

Module

3

Embedded Systems I/O

Lesson

19

Analog Interfacing

## Instructional Objectives

After going through this lesson the student would be able to

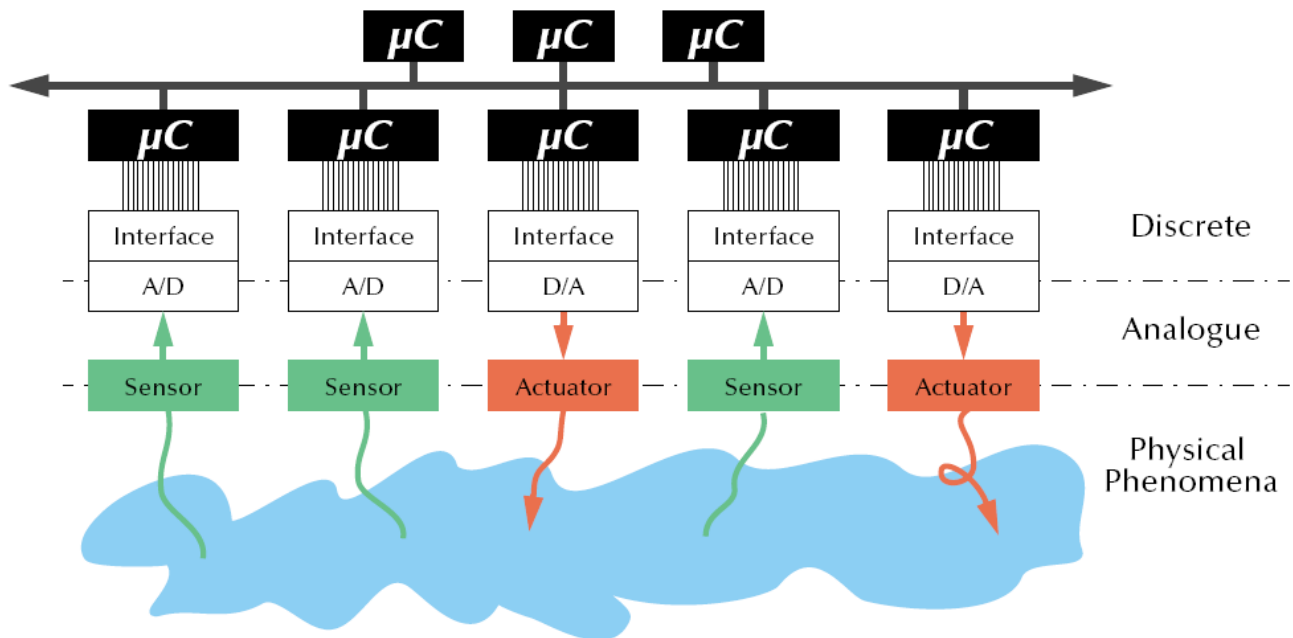
- Know the interfacing of analog signals to microcontrollers/microprocessors
- Generating Analog Signals
- Designing AD and DA interfaces
- Various Methods of acquiring and generating analog data

