Leveraging ADC Technology to Capture, Compute and Communicate in Microcontroller-Based Embedded Systems

Introduction

System designers using microcontrollers (MCUs) continue to benefit from Moore's Law with ever-increasing, feature-rich functionality available in smaller footprints and at lower costs. Today's MCU suppliers are busy analyzing the marketplace for opportunities to improve or increase the MCU's integrated peripheral functions and ease of use while expanding connectivity to the Internet of Things. The embedded system designer and the MCU manufacturer are interested in at least three basic data acquisition system functions: capture, compute and communicate. While it is helpful to understand each of these functions, this white paper will focus primarily on the capture phase of a data acquisition system.

Capture

Sophisticated mixed-signal MCU devices must be able to *capture* some aspect of the analog world and convert a continuous-time signal into discrete digital form. The analog-to-digital converter (ADC) is the all-important microcontroller peripheral for this task, and as such the ADC's capabilities will often determine which MCU is appropriate for an application. The microcontroller can also capture system information already in digital form from an outside source via various serial or parallel digital I/O interfaces.

Compute

After signal acquisition, something must be done with the acquired data; sometimes the analog-to-digital conversion alone is all that is needed, but it is more likely that some *computation* must be done with the acquired data sample. There is an ongoing digital evolution within the microcontroller industry, giving system designers more sophisticated levels of signal processing and higher processor speeds. As a result, embedded developers now have access to a larger selection of 8-bit, 16-bit and 32-bit microcontroller ICs that fit various price/performance targets. Developers also have more on-chip options available to them to accomplish system tasks. For example, designers may simply choose to use program code to process the sampled data, or they may also take advantage of the many integrated improvements within MCU peripherals. In addition, MCU hard-macro peripherals are becoming more autonomous in nature as functional state machines are integrated into them to offload common tasks such as filtering from the processor.

Communicate

Finally, some form of *communication* is necessary to exchange information within a control process. This function may be as simple as a single port-bit toggle indicating that "the tank is full," or it can get very complicated such as triggering state-dependent operations via various serial, parallel or radio interfaces. The communication may even be presented as an analog output of voltage or current by converting the captured and processed data back into analog form using a digital-to-analog converter (DAC).

Versatile MCU-Based Data Acquisition Systems

The backbone of an MCU's data acquisition system is the ADC. Probably the most familiar type of ADC in the electronics world is the successive-approximation ADC (SARADC). Many MCUs use the SARADC because of its versatility in speed and performance. SARADCs with accuracies from 8 bits to 16 bits and throughput capabilities ranging from very slow on-demand conversion requests to more than 1 million conversions per second are available on MCUs. But the ADC is only one part of a complete data acquisition system. The other parts of the data acquisition system that contribute to its versatility include the signal input interface, the reference voltage interface, the clocking and sampling system for the ADC, and the data management capabilities for the converted ADC output data.

The analog input interface is usually shared among common input-output (I/O) pad structures, which may be configured for digital or analog use and sometimes for both domains. For example, Silicon Labs' Precision32™ SiM3U1xx, SiM3C1xx and SiM3L1xx 32-bit microcontroller devices provide the system designer with a large input multiplexer, supporting up to 31 inputs. The purpose of such a large input multiplexer is not only to provide the ability to monitor a large number of input sources but also to aid in the configurability of the embedded system design. Versatile configurability is accomplished by allowing the designer to partition analog and digital functions among various package pin locations according to their system requirements. As part of the input multiplexer, one of the most common auxiliary inputs is the on-chip temperature sensor; other important inputs include internal voltages such as supply voltages or ground signals.

Once the system's input channels are configured, the embedded designer may use program code to select any channel and request ADC conversions. The designer may also choose to offload program codes and allow the ADC channel sequencer (a feature available on Precision32 MCUs) to cycle through predetermined channel inputs until something of importance is detected.

