

Adding a C to the CLA

Chris Clearman, Motor Control Solutions Manager, Texas Instruments



All motor control designers want to deliver the best possible motor control solution quickly. Part of this depends on the gauged difficulty of design. If you've investigated TI microcontrollers for your motor control designs, you may have run across something called a Control Law Accelerator (CLA). The CLA is in our C2000 32-bit Piccolo microcontrollers. The integrated Control Law Accelerator (CLA) is a 32-bit floating-point math accelerator that effectively offloads complex, high-speed FULL control loops (or just processing / algorithms) from the main CPU core. This offloading frees the CPU to handle additional tasks such as communication, diagnostics or integrate more control loops on the same processor. The CLA more than doubles the computational bandwidth available on the controller. Similar to an independent core, the CLA has its own instruction set and memory space, allowing it to operate completely independent of the CPU and enables efficient software partitioning. The CLA also provides direct access to on-chip peripherals for parallel execution of algorithms to respond to peripheral interrupts without having to go through the CPU core.

A typical control loop has three main components; ADC feedback, CPU based compensation, and PWM feed-forward. Without the CLA, the PWM triggers the ADC, which fires an interrupt when it finishes the conversion. The interrupt is then scheduled by the Interrupt Manager, and when the CPU receives the interrupt request it has to switch contexts, handle the interrupt, and switch back to resume what it was doing.

With the CLA, time-consuming context switches are taken out of the equation and a parallel feedback, processing, and feedforward path is created! The CLA has direct communication with the ADC (and motor sensor feedback on some devices) and ePWM peripherals, eliminating the need for the CPU to even acknowledge the interrupt request. The CLA has its own interrupt controller that simply organizes incoming requests in a priority queue, reducing response latency to just 5 cycles. Additionally, once a task begins in the CLA, it is run to completion, meaning there are no nested interrupts. The CPU is free to handle other system processes, or handle other control loops in multi-loop systems.

Adding a C to the CLA

Published on Electronic Component News (<http://www.ecnmag.com>)

The addition of a CLA minimally affects the CPU and code organization. CPU responsibilities remain the same, with the addition of CLA initialization and CLA communication through shared message RAMs and interrupts. All C28x CPU code is still done in C. The CLA can only run its own control code, as well as access memory, interrupts, and the shared message RAMs. It can also access certain peripheral registers to read/write data, but it cannot configure the peripherals.

Adding a parallel floating-point unit such as the CLA sounds like there's potential for big headaches in development. However, Piccolo's CLA has been designed to support simple and user-friendly debugging. The CLA can halt, single-step and run independently from the main CPU, resulting in a clean and easy-to-use real-time debugging interface very similar to the main CPU. Additionally, both the CLA and the main CPU are debugged from the same JTAG port. Code Composer Studio™ ([CCStudio](#) [1]) Integrated Development Environment (IDE) includes a parallel debug manager window to allow programmers to independently monitor both the C28x core and the CLA.

But the big news is the recent announcement of a C-compiler for the CLA! Until now all of the code had to be written in CLA Assembly or based on pre-packaged TI provided assembly macros. While this works extremely well for small, dedicated, high frequency tasks in power conversion it has not provided enough flexibility for motor control or the "C artist" in all of us. :)

Stay tuned for further special C-based CLA libraries from TI engineers, and please share with the community what you are NOW able to do with this powerful parallel processor that we call the CLA.

For more information, please visit <http://www.ti.com/c2x-f2803-6x-pr-pf1> [2]

To view more informative blogs about motor control, visit www.ti.com/mcblog-ecm [3].

Source URL (retrieved on 09/23/2014 - 11:36pm):

<http://www.ecnmag.com/blogs/2012/04/adding-c-cla>

Links:

[1] <http://www.ti.com/tool/CCStudio>

[2] <http://www.ti.com/c2x-f2803-6x-pr-pf1>

[3] <http://www.ti.com/mcblog-ecm>