Analog-to-digital Converter in the AT91M55800A

Introduction

Analog-to-digital converters translate analog measurements, characteristic of most phenomena in the real world, to digital format, used in information processing, computing, data transmission, and control systems.

Terminology

Table 1. Terminology

Category	Code	Description
Related to Numeric Base	0b	Binary Notation
	0x	Hexadecimal Notation
Related to Application		
	LSB	Least Significant Bit
	MSB	Most Significant Bit
	FS	Full Scale
	A/D	Analog-to-Digital
	D/A	Digital-to-Analog
	S/H	Sampling and Hold
	DNL	Differential Non Linearity
	INL	Integral Non Linearity
	SWRST	Software Reset
	EOC	End Of Conversion
	TIOA	Timer Input/Output A
	PIO	Parallel Input/Output
	PIO_PDR	PIO Disable Register

AT91 ARM® Thumb® Microcontrollers

Application Note

6039A–ATARM–29-Jan-04

General Information

Digital Quantities

The best-known code is natural binary (base 2). Binary codes are most familiar in presenting integers; i.e., in a natural binary integer code having n bits, the LSB (Least Significant Bit) has a weight of 2^0 (i.e., 1), the next bit has a weight of 2^1 (i.e., 2), and so on up to the MSB (Most Significant Bit), which has a weight of 2^{n-1} (i.e., $2^n/2$). The integer value of the binary number is obtained by adding up the weights of all non-zero bits.

In converter technology, because full scale (i.e., the converter's reference) is independent of the number of bits of resolution, a more useful coding is fractional binary, which is always normalized to full scale. Integer binary can be interpreted as fractional binary if all integer values are divided by 2^n. For example, the MSB has weight of 1/2 (i.e., $2^{(n-1)}/2^n = 2^{-1}$), the next bit has a weight of 1/4 (i.e., $2^{(n-2)}/2^n = 2^{-2}$), and so forth down to the LSB, which has a weight of $1/2^n$ (i.e., 2^{-n}). When the weighted bits are added up, they form a number with any of 2^n values, from $(1 - 2^{-n})$ of the full-scale. When all bits are "1" in natural binary, the fractional number value is $1-2^{-n}$, or normalized full-scale less 1 LSB (1- 1/16 = 1/15 in the 4-bit converter for example). The normalized numerical value of 10111001, an 8-bit code, is:

$$10111001 = 2^7 + 2^5 + 2^4 + 2^3 + 2^0 = 128 + 32 + 16 + 8 + 1 = 185$$

in integer binary and

$$\frac{2^7}{2^8} + \frac{2^5}{2^8} + \frac{2^4}{2^8} + \frac{2^3}{2^8} + \frac{2^0}{2^8} = \frac{128}{256} + \frac{32}{256} + \frac{16}{256} + \frac{8}{256} + \frac{1}{256} = \frac{185}{256}$$

$$2^{-1} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-8} = \frac{1}{2} + \frac{1}{8} + \frac{1}{16} + \frac{1}{32} + \frac{1}{256} = \frac{185}{256}$$

$$0.5 + 0.125 + 0.0625 + 0.0313 + 0.0039 = 0.7227$$

in fractional binary.

Bit numbering for microprocessor buses is based on whole numbers, not binary fractions. In such systems, the LSB is always Bit 0 (2^0), the MSB is always Bit n-1 (2^{n-1}). Setting up a correspondence between the bit numbers used in words in the two systems, we obtain

Bit(n - 1): (2^{n-1}), Bit(n - 2): (2^{n-2}), …. Bit(n - n) (i.e., 0): ($2^0 = 1$) in integer binary and

Bit 1: (2^{-1}), Bit 2: (2^{-2}) …..Bit n: (2^{-n}) in fractional binary.