## Pre-Requisite

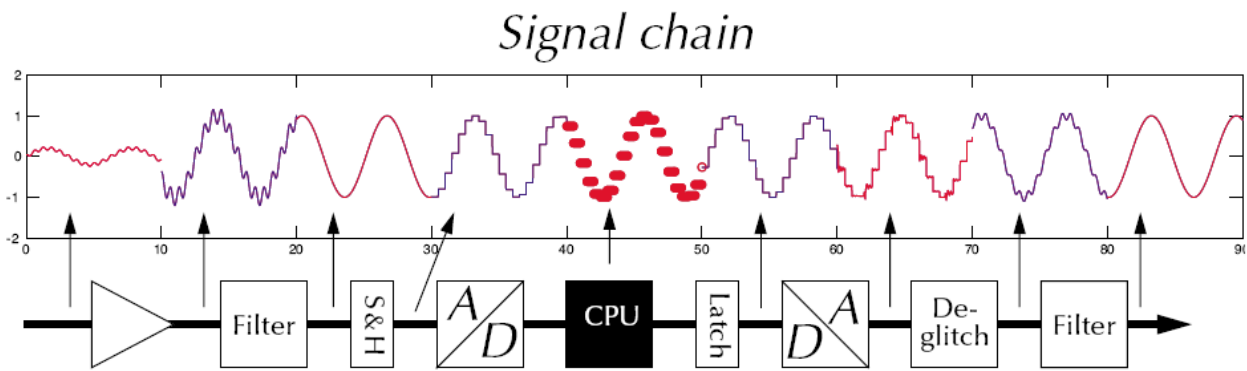
Digital Electronics, Microprocessors

### 19(I) Introduction

Fig.19.1 shows a typical sensor network. You will find a number of sensors and actuators connected to a common bus to share information and derive a collective decision. This is a complex embedded system. Digital camera falls under such a system. Only the analog signals are shown here. Last lesson discussed in detail about the AD and DA conversion methods. This chapter shall discuss the inbuilt AD-DA converter and standalone converters and their interfacing.



**Fig. 19.1 The Analog Interfacing Network**



**Fig. 19.2 The Analog-Digital-Analog signal path with real time processing**

## Different Stages of Fig.19.2

Stage-1 Signal Amplification and Conditioning; Stage-2 Anti-aliasing Filter; Stage-3 Sample and Hold;

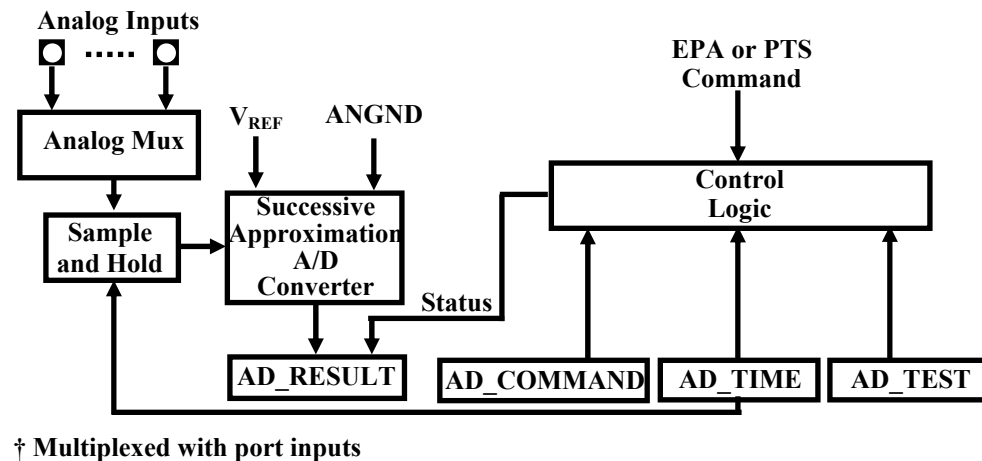
Stage-4 Analog to Digital Converter; Stage-5 Digital Processing and Data manipulation in a Processor;

Stage-6 Processed Digital Values are temporarily stored in a latch before D-A conversion;

Stage-7 Digital to Analog Conversion; Stage-8 Removal of Glitches and Spikes; Stage-8 Final Low pass filtering

## 19(II) Embedded AD Converters in Intel 80196

Fig.19.3 shows the block diagram of the AD converter inbuilt to 80196 embedded processor. The details of the subsystems are given as follows:



**Fig. 19.3 The block diagram of the Internal AD converter**

*Analog Inputs:* There are 12 input channels which are multiplexed with the Port P0 and Port P1 of the processor.

*ANGND*: It is the analog ground which is separately connected to the circuit from where analog voltage is brought inside the processor.

*Vref*: It is reference voltage which decides the range of the input voltage. By making it negative bipolar inputs can be used.

## EPA: Event Processor Array

Control applications often require high-speed event control. For example, the controller may need to periodically generate pulse-width modulated outputs or an interrupt. In another application, the controller may monitor an input signal to determine the status of an external device. The event processor array (EPA) was designed to reduce the CPU overhead associated with these types of event control. This chapter describes the EPA and its timers and explains how to configure and program them. The EPA can control AD converter such as generating timing pulses, start conversion signals etc.

