EE 354 **Fall 2016**
A to D and D to A Conversion

Analog to Digital Conversion:

There are four main types of Analog to Digital converters and probably four to eight more types that are slower and get used less frequently. The four types of conversion that we will look at here are:

1. Successive Approximation Register (SAR)
2. Dual Slope
3. Flash Converter
4. Sigma-Delta Converter

Successive Approximation Register (SAR) Converters:
This algorithm is akin to doing division when you are given only a multiplier. Figure 1 shows such a scheme.

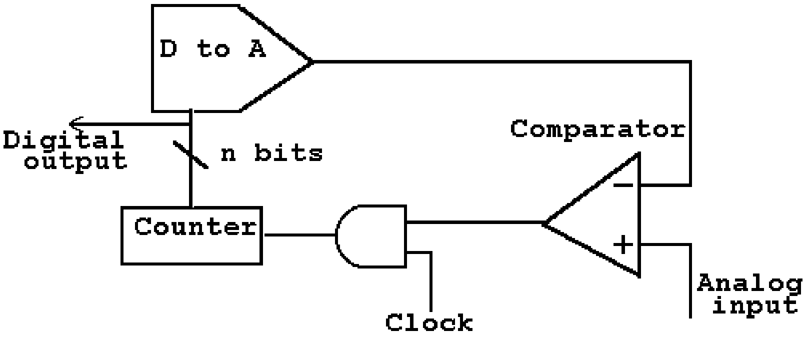

Figure 1
A feedback converter. The block labeled counter can be replaced with hardware to allow a binary search for the correct answer.

In Figure 1, take the counter to be started at 0 so the output of the D to A converter is zero. The D to A output goes to a comparator which compares it to the analog input. If the comparator finds that the analog input is higher than the D to A input it outputs a 1 allowing the counter to be clocked to a higher number. A new D to A conversion is performed and this process continues until such time as the D to A output is just above the Analog input at which time the comparator output goes to zero and the counting stops.

In an SAR converter the counting logic is replaced by an SAR register. This register and hardware effectively do a binary search for the answer by guessing from the most significant bit to the least significant bit. This process guarantees that for n-bit resolution we need only guess n times at the result.

Dual Slope Conversion:
Dual slope converters are also called integrating converters. These work by producing a ramp function which is compared to the analog input. Note that a ramp function can be created by integrating a square wave. A timer is started when the ramp begins and is stopped when the ramp reaches the value of the input analog signal. The time on the counter is proportional to the

height of the analog input signal. Thus, for this converter there is no need for a separate D to A converter. There is however, a problem with calibration. To fix this two ramps are used and that's where the name "dual slope" converter comes from. The first ramp can be generated by charging a capacitor with a constant current source. When the capacitor voltage reaches the analog input level the capacitor is discharged to produce a negative going ramp which is stopped when its value hits the original capacitor value (somewhere near zero). This process removes such factors as the capacitor leakage and nonlinear charging and makes the conversion more accurate.

Flash Converter:

The flash converter that has gained wide acceptance and is straight forward to understand. The flash converter employs an array of comparators and compares the analog input to successively higher analog values “in parallel”. This scheme is shown in Figure 2. In this type of converter there are 2^{B-1} comparators where the output is an B-bit binary number.   So an 8-bit converter requires 255 comparators and a 10 bit converter requires 1023 comparators. The only time required for a complete conversion is that needed for the electrical signal to flow through the comparators and the logic. This is typically a matter of a few tens of nano-seconds. Flash converters are used in video applications where very high speed is required but 6 to 8 bits of precision is adequate (say for gray scale levels).

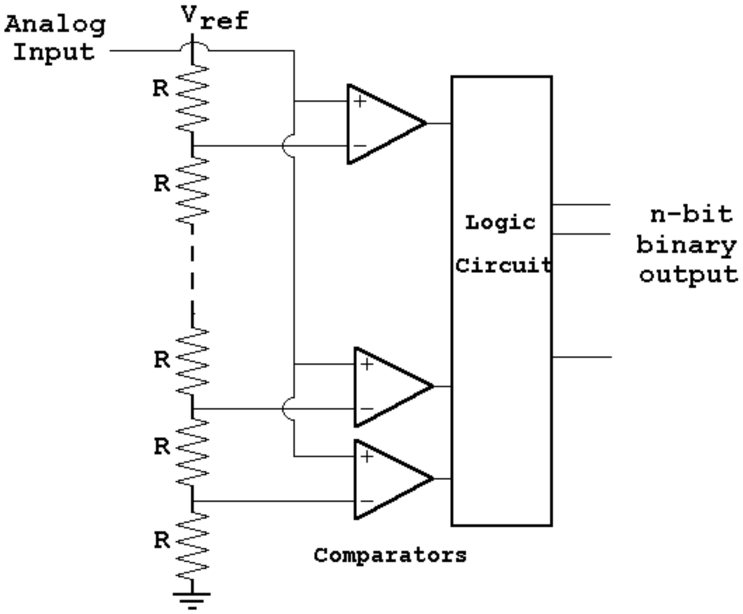

Figure 2
A flash converter.