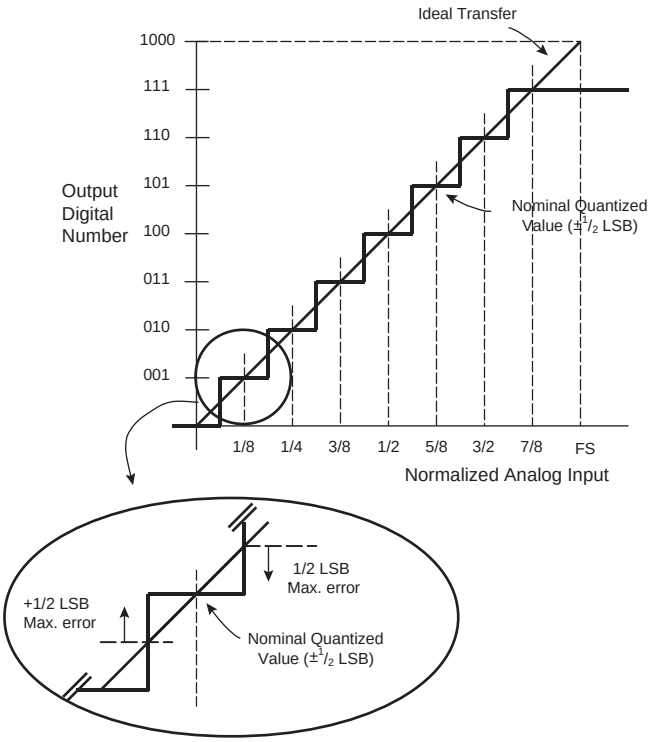
Basic Conversion Relationships

A graph is often the most effective way of indicating in detail the relationship between analog and digital quantities in a conversion.

Figure 1 shows the graph for an ideal 3-bit converter. Since all values of the analog input are presumed to exist, they must be quantized by partitioning the continuum into 8 discrete ranges.

There is, therefore, in the A/D conversion process, an inherent quantization uncertainty of ± ½ LSB (see Figure 1), in addition to the conversion errors.

Figure 1. Ideal A/D Conversion



An n-bit converter is able to discriminate 2^n different states. We say that the resolution is equal to n bits. The relative accuracy is given by:

$$r = 2^{-n}$$

For example, the resolution of a 3-bit analog-to-digital converter is equal to:

$$r = \frac{1}{8} = 12.5\%$$

An absolute resolution corresponds to this relative resolution depending on the full scale. For example, for a 3-bit analog-to-digital converter with a full scale equal to 2.5V, the absolute resolution is:

$$r = \frac{2.5}{8} = 0.3125V$$

Therefore, this resolution defines the smallest voltage input value that the ADC is able to discriminate: this is the quantum of conversion.

As a result, in theory, the error induced by the conversion performed according to the analog value is ±1/2 LSB or $\pm 1/2^{n+1}\%$.

A/D Conversion Error Calculation

Differential Nonlinearity

Differential nonlinearity (DNL) in an ADC is defined as the deviation in code width from the value of 1 LSB (i.e., $V_{FS}/2^n$).

Figure 2 illustrates an error equal to -0.8 LSB between the digital code 0b010 and 0b011: the gap between these codes is 0.2 LSB instead of 1 LSB, the DNL error is 0.8 LSB in this case. In this example, the DNL error is inferior to ±1 LSB and there is no missing code.

Figure 2. DNL Error Inferior to ± 1 LSB

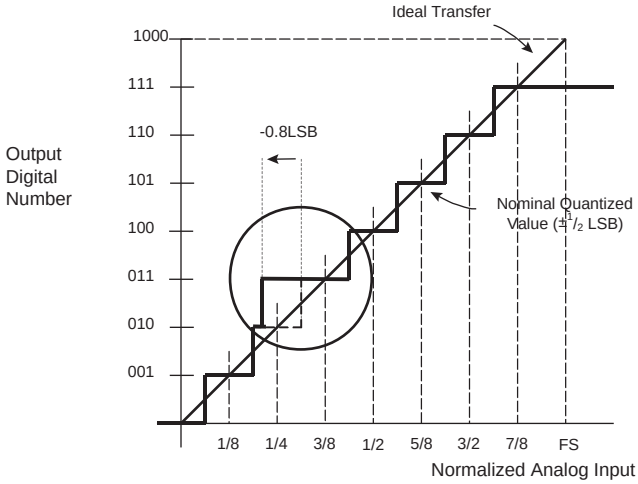
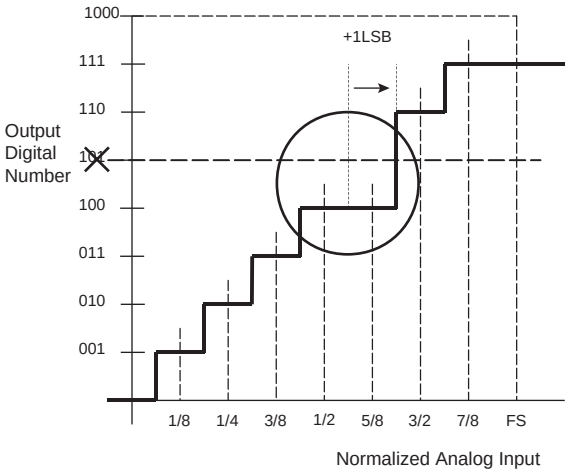