## PTS: Peripheral Transaction Server

The microcontroller's interrupt-handling system has two components: the programmable interrupt controller and the peripheral transaction server (PTS). The programmable interrupt controller has a hardware priority scheme that can be modified by the software. Interrupts that go through the interrupt controller are serviced by interrupt service routines that you provide. The upper and lower interrupt vectors in special-purpose memory contain the interrupt service routines' addresses. The peripheral transaction server (PTS), a microcoded hardware interrupt processor, provides high-speed, low-overhead interrupt handling; it does not modify the stack or the Processor Status Word. The PTS supports seven microcoded routines that enable it to complete specific tasks in lesser time than an equivalent interrupt service routine can. It can transfer bytes or words, either individually or in blocks, between any memory locations; manage multiple analog-to-digital (A/D) conversions; and transmit and receive serial data in either asynchronous or synchronous mode.

## Analog Mux: Analog Multiplexer

It selects a particular analog channel for conversion. Only after completing conversion of one channel it switches to subsequent channels.

## The associated Registers

**AD\_COMMAND** register

This register selects the A/D channel, controls whether the A/D conversion starts immediately or is triggered by the EPA, and selects the operating mode.

## AD\_RESULT

For an A/D conversion, the high byte contains the eight MSBs from the conversion, while the low byte contains the two LSBs from a 10-bit conversion (undefined for an 8-bit conversion), indicates which A/D channel was used, and indicates whether the channel is idle. For a

threshold-detection, calculate the value for the successive approximation register and write that value to the high byte of AD\_RESULT. Clear the low byte or leave it in its default state.

#### AD\_TEST A/D Conversion Test

This register specifies adjustments for zero-offset errors.

#### AD\_TIME A/D Conversion Time

This register defines the sample window time and the conversion time for each bit.

#### INT\_MASK Interrupt Mask

The AD bit in this register enables or disables the A/D interrupt. Set the AD bit to enable the interrupt request.

#### INT\_PEND Interrupt Pending

The AD bit in this register, when set, indicates that an A/D interrupt request is pending.

## A/D Converter Operation

An A/D conversion converts an analog input voltage to a digital value, stores the result in the AD\_RESULT register, and sets the A/D interrupt pending bit. An 8-bit conversion provides 20 mV resolution, while a 10-bit conversion provides 5 mV resolution. An 8-bit conversion takes less time than a 10-bit conversion because it has two fewer bits to resolve and the comparator requires less settling time for 20 mV resolution than for 5 mV resolution. Either the voltage on an analog input channel or a test voltage can be converted. Converting the test inputs is used to calculate the zero-offset error, and the zero-offset adjustment is used to compensate for it. This feature can reduce or eliminate off-chip compensation hardware. Typically, the test voltages are converted to adjust for the zero-offset error before performing conversions on an input channel. The AD\_TEST register is used to program for zero-offset adjustment. A threshold-detection compares an input voltage to a programmed reference voltage and sets the A/D interrupt pending bit when the input voltage crosses over or under the reference voltage. A conversion can be started by a write to the AD\_COMMAND register or it can be initiated by the EPA, which can provide equally spaced samples or synchronization with external events.

Once the A/D converter receives the command to start a conversion, a delay time elapses before sampling begins. During this sample delay, the hardware clears the successive approximation register and selects the designated multiplexer channel. After the sample delay, the device connects the multiplexer output to the sample capacitor for the specified sample time. After this sample window closes, it disconnects the multiplexer output from the sample capacitor so that changes on the input pin will not alter the stored charge while the conversion is in progress. The device then zeros the comparator and begins the conversion. The A/D converter uses a successive approximation algorithm to perform the analog-to-digital conversion. The converter hardware consists of a 256-resistor ladder, a comparator, coupling capacitors, and a 10-bit successive approximation register (SAR) with logic that guides the process. The resistive ladder provides 20 mV steps ( $V_{REF} = 5.12$  volts), while capacitive coupling creates 5 mV steps within the 20 mV ladder voltages. Therefore, 1024 internal reference voltage levels are available for comparison against the analog input to generate a 10-bit conversion result. In 8-bit conversion mode, only the resistive ladder is used, providing 256 internal reference voltage levels. The successive approximation conversion compares a sequence of reference voltages to