Sigma-Delta Converter

Sigma-Delta modulation is a technique which uses oversampling to create an A to D (or a D to A) converter. The concepts used in sigma-delta (Σ-Δ) modulation were first introduced in 1962. The technique samples and analog signal at a very high sampling rate and at very low precision (typically only one-bit). These low resolution signals are then averaged over time to increase the precision at a decreased sampling rate. In practice, Σ-Δ A to D and D to A converters have conversion frequencies suitable for the audio range with 16 to 20 bits of precision, making them useful for numerous consumer audio applications (such as the CD player).

The "delta" portion of the Σ-Δ converter refers to a "difference" module which produces a signal equal to the difference between the actual input signal and a current estimate of that input signal. This difference signal can be thought of as a modulated form of the input signal where the modulation technique is *pulse code modulation* (PCM) and the circuit which generates the differences is called a delta modulator. A block diagram for such a modulator is shown in Figure 3.

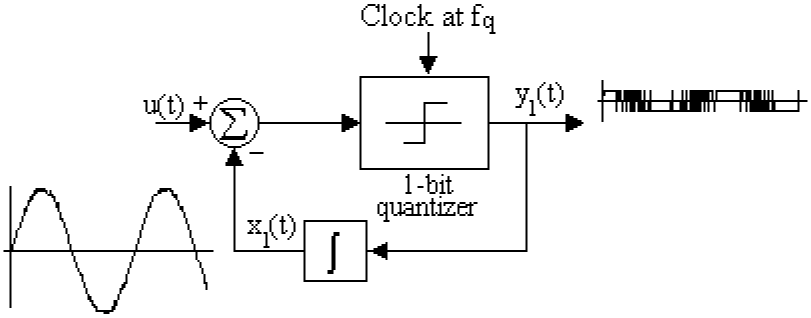

Figure 3

A Delta modulator. The input u(t) is a sinusoid. The estimated input, $x_1(t)$ and the modulated output, $y_1(t)$ are shown. The signal f_q is the clock for the quantizer.

In Figure 3 the input to the delta modulator is a sine wave. The 1-bit quantizer puts out either +1 or –1 depending on whether the difference it receives is positive or negative. The quantizer output is fed into an integrator to sum up the difference signals and produce an estimate of the input signal. The resulting output $y_1(t)$ is a delta modulated signal. The signal labeled "Clock at f_q" determines when the difference signal is quantized. Thus the integrator receives signals which are either +1 or –1 and may be changing as fast as the clock rate. If the output signal of the delta modulator is transmitted over a channel the original signal may be recovered by a demodulator circuit like that shown in Figure 4. In this circuit the integrator recovers the estimated signal. A low pass filter then blocks out the high frequency content introduced by the delta modulation and a decimator reduces the sampling rate. The resulting output is the original signal in discrete sampled form. Therefore, if we combine a delta modulator and a sigma demodulator we have an A to D converter since the input can be a continuous time signal and the output is digital.

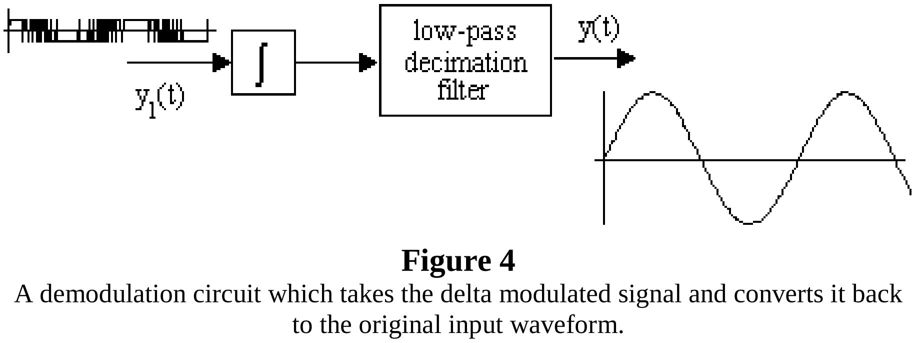

Figure 4

A demodulation circuit which takes the delta modulated signal and converts it back to the original input waveform.

In combining the delta modulator with the sigma demodulator we can make some simplifications as shown in Figure 5.