Another important aspect of a data acquisition system is the ADC reference voltage (VREF). The reference voltage sets the input dynamic range or full scale of the system and can dramatically affect overall noise performance. A multiplexer is commonly used to select VREF from various on-board and external reference voltage inputs. Commonly used voltages to select from include multiples of a buffered bandgap voltage generated internally on the MCU, an externally generated precision voltage reference and various external supply voltages, which are compatible with the I/O pad structure and ADC limitations.

With so many available input channels, one can easily imagine that in some systems the input dynamic range of one or more of the inputs may not be compatible with a single VREF voltage. To address this concern, the Precision32 32-bit microcontroller devices integrate an input gain stage with a gain of 0.5x or 1x as part of the ADC in order to adjust the input signal to be more compatible with the VREF selection. In addition, Silicon Labs' 8051-based C8051F50x/F51x MCUs implement a selectable gain adjustment option that enables the developer to select gain values between 0 and 1.016. These gains are useful in attenuating an input signal's dynamic range to be closer to the ADC's full scale limit as defined by VREF.

ADC Conversion Cycle

The ADC's clocking system must also be very configurable in order to support the general-purpose nature of MCU applications. The SAR converter falls into the category of Nyquist-rate converters, which require the system designer to take care to meet the Nyquist criterion of sampling with a sample rate of at least twice the input signal's bandwidth. The developer also must consider two time keeping tasks when configuring the ADC. These tasks relate to the ADC conversion cycle and to the other available clock sources within the microcontroller system. The conversion cycle consists of two parts: a tracking period and a conversion period, as shown in Figure 1.

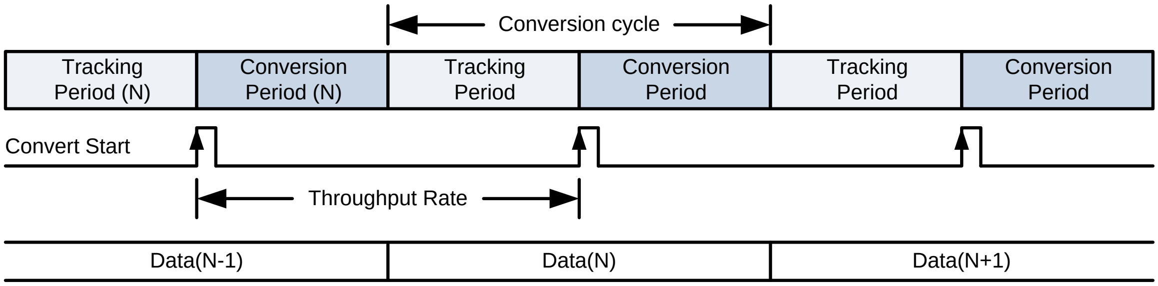

Figure 1. ADC Conversion Cycle

The tracking period is that part of the conversion cycle when the ADC input circuitry is connected to the input signal. The input sample occurs at the moment in time when the tracking period ends and the input circuitry is disconnected from the input source. This moment is caused by a digital control signal to the ADC called convert-start (CNVST). Aptly named, CNVST marks the end of the tracking period and the beginning of the conversion period.

The conversion period is that part of the conversion cycle when the ADC is called upon to execute the successive-approximation-register (SAR) algorithm. For example, the SAR successively resolves one bit at a time from the most-significant-bit (MSB), usually half-scale, to the least-significant-bit (LSB). Each SAR decision result is clocked into its respective position in the SAR-register (SARREG), ultimately forming the completed digital result. An N-bit SARADC will use N ADC clock cycles to resolve the N bit decisions in addition to a few more overhead clock cycles to synchronize to the MCU system clock.

The amount of time that the ADC must use to track an input signal is related to the input load characteristics of the ADC, the drive capabilities of the signal source and the degree of accuracy required of the measurement. The microcontroller device specification sheet will list the ADC input model, giving values for input capacitance, resistance and leakage current at the ADC input. For accurate measurements, the developer should allow enough track time for the input signal to settle to better than 0.5 LSB. Failure to track for a sufficient amount of time will result in measurement inaccuracies such as gross signal errors, crosstalk with previously sampled inputs and distortion.

Since the conversion period is usually a configurable amount of time related to the SARADC clock period, i.e. for each bit decision, it may be best to describe the track time as the amount of time between CNVST requests that is not used for the SAR conversion period. Simply put, if the ADC is not converting, it is tracking. A long time between convert requests will result in excessive track times. To address this issue, Silicon Labs’ MCU products offer the ability to power down the tracking circuitry between convert requests, thereby reducing system power.

The ADC conversion throughput rate is the frequency at which conversions are requested and is usually designated with the symbol Fs. The maximum throughput rate is set by the ADC timing constraints of minimum track time, in addition to the minimum convert time. A consistent throughput rate is accomplished by issuing a stream of convert start requests spaced equally in time. This is where a configurable ADC clock system is essential to manage two critical time keeping tasks.

One of the time keeping tasks is the generation of the clock used during the conversion period to execute the SAR algorithm. The SAR clock (SARCLK) associated with the conversion period is usually derived from the MCU system clock. The SARCLK configurability must accommodate MCU system clocks that range in frequency from less than 1 MHz to more than 100 MHz. Due to the ADC’s internal comparator design, there is a maximum rate at which the SAR conversion algorithm can be clocked. The system designer must configure the SARCLK frequency to avoid violating its maximum clock rate specification. The other time keeping task is to generate a convert request sample rate that does not violate the maximum throughput rate of the ADC converter given a properly configured conversion period.

Aperture Jitter and Delay

The convert start request signal should be thought of as the sample clock because it defines the moment in time when the ADC sample-and-hold circuitry actually captures the input signal. Two well-known specifications relating to sample-and-hold circuits must be considered when configuring the time-base for the ADC convert requests. These specifications–aperture jitter and aperture delay–affect the accuracy of the input sample for input signals that are changing rapidly relative to the aperture time delays, as shown in Figure 2.

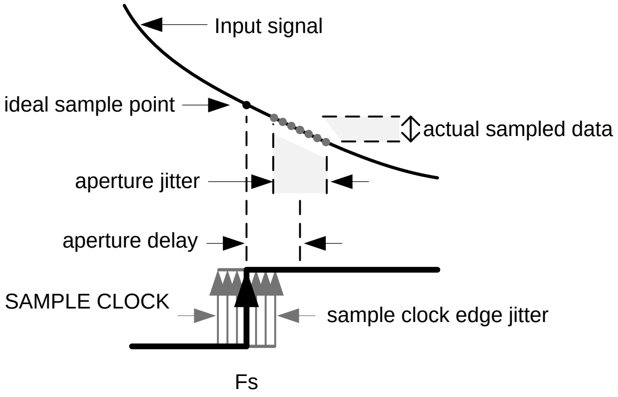

Figure 2. Aperture Jitter and Delay

The aperture jitter results from inaccuracies, i.e. clock jitter, in the clock system and other circuitry that generates the convert start signal, and aperture delay results from circuit delays between the convert start signal and the sample switch. Aperture jitter can introduce noise and distortion into a data acquisition system. Aperture delay is managed internally by the MCU designer and should be minimized to avoid the risk of adding even more jitter due to long delays. Aperture delay introduces a delay error into the data acquisition system. An aperture delay that is too long is analogous to a water tank that is overflowing before the “the tank is full” signal is issued.

As with many advanced MCUs, Silicon Labs’ microcontroller products allow conversions to be initiated by internal and external convert start trigger sources. For example, internal convert request sources include on-demand requests from program code or various timer peripheral overflow indications. External conversion request sources come in through the MCU I/O interface and are connected directly to the ADC sample circuit. Silicon Labs’ MCUs do not resynchronize the external convert signal prior to sampling because this would result in aperture jitter and thus cause measurement error.