the analog input, performing a binary search for the reference voltage that most closely matches the input. The  $\frac{1}{2}$  full scale reference voltage is the first tested. This corresponds to a 10-bit result where the most-significant bit is zero and all other bits are ones (011111111). If the analog input was less than the test voltage, bit 10 of the SAR is left at zero, and a new test voltage of  $\frac{1}{4}$  full scale (001111111) is tried. If the analog input was greater than the test voltage, bit 9 of SAR is set. Bit 8 is then cleared for the next test (010111111). This binary search continues until 10 (or 8) tests have occurred, at which time the valid conversion result resides in the AD\_RESULT register where it can be read by software. The result is equal to the ratio of the input voltage divided by the analog supply voltage. If the ratio is 1.00, the result will be all ones. The following A/D converter parameters are programmable:

- conversion input — input channel
- zero-offset adjustment — no adjustment, plus 2.5 mV, minus 2.5 mV, or minus 5.0 mV
- conversion times — sample window time and conversion time for each bit
- operating mode — 8- or 10-bit conversion or 8-bit high or low threshold detection
- conversion trigger — immediate or EPA starts

### 19(III) The External AD Converters (AD0809)

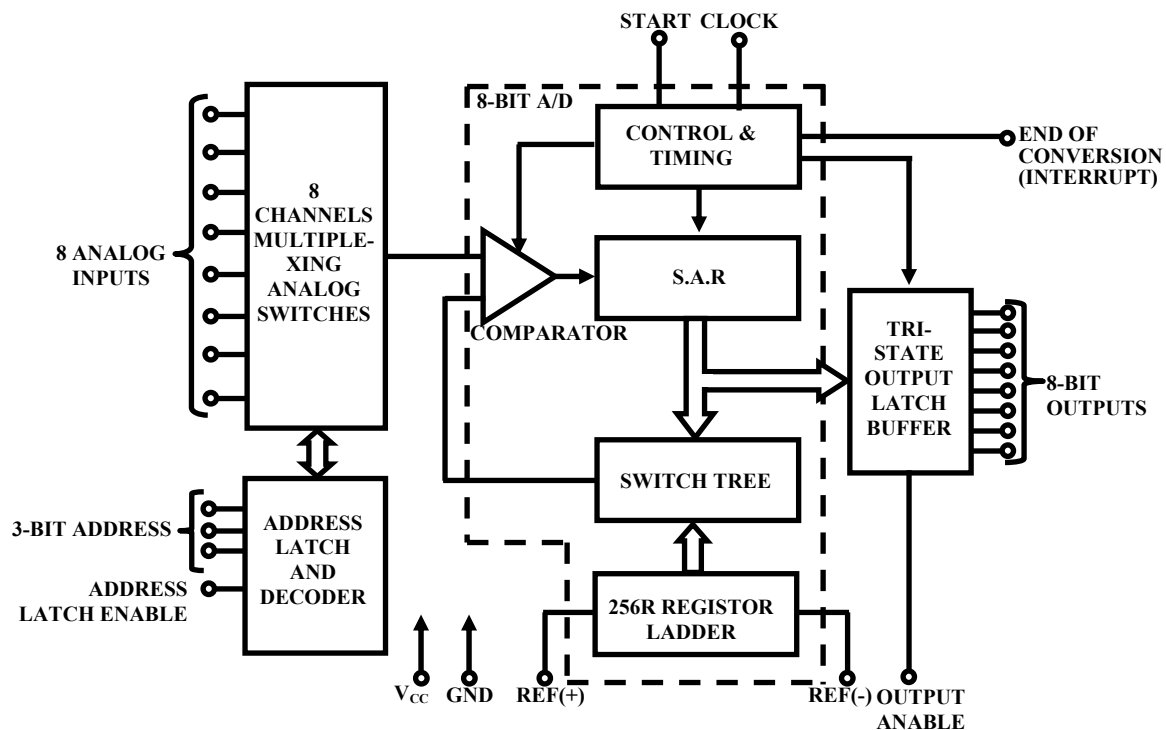
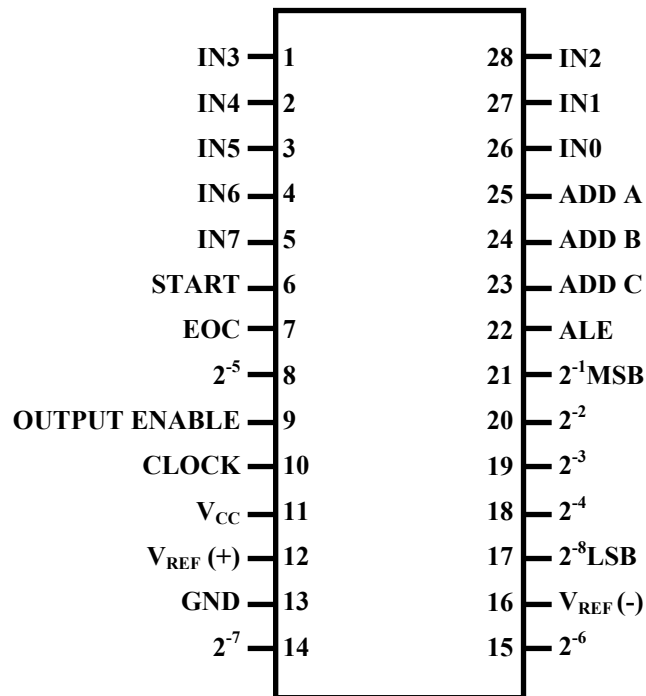