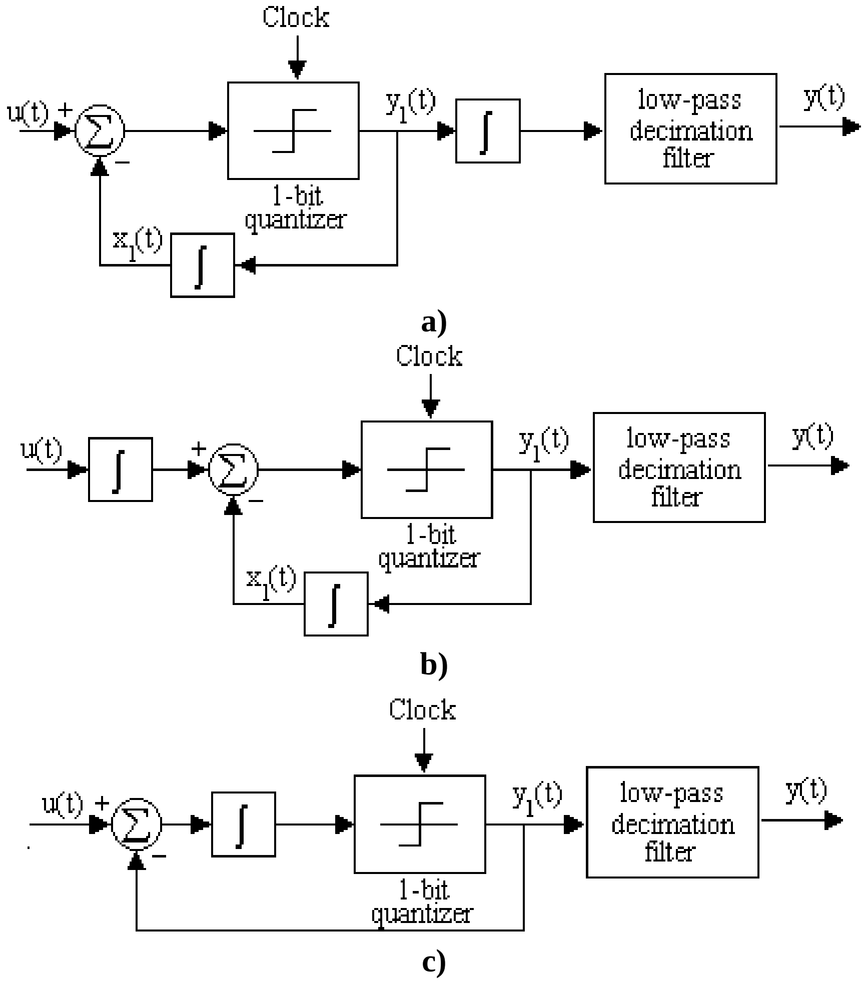

Figure 5

a) A delta modulator combined with a sigma demodulator. b) The integrator is moved in front t
form a sigma delta circuit c) Part b can be simplified by eliminating one integrator.

In Figure 5a, the delta modulator and the sigma demodulator are combined to form a single circuit. Because this circuit is linear we can relocate the integrator without changing the transfer function of the circuit. This relocation is done in Figure 5b and the result is called a sigma-delta converter. If we take the integrator to have a transfer function I, the quantizer to have a transfer function Q, and the low-pass filter to have a transfer function of L, the transfer function of the circuit in Figure 5b is given by the following:

$$H = I \cdot \frac{Q}{1+QI} \cdot L = \frac{QI}{1+QI} \cdot L$$

From this equation it is clear that the circuit shown in Figure 5b can be reduced to that shown in Figure 5c which has a single integrator and unity feedback. Figure 5c is the traditional schematic for a sigma-delta converter.

In Figure 5c, the input signal, u(t), the summing junction, and the integrator are analog components. The low pass decimation filter must be a digital component in order to obtain a digital output. To implement the clock on the quantizer we can add a unit delay in the feedback path with the delay time set to the clock period. The quantizer becomes an analog comparator. This practical modification is shown in Figure 6. The block labeled "One-bit D toA Converter" contains a unit delay register driven at the clock frequency with appropriate level shifting so that its output is equivalent to the most significant bit of the input range. Thus, the output of the D to A converter is a one bit approximation to the input signal.

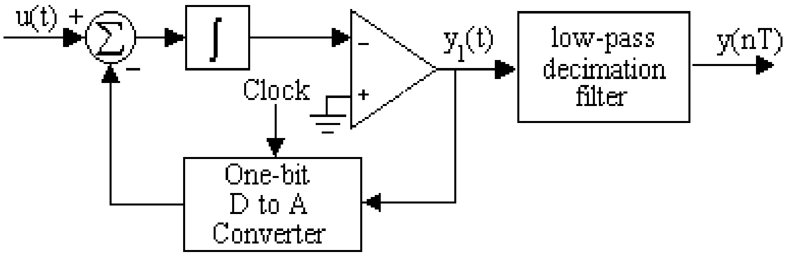

Figure 6
A sigma-delta A to D converter with a one-bit D to A converter in the feedback path.

The clock frequency of the sigma-delta loop is necessarily very high. In many implementations, its frequency is high enough that a sample and hold circuit is not necessary on the analog input. The clock frequency for the one-bit D to A converter, f_Σ, is typically many times larger than f_s, the desired sample frequency for the input signal.

The low-pass decimation filter on the output of the A to D converter has two functions. The first of these is to eliminate the high frequency content of the signal coming in. This high frequency content is a result of the high sampling rate of the Σ-Δ loop and the fact that the output of this loop is a series of pulses and not smooth sinusoids. The second function of the low-pass decimation filter is to reduce the sample rate of the Σ-Δ loop to the desired sampling rate. This is the decimation process. In most practical converters this double function is combined into a single circuit.

