CpE 390: Microprocessor Systems

Lecture 11

68HC12 Analog to Digital Converter (1)

A/D Converter

- A *transducer:* convert a non-electrical quantity into an electrical voltage
- A *voltage shifter* and *scaler:* transform the transducer output voltage into a range that handled by the A/D converter.
- An analog to digital (A/D) converter can convert a electrical voltage to a digital value.

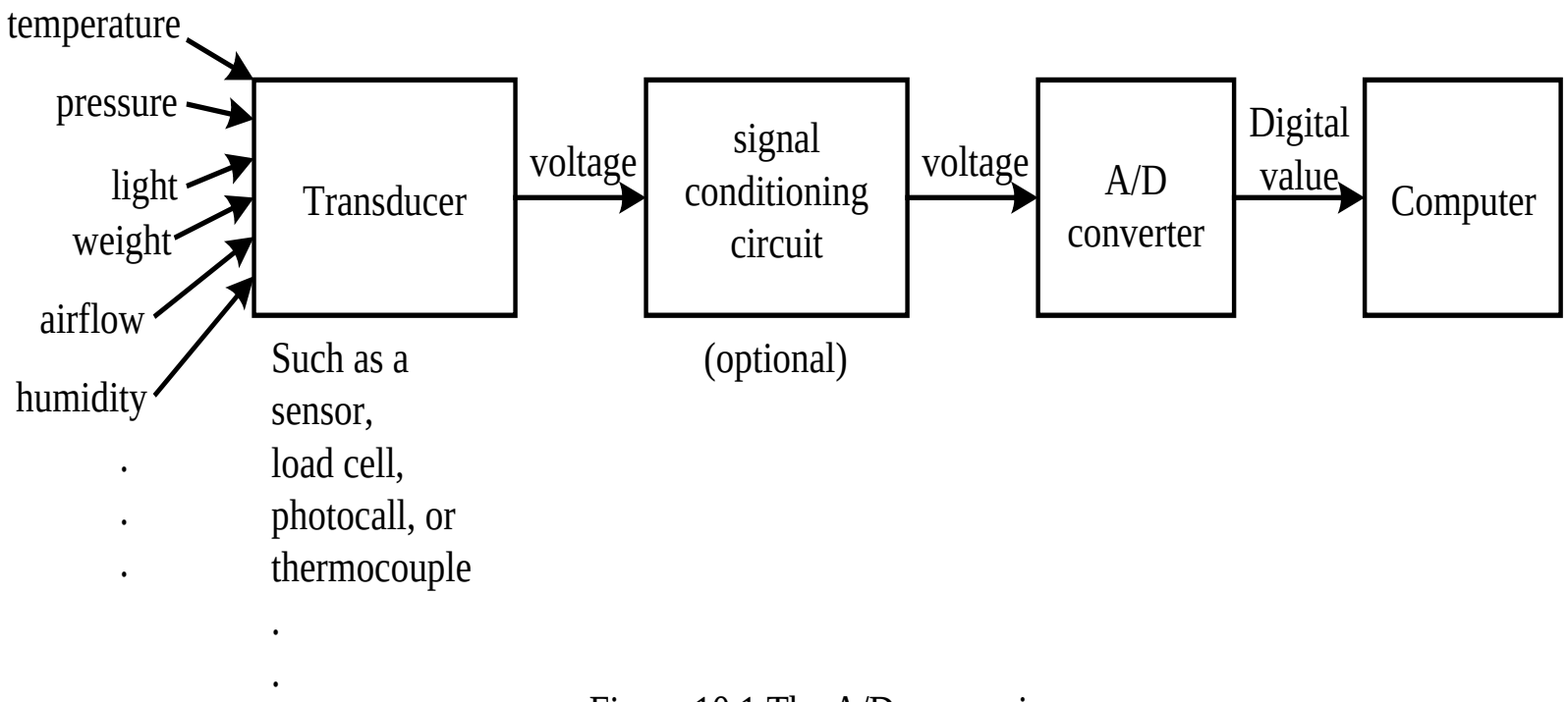


Figure 10.1 The A/D conversion process

68HC12 A/D conversion method: Successive Approximation Method

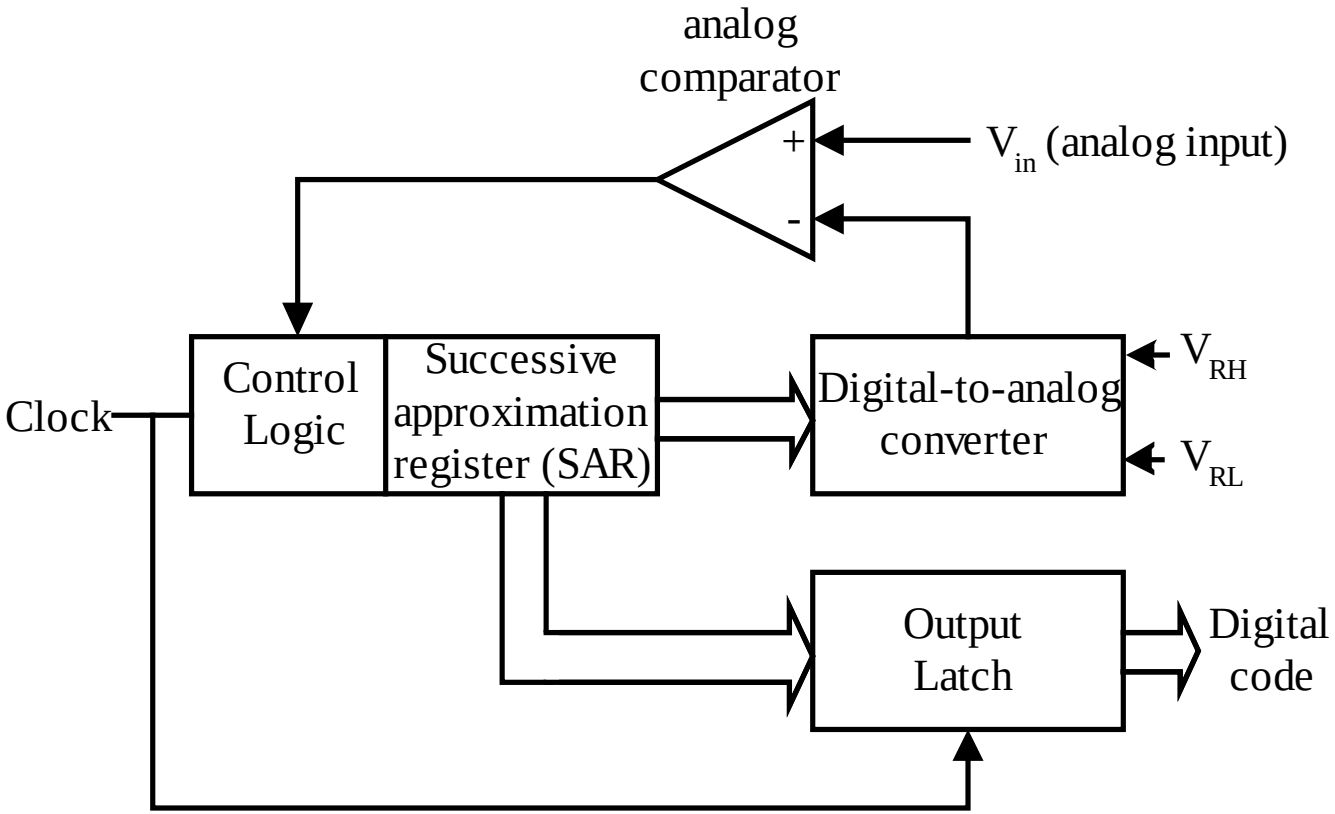


Figure 10.2 Block diagram of a successive approximation A/D converter