Fig. 19.4 The internal architecture of 0809 AD converter



**Fig. 19.5 The signals of 0809 AD converter**

## Functional Description

### Multiplexer

The device contains an 8-channel single-ended analog signal multiplexer. A particular input channel is selected by using the address decoder. Table 1 shows the input states for the address lines to select any channel. The address is latched into the decoder on the low-to-high transition of the address latch enable signal.

**TABLE 1**

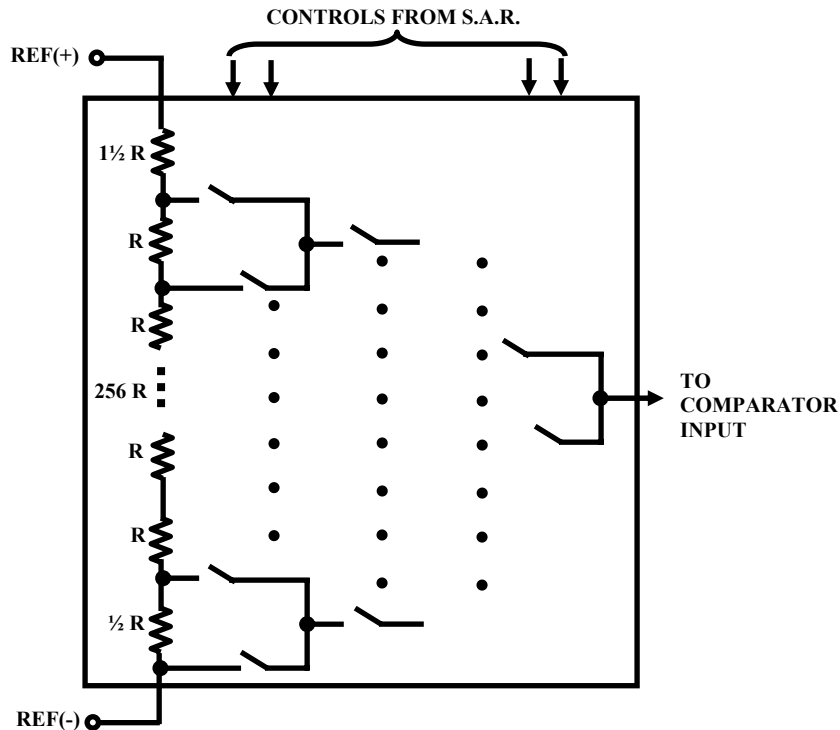
SELECTED ANALOG CHANNEL	ADDRESS LINE		
	C	B	A
IN0	L	L	L
IN1	L	L	H
IN2	L	H	L
IN3	L	H	H
IN4	H	L	L
IN5	H	L	H
IN6	H	H	L
IN7	H	H	H

### The Converter

This 8-bit converter is partitioned into 3 major sections: the 256R ladder network, the successive approximation register, and the comparator. The converter's digital outputs are positive true. The



256R ladder network approach (Figure 1) was chosen over the conventional R/2R ladder because of its inherent monotonicity, which guarantees no missing digital codes. Monotonicity is particularly important in closed loop feedback control systems. A non-monotonic relationship can cause oscillations that will be catastrophic for the system. Additionally, the 256R network does not cause load variations on the reference voltage.