Analog to digital converters which use sigma-delta modulation are referred to as sigma-delta converters, or in some consumer products they are advertised as "one-bit" converters because of the signum function which converts one bit at a time. Texas Instruments is one of several manufacturers making a sigma-delta converter as a one-chip integrated circuit. Converters with 18 bits of precision at 48khz are commonly available.

AT89C51CC03 A to D Conversion:
The A to D converter on the AT89C51CC03 is an 8-channel 10-bit converter of the SAR type. A block diagram for the converter is shown in Figure 7. The conversion time is about 16 µsec. The A to D converter has multiple operating modes that allow the user to scan a single channel, automatically scan multiple channels, single step through multiple channels, and variations of these methods. For this class we will look only at converting a single channel called the "Fixed Channel Single Conversion Mode".

Figure 59. ADC Description

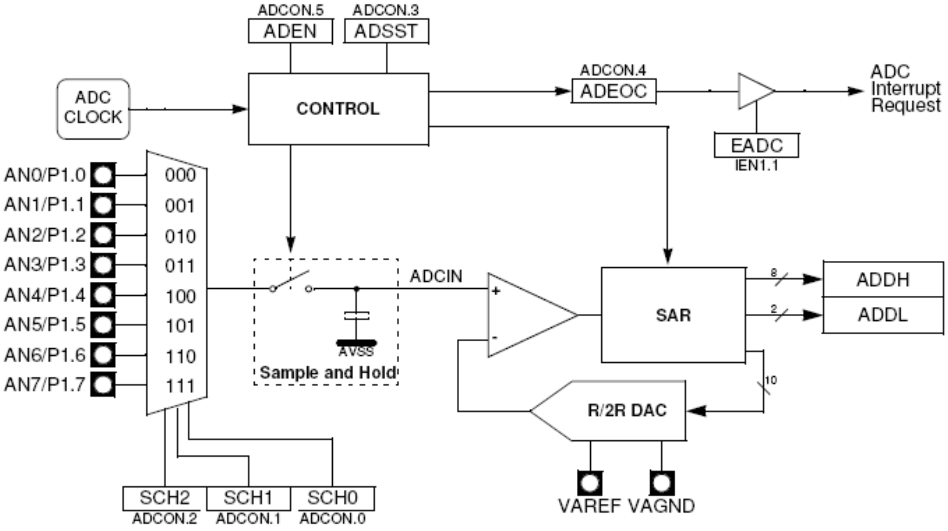

Figure 7
The ADC on the AT89C51CC03.

The analog input channels are named AN0 to An7 and are input through the bits on Port 1. The converter places the 10-bit result in the two registers ADDH and ADDL.

To select a channel for a single conversion you must set bits 2, 1, and 0 of the A to D control register ADCON. The bits in the ADCON register are shown in Figure 8.

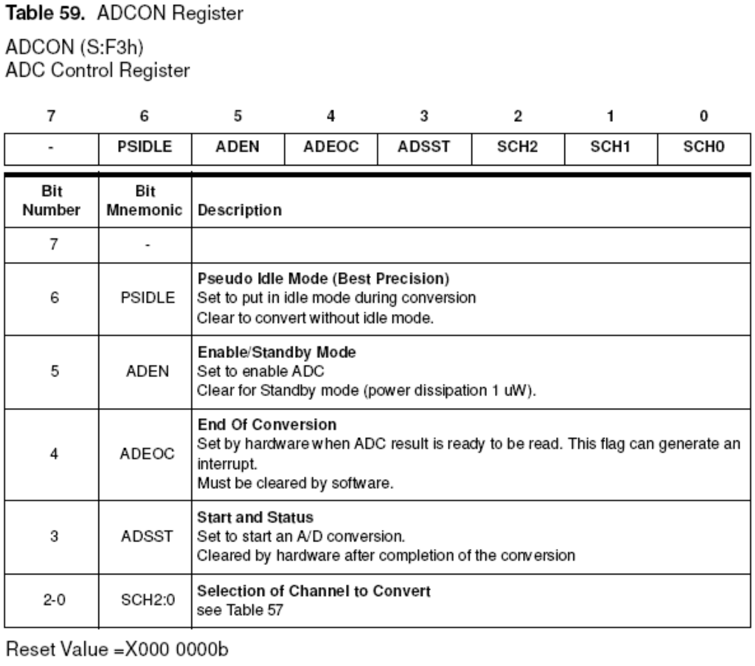

Table 59. ADCON Register

ADCON (S:F3h)
ADC Control Register

7	6	5	4	3	2	1	0
-	PSIDLE	ADEN	ADEOC	ADSST	SCH2	SCH1	SCH0

Bit Number	Bit Mnemonic	Description
7	-	
6	PSIDLE	**Pseudo Idle Mode (Best Precision)** Set to put in idle mode during conversion Clear to convert without idle mode.
5	ADEN	**Enable/Standby Mode** Set to enable ADC Clear for Standby mode (power dissipation 1 uW).
4	ADEOC	**End Of Conversion** Set by hardware when ADC result is ready to be read. This flag can generate an interrupt. Must be cleared by software.
3	ADSST	**Start and Status** Set to start an A/D conversion. Cleared by hardware after completion of the conversion
2-0	SCH2:0	**Selection of Channel to Convert** see Table 57

Reset Value =X000 0000b

Figure 8
A to D Control register for the AT89C51CC03 processor.