Figure 3 illustrates a DNL error between the code 0b100 and 0b110. The DNL error is equal to the real gap value between the code 0b100 and the next minus the 1 LSB theoretically expected: 2 LSB - 1 LSB = +1 LSB of DNL error in this case.

Figure 3. DNL Error Greater Than 1 LSB



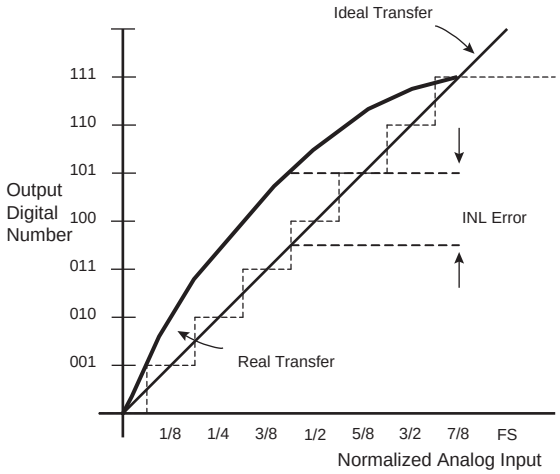
In this last example, the DNL error is greater than 1 LSB, the code width vanishes entirely and the ADC has at least one missing code.

Integral Nonlinearity

Integral nonlinearity (INL) is the DNL error integration. The INL parameter indicates the gap value from the real converter result and the ideal transfer function. For example, in a 10-bit system, a ±2 LSB INL error advises the user a maximum nonlinearity equal to:

$$\frac{2}{1024} = 0.00195 \text{ or } 0.2\%$$

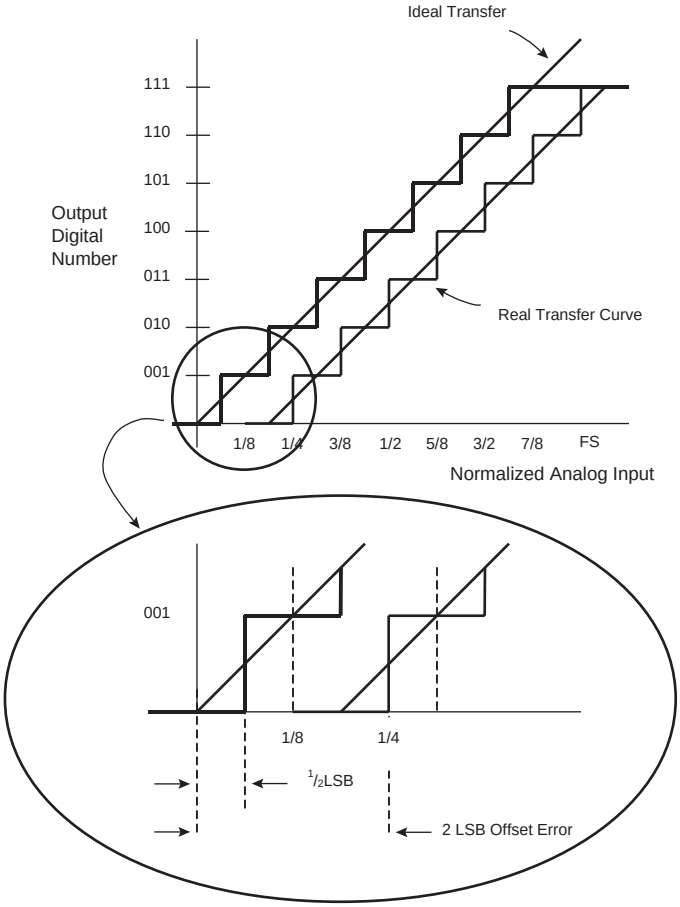
Figure 4. Integral Nonlinearity Example



Offset Error

An offset error is defined as a common deviation from the ideal transition voltages. It is usually tested by finding the error of the first (LSB) transition, because it is likely that the offset error is the only significant error present in that transition.