**Fig. 19.6 The 256R ladder network**

The bottom resistor and the top resistor of the ladder network in Fig.19.6 are not the same value as the remainder of the network. The difference in these resistors causes the output characteristic to be symmetrical with the zero and full-scale points of the transfer curve. The first output transition occur when the analog signal has reached +1/2 LSB and succeeding output transitions occur every 1 LSB later up to full-scale. The successive approximation register (SAR) performs 8-iterations to approximate the input voltage. For any SAR type converter, n-iterations are required for an n-bit converter. Fig.19.7 shows a typical example of a 3-bit converter. The A/D converter's successive approximation register (SAR) is reset on the positive edge of the start conversion (SC) pulse. The conversion is begun on the falling edge of the start conversion pulse. A conversion in process will be interrupted by receipt of a new start conversion pulse. Continuous conversion may be accomplished by tying the end-of-conversion (EOC) output to the SC input. If used in this mode, an external start conversion pulse should be applied after power up. End-of-conversion will go low between 0 and 8 clock pulses after the rising edge of start conversion. The most important section of the A/D converter is the comparator. It is this section which is responsible for the ultimate accuracy of the entire converter.



## The timing Diagram (Fig.19.9)

The address latch enable signal and the start conversion are almost made high at the same time as per the connections in Fig.19.8. The analog input should be stable across the hold capacitor for the conversion time( $t_c$ ). The digital outputs remain tri-stated till the output is enabled externally by the Output Enable(OE) signal. The comparator input changes by the SAR counter and switch tree through the ladder network till the output almost matches the voltage at the selected analog input channel.