Bit-5 of the ADCON register is the A to D Enable bit and must be set to 1 to enable the converter. Bit-3 of the ADCON register is the A to D start and status bit. This bit is set to 1 to begin the conversion and is reset to 0 by hardware when the conversion is complete. Bit-4, ADEOC is the end of conversion signal. ADEOC must be set to zero before a conversion begins. This bit goes to one when the conversion ends and may be used to trigger an interrupt.

Note that since Port 1 is multiplexed it can be used as either an analog input port for the A to D converter or it can be used as a normal programmable I/O port. The register that determines how Port 1 is used is the A to D configuration (ADCF) register. Each bit in the ADCF register individually controls a pin on port 1. If, for example, bit 0 in the ADCF register is set to 1 then the converter takes bit 0 of Port 1 to be in the analog input mode. The bits in ADCF must be set in your program prior to using the converter.

The A to D Converter has a clock signal generated by the CPU clock and prescaled as shown in Figure 9.

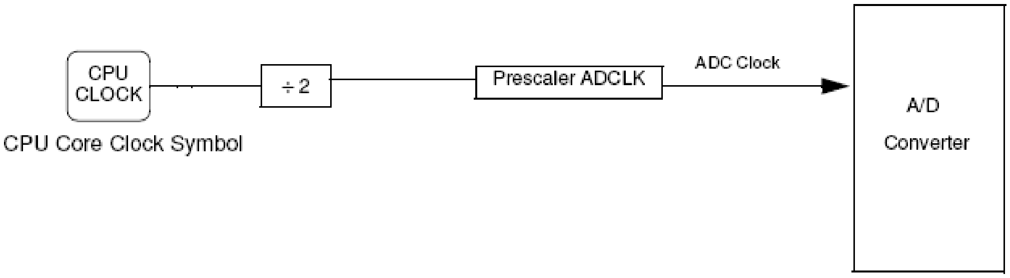

Figure 9

ADC clock.

The typical ADC Clock is 700 KHz. The user must set the prescaler such that the ADC clock is near this frequency. The latest specification sheet gives not maximum so it is likely the converter will run faster if needed.

$f_{ADC} = f_{CPU}/(4*\text{Prescaler})$ in normal mode

$f_{ADC} = f_{CPU}/(2*\text{Prescaler})$ in x2 mode

If the prescaler is 0 then $f_{ADC} = f_{CPU}/128$ in normal mode

If the prescaler is 0 then $f_{ADC} = f_{CPU}/64$ in X2 mode

The values which the user can load into the prescaler are limited as shown in Figure 10.

For the board used in this class the CPU clock is running at 29.4 MHz. 29.4 MHz/64 = 0.459375 MHz . A complete A to D conversion requires 4 µsec for set up time plus 11 clock cycles. With the prescaler set to 0 for our system the conversion time is:

$$\text{ADC Conversion time} = 4\mu\text{sec} + 11*(1/.459375*10^6) = 27.9\ \mu\text{sec}$$

