Technical Article Customizing On-chip Peripherals Defies Conventional Logic



Eric Best



It is a familiar scene in labs across the globe: A design engineer is pushing the envelope, seeking to enhance functionality or improve performance. Unfortunately, while digging deep into low-level system timing, a design stalemate occurs. The potential solution demands the alteration of key control signals. These situations can impede creativity, leaving engineers with one wish: "If only there was a way to customize logic and timing within a control peripheral." That wish is now a reality with the configurable logic block (CLB) for C2000[™] microcontrollers (MCUs).

What is the CLB? The CLB integrates an optimized piece of high-speed programmable logic into a real-time microcontroller like a C2000 MCU. It provides intelligent input/output (I/O) routing to key peripherals such as enhanced pulse-width modulators (PWMs), enhanced capture, enhanced quadrature encoder pulse and general-purpose I/O. The result is an intellectual property enhancement offering system-level differentiation.

As shown in Figure 1, the CLB tool comprises either two, four or eight tiles. Within each tile is an optimized group of programmable logic cell blocks that contains:

- Three 4-input look-up tables (LUT4).
- Three 32-bit counters (Ctr32).
- Three 4-state finite state machines (FSMs).
- Eight 3-input output look-up tables (L3).
- One high-level controller (HLC).

1



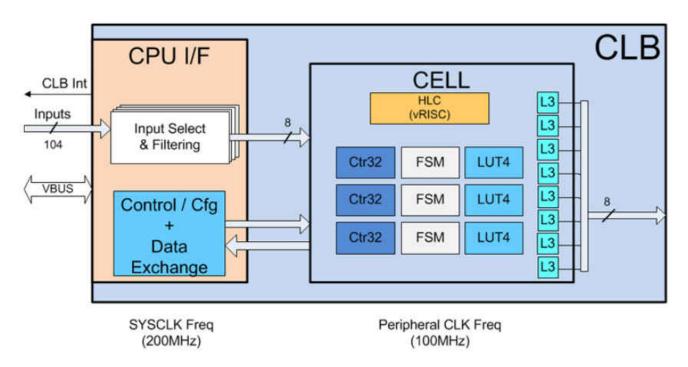


Figure 1. Example of CLB tool tile cell blocks

By individually configuring and wiring these cell blocks together, you can generate a custom logic scheme, providing novel solutions to difficult system problems like programming application-specific logic for trip zones, enhancing soft-switching PWM schemes or implementing condition-based quadrature encoder position capture.

To make development easy, TI created a graphical CLB configuration tool (shown in Figure 2) that is integrated directly into TI's Code Composer Studio[™] integrated development environment for code development and debugging.

>FSM_2(<u>hide</u>)		
eqn_out	e1 e0	
eqn_s0	e0 & e1	
eqn_s1	s0 s1	
e0	LUT_2.0UT	•
e1	COUNTER_2.count_zero	•
xe0	0	•
xel	0	•
> COUNTER_0(hide)		
reset	COUNTER_0.count_match1	•
event	0	*
mode0	1	
mode1	1	
match1_val	1000	
match2_val	100	
event_load_val	0	
What action should be taken on an event trigger?	None	*

Figure 2. CLB tool graphical configuration view

Once configured, the generation of the CLB tool output logic is automatic. The project build output is a *.h header file describing the CLB tool configuration. During run-time initialization, the header file descriptors configure the tiles using C2000Ware DriverLib application programming interfaces.

Design engineers are making use of the CLB tool and exploring new areas of product differentiation. Matt Parnell Sr., principal hardware engineer for lifestyle audio at Harman International, has been using the CLB to enhance products.

According to Matt, they are always looking to "bend the limits of engineering."

"For example, we can now offload custom control logic and diagnostics from the main software control loop to the CLB. At 400 kHz, every cycle counts, so this reduction in overhead is extremely welcome. These cycle savings open up bandwidth that can be directly applied to more innovative control features," Matt said.

He also shared how the signal integration of the enhanced PWM (ePWM) peripheral is "profound."

"Within the CLB, designers can intercept internal ePWM signals such as action qualifiers and dead band, condition with logic, and basically create their own version of a submodule. This feature has already saved the day here at Harman. The CLB adds an incredible amount of customization to an already extremely capable device."

The new CLB peripheral is breaking barriers in control-oriented design, offering a novel tool set for crafting differentiated solutions. The latest C2000Ware revision includes 15 CLB examples; encoder examples are in the MotorControl sofware development kit. Check out the training and application notes. See how the CLB can defy conventional logic.

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 Copyright © 2023, Texas Instruments Incorporated