Important Specifications

8- time-multiplexed analog channels

Resolution 8 Bits

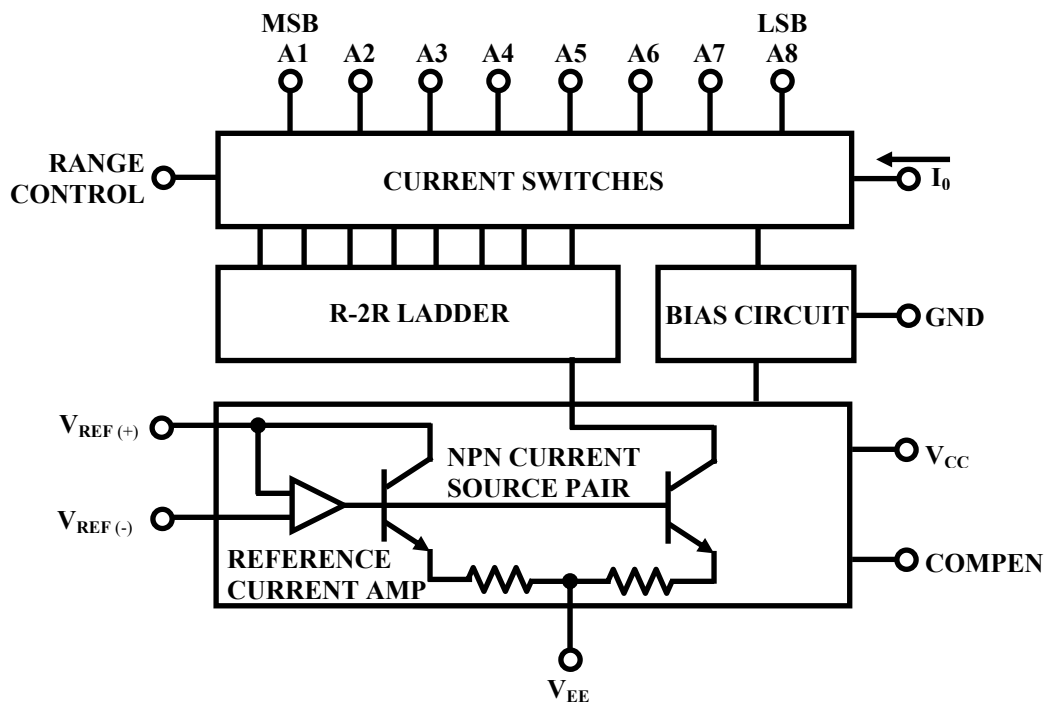
Supply 5 VDC

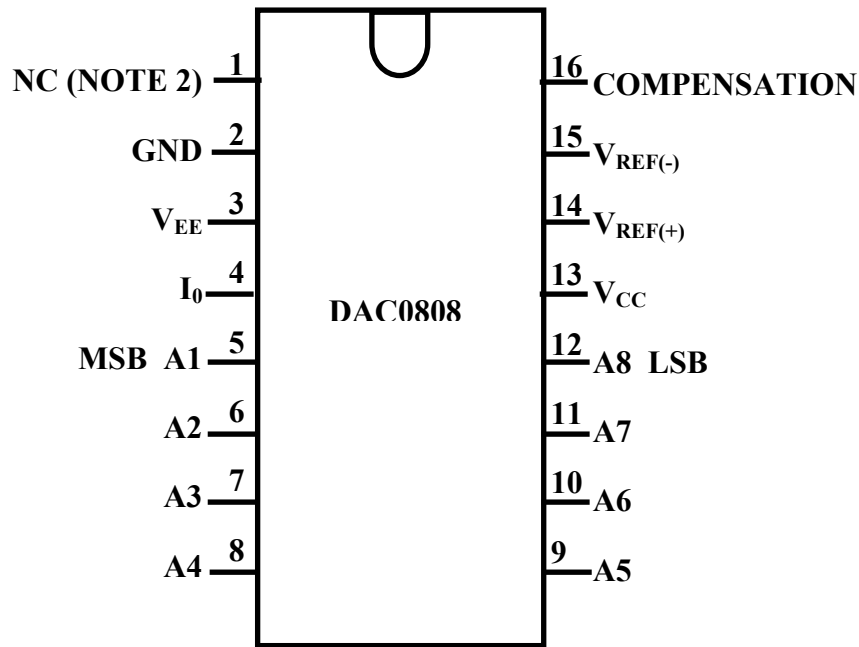
Average Power consumption 15 mW

Conversion Time 100  $\mu$ s

## 19(IV) The DA Converter DAC0808

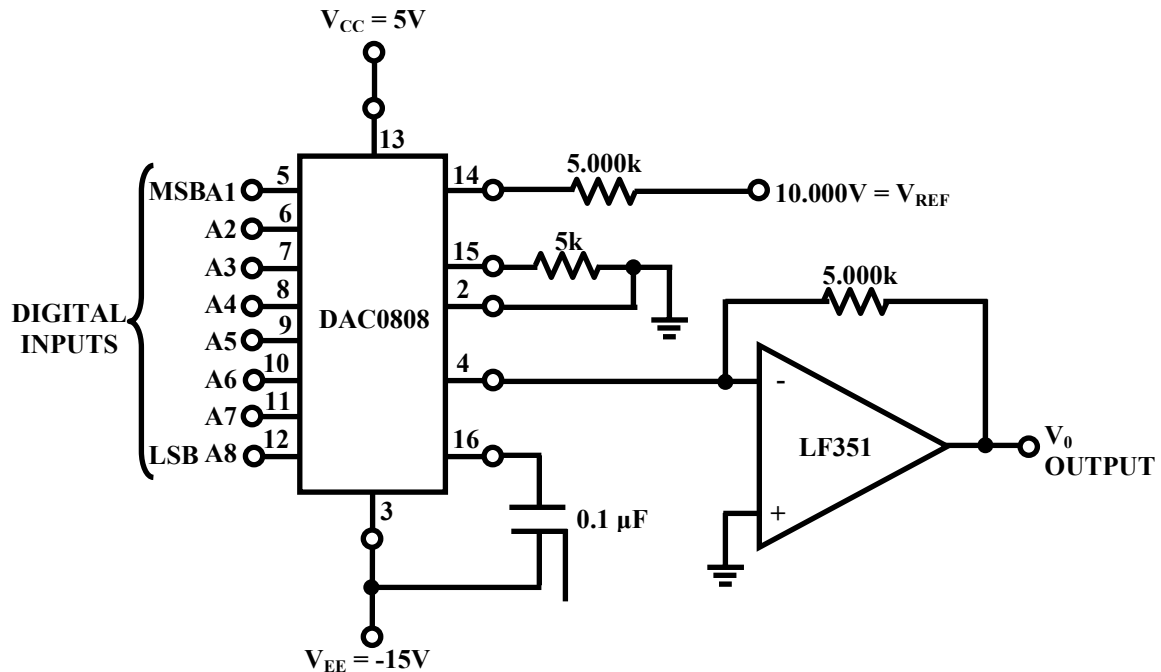
The DAC0808 is an 8-bit monolithic digital-to-analog converter (DAC). Fig.19.9 shows the architecture and pin diagram of such a chip.