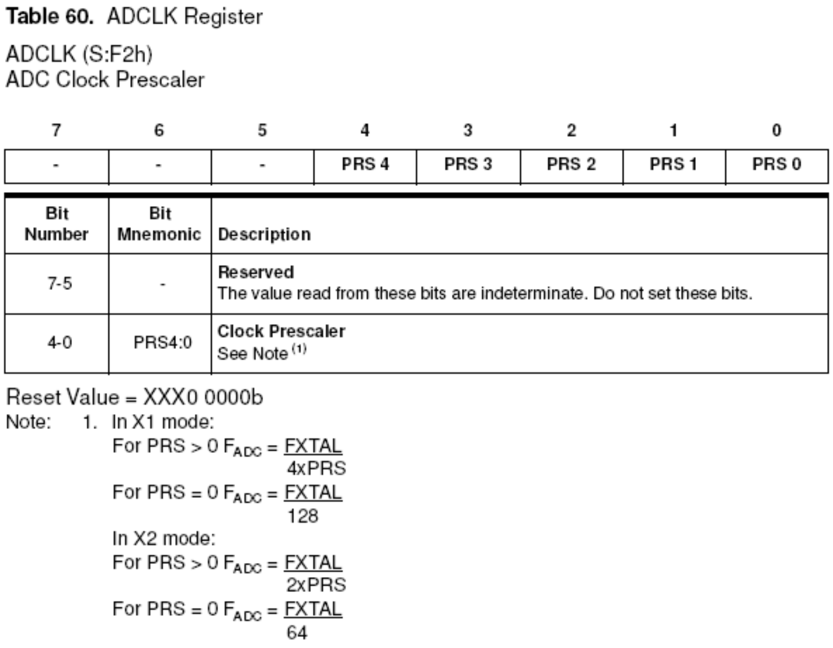

Table 60. ADCLK Register

ADCLK (S:F2h)
ADC Clock Prescaler

7	6	5	4	3	2	1	0
-	-	-	PRS 4	PRS 3	PRS 2	PRS 1	PRS 0

Bit Number	Bit Mnemonic	Description
7-5	-	**Reserved** The value read from these bits are indeterminate. Do not set these bits.
4-0	PRS4:0	**Clock Prescaler** See Note [(1)]

Reset Value = XXX0 0000b

Note: 1. In X1 mode:

For PRS > 0 $F_{ADC} = \frac{FXTAL}{4xPRS}$

For PRS = 0 $F_{ADC} = \frac{FXTAL}{128}$

In X2 mode:

For PRS > 0 $F_{ADC} = \frac{FXTAL}{2xPRS}$

For PRS = 0 $F_{ADC} = \frac{FXTAL}{64}$

Figure 10
The ADCLK register.

The 10-bit result of the A to D converter come out in ADDH and ADDL. ADDH holds bits 2 to 9 and ADDL holds bits 0 and 1. In C we can use ADDH as an 8-bit conversion value or, we can put all 10 bits into an integer like this:

```
result = (ADDH << 2) + ADDL
```

The 10-bit binary number that comes from the A/D converter is scales the voltage range from 0 to VAREF from 0 to 1023 in a linear fashion. In equation form we can write:

$\frac{v_{in}}{VAREF} = \frac{x}{1023}$ or $x = v_{in}\frac{1023}{VAREF}$ where x is the number produced by the A/D converter.

The value of VAREF must be between 2.4 volts and 3.0 volts. The AT89C51CC03 board has it set to 2.5 volts.

Digital to Analog Conversion

Electrically, digital to analog conversion is a much simpler process than analog to digital conversion. Sequential circuitry is not needed and the result can be obtained much faster. The AT89C51CC03 has an on board D to A converter but it is used exclusively by the A to D converter circuitry and is not available for general purpose use by the user. A separate 8-bit D to A converter has been added to the AT89C51CC03 circuit board by way of the AD7524 chip.

The simplest kind of D to A converter to understand is the weighted resistor converter. Consider a binary number Y_B, having a total of B bits that can be represented as:

$Y_B = b_{B-1}b_{B-2}b_{B-3}\cdots b_1b_0$ where b_i is a binary digit (1 or 0)

Since the individual bits in the binary numbers represent powers of two we can write the "value" of Y_B as

$$Value\ of\ Y_B = 2^{B-1}b_{B-1} + 2^{B-2}b_{B-2} + \cdots + 2b_1 + b_0$$

Thus we see that each bit of a binary number has a weighting factor which is a power of two. The particular power of two that is used depends on the position of the bit in the binary number.

As an electrical circuit we can take the binary digits to be represented by discrete voltage levels - say 0 volts = binary zero and +5 volts = binary one. Zero volts across any resistor produces zero current and + 5 volts across any resistor produces a current inversely proportional to the resistance value. Thus to make D to A converter based on this concept we can change the resistor values to form weighting factors corresponding to powers of two and use an op amp to sum the currents. Figure 4.11 shows a schematic for how this might work. This figure is somewhat simplified from what would be done in practice. The binary bits have been assumed to be either at 0 volts or at +5 volts. In practice the bits would be used to control switches which would apply a reference voltage to the resistors. The resistor R_g is a gain resistor whose value is chosen so that the output voltage is in the appropriate range.

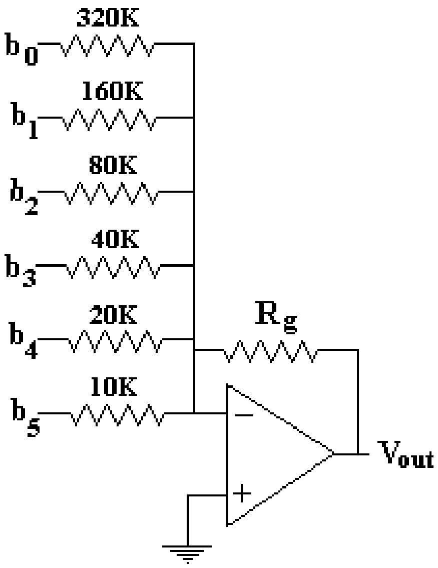

Figure 11

A weighted resistor type of D to A converter. In this converter the resistors are the weighting factors and the op amp sums the current.

Unlike the A to D process, digital to analog conversion does not require any counting, guessing, or other sequential process to arrive at a result. The speed is determined by the time it takes the signal to propagate through the components and no clock is needed.