Algorithm of Successive Approximation Method

- Initialize the SAR register to 0.
- Starting from the most significant bit of SAR and work toward the least significant bit, for each bit

  1. Guess the bit to be a 1.
  2. Converts the value of the SAR to an analog voltage
  3. Compares the D/A output with the analog input.
  4. Clears the bit to 0 if the D/A output is larger.

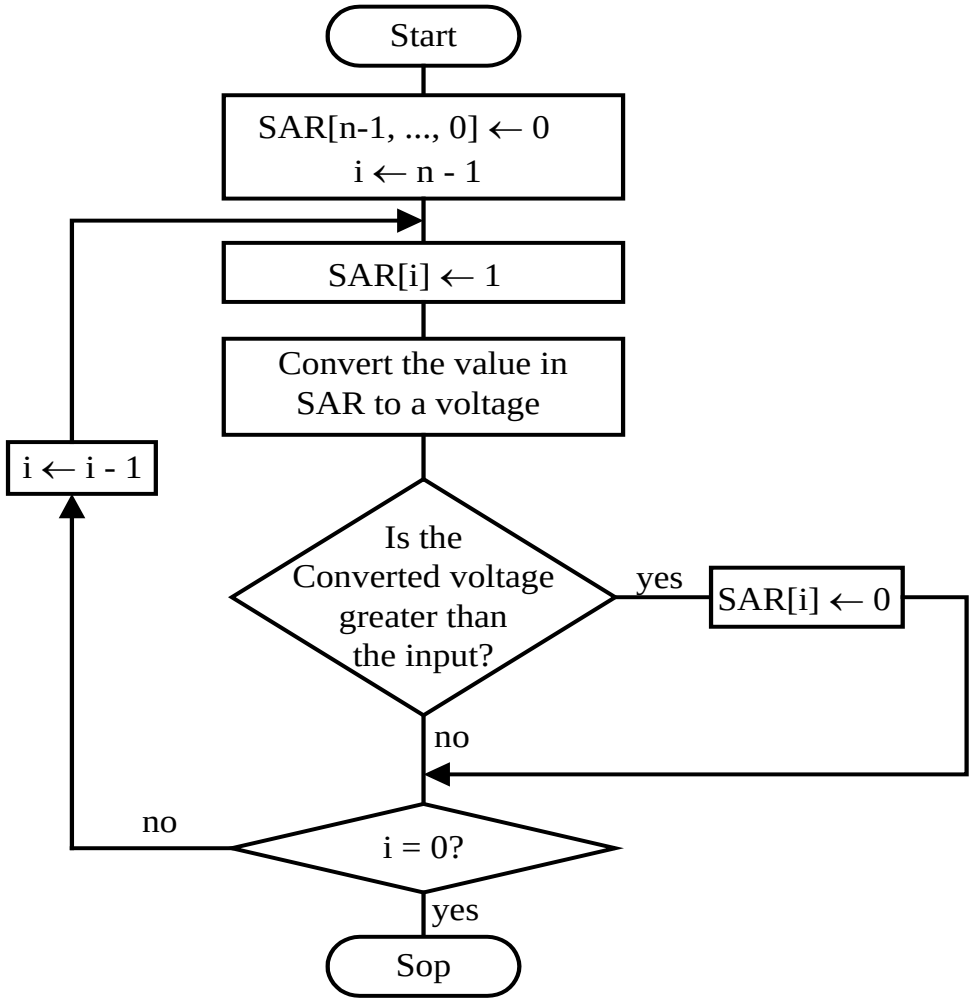


Figure 10.3 Successive approximation A/D conversion method

Signal Conditioning Circuits

Optimal Voltage Range for the A/D Converter

- A/D converter needs a low reference voltage VLREF (is often set to 0) and a high reference voltage VHREF (is often set to VCC) to operate.