**Fig. 19.9 The DAC 0808 Signals**

The pins are labeled A1 through A8, but note that A1 is the Most Significant Bit, and A8 is the Least Significant Bit (the opposite of the normal convention). The D/A converter has an output current, instead of an output voltage. An op-amp converts the current to a voltage. The output current from pin 4 ranges between 0 (when the inputs are all 0) to  $I_{max} * 255/256$  when all the inputs are 1. The current,  $I_{max}$ , is determined by the current into pin 14 (which is at 0 volts). Since we are using 8 bits, the maximum value is  $I_{max} * 255/256$ . The output of the D/A converter takes some time to settle. Therefore there should be a small delay before sending the next data to the DA. However this delay is very small compared to the conversion time of an AD Converter, therefore, does not matter in most real time signal processing platforms. Fig.19.10 shows a typical interface.



**Fig. 19.10 Typical connection of DAC0808**

LF351 is an operational amplifier used as current to proportional voltage converter. The 8-digital inputs at A8-A1 is converted into proportional current at pin no.4 of the DAC. The reference voltages(10V) are supplied at pin 14 and 15(grounded through resistance). A capacitor is connected across the Compensation pin 16 and the negative supply to bypass high frequency noise.

#### Important Specifications

±0.19% Error

Settling time: 150 ns

Slew rate: 8 mA/μs

Power supply voltage range: ±4.5V to ±18V

Power consumption: 33 mW @ ±5V

## 19(V) Conclusion

In this lesson you learnt about the following

The internal AD converters of 80196 family of processor

The external microprocessor compatible AD0809 converter

A typical 8-bit DA Converter

Both the ADCs use successive approximation technique. Flash ADCs are complex and therefore generate difficult VLSI circuits unsuitable for coexistence on the same chip. Sigma-Delta need very high sampling rate.

## Question Answers

Q.1. What are the possible errors in a system as shown in Fig. 19.2?

**Ans:**

*Stage-1 Signal Amplification and Conditioning* This can also amplify the noise.

*Stage-2 Anti-aliasing Filter* Some useful information such as transients in the real systems cannot be captured.

*Stage-3 Sample and Hold* The leakage and electromagnetic interference due to switching

*Stage-4 Analog to Digital Converter* Quantization error due to finite bit length

*Stage-5 Digital Processing and Data manipulation in a Processor:* Numerical round up errors due to finite word length and the delay caused by the algorithm.

*Stage-6 Processed Digital Values* are temporarily stored in a latch before D-A conversion: Error in reconstruction due to zero-order approximation

Q.2 Why it is necessary to separate the digital ground from analog ground in a typical ADC?

**Ans:** Digital circuit noise can get to analogue signal path if separate grounding systems are not used for digital and analogue parts. Digital grounds are invariably noisier than analog grounds because of the switching noise generated in digital chips when they change state. For large current transients, PCB trace inductances causes voltage drops between various ground points on the board (ground bounce). Ground bounce translates into varying voltage level bounce on signal lines. For digital lines this isn't a problem unless one crosses a logic threshold. For analog it's just plain noise to be added to the signals.

Source:

<http://www.nptel.ac.in/courses/Webcourse-contents/IIT%20Kharagpur/Embedded%20systems/Pdf/Lesson-19.pdf>