One practical problem with the D to A converter shown in Figure 11 is that for a large number of bits the resistor values become unwieldy. For example, for a 10 bit converter, if the resistor for bit b_9 is 10K, the resistor for b_0 is 51.2 mega ohms. For a 12 bit converter the resistor for the b_{11}

bit would be 204.8 mega ohm. The large values also need to be specified to high precision. For example a 10 bit converter can receive numbers known to 1 part in 2^{10} or 0.1%. Such precise values on a wide range of resistors cannot be achieved except by trimming.

A second circuit for a D to A converter that overcomes some of the problems of precise multi-valued resistors is shown in Figure 12. This converter is referred to as a ladder type or sometimes as an R/2R resistor network type converter. There are only two resistor values used. In building a monolithic circuit it is much easier to construct two nearly identical resistors (within 1% of each other) than it is to construct two 1% resistors of particular but different values.

In Figure 12 the mechanical switches shown would be electronic switches controlled by the bits b_0 through b_5. The current reaching any switch is exactly one half of the amount of current reaching the previous switch because of the voltage divider network.

In practice D to A conversion time is rarely a significant factor in the design of signal processing units since converters are typically very fast. Sixteen bit converters that are suitable for audio applications are readily available. A typical 10 bit D to A converter may have a settling time of 200 nsec.

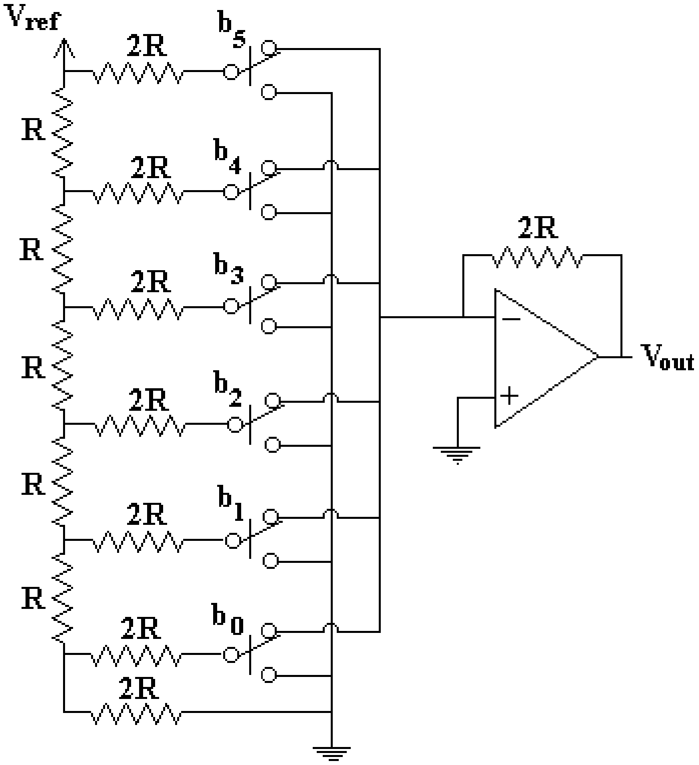

Figure 12

A six bit ladder type D to A converter. Note that only two different values of resistors are used. The digital input bits (b_0 to b_5) set the switches. The switches are shown in the "1" bit position.

The AD7524 chip used on the AT89C51CC03 board makes use of an R/2R ladder network. A block diagram of the AD7524 chip is shown in Figure 13. Note that there are 8 data bits plus, two output bits, a chip select, a reference voltage input and a write line. The analog output comes out as a current and op amps must be attached to convert this to a voltage signal. We have only one D to A converter in our system to the chip select can be tied to ground making the converter always enabled. The write line is active low. To use the converter we must apply a digital number (from port 2) to the digital inputs and put a low going pulse on the write line. The DAC 7524 converts and 8-bit number to an analog signal in 400 nsec.