For these reasons, an accurate time base is needed for the generation of stable convert start request timing. MCUs offer a range of on-board clock or external sources from which to choose the system clock. The system designer must be careful to choose a clock source with sufficient accuracy to meet the demands of their data acquisition system. For high-speed input sources, this might be the crystal oscillator, which is very accurate. On the other hand, direct current (dc) or very slow inputs are more immune to clock system errors but still do require sufficient settling time between conversions.

Burst Mode Features

Two particularly useful features available in Silicon Labs’ MCU products are burst mode and post-track mode. Burst mode produces an accumulated or averaged result from a programmable number of consecutive ADC conversions all triggered from one convert request. Post track mode changes the operation of the convert start request to offload the required track time management from the MCU system. Normally convert start marks the end of the tracking period and the beginning of the conversion period. But in post track mode, the convert start request instead triggers the start of a track period, which then proceeds for a programmable amount of time based upon the preconfigured SARADC clock period, and then the conversion begins. Burst mode with post tracking can be configured to achieve an accumulated ADC result within a single MCU clock cycle for MCUs running at low frequencies, thus saving system cycles and energy, as shown in Figure 3.

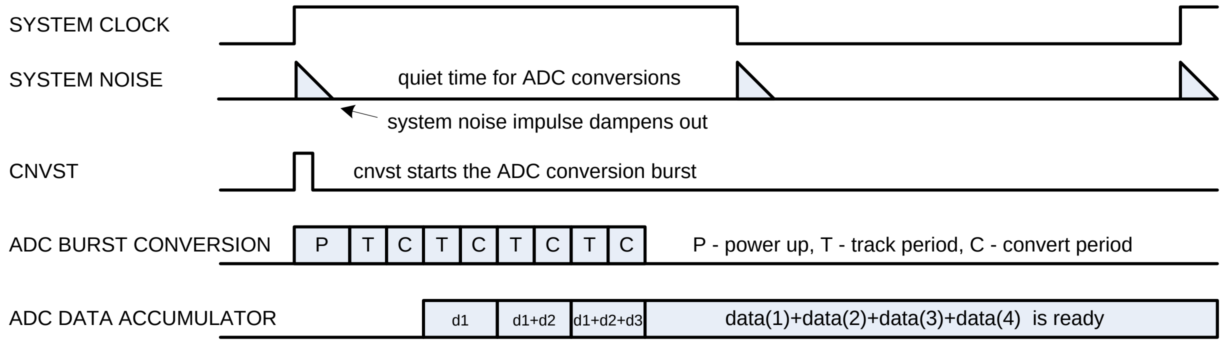

Figure 3. ADC Burst Mode, 4X Data Accumulations in One System Clock

ADC Data Windowing

Silicon Labs’ 8- and 32-bit microcontroller devices feature an ADC output data window comparator. The ADC output data is compared against programmable high and low limits and can automatically generate program interrupts for ADC output data inside, outside, above or below the set limits. Using the data window comparator, the designer can configure the ADC to autonomously check on the “tank is full” level sensor input until the data window comparator issues an interrupt to the MCU program. When this interrupt is issued, the MCU can break from whatever it was doing at the time and switch to the task of closely controlling the tank system.

Conclusion

Embedded system developers continue to benefit from increased silicon integration, enabling a plethora of configuration options that enhance data acquisition systems. MCU peripherals such as the ADC can be easily configured to serve a wide range of embedded applications that formerly required custom ASICs. Mixed-signal MCU peripherals increasingly operate more autonomously, allowing program code to perform other tasks until a peripheral interrupt is issued. For more details about Silicon Labs’ microcontroller devices, please visit www.silabs.com/mcu.

# # #

Silicon Labs invests in research and development to help our customers differentiate in the market with innovative low-power, small size, analog intensive mixed-signal solutions. Silicon Labs' extensive patent portfolio is a testament to our unique approach and world-class engineering team. Patent: www.silabs.com/patent-notice