- The A/D conversion result k corresponds to an analog voltage given by

$$\mathbf{Vx = V_{LREF} + (range \times k) \div (2^n - 1)}$$

where, n is the number of bits the A/D converter uses to represent a conversion result, n is also called the resolution of the A/D converter, range = $V_{HREF} - V_{HREF,}$

Example 10.1 Suppose there is a 12-bit A/D converter with V_{LREF} = 0.5V and V_{HREF} = 3.5V. Find the corresponding voltage values for A/D conversion results of 20, 100, 800, 1200, 2400, and 3600.

Solution:

range = $V_{HREF} - V_{LREF}$ = 3 V.

V(20) = $0.5V + (20 \times 3) \div (2^{12} - 1) = 0.515$ V
V(100) = $0.5V + (100 \times 3) \div (2^{12} - 1) = 0.573$ V
V(800) = $0.5V + (800 \times 3) \div (2^{12} - 1) = 1.086$ V
V(1200) = $0.5V + (1200 \times 3) \div (2^{12} - 1) = 1.379$ V
V(2400) = $0.5V + (2400 \times 3) \div (2^{12} - 1) = 2.258$ V
V(3600) = $0.5V + (3600 \times 3) \div (2^{12} - 1) = 3.137$ V

Voltage Scaling and Shifting Circuit

- Some transducers have output in the range of V_1~V_2.

- V_1 may be negative and V_2 may be different from the power supply.

- To take advantage of the whole dynamic range of the 68HC12 A/D converter, the scaling and shifting circuit will shift and scale the transducer output to the range of 0~V_{DD}.

- By choosing appropriate values for V_1 and resistors, the transducer output can be scaled and shifted to 0~V_{DD}.

Voltage Scaling Circuit

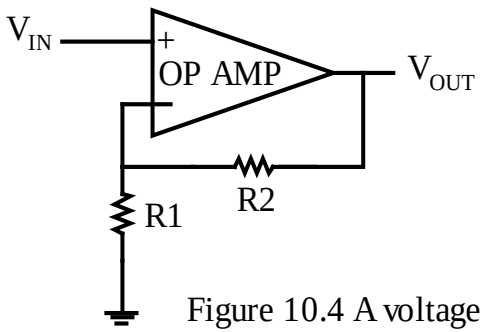


Figure 10.4 A voltage scaler

$$A_V = \frac{R1 + R2}{R1} = 1 + \frac{R2}{R1} \qquad 10.2$$

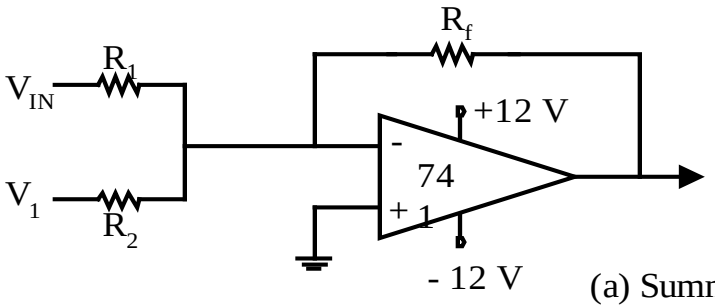
Example 10.2 Convert the output voltage of an A/D converter from 0-100mV to the range of 0-5V.

Solution:

$A_V = 1 + (R2/R1) = (5V/0.1V) = 50$

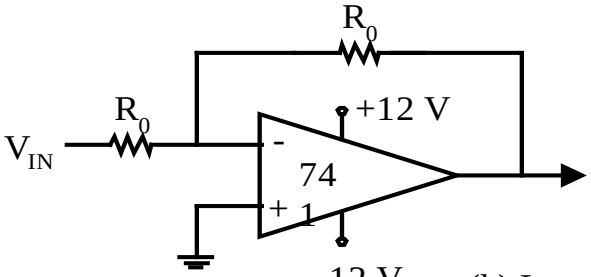
$\therefore$ R2/R1 = 49

Choose 6.8K for R1, then R2 = 330K. The R2/R1 ratio is 48.53. Error is smaller within 0.3%.



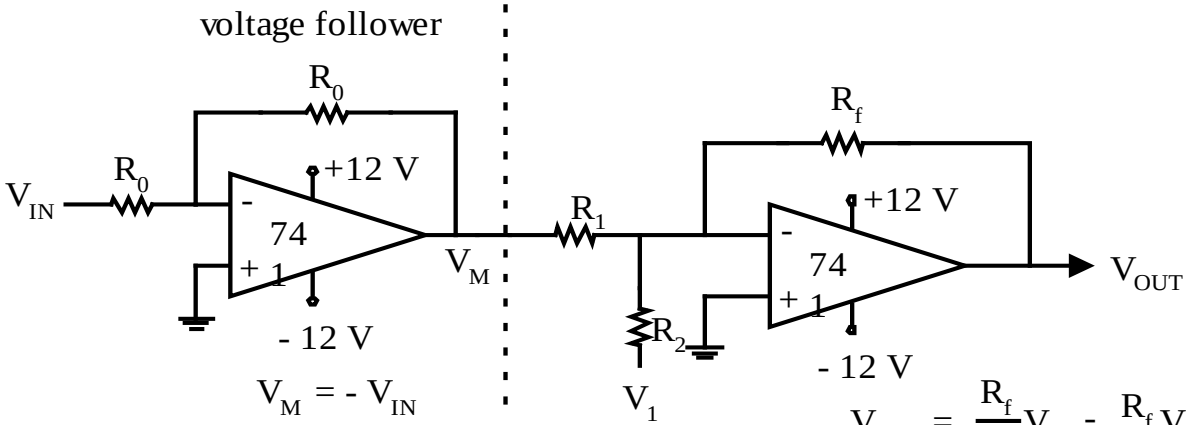
$$V_{OUT} = -\frac{R_f}{R_1}V_{IN} - \frac{R_f}{R_2}V_1 \qquad (10.3)$$

(a) Summing circuit



$$V_{OUT} = -V_{IN} \qquad (10-4)$$

(b) Inverting voltage follower



$$V_{OUT} = \frac{R_f}{R_1}V_{IN} - \frac{R_f}{R_2}V_1 \qquad (10-5)$$

(c) Level shifter and scaler

Figure 10.5 Level shifting and scaling circuit

Example 10.3 Choose the appropriate values of resistors and the adjusting voltage so that the circuit in Figure 10.5c can shift the voltage from –1.5V ~ 3.5V to 0V ~ 5V.

Solution:

$$0 = -1.5 \quad (R_f/R_1) - (R_f/R_2) \quad V_1$$
$$5 = 3.5 \quad (R_f/R_1) - (R_f/R_2) \quad V_1$$

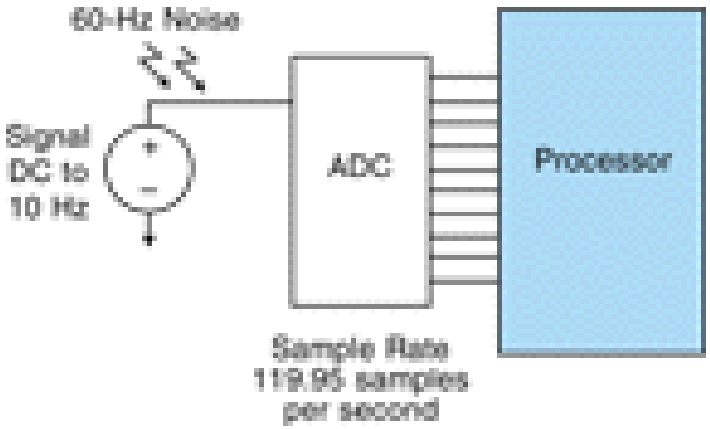
By choosing V_1 = - 12V and $R_f = R_1 = R_0$ = 15KΩ, R_2 is solved to be 120KΩ.

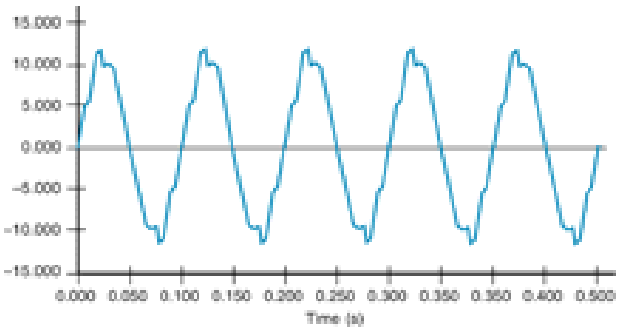
Aliasing in an ADC system

- Aliasing most often appears as a very low-frequency roll in the data, almost indistinguishable from a DC drift.
- The aliasing is caused by noise whose frequency is greater than the Nyquist frequency.
- The noise is undersampled, and the difference between the noise and the Nyquist frequency appears in the data.
- The amplitude of the false signal depends on the amplitude of the noise.

The Nyquist Rule

- The minimum sample rate required to describe a signal is at least two times the frequency of interest.
- This rule only works when the signal to be sampled has no frequency component greater than half the sample rate.
- In the real world, everything has noise. So, every frequency above half the sample rate is aliased or folded back and appears as low-frequency components of the signal. The closer in frequency the noise is to the Nyquist frequency, the lower the frequency of the contamination of the data.



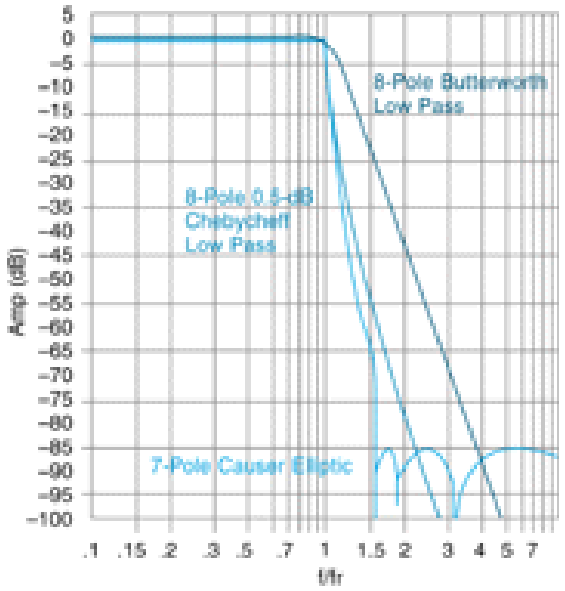


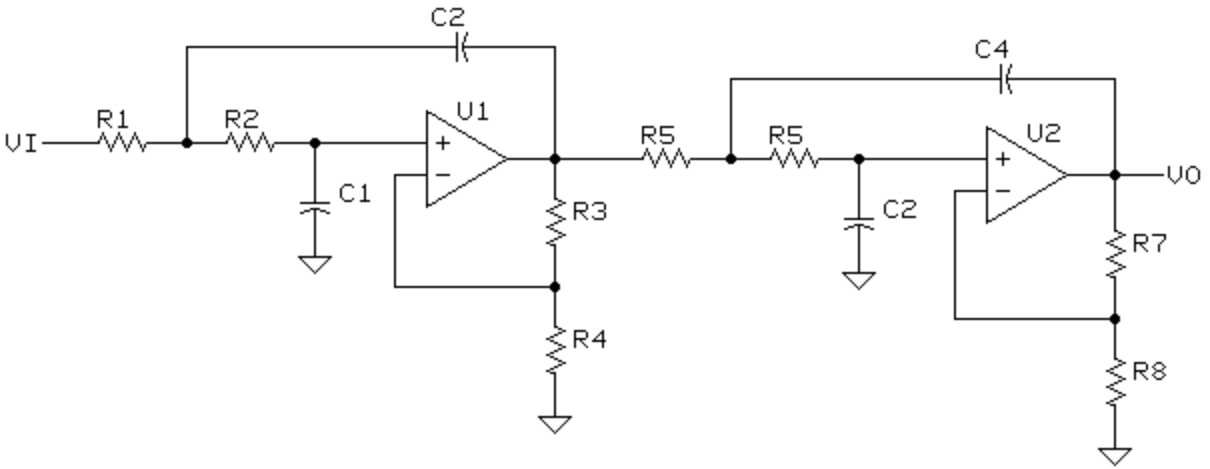
As the sample rate, let's choose 119.95 samples per second. What will happen?

• The Nyquist frequency of the system is 59.975 Hz. The 60-Hz noise is undersampled and folds back.

• The beat frequency between the sample rate and the noise (i.e., 60.000 – 59.975 = 0.025 Hz) appears in the data. Because 0.025 Hz is in the frequency range you're transducing, you aren't able to distinguish between the aliased signal and a real signal

Generally, an ADC system has some type of amplifier sitting between the signal source and the ADC, and this amplifier can sometimes serve as a filter and there are a lot of implementations of active filters using op-amps.





Typical low-pass filter transfer function

An Overview of the 68HC12 A/D Converter

- The successive-approximation method is used to implement the A/D converter.

- After the A/D converter has been enabled, the A/D conversion will be started when the ATDCTL5 register is written into.

-A clock signal is required to control the A/D conversion that must have a frequency between 2 MHz and 500KHz.

-Reference voltages are required for the conversion: one is high reference (V_{RH}) voltage (between VDDA/2 and VDDA),the other is low reference (V_{RL}) voltage (between VSSA and VDDA/2).

- Accuracy is only guaranteed for V_{RL} = 0 V and V_{RH} = 5 V.

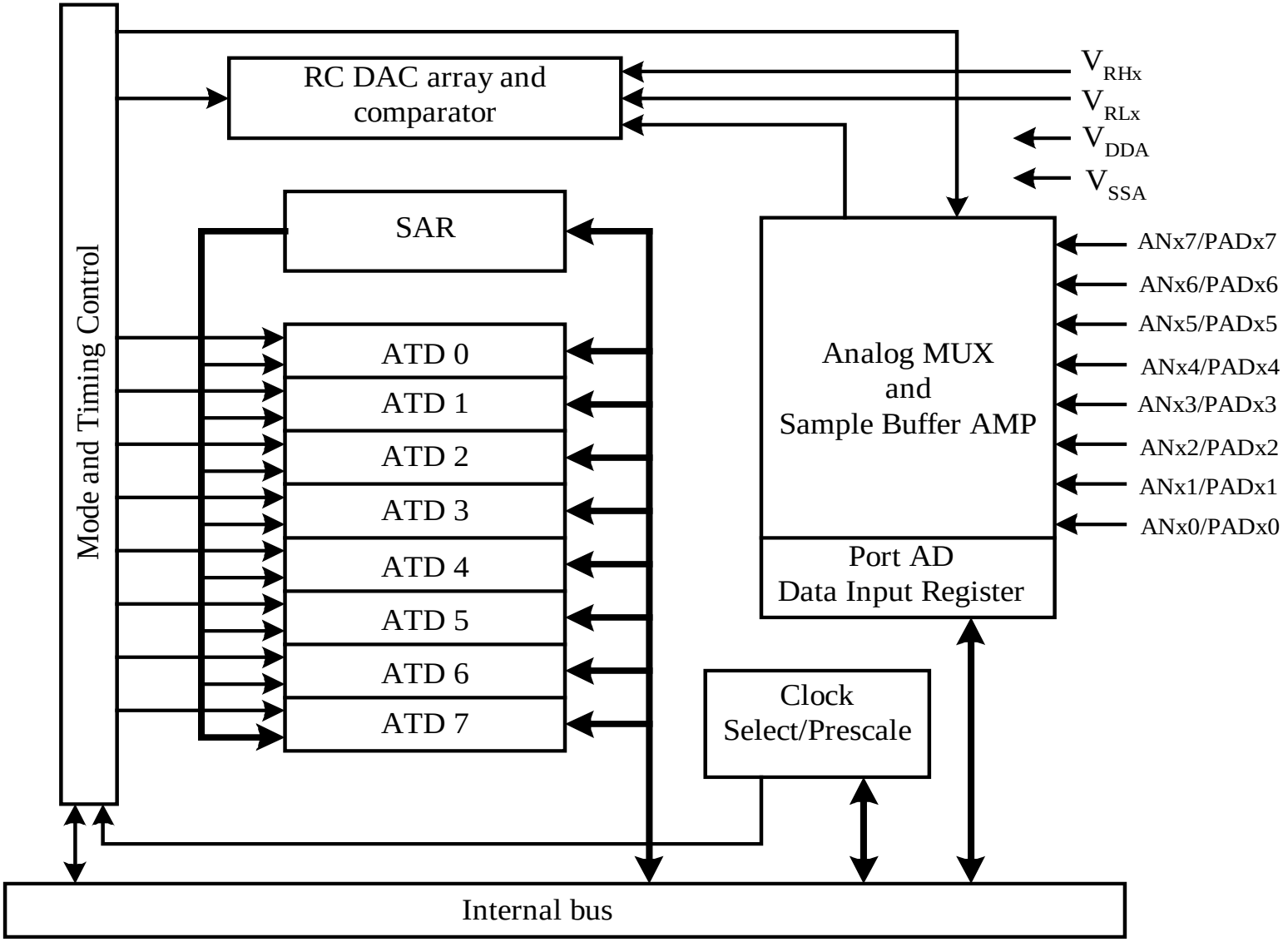


Figure 10.6 68HC12 ATD block diagram

ATD Registers

-Each of the two 68HC12 A/D converters has six control registers for configuring the A/D parameters.

-Among these six registers, the first two are not implemented.

-For 812A4 and B family members, the mnemonics of these six control registers are ATDCTL0, …, ATDCTL5.

- For those members that have two A/D converters, the control registers associated with ATD0 are ATD0CTL0, …, ATD0CTL5 whereas those associated with ATD1 are ATD1CTL0, …, ATD1CTL5.

ATD Control Register 2 (ATDCTL2, ATD0CTL2, and ATD1CTL2)

	7	6	5	4	3	2	1	0
	ADPU	AFFC	AWAI	0	0	0	ASCIE	ASCIF
value after reset:	0	0	0	0	0	0	0	0

ADPU: ATD disable bit
- 0 = Disables ATD and abort any conversion in progress
- 1 = Enables ATD

AFFC: ATD fast flag clear bit
- 0 = ATD flag is cleared normally, i.e., read the status register before reading the result register.
- 1 = Changes all ATD conversion complete flags to a fast clear sequence. Any access to a result register will cause the associated CCF flag to clear automatically if it is set at the time.

AWAI: ATD stop in wait mode bit
- 0 = ATD continues to run when the 68HC12 is in wait mode.
- 1 = ATD stops to save power when in wait mode.

ASCIE: ATD sequence complete interrupt enable bit
- 0 = disables ATD interrupt
- 1 = enables ATD interrupt on sequence complete

ASCIF: ATD sequence complete interrupt flag
- 0 = no ATD interrupt occurred
- 1 = ATD sequence complete

Figure 10.7 ATD control register 2

Control Register 3 (ATDCTL3, ATD0CTL3, and ATD1CTL3)

- This register is used to select freeze control.

	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	FRZ1	FRZ0
value after reset:	0	0	0	0	0	0	0	0

FRZ1 and FRZ0: background debug (freeze) enable bit
- 00: contiue conversions in active background mode
- 01: reserved
- 10: finish current conversion, then freeze
- 11: freeze when background mode is active

Figure 10.8 ATD control register 3

ATD Control Register 4 (ATDCTL4, ATD0CTL4, and ATD1CTL4)

- The ATD control register 4 selects the A/D resolution, sample times, and sets up the prescaler.

	7	6	5	4	3	2	1	0
	S10BM	SMP1	SMP0	PRS4	PRS3	PRS2	PRS1	PRS0
value after reset:	0	0	0	0	0	0	0	0

S10BM: ATD 10-bit mode control bit
0 = 8-bit operation
1 = 10-bit operation

SMP1 and SMP0: select sample time bits
These bits are used to select one of four sample times after the buffered sample and transfer has occurred. See Table 10.1.

PRS4--PRS0: select divide-by factor for ATD P-clock prescaler bits
The P clock is divided by this value plus one, and then fed into a divide-by-two circuit to generate the ATD module signal. Table 10.2 shows the divide-by operation and the appropriate range of system clock frequencies. The ATD conversion frequency must be between 500KHz and 2 MHz.

Figure 10.9 ATD control register 4

Table 10.1 Final sample time selection

SMP1	SMP0	Final sample time	Total 8-bit conversion time	Total 10-bit conversion time
0	0	2 ATD clock periods	18 ATD clock periods	20 ATD clock periods
0	1	4 ATD clock periods	20 ATD clock periods	22 ATD clock periods
1	0	8 ATD clock periods	24 ATD clock periods	26 ATD clock periods
1	1	16 ATD clock periods	32 ATD clock periods	34 ATD clock periods

Table 10.2 Clock prescale values

Prescale value	Total Divisor	Max P clock	Min P clock
00000	2	4 MHz	1 MHz
00001	4	8 MHz	2 MHz
00010	6	8 MHz	3 MHz
00011	8	8 MHz	4 MHz
00100	10	8 MHz	5 MHz
00101	12	8 MHz	6 MHz
00110	14	8 MHz	7 MHz
00111	16	8 MHz	8 MHz
01xxx 1xxxx	Do not use		

1. Maximum conversion frequency is 2 MHz. Maximum P clock divisor value becomes maximum conversion rate that can be used on this ATD module.
2. Minimum conversion frequency is 500 KHz. Minimum P clock divisor value becomes minimum conversion rate that this ATD can perform.

ATD Control Register 5 (ATDCTL5, ATD0CTL5, and ATD1CTL5)

-The ATD control register 5 is used to select the conversion modes, the conversion channel (s), and initiate conversions.

-A write to ATDCTL5 starts a new conversion sequence.

-If a conversion is in progress when a write occurs, that sequence is aborted and the SCF and CCF bits are reset.

- The A/D converter allows you to select a single channel, a group of four channels, or a group of eight channels to perform the conversion.

	7	6	5	4	3	2	1	0
	0	S8CM	SCAN	MULT	CD	CC	CB	CA
value after reset:	0	0	0	0	0	0	0	0

S8CM: select 8-channel mode bit
0 = conversion sequence consists of four conversions
1 = conversion sequence consists of eight conversions
SCAN: enable continuous channel scan bit
0 = single conversion sequence
1 = continuous conversion sequences (scan mode)
MULT: enable multichannel conversion bit
0 = ATD sequencer runs all four or eight conversions on a single input channel selected via the CD, CC, CB, and CA bits
1 = ATD sequencer runs each of the four or eight conversions on sequential channels in a specific group. Refer to Table 10.3.
CD, CC, CB, and CA: channel select for conversion bits
The channel selection is shown in Table 10.3.

Figure 10.10 ATD control register 5

When MULT=0, all four bits (CD, CC, CB, and CA) must be specified and a conversion sequence consists of four or eight consecutive conversions of the single specified channel.
When MULT=1, ATD sequencer runs each of the four or eight conversions on sequential channels in a specified group. Refer to Table 10.3.

Table 10.3 Multichannel mode result register assignment

S8CM	CD	CC	CB	CA	Channel Signal	Result in ADRx if MULT = 1
0	0	0	0	0	AN0	ADR0
			0	1	AN1	ADR1
			1	0	AN2	ADR2
			1	1	AN3	ADR3
0	0	1	0	0	AN4	ADR0
			0	1	AN5	ADR1
			1	0	AN6	ADR2
			1	1	AN7	ADR3
0	1	0	0	0	Reserved	ADR0
			0	1	Reserved	ADR1
			1	0	Reserved	ADR2
			1	1	Reserved	ADR3
0	1	1	0	0	VRH	ADR0
			0	1	VRL	ADR1
			1	0	$(V_{RH} + V_{RL})/2$	ADR2
			1	1	Test/reserved	ADR3
1	0	0	0	0	AN0	ADR0
			0	1	AN1	ADR1
			1	0	AN2	ADR2
			1	1	AN3	ADR3
		1	0	0	AN4	ADR4
			0	1	AN5	ADR5
			1	0	AN6	ADR6
			1	1	AN7	ADR7
1	1	0	0	0	Reserved	ADR0
			0	1	Reserved	ADR1
			1	0	Reserved	ADR2
			1	1	Reserved	ADR3
		1	0	0	VRH	ADR4
			0	1	VRL	ADR5
			1	0	$(V_{RH} + V_{RL})/2$	ADR6

ATD Status Registers (ATDSTAT0, ATDSTAT1, ATD0STAT0, ATD0STAT1, ATD1STAT0, and ATDSTAT1**)**

- Each A/D channel has two status registers for indicating whether a sequence of conversion is complete.

	7	6	5	4	3	2	1	0
	SCF	0	0	0	0	CC2	CC1	CC0
value after reset:	0	0	0	0	0	0	0	0

(a) ATD status register 0 (ATDSTAT0, ATD0STAT0, and ATD1STAT0

	7	6	5	4	3	2	1	0
	CCF7	CCF6	CCF5	CCF4	CCF3	CCF2	CCF1	CCF0
value after reset:	0	0	0	0	0	0	0	0

(b) ATD status register 1 (ATDSTAT1, ATD0STAT1, and ATD1STAT1)

SCF: sequence complete flag

This bit is set at the end of the conversion sequence when in the single conversion sequence mode and is set at the end of the first conversion sequence when in the continuous conversion mode. When AFFC = 0, SCF is cleared when the ATD control register 5 is being written to initiate a new conversion sequence. When AFFC = 1, SCF is cleared after the first result register is read.

CC2--CC0: conversion counter bits for current 4 or 8 conversions

This 3-bit value reflects the contents of the conversion counter pointer in a four or eight count sequence. This value also reflects which result register is written next, indicating which channel is currently being converted.

CCF7--CCF0: conversion complete flag

Each CCF bit is associated with an individual ATD result register. For each register, this bit is set at the end of conversion for the associated ATD channel and remains set until that ATD result register is read.

Figure 10.11 ATD status register

ATD Test Registers (ATDTESTH/L, ATD0TESTH/L, and ATD1TESTH/L)

- Each A/D converter has a test register to facilitate the factory testing.

Port AD Data Input Register (PORTAD, PORTAD0/PORTAD1)

- When the A/D converter is not enabled, the associated port can be used as an input port.
- The values of these pins can be read from the port AD data register.

Result Registers (ADR0H/L..ADR7H/L, ADR00H/L..ADR07H/L, ADR10H/L..ADR17H/L**)**

- Each ATD channel has eight result registers to hold the conversion result.
- Each result register is 16-bit and consists of a high and a low result register.
- All except the 812A4 can choose to use either 8-bit or 10-bit conversion mode.
- When 8-bit mode is chosen, the conversion result is stored in the high result register.
- In 10-bit mode, the high result register holds the upper eight bits of the conversion whereas the highest two bits (bit 7 and 6) of the low result register hold the lowest two bits of the conversion result.

The Procedure for Using the A/D Converter

Step 1
Connect the hardware properly. The A/D-related pins must be connected as follows:
V_{DDA}: connect to 5 V.
V_{SSA}: connect to 0 V.
V_{RH} (or V_{RHx}, x = 0 or 1): 5 V
V_{RL} (or V_{RLX}, x = 0 or 1): 0 V

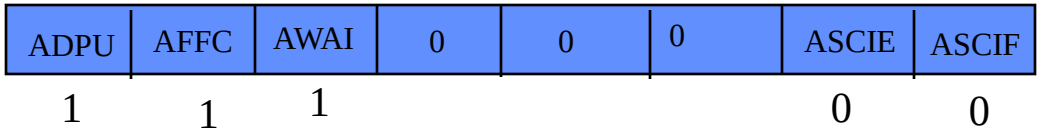
If the transducer output is not in the appropriate range, then we should use a signal conditioning circuit to shift and scale it to between V_{RL} and V_{RH}.
Step 2
Configure ATD control registers 2 to 4 properly and wait for the ATD to stabilize (need to wait for 5 μs).
Step 3
Select the appropriate channel (s) and operation modes by programming the ATD control register 5. Writing into the ATD control register 5 also starts the A/D conversion.
Step 4
Wait until the SCF flag of the status register ATDSTAT0 (ATD0STAT0 or ATD1STAT0 for members having two ATD modules) is set, then collect the A/D conversion results and store them in memory.

Example 10.4 Write a subroutine to initialize the ATD converter for the 912BC32 and start the conversion with the following parameters:

- Nonscan mode
- Select channel 7
- enable ATD fast flag clear
- Stop ATD in wait mode
- Disable interrupt
- Finish current conversion then freeze when BDM becomes active
- 10-bit operation and 2 A/D clock periods of sample time
- Choose 2 MHz as the conversion frequency for 8 MHz E clock

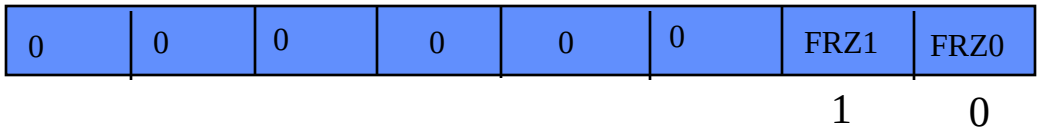
Solution: The setting of ATD control register 2 to 5 are as follows:

- Enable ATD (set bit 7 to 1)
- Enable fast flag clear (set bit 6 to 1)
- Stop ATD when in wait mode (set bit 5 to 1)
- Disable ATD interrupt (set bit 1 to 0)
- Clear all other bits
- Write the value $E0 to the ATDCTL2



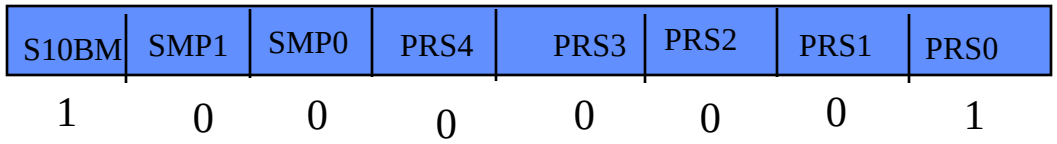
ATDCTL3

- Complete the current instruction then freeze when BDM becomes active
- Write the value $02 to this register



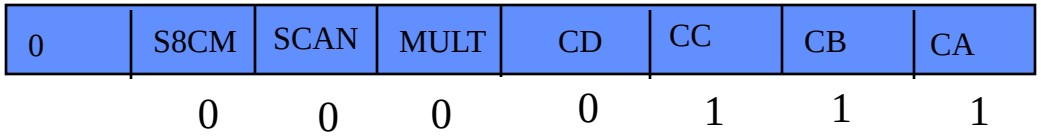
ATDCTL4

- Select 10-bit operation (set bit 7 to 1)
- Set sample time to two ATD clock period (set bits 6 & 5 to 00)
- Set clock prescale factor to 4 to select 2 MHz as the ATD clock frequency. Set bits 4 to 0 to 00001.
- Write the value $81 to this register.



ATDCTL5

- Select four conversions as the conversion sequence length (set bit 6 to 0)
- Select nonscan mode (set bit 5 to 0)
- Select single channel mode (set bit 4 to 0)
- Select channel 7 (set bits 3..0 to 0111)
- Write the value $07 to this register



The following subroutine will perform the desired initialization:

```
#include   "d:\miniide\hc12.inc"
ATD_init   movb    #$E0,ATDCTL2
           ldaa    #10
; the next two instructions create 5 us delay
wait       deca                    ; 1 E clock cycle execution time
           bne     wait            ; 3/1 E clock cycles execution time
           movb    #$02,ATDCTL3
           movb    #$81,ATDCTL4
           rts
```

Notes: this routine does not write into the ATDCTL5 register because that will start the conversion. We should write into the ATDCTL5 register only when we want to perform the conversion

Don’t get up

- Lecture 12 now

Homework #9

- See course website: http://390.revan.us
  - click homework tab
- Please submit a hard copy