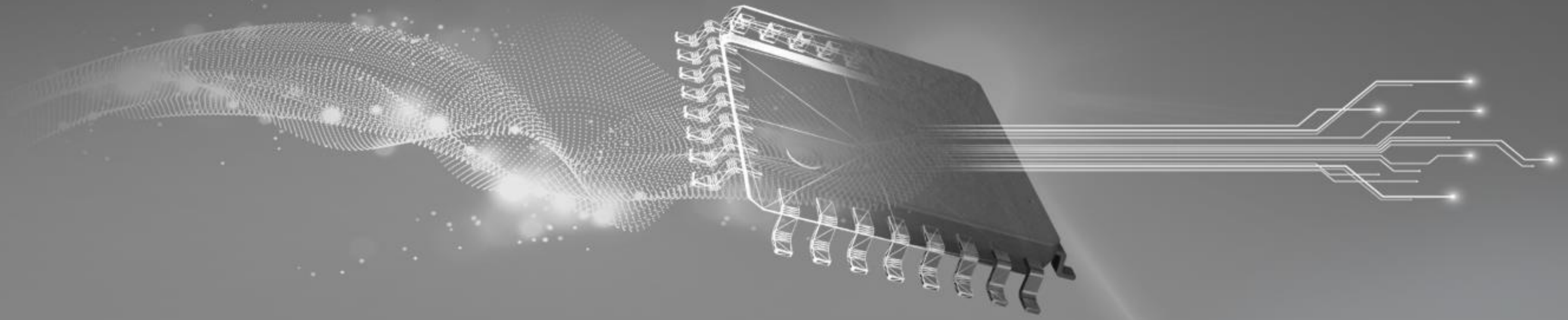
TI TECH DAYS



Enable Differentiation and win with CLB in various applications, using the newly launched CLB configuration and simulation tool

Nima Eskandari

C2000 real-time microcontrollers

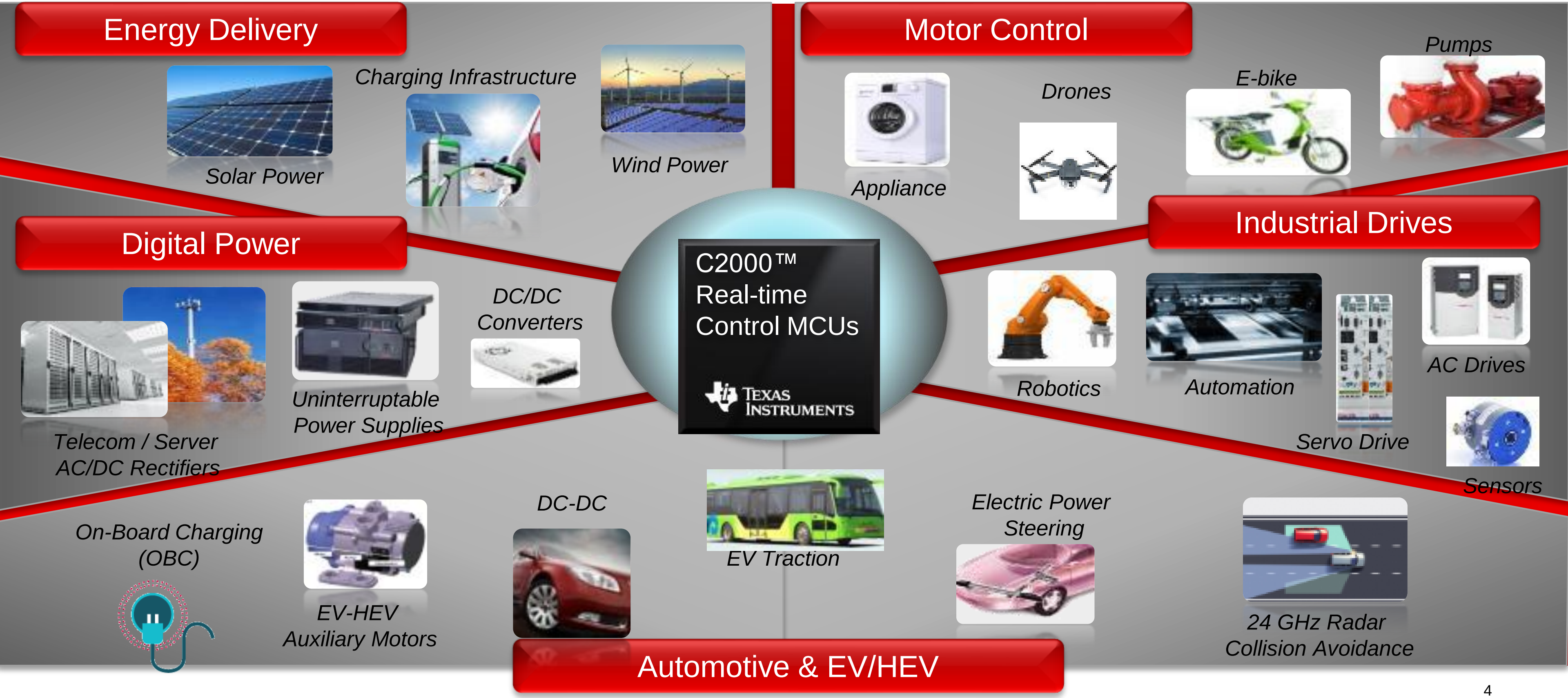
All of the content described are available at:

- Documents are available:
  - C:\ti\c2000\C2000Ware_VERSION\utilities\clb_tool\clb_syscfg\doc
  - CLB Tool User guide https://www.ti.com/lit/pdf/spruir8
  - Device TRMs with CLB included
    - F28X7x (F2837xS, F2837xD, F2807x)
    - F28004x
    - F2838x
    - F28002x
  - App. Note (SPRACL3) Designing with the C2000 Configurable Logic Block
    - https://www.ti.com/lit/pdf/spracl3
  - App. Note (SPRACO2) How to Migrate Custom Logic From an FPGA/CPLD to C2000™ Microcontrollers
    - https://www.ti.com/lit/pdf/spraco2
  - More information available here

Agenda

- C2000^{TM} Real-Time Controller Overview
- CLB Overview
- CLB Architecture
- CLB Connectivity
- SysConