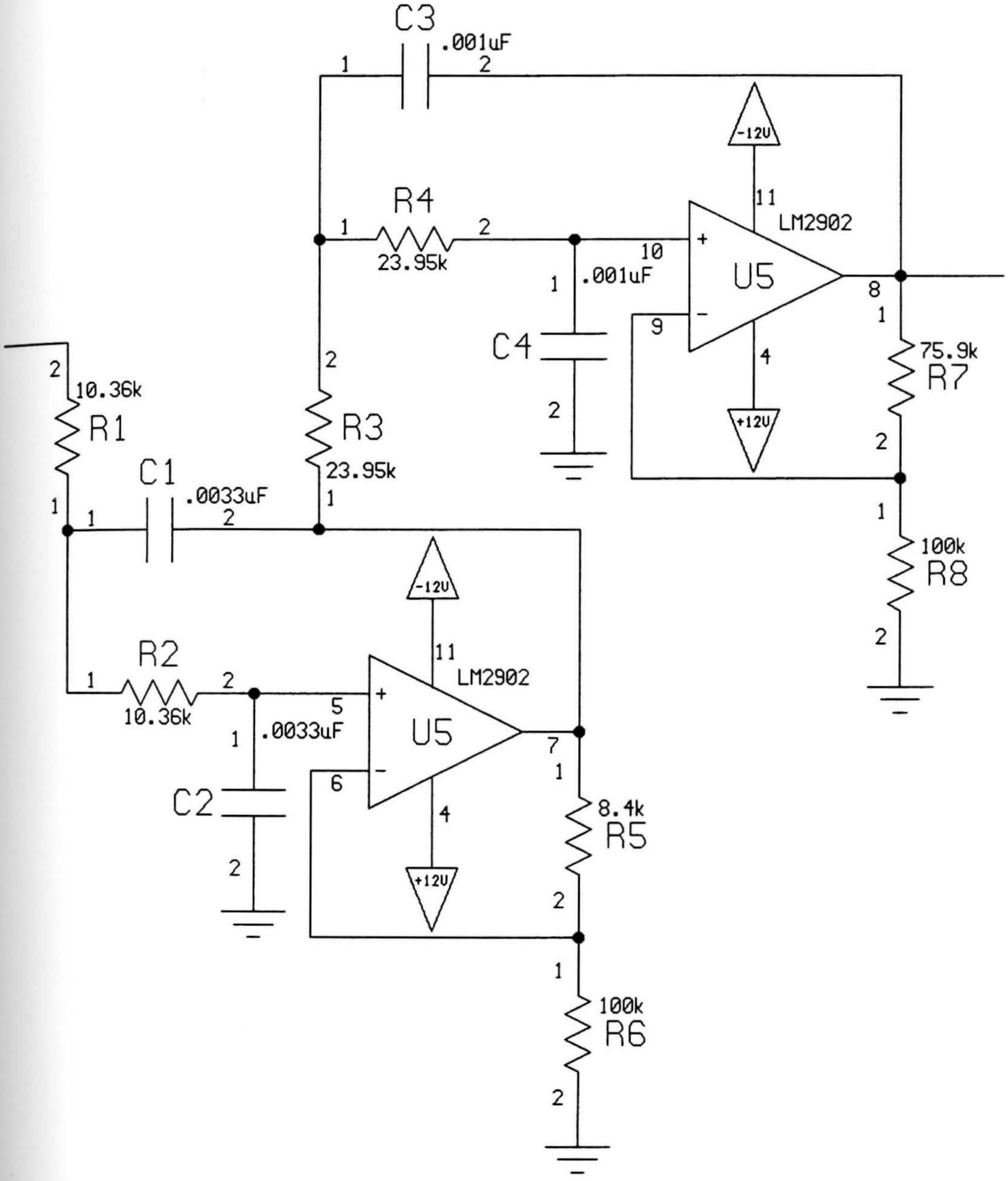


Figure 15. LOW-PASS INPUT FILTER FOR CHANNEL 1

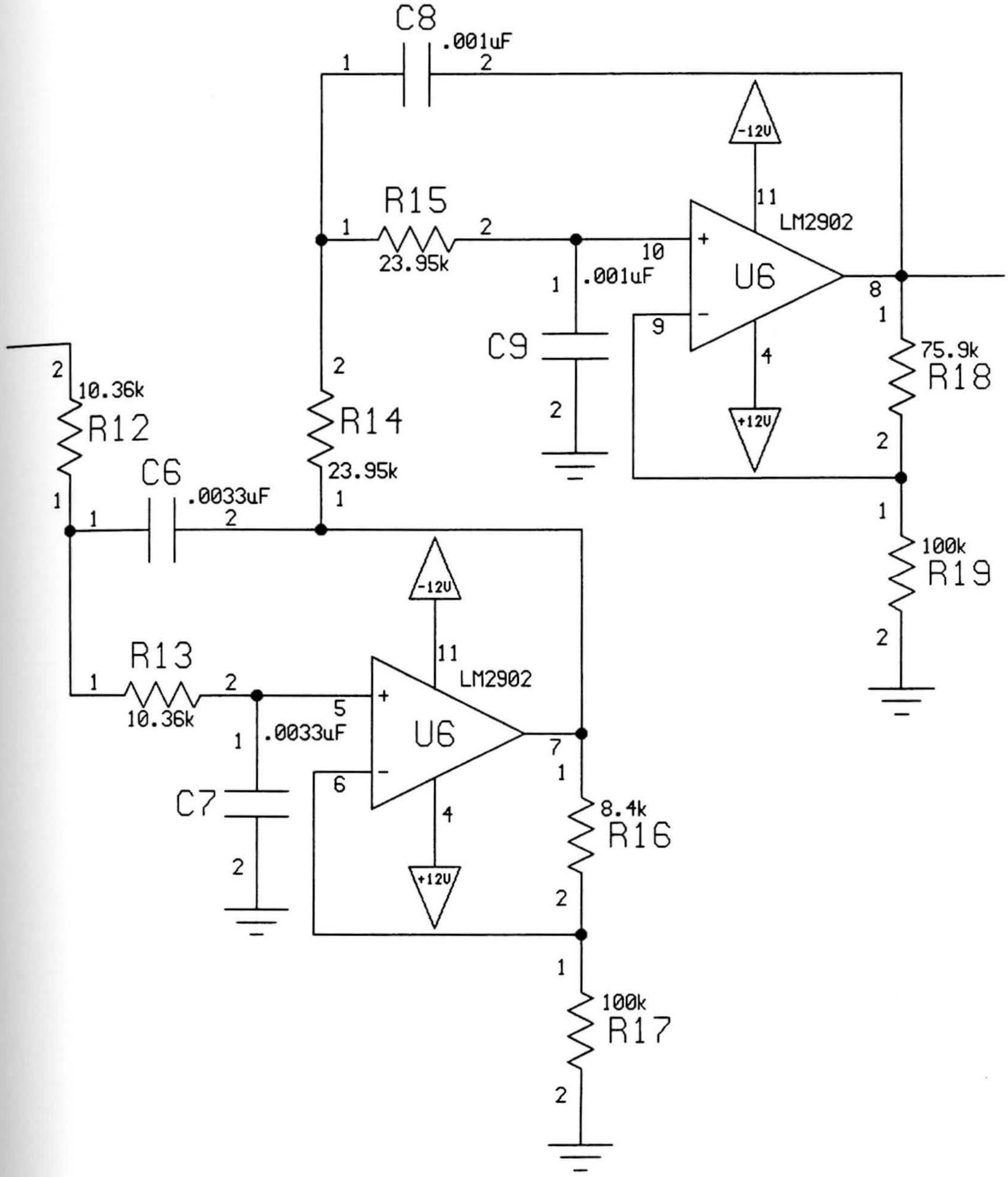


Figure 16. LOW-PASS INPUT FILTER FOR CHANNEL 2

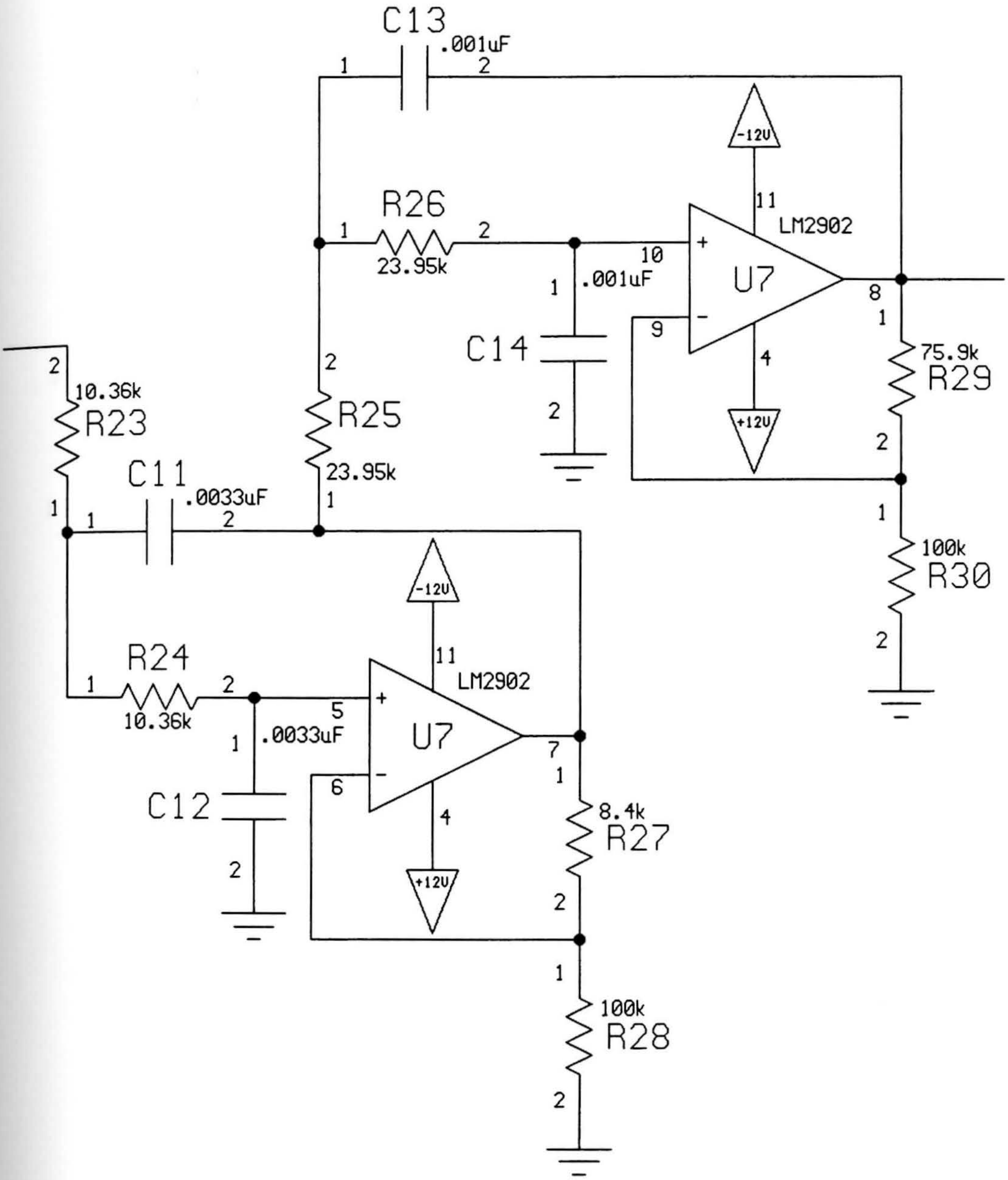


Figure 17. LOW-PASS INPUT FILTER FOR CHANNEL 3

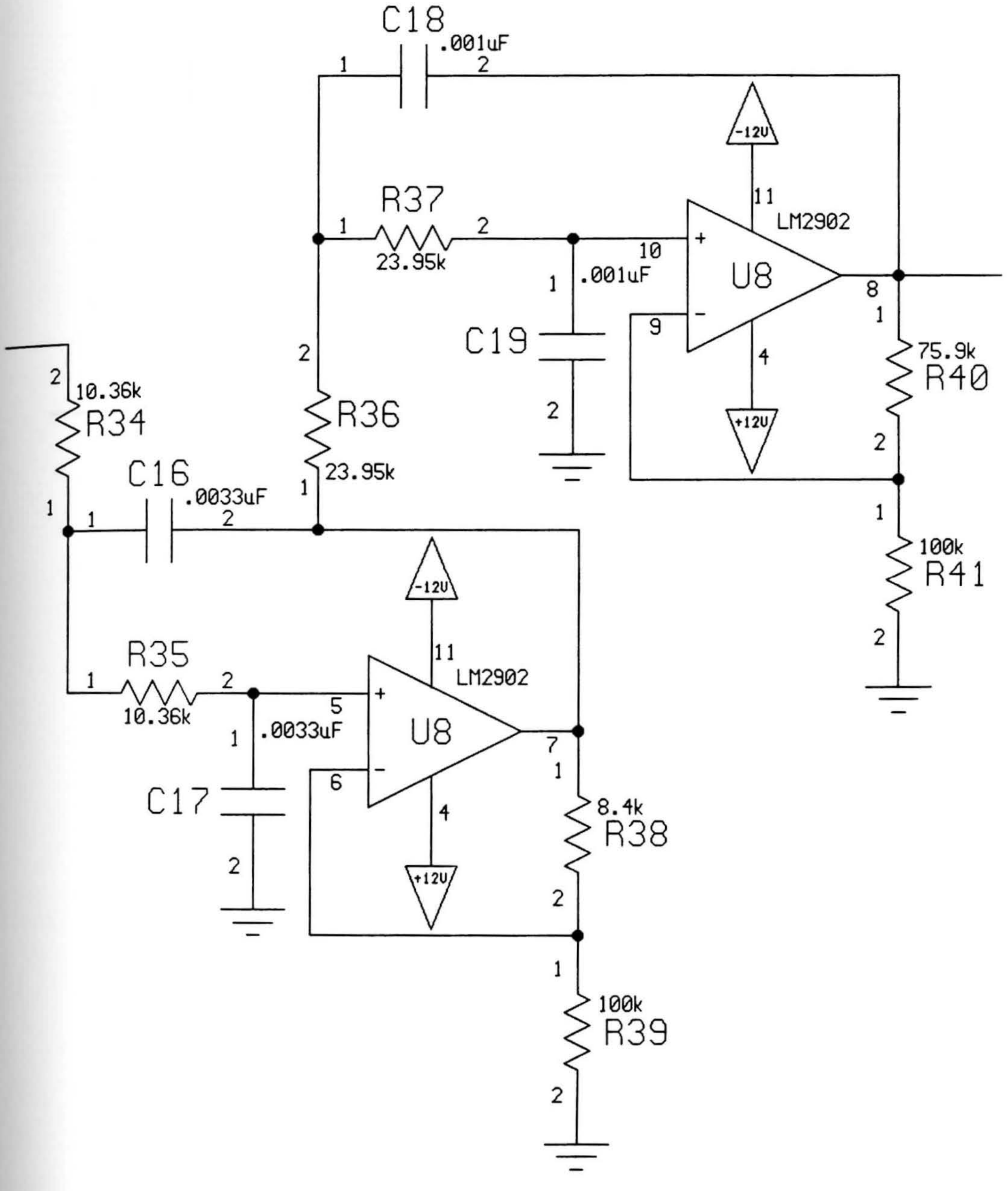


Figure 18. LOW-PASS INPUT FILTER FOR CHANNEL 4

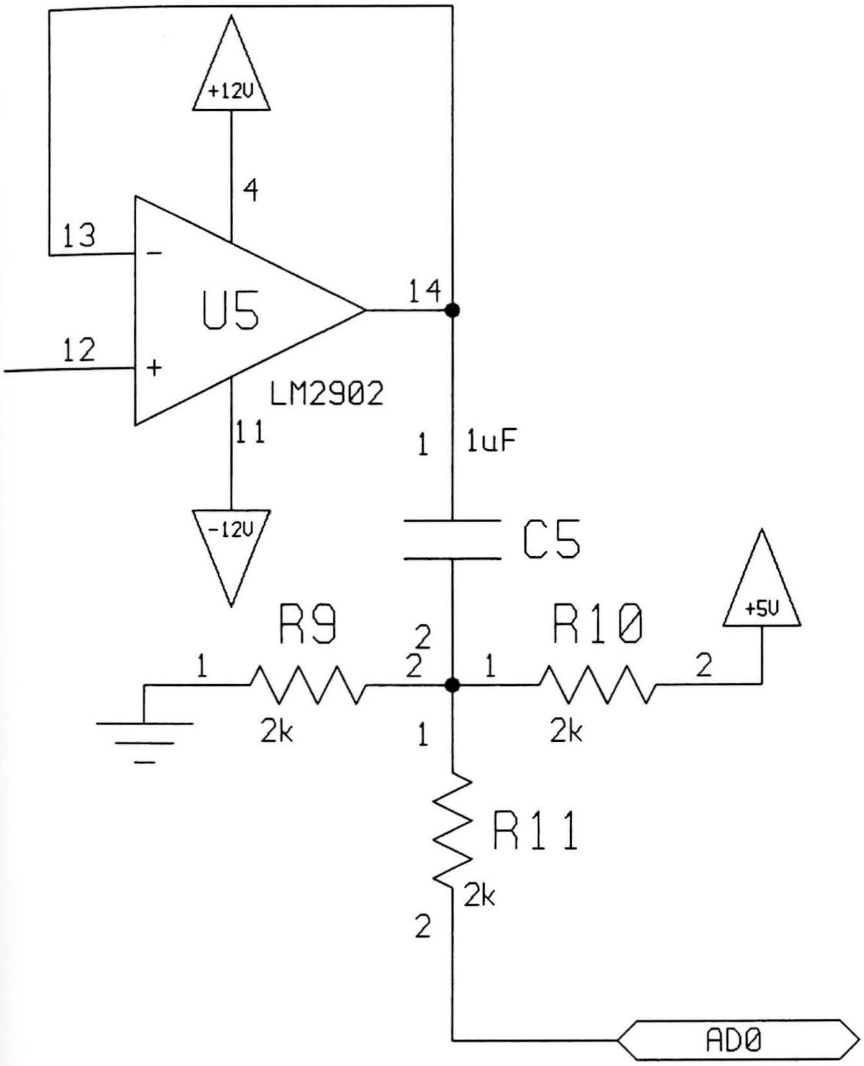


Figure 19. INPUT BUFFER FOR CHANNEL 1

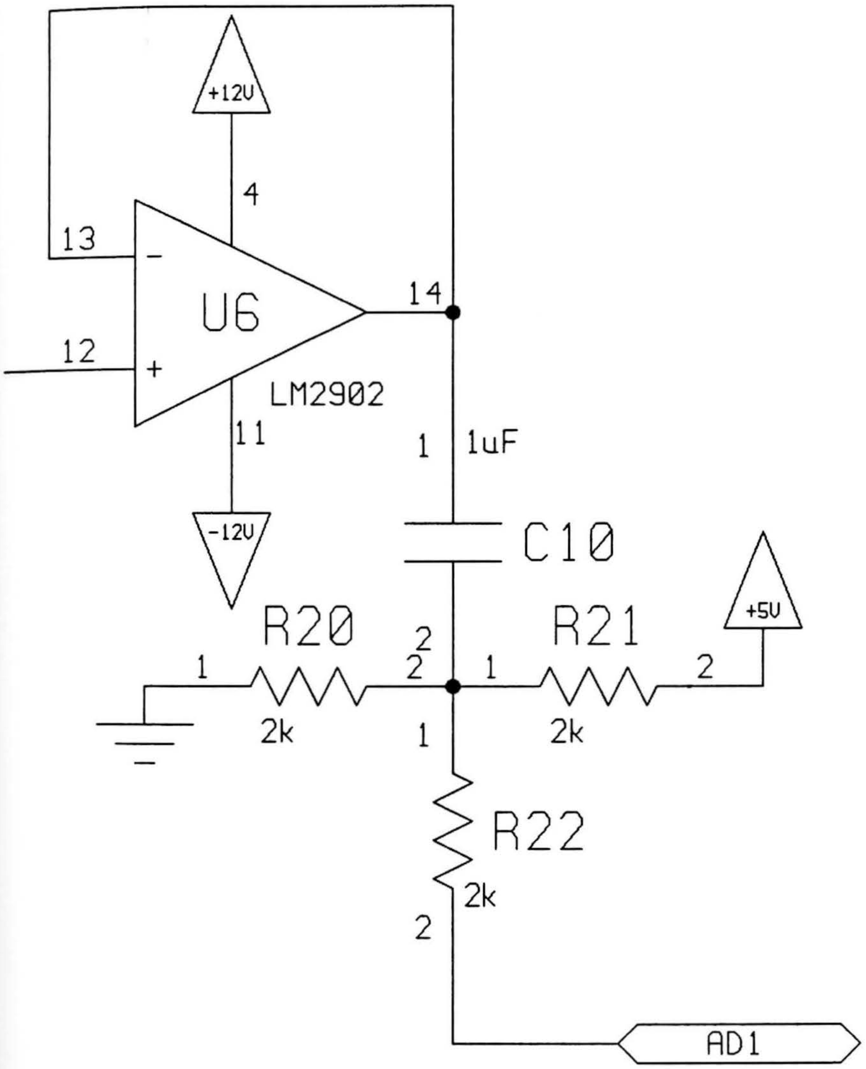


Figure 20. INPUT BUFFER FOR CHANNEL 2

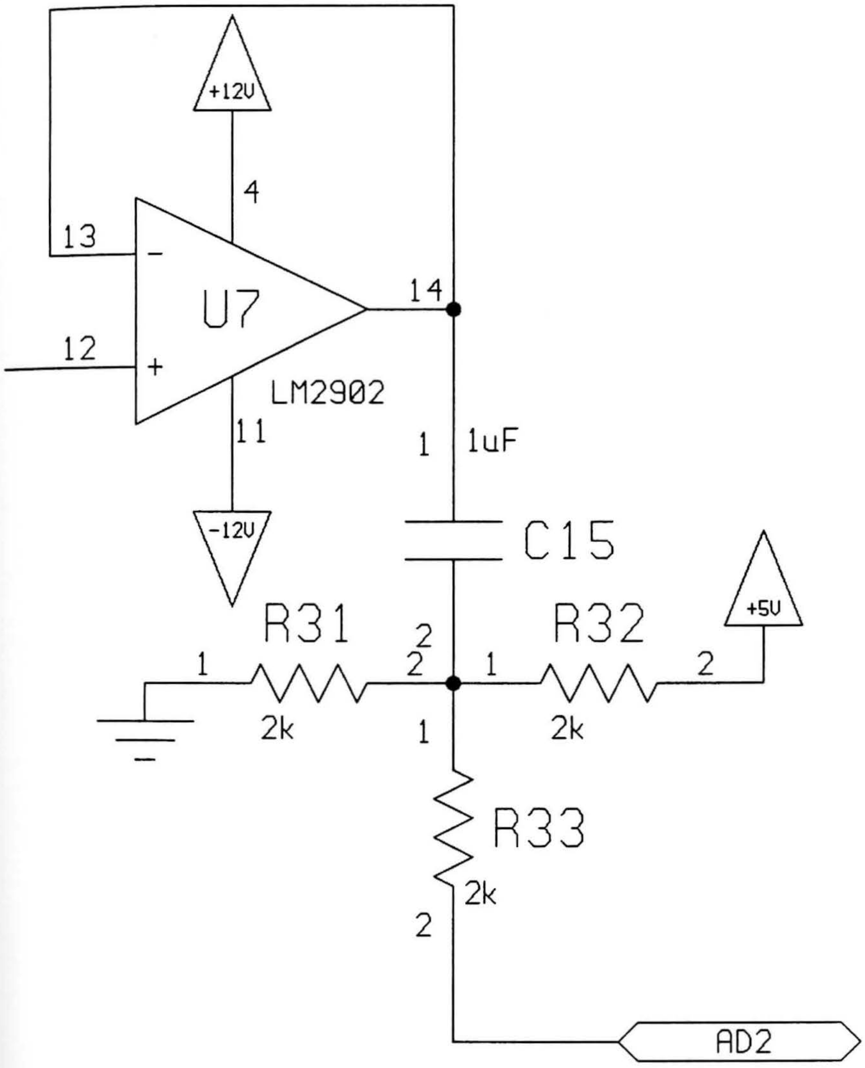


Figure 21. INPUT BUFFER FOR CHANNEL 3

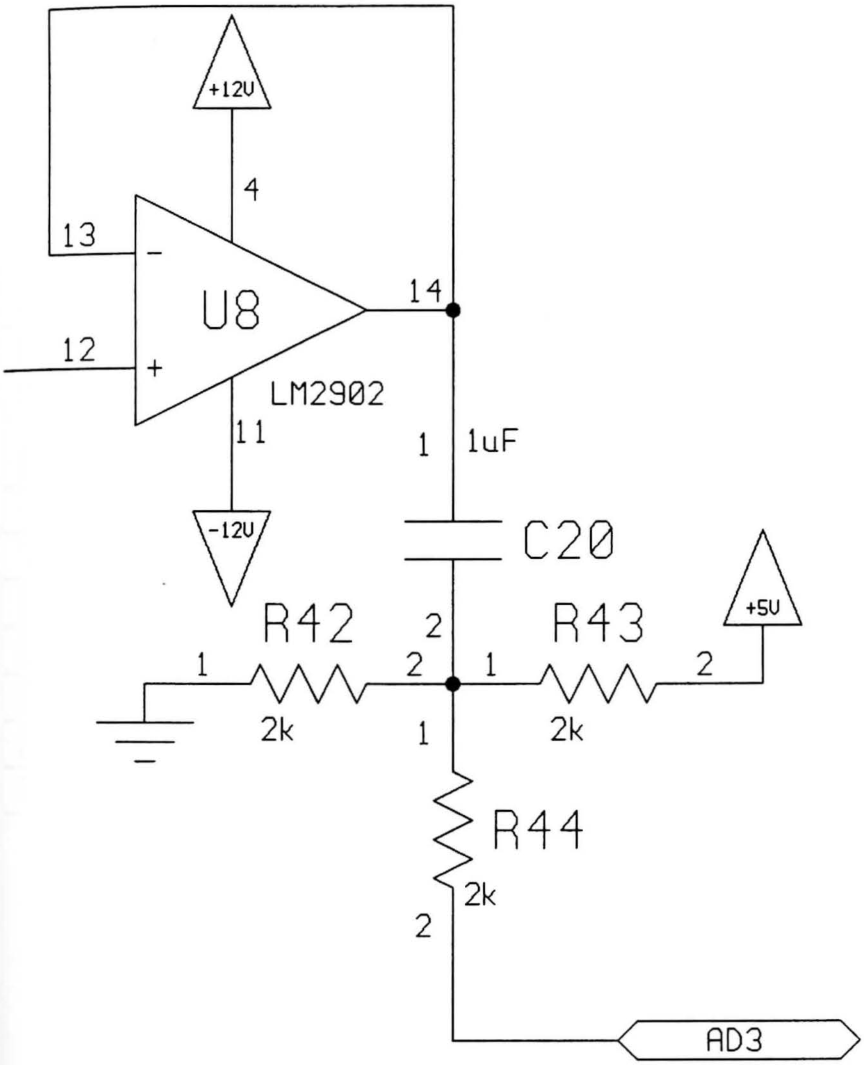


Figure 22. INPUT BUFFER FOR CHANNEL 4

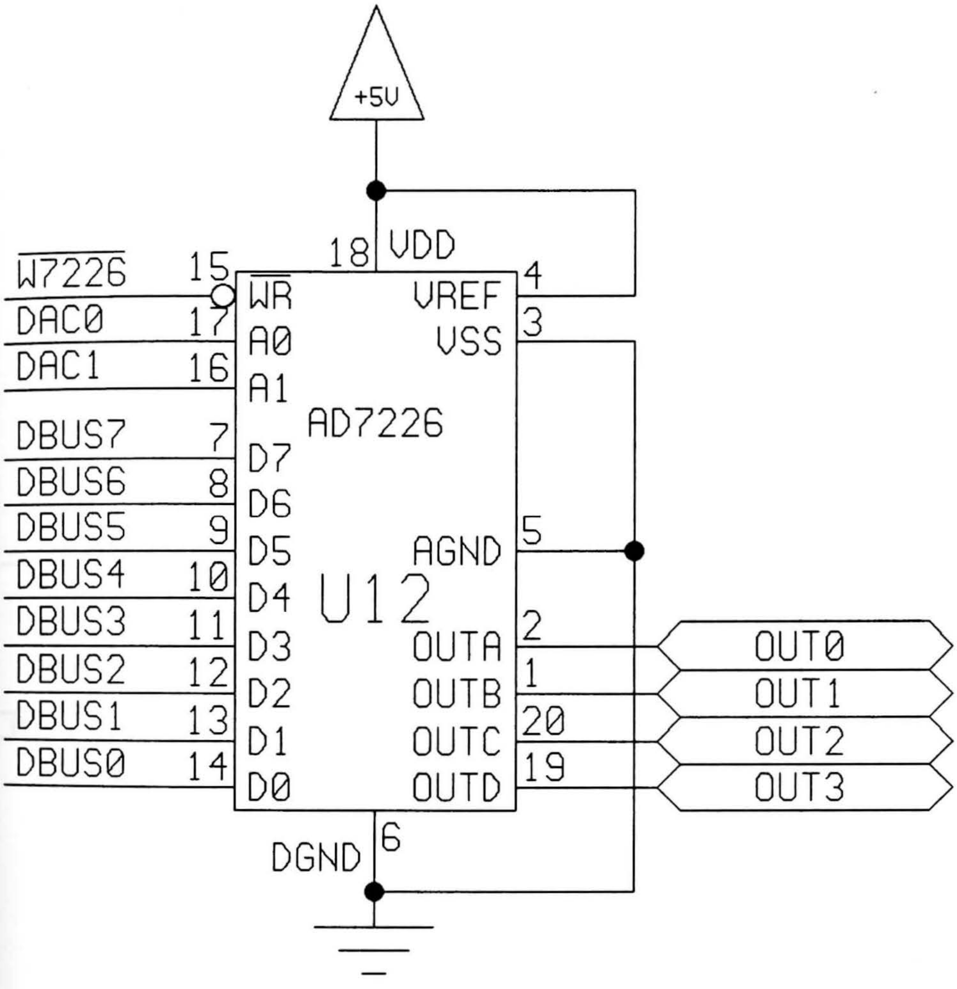


Figure 23. D/A CONVERTER FOR ALL OUTPUT CHANNELS

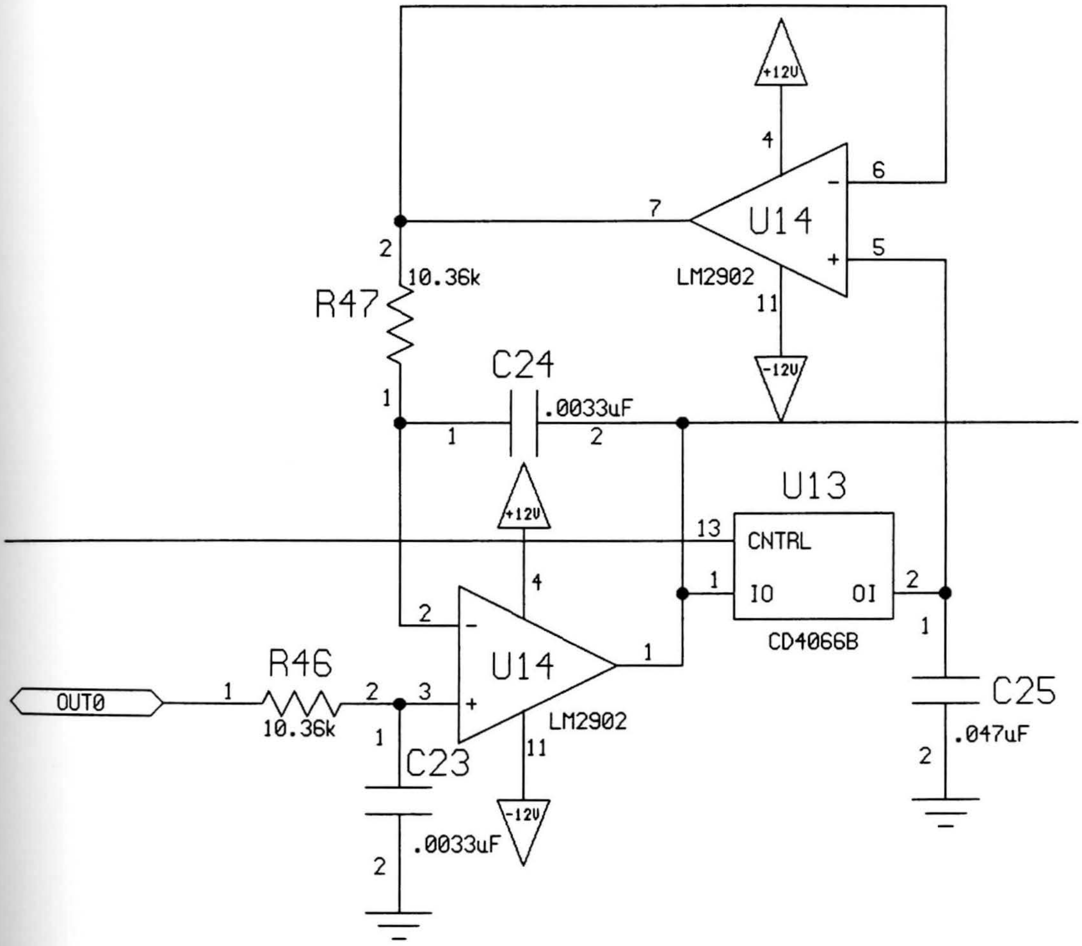


Figure 24. SDI FOR OUTPUT CHANNEL 1

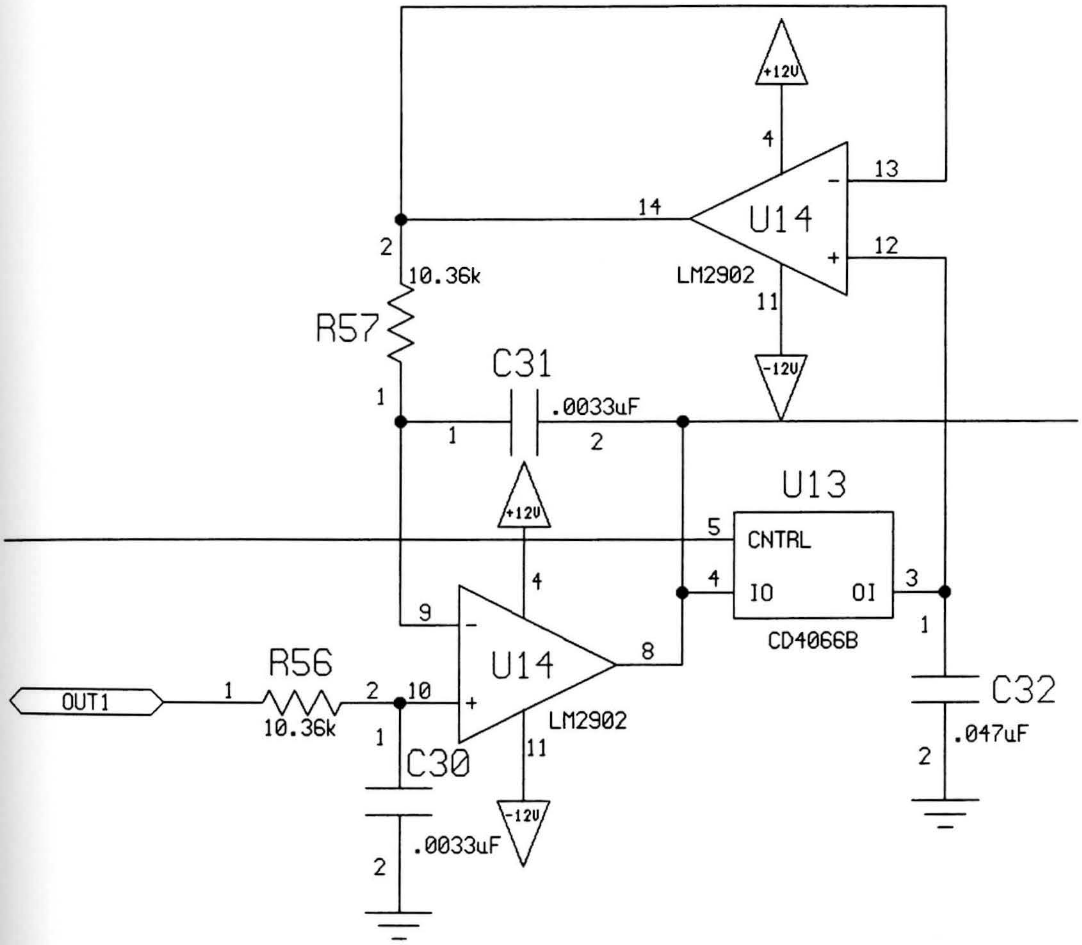


Figure 25. SDI FOR OUTPUT CHANNEL 2

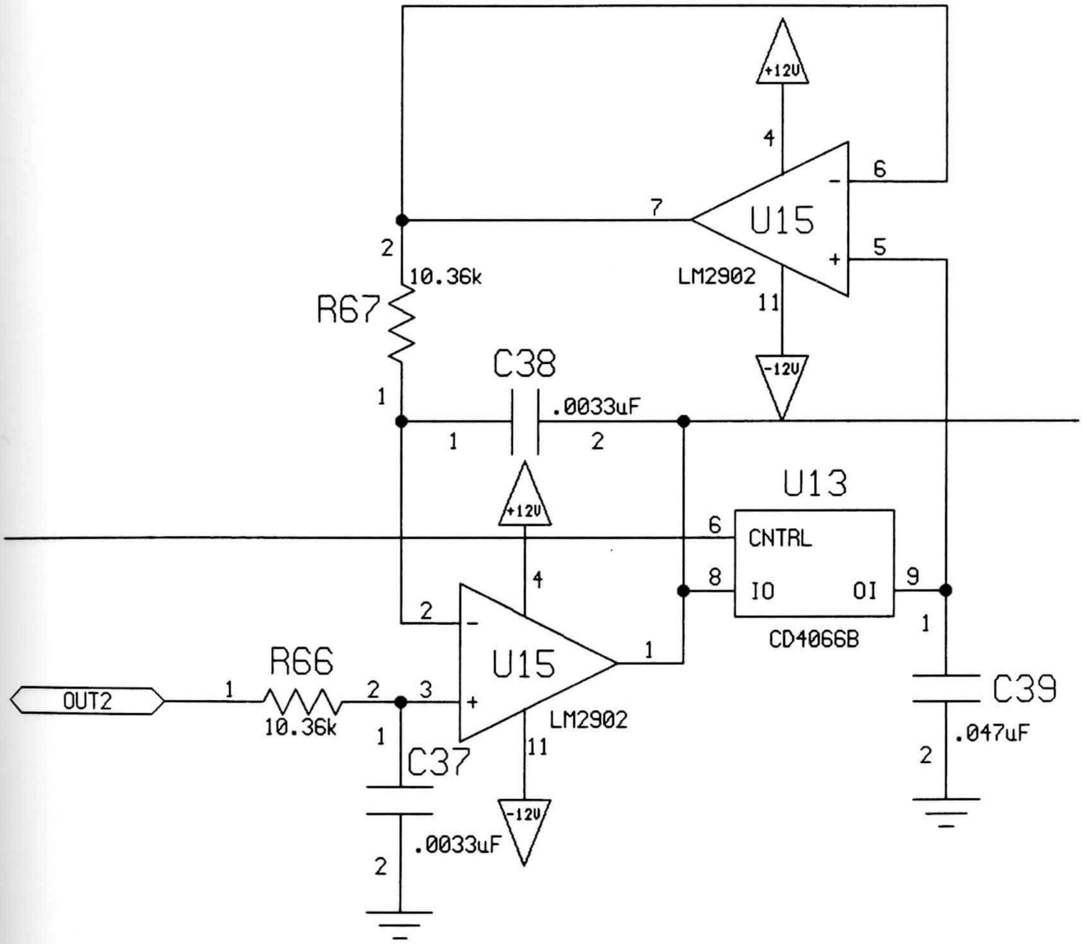


Figure 26. SDI FOR OUTPUT CHANNEL 3

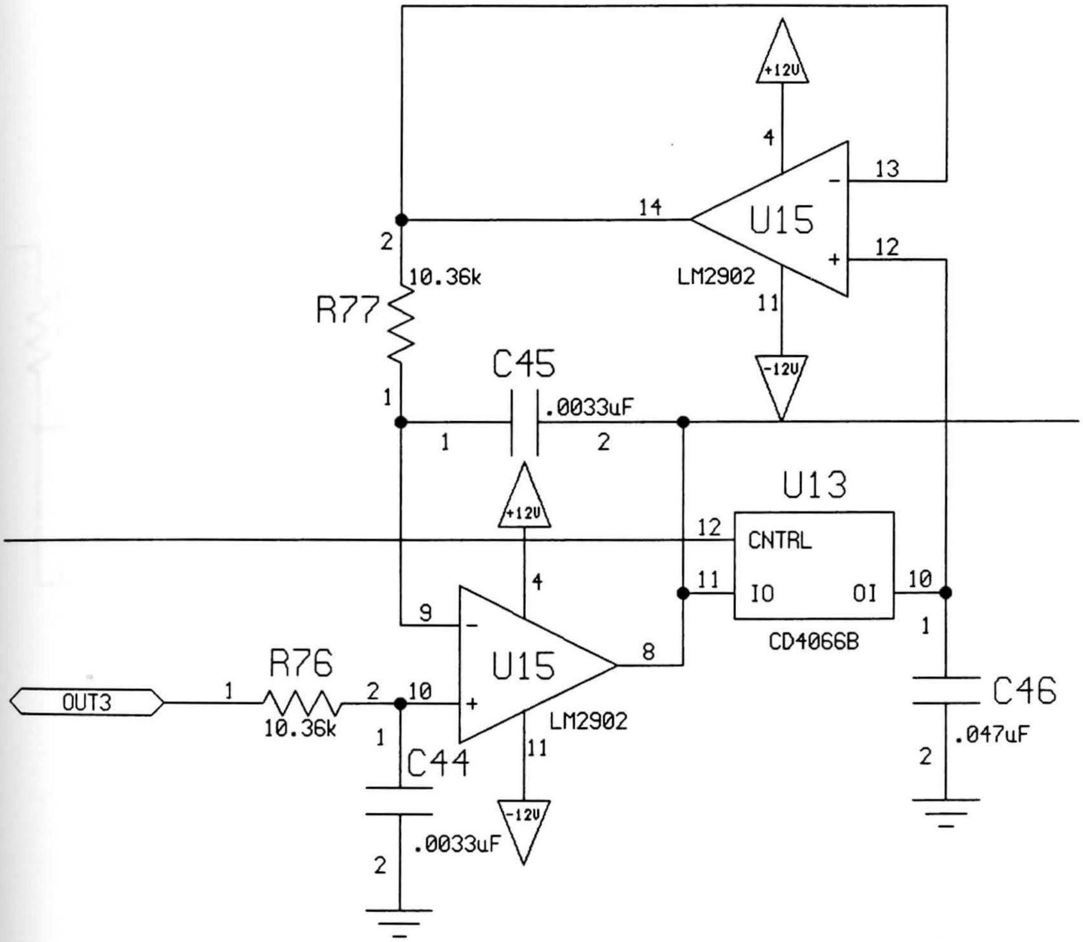


Figure 27. SDI FOR OUTPUT CHANNEL 4

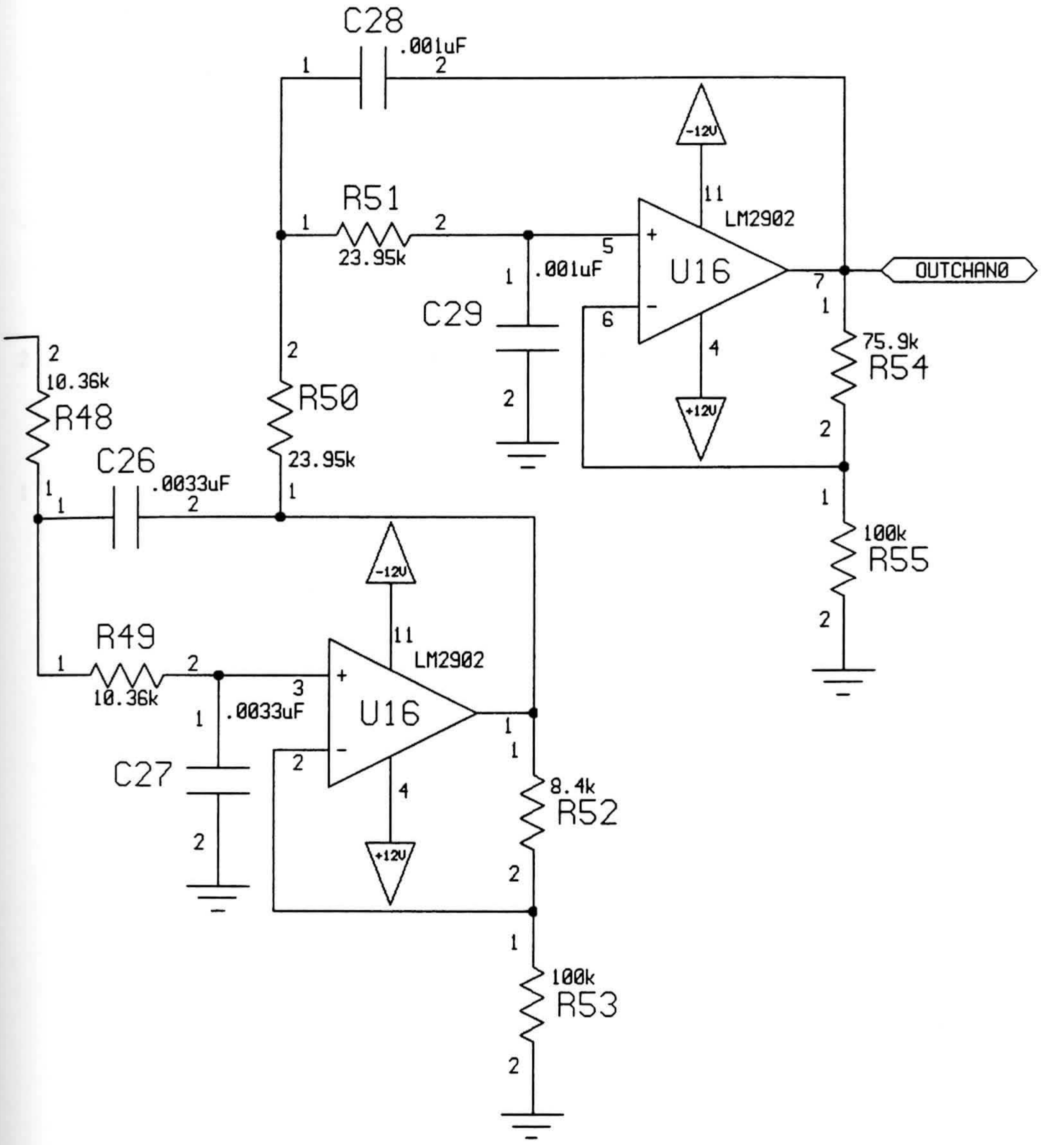


Figure 28. LOW-PASS OUTPUT FILTER FOR CHANNEL 1

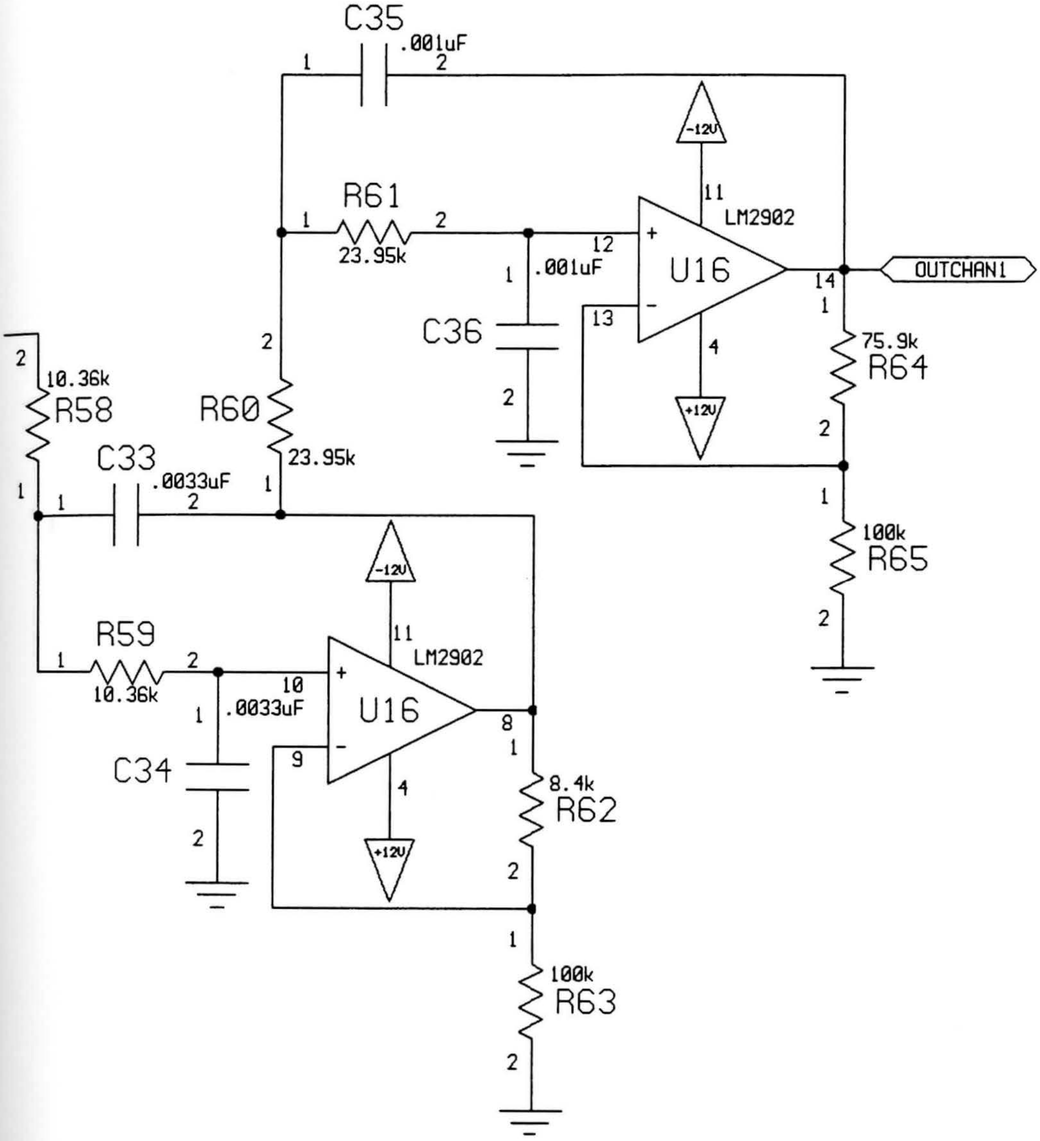


Figure 29. LOW-PASS OUTPUT FILTER FOR CHANNEL 2

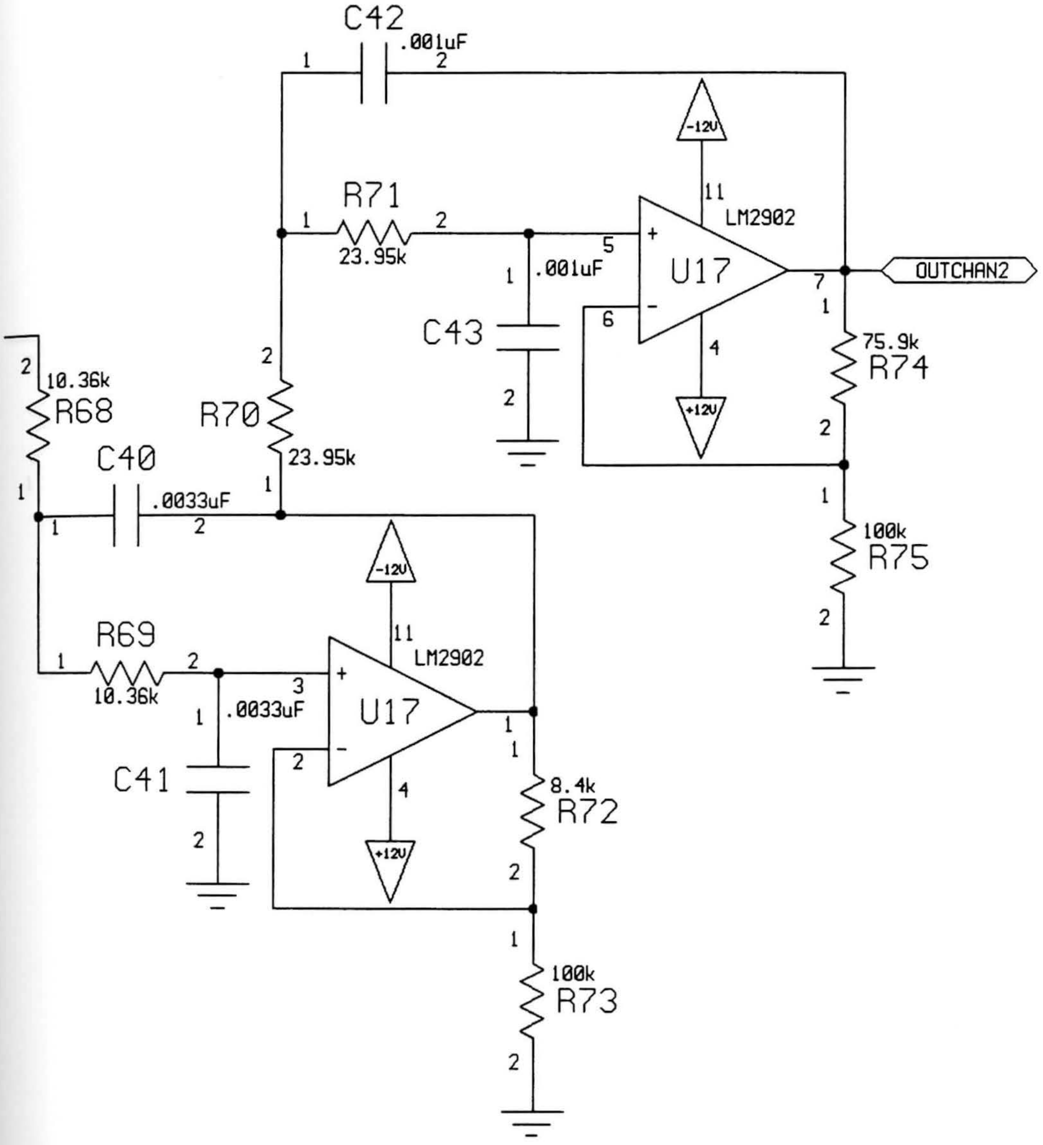


Figure 30. LOW-PASS OUTPUT FILTER FOR CHANNEL 3

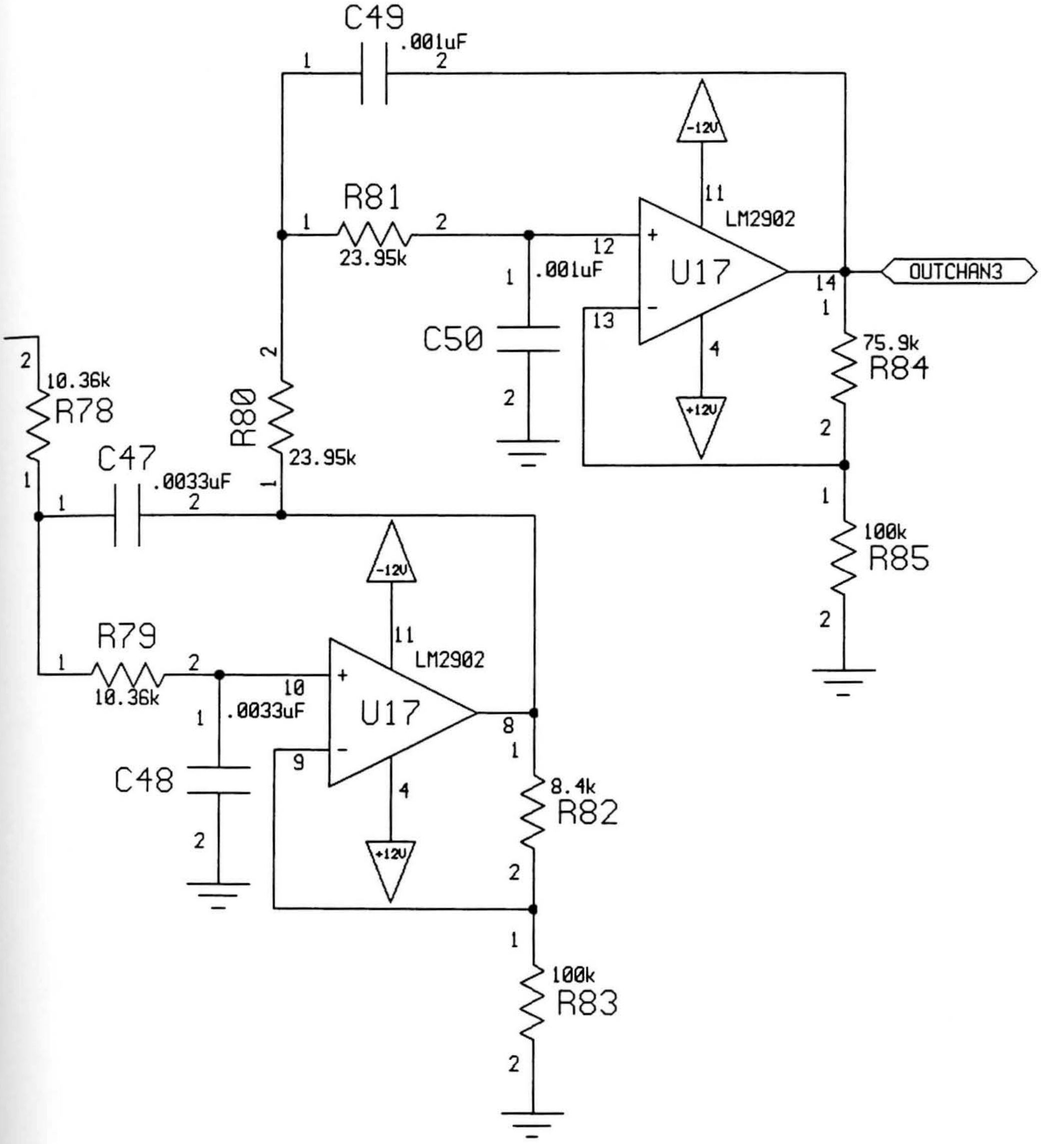


Figure 31. LOW-PASS OUTPUT FILTER FOR CHANNEL 4

APPENDIX C

FINAL SOFTWARE LISTING

```
include ''../68hc11.h''
*
* select bit, used to select either the DAC chip (AD7226)
* or the ADC chips (AD7528).  This bit is in port d.
*
sel7226 equ     0x20
sel7528 equ     0x00
*
* These values are used to set up port d for output DAC scans.
*
oscan1  equ     sel7226+0x00
oscan2  equ     sel7226+0x01
oscan3  equ     sel7226+0x02
oscan4  equ     sel7226+0x03
*
* used to set up port b for DCGA scans
*
* since the chip select lines are active low, port b
* must be used to drive low the selects for the AD7528
* desired, and to address either DACA or DACB.
*
seldac1 equ     0xfc            bit 1 of port b
seldac2 equ     0xfa            bit 2 of port b
seldac3 equ     0xf6            bit 3 of port b
seldac4 equ     0xee            bit 4 of port b
daca    equ     0x01            bit 0 of port b
dacb    equ     0x00

dcga1a  equ     seldac1+daca
dcga1b  equ     seldac1+dacb

dcga2a  equ     seldac2+daca
dcga2b  equ     seldac2+dacb

dcga3a  equ     seldac3+daca
dcga3b  equ     seldac3+dacb

dcga4a  equ     seldac4+daca
dcga4b  equ     seldac4+dacb

        org     rambase
*
* Storage for last values read from A/D convertor
*
old1:   rmb     1
old2:   rmb     1
old3:   rmb     1
```

```
old4:   rmb     1
*
* Storage for minimum values read from A/D convertor
* since last time the gain was adjusted.
*
min1:   rmb     1
min2:   rmb     1
min3:   rmb     1
min4:   rmb     1
*
* Storage for gain values for DCGA's.  One AD7528 is
* used for each channel.  The input signal is applied
* to REFA and REFB is tied to the op-amp output.
* OUTA and OUTB are grounded.  AGND is tied to the
* inverting input of the op-amp.  The actual resistance
* between a given reference (REFA or REFB) is proportional
* to the COMPLEMENT of the value stored to its DAC latch.
* Also, to simplify software, the value stored to DACB is
* always the complement of the value stored to DACA.  This
* has the net effect of creating 256 distinct gain values
* ranging from -256 to -1/256, or from 48 dB of gain to 48
* dB of attenuation in an inverting amplifier.  A zero in
* one of the following locations corresponds to minimum gain.
* Minimum gain (-1/256) is achieved by setting DACB to zero
* and DACA to 255.  This sets minimum resistance from
* REFB to AGND and maximum resistance from REFA to AGND.
*
gain1:  rmb     1
gain2:  rmb     1
gain3:  rmb     1
gain4:  rmb     1
*
* Temporary storage for A/D convertor input values.
*
temp1:  rmb     1
temp2:  rmb     1
temp3:  rmb     1
temp4:  rmb     1

        org     0xfffe
        fdb     reset

        org     0xfff0
        fdb     rtiisr

        org     0xd000
reset:
```

```
*
* Turn on A/D convertor
*
        ldaa    #adpu
        staa    option
*
* Set up data direction registers and initialize ports.
* Set up real-time interrupt to 8.19msec,
* or about 122 times/second.
*
        ldaa    #0xff
        staa    porta
        staa    ddrc
        staa    ddrd
        staa    portc

        ldaa    #ddra7+rtr0
        staa    pactl
*
* Enable SPI interface.
*
        ldaa    #spe+mstr
        staa    spcr
*
* Disable output compares, set strobe for active low,
* on writes to port b.
*
        clr     tctl1
        clr     pioc
*
* Clean out ram.
*
        ldx     #0
        clra
        clrb
clrram:
        staa    0,x
        inx
        cpx     #512
        blo     clrram
*
* Enable real-time interrupts.
*
        ldaa    #rtii
        staa    tmsk2
        cli
main:
```

```
*
* Start a scan on the A/D convertor.  Assume it has
* valid data for the first time around.  Since a scan
* takes less time than the loop time, each time through
* the loop after the first, the data will be valid.
*
       ldaa    #mult
       staa    adctl
*
* Grab the A/D values, start the first byte out the SPI port.
*
       ldd     adr1
       staa    spdr
       std     temp1
       ldd     adr3
       std     temp3
process1:
*
* Select DAC1
*
       ldaa    #oscan1
       staa    portd
*
* Grab new data from SPI port.  The load double reads
* the spsr first to clear the transfer complete flag,
* then reads the spdr to retrieve the data.
* The status will be in regA, the data in regB.
*
       ldd     spsr
*
* start the next A/D value on its way.
*
       ldaa    temp2
       staa    spdr
*
* Output value just received.
*
       tba
       std     portc
*
* note: the store double to port c stores garbage to
* port b, but the AD7528's are deselected right now,
* so all that happens is the write line gets strobed.
*


*
* Update min if this one is lower.
```

```
*
        cmpb    min1
        bhs     process2
        stab    min1

process2:
*
* Select DAC2
*
        ldaa    #oscan2
        staa    portd
*
* Grab new data from SPI port.  The load double reads
* the spsr first to clear the transfer complete flag,
* then reads the spdr to retrieve the data.
* The status will be in regA, the data in regB.
*
        ldd     spsr
*
* start the next A/D value on its way.
*
        ldaa    temp3
        staa    spdr
*
* Output value just received.
*
        tba
        std     portc
*
* note: the store double to port c stores garbage to
* port b, but the AD7528's are deselected right now,
* so all that happens is the write line gets strobed.
*

*
* Update min if this one is lower.
*
        cmpb    min2
        bhs     process3
        stab    min2

process3:
*
* Select DAC3
*
        ldaa    #oscan3
        staa    portd
```

```
*
* Grab new data from SPI port.  The load double reads
* the spsr first to clear the transfer complete flag,
* then reads the spdr to retrieve the data.
* The status will be in regA, the data in regB.
*
        ldd     spsr
*
* start the next A/D value on its way.
*
        ldaa    temp4
        staa    spdr
*
* Output value just received.
*
        tba
        std     portc
*
* note: the store double to port c stores garbage to
* port b, but the AD7528's are deselected right now,
* so all that happens is the write line gets strobed.
*


*
* Update min if this one is lower.
*
        cmpb    min3
        bhs     process4
        stab    min3

process4:
*
* Select DAC4
*
        ldaa    #oscan4
        staa    portd
*
* Grab new data from SPI port.  The load double reads
* the spsr first to clear the transfer complete flag,
* then reads the spdr to retrieve the data.
* The status will be in regA, the data in regB.
*
        ldd     spsr
*
* Output value just received.
*
        tba
        std     portc
*
* note: the store double to port c stores garbage to
* port b, but the AD7528's are deselected right now,
* so all that happens is the write line gets strobed.
*

*
* Update min if this one is lower.
*
        cmpb    min4
        bhs     main
        stab    min4
        bra     main

rtiisr:
        ldaa    portd           select the AD7528's
        psha
        ldaa    #sel7528
        staa    portd
tstmin1:
        ldaa    gain1
        ldab    min1
        beq     down1           if clipping, reduce gain.
        cmpb    #32
        blo     tstmin2         else if in window, do next one.
up1:                    ;       otherwise, increase gain.
        inca
        beq     tstmin2         if can't increase, do next one.
adj1:
        staa    gain1             save new gain value
        ldab    #dcga1a           load the addressing value
        std     portc             and set DACA
        coma                      generate the value for DACB
        ldab    #dcga1b
        std     portc             and set it.
        bra     tstmin2
down1:
        deca
        bhs     adj1
tstmin2:
        ldaa    gain2
        ldab    min2
        beq     down2           if clipping, reduce gain.
        cmpb    #32
        blo     tstmin3         else if in window, do next one.
up2:                    ;       otherwise, increase gain.
```

```
        inca
        beq     tstmin3         if can't increase, do next one.
adj2:
        staa    gain2             save new gain value
        ldab    #dcga2a           load the addressing value
        std     portc             and set DACA
        coma                      generate the value for DACB
        ldab    #dcga2b
        std     portc             and set it.
        bra     tstmin3
down2:
        deca
        bhs     adj2
tstmin3:
        ldaa    gain3
        ldab    min3
        beq     down3           if clipping, reduce gain.
        cmpb    #32
        blo     tstmin4         else if in window, do next one.
up3:                    ;       otherwise, increase gain.
        inca
        beq     tstmin4         if can't increase, do next one.
adj3:
        staa    gain3             save new gain value
        ldab    #dcga3a           load the addressing value
        std     portc             and set DACA
        coma                      generate the value for DACB
        ldab    #dcga3b
        std     portc             and set it.
        bra     tstmin4
down3:
        deca
        bhs     adj3
tstmin4:
        ldaa    gain4
        ldab    min4
        beq     down4           if clipping, reduce gain.
        cmpb    #32
        blo     rtidone         else if in window, done.
up4:                    ;       otherwise, increase gain.
        inca
        beq     rtidone         if can't increase, done.
adj4:
        staa    gain4             save new gain value
        ldab    #dcga4a           load the addressing value
        std     portc             and set DACA
        coma                      generate the value for DACB
        ldab    #dcga4b
        std     portc              and set it.
        bra     rtidone
down4:
        deca
        bhs     adj4
rtidone:
        pula                     return portd to original value.
        staa    portd

        ldaa    #rtif
        staa    tflg2
        rti
```

APPENDIX D

LIST OF PARTS REQUIRED

```
%*********************************************************************
%                                                                    *
%     Program  :  PC-FORM VERSION 4.50                               *
%     Date     :  Jun 03 1991                                        *
%     Time     :  01:57:59 PM                                        *
%     File In  :  D:THESIS.NLT                                       *
%     File Out :  D:THESIS.MAT                                       *
%     Format   :  P-CAD MATERIALS LIST                               *
%                                                                    *
%*********************************************************************



ITEM  QTY  COMP-NAME  REFERENCE-DESIGNATOR  DESCRIPTION
----  ---  ---------  -------------         -----------

1     8    LM2902     U5 U6 U7 U8 U14       QUAD OP AMP
                      U15 U16 U17

2     4    AD7528     U1 U2 U3 U4           DUAL D/A, CURRENT

3     1    68HC11A8   U10                   MICRO-CONTROLLER

4     1    AD7226     U12                   QUAD D/A, VOLTAGE

5     1    C4066B     U13                   QUAD ANALOG SWITCH

6     1    74HC00     U11                   QUAD 2-INPUT NAND

7     1    XTAL       X1                    VAL=16MHz

8     16   RESISTOR   R41 R39 R30 R28       VAL=100k
                      R19 R17 R8 R6 R85
                      R83 R75 R73 R65
                      R63 R55 R53

9     16   RESISTOR   R36 R37 R25 R26       VAL=23.95k
                      R14 R15 R3 R4 R80
                      R81 R70 R71 R60
                      R61 R50 R51

10    8    CAPACITOR  C19 C14 C9 C4 C50     VAL=.001uF
                      C43 C36 C29
```

```
ITEM  QTY  COMP-NAME  REFERENCE-DESIGNATOR  DESCRIPTION
----  ---  ---------  -------------         -----------

11    24   RESISTOR   R35 R34 R24 R23       VAL=10.36k
                      R13 R12 R2 R1 R76
                      R77 R67 R66 R79
                      R78 R69 R68 R57
                      R56 R59 R58 R47
                      R46 R49 R48

12    16   CAPACITOR  C17 C12 C7 C2 C44     VAL=.0033uF
                      C45 C38 C37 C48
                      C41 C31 C30 C34
                      C24 C23 C27

13    4    CAPACITOR  C46 C39 C32 C25       VAL=.047uF

14    8    CAPACITOR  C16 C11 C6 C1 C47     VAL=.0033uF
                      C40 C33 C26

15    8    RESISTOR   R38 R27 R16 R5 R82    VAL=8.4k
                      R72 R62 R52

16    8    RESISTOR   R40 R29 R18 R7 R84    VAL=75.9k
                      R74 R64 R54

17    8    CAPACITOR  C18 C13 C8 C3 C49     VAL=.001uF
                      C42 C35 C28

18    4    CAPACITOR  C20 C15 C10 C5        VAL=1uF

19    2    CAPACITOR  C21 C22               VAL=22pF

20    8    RESISTOR   R42 R43 R31 R32       VAL=2k
                      R20 R21 R9 R10

21    1    RESISTOR   R45                   VAL=1M

22    4    RESISTOR   R44 R33 R22 R11       VAL=2k
```

BIBLIOGRAPHY

BOOKS

The Engineering Staff of Analog Devices, Inc., *Analog-Digital Conversion Handbook.* Edited by Daniel H. Sheingold. Englewood Cliffs, New Jersey: Prentice Hall, 1986.

Jacob, J. Michael, *Applications and Design with Analog Integrated Circuits.* Reston, Virginia: Reston Publishing Company, Inc., 1982.

Lipovski, G. J., *Single- and Multiple-Chip Microcomputer Interfacing.* Englewood Cliffs, New Jersey: Prentice Hall, 1988.

Motorola, Inc., *M68HC11 Reference Manual.* Englewood Cliffs, New Jersey: Prentice Hall, 1989.

Poularikas, Alexander D. and Seely, Samuel, *Signals and Systems.* Boston, Massachussetts: PWS Engineering, 1985.

Ziemer, Rodger E., Tranter, William H., and Fannin, D. Ronald, *Signals and Systems: Continuous and Discrete.* New York, New York: Macmillan Publishing Co., Inc., 1983.

MANUFACTURERS DATA BOOKS

Motorola Inc., *Linear and Interface Integrated Circuits* Phoenix, Arizona: Motorola Inc., 1990.

National Semiconductor Corporation, *CMOS Logic Databook.* Santa Clara, California: National Semiconductor Corporation, 1988.

Precision Monolithics Inc., *Analog IC Databook.* Santa Clara, California: Precision Monolithics Inc., 1988.

Texas Instruments, *Interface Circuits Data Book.* Dallas, Texas: Texas Instruments, 1987.