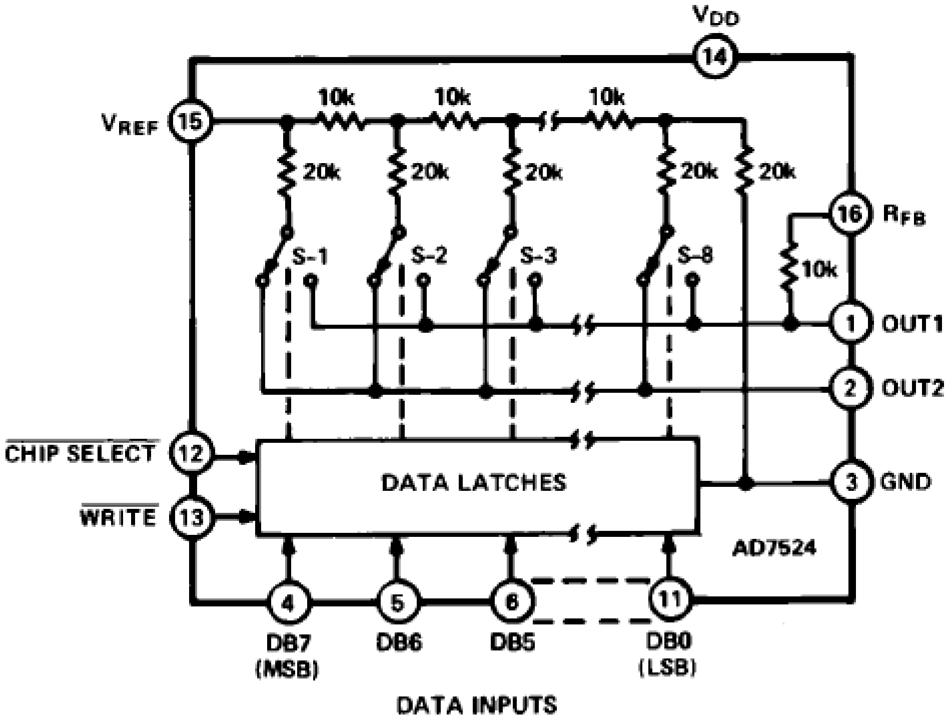

Figure 13
The AD7524 D to A converter chip.

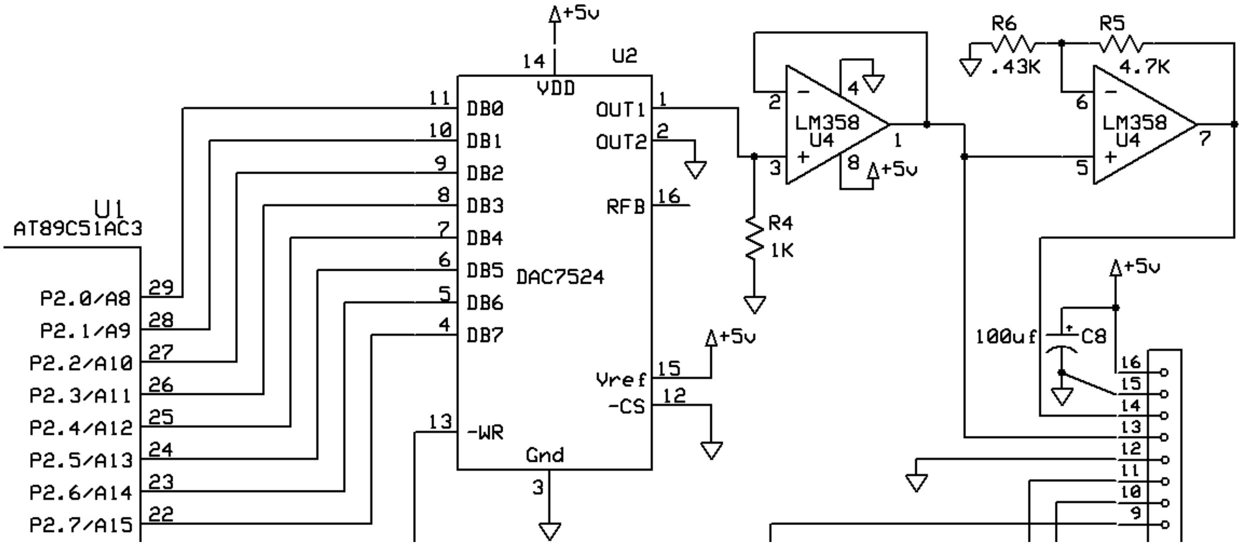

Figure 14
The DAC 7524 as implemented on the AT89C51CC03 board. Pin 13, the write line, is driven from P4.0 and is active low.

Figure 15 shows a complete program in C which does an analog to digital conversion on channel 0 and sends the result to the digital to analog converter on P2.

```
//AtoDTest.c
#include <at89c51cc03.h>
//Takes the input from the A/D converter channel 0 and
// sends it to P2 and P3.  Assumes P2 has D/A triggered
// by bit P4.0.  P2 holds 8 MSBs.
void main (void)
   {unsigned char tmp;
    ADCF  = 0x01;        // P1.0 = ADC[0]
    ADCON = 0x20;        // Enable ADC Function
    ADCLK = 0x00;        // Prescaler to 0
    EA = 0;              //Turn off interrupts
    while(1)             // Loop Forever
     {ADCON &= 0xF8;     // Reset ADC Channel Select
      ADCON |= 0x00;     // Select ADC = Ch0
      ADCON |= 0x20;     // Use Standard mode
      ADCON |= 0x08;     // Start ADC Convert
      tmp = (ADCON & 0x10);    // Get done bit
      while(tmp != 0x10)       // Loop until complete
        tmp = (ADCON & 0x10);
      P2 = ADDH;               // Send 8 MSB to P2
      P3 = ADDL;
      P4_0 = 0;                // Low going pulse to D to A
      P4_0 = 1;                //  write line
      ADCON &= 0xEF;           // Clear ADEOC = 0
     }
   }
```

Figure 15

A to D and D to A conversion on the AT89C51CC03 board.