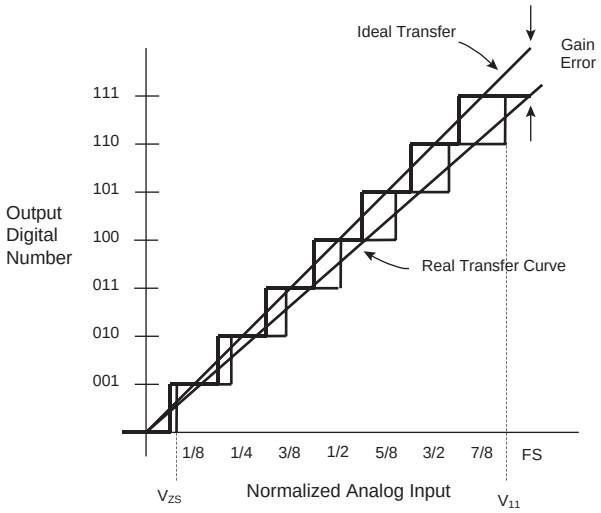
Figure 5. Offset Error Example



Gain Error

Gain error appears as a change in slope of the transfer function. Thus, gain error is the same as full-scale error, except that the offset error is subtracted. Gain error affects each code in an equal ratio.

Figure 6. Gain Error Example



To determine the gain error of an ADC, measure the final transition voltage and subtract the first transition voltage. Since this interval is ideally equal to (V_{FS} - 2 LSB), the deviation of the difference of the measured values from the value of (V_{FS} - 2 LSB) is the full-scale gain error. The gain error can be expressed in LSBs. The gain error of Figure 6 may be determined by the following equation:

$$\text{Gain Error} = (V_{FS} - 2\text{LSB}) - (V_{11} - V_{ZS})$$

where:

- V_{11} = final transition voltage
- V_{ZS} = first transition voltage
- V_{FS} - 2 LSB = full-scale range of the ADC minus 2 ideal LSBs.

Basic Characteristics of the Analog-to-digital Converter in the AT91M55800A

Successive Approximations

Successive-approximation A/D converters are widely used in the microcontroller domain. Conversion time is fixed and independent of the magnitude of the input voltage. Each conversion is unique and independent of the results of previous conversion because the internal logic is cleared at the start of a conversion.

The conversion technique consists of comparing the unknown input against a precise voltage generated by a digital-to-analog converter. The input of the D/A converter is the digital number at the A/D converter's output.

After the conversion command is applied and the converter has been cleared, the D/A converter's MSB output (½ full-scale) is compared with the input. If the input is greater than the MSB, it remains ON (i.e., 1 in the output register), and the next bit (1/4 full scale) is tried. If the input is less than the MSB, it is turned OFF (i.e., 0 in the output register), and the next bit is tried. If the second bit does not add enough weight to exceed the input, it is left ON (1), and the third bit is tried. The process continues in order of descending bit weight until the last bit has been tried. When the process is completed, the status line changes state to indicate that the contents of the output register now constitute a valid conversion. The content of the output register form a binary digital code corresponding to the input signal's magnitude.

In terms of conversion time, this method gives the final result in n clock cycles. For example, with a 10-bit ADC, the conversion time is equal to 10 clock cycles.

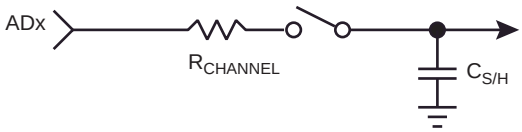
The embedded ADC in the AT91M55800A has an 8-bit conversion mode. In this mode, analog conversion is done over 10 bits and the 2 lowest significant bits are ignored. Thus, conversion time is not reduced, i.e., is equal to 10 cycles of the ADC clock.

