Technical Article

Cycle Scavenging on C2000™ MCUs, Part 4: ADC Post-processing Blocks

Jitin George

The idea of “cycle scavenging” is built into the DNA of C2000 microcontrollers (MCUs), which enables them to minimize latency at every stage of real-time control without compromising performance.

In previous installments of this series, I looked at a number of features on C2000 MCUs, including zero-wait-state analog-to-digital converter (ADC) transfers, multiport ADC reads, start-of-conversion timing and configurable ADC interrupt delays that are tailored to scavenge cycles at the sensing stage of real-time closed-loop control systems. In this installment, I’ll take a closer look at how the ADC post-processing blocks shown in Figure 1 provide a unique way of scavenging cycles from the main processors that would otherwise be required in the middle of your most time-critical loops.

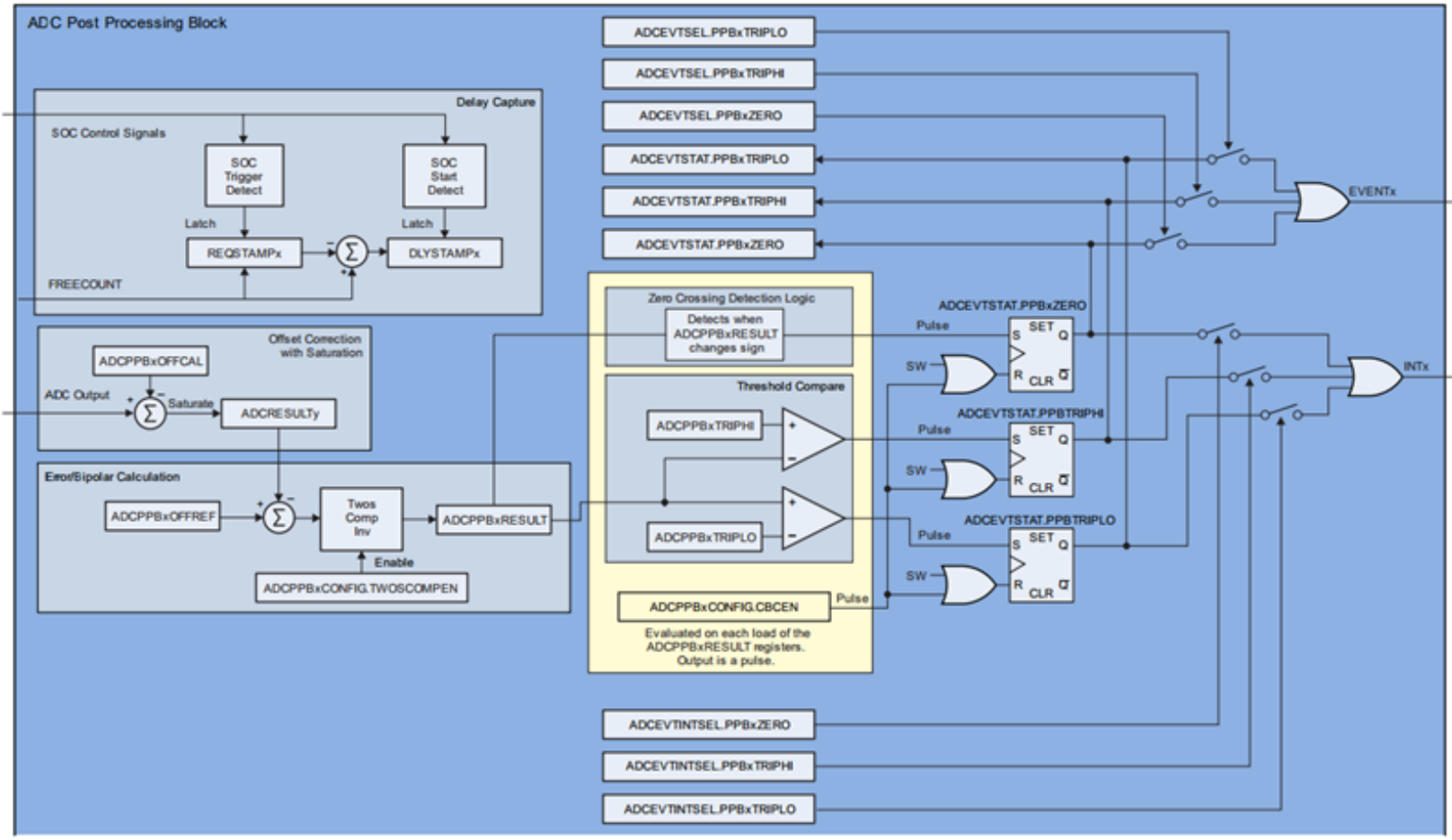

Figure 1. ADC Post-processing Block Diagram

Traditionally, the main processors of an MCU handled sample-processing routines such as offset correction, error calculation and threshold comparisons. This took place during the processing stage of closed-loop systems. With the addition of ADC post-processing blocks on new C2000 MCUs, these routines are now performed during the sensing stage, with zero software overhead.

Offset correction

In many applications, the use of external signals and signal sources produces an offset on the input ADC channels. In such situations, global trimming is not enough to compensate for these offsets because they may vary from channel to channel. The typical approach would be to remove these offsets in software at the processing stage of the control loop. The downside to such an approach is that it increases the burden on the central processing unit (CPU), burning more cycles. With ADC post-processing blocks, these offset corrections can occur in hardware, saving numerous cycles in the process.

Error computation

In almost all closed-loop control applications, an error from a desired set point must be computed from the digital output of the ADC conversion in order to actuate an appropriate system response. Most general-purpose MCUs perform these functions at the beginning of an interrupt service routine (ISR), burning several cycles doing so. The ADC post-processing blocks on C2000 MCUs can perform these error computations automatically in hardware, reducing not only software overhead but also sample-to-output latency.

Threshold detection

Post-processing blocks have the ability to generate enhanced pulse-width modulator (ePWM) trips based on an out-of-range ADC conversion without any CPU intervention. Being part of the ADC module, the post-processing blocks can automatically perform a check against a high and low ADC limit and generate a trip to the PWM and/or an interrupt based on the comparisons. This invariably lowers the sample-to-ePWM latency and significantly reduces software overhead compared to other general-purpose MCUs.

While most general-purpose MCUs rely on the main processors to handle the above-mentioned sample-processing routines, C2000 MCUs differentiate themselves by providing the ability to do the same in hardware with the ADC post-processing blocks. As you can see, the post-processing block is a very powerful feature that enables C2000 MCUs to realize significant cycle savings, thereby adding remarkable value from a real-time control perspective.

Moving on from sensing, in the next installment of this blog series I will address processing, discussing the trigonometric math unit and control law accelerator and their abilities to scavenge cycles at the processing stage of real-time control systems.

Additional Resources

- Download the “TMS320F2837xD Dual-Core Delfino™ Microcontrollers Technical Reference Manual” and “The TMS320F2837xD Architecture: Achieving a New Level of High Performance” technical brief.
- Download the “TMS320F28004x Piccolo™ Microcontrollers Technical Reference Manual.”
- Read the blog post, “Achieve unprecedented current-loop performance from an off-the-shelf MCU.”

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265