In the AT91M55800A, the sample and hold circuitry is integrated into the ADC cell. As a result, the full conversion process requires 11 clock cycles.

Sample-Hold Circuit

A circuit for an analog input channel in the AT91M55800A is shown in Figure 7.

Figure 7. AT91M55800A Analog Input Channel



where:

- ADx is an ADC input on the AT91M55800A
- $C_{S/H}$ is the sum of the parasitic capacitances due to package, pin-to-pin, and pin-to-package base coupling and the sampling capacitor circuit
- $R_{CHANNEL}$ is the equivalent resistance through
  - the multiplexer circuitry
  - the Sample/Hold switch

When the switch is closed, the capacitor charges or discharges exponentially up to the input voltage. When the switch is opened, the charge or discharge remains on the capacitor's voltage. The charging time depends on the series resistance and the current available to charge the capacitor. Once the charge is acquired with the appropriate accuracy, the switch can be opened, even though the amplifier has not yet settled, without affecting the final output value.

A disadvantage of this circuit is that the switched capacitor dynamically loads the input source, which may not have low enough output impedance and sufficient current-drive capacity in the given delay defined by the sampling period.

The analysis of this constraint concerning the input equivalent impedance must take the requested accuracy and the sampling time into account over a full scale step of voltage at the input level.

By analyzing the circuit as a first-order R-C system, the maximum impedance of the analog circuit driving an ADC input is defined by the formula:

$$Z_{max} = \frac{2}{107 \times 10^{-12} \times \ln(2) \times n} \times \left(\frac{1}{2 \times F_{ADC}} - \frac{1}{F_{MCK}} \right) - 460$$

where:

- n is equal to 8 or 10 depending on the selected resolution
- F_{ADC} is the ADC clock frequency
- F_{MCK} is the Master Clock frequency of the microcontroller

Dynamic and Static Measurements

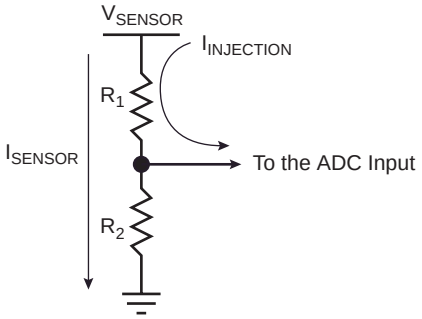
The formula above gives the maximum output impedance of the analog circuitry seen on the AT91M55800A's ADC input. Below this limit, the user is sure that the track voltage is correct, i.e., the charge transfer to the sampling capacitance has been fully completed.

In order to stay below this limit in the dynamic domain, the use of an operational amplifier is recommended for two reasons:

- it has low output impedance (impedance adaptation)
- it allows adaptation voltage range input
- it allows implementation of an anti-alias filter (in dynamic domain)

When the ADC is used in continuous domain as supply voltage, battery, temperature, etc., the application must resolve the discrepancy between the ADC full range input capability and the maximum level voltage of the information. The most efficient means for achieving this, taking into account ease of installation, cost and board space, is the use of a resistor bridge. See Figure 8.

Figure 8. Basic Resistor Bridge

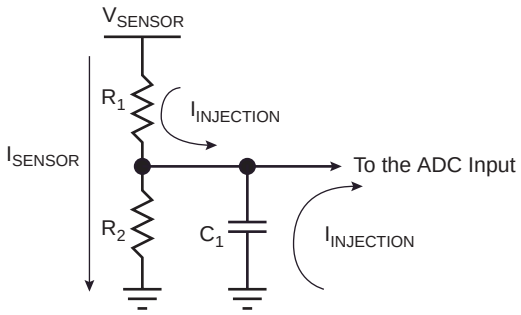


In order to decrease the continuous I_{SENSOR} leakage through R1 and R2, the user selects the higher bridge value without exceeding the previously given Z_{MAX} so that $I_{INJECTION}$ is sufficient.

Adding a capacitance (see Figure 9) reduces the impedance of the circuitry in the tracking period. The charge injection, during this period, is not only completed through the

bridge resistor but also through the capacitor. By choosing a higher capacitance value compared to the sampling capacitor, it can be seen as a perfect voltage generator.

Figure 9. Resistor Bridge with Added Capacitor



The value of the capacitor must be sufficient to transfer the necessary charges and show a voltage variation less than a quantum of conversion. Inversely, C_1 must not be overvalued: Out of the track period (between two consecutive measurement or a measurement sequence), the full load must be reach through the equivalent impedance equal to:

$$R_{equivalent} = \frac{R_1 R_2}{R_1 + R_2}$$

The minimum required value can be expressed by the formula:

$$C_1 = 2^n \times C_{S/H} \times V_{REF}$$

where:

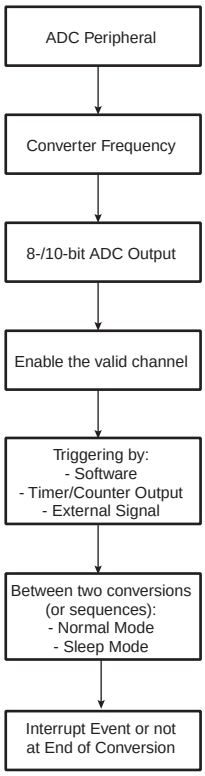
- *n* is equal to 8 or 10 depending on the selected resolution
- $C_{S/H}$ is the sampling capacitor
- V_{REF} is the reference voltage of the ADC

Working with the Analog-to-digital Converter in the AT91M55800A

Overview

Figure 10 summarizes the settings required for an A/D conversion in the AT91M55800A. These settings are made via the User Interface.

Figure 10. ADC Settings



Detailed Sampling Period

The ADC peripheral can provide 8- or 10-bit output data. Note that the 8-bit mode does not decrease conversion time; in both modes, 11 ADC clock cycles are required to perform a full conversion. The major difference between the two modes is that in 8-bit mode data handling is easier when the software needs to work in byte mode.

In 8-bit or 10-bit mode, an end of conversion (EOC) occurs for each channel and can generate an interrupt.

The start of each conversion, or one sequence of analog-to-digital conversion, can be:

- software-triggered via the bit START in the register ADC_CR
- hardware-triggered via an embedded Timer/Counter output
- hardware-triggered via a dedicated input pin, AD0TRIG or AD1TRIG, depending on the targeted ADC

The act of conversion can be for either:

- only one valid channel
- all valid channels

In all modes, the conversion can be divided into two steps:

- sampling period
- conversion period

During the sampling period, the time during which the ADC's input tracks the analog voltage is equal to one ADC clock cycle minus two master clock cycles. See Figure 11.

Figure 11. Sampling Period

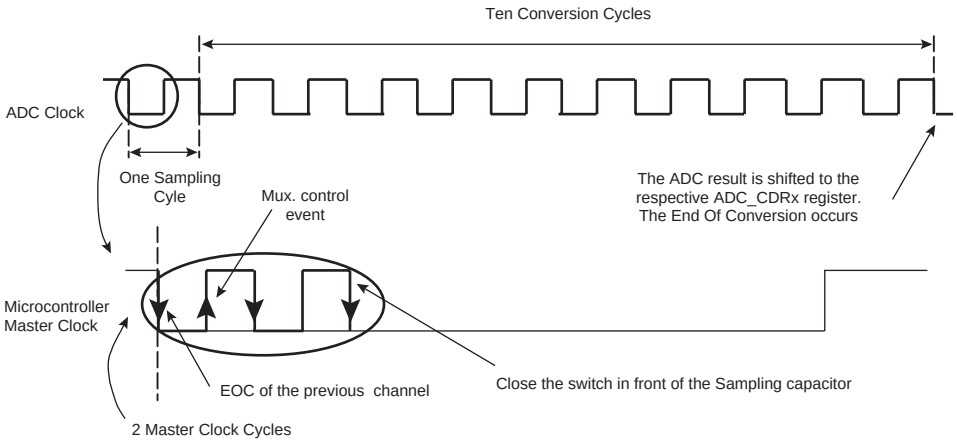


Figure 11 also shows that, in order to avoid a deterioration in sampling time due to both master clock cycles, it is important that the ADC frequency is lower than the master clock frequency. The calculation of input impedance takes this constraint into account.

Enabling the ADC in the Advanced Power Management Peripheral

After reset, in order to optimize power consumption, all peripheral clocks are stopped at the Advanced Power Management Controller level. As for other peripherals, before starting an analog-to-digital conversion, the user must enable the dedicated ADC clock in the APMC_PCER register (Peripheral Clock Enable Register).

Mode Register Settings

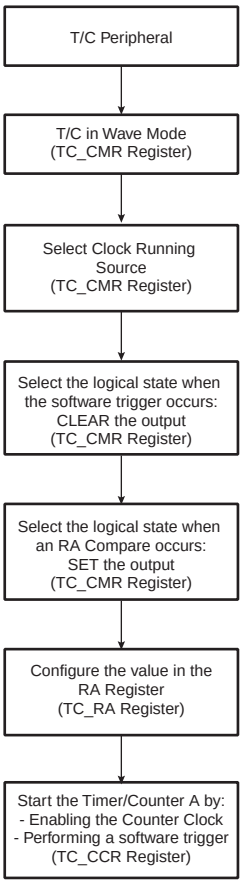
The ADC cell features a sleep mode to optimize power consumption. Sleep mode is enabled by setting the SLEEP bit in the Mode Register before starting the conversion. At the end of conversion, if sleep mode is enabled, the reference voltage used by the ADC is disconnected internally.

It is important to note that setting the Software Reset bit (SWRST) in ADC_CR clears all status registers and all fields in the Mode Register.

Trigger from a Timer/Counter Output

Figure 12 summarizes the configuration for the Timer/Counter when it is used to trigger an A/D conversion.

Figure 12. Timer/Counter Settings



When a Timer/Counter output is used to trigger the conversion, the multiplexed I/O controller must first be disabled on the targeted pin.

The Timer/Counter outputs able to drive the ADC trigger inputs are: TIOA0, TIOA1, TIOA2, TIOA3, TIOA4, TIOA5. The PIO multiplexed lines are on the Timer/Counter output are, respectively: PB20, PB23, PB26, PA1, PA4 and PA7. The PIO_PDR register is used to disable the targeted PIO line and, at the same time, enable the multiplexed peripheral.

Thus, at the PCB level, the user should not use a pin dedicated as ADC trigger signal.

The user must enable the peripheral clock at the APMC level via the APMC_PCER register.

Warning: Conversion on the First Valid Channel

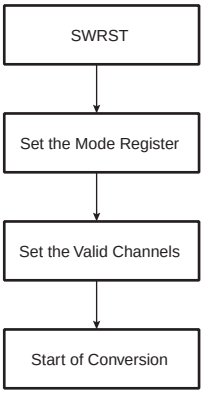
This Application Note gives the user the information required to calculate all the necessary external (analog characteristics) and internal (clock frequency and sampling time) parameters in order to use the ADC peripheral correctly.

However, the behavior of the first valid channel during a conversion - in particular, at sampling time - shows the uncertainty about the duration of this step.

Figure 11 shows that the ADC clock is generated by dividing the Master Clock by a counter. This clock divider works as soon as a valid clock has been enabled on its input and cannot be reset other than by a hardware reset or the Software Reset bit (SWRST). For this reason, after the start of conversion, independent of the threshold source or the mode used in the ADC, it is not possible to know the duration of the sampling time. As a result, the required input characteristics cannot be specified.

However, the use of one channel is possible by setting the SWRST bit in the Control Register and by starting the conversion immediately afterwards. In this case, the counter is reset and the sampling time can be anticipated. Figure 13 shows the steps necessary, and the fetch and execution time of those instructions reduce the sampling time. Therefore, it is important to evaluate this time correctly.

Figure 13. Programming Steps



In addition, the user must assure that these steps are completed without interruption.

Document Details

Title	ADC in the AT91M55800A
Literature Number	6039

Revision History

Version A	Publication Date: 29-Jan-04

Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

Regional Headquarters

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

Literature Requests
www.atmel.com/literature

6039A–ATARM–29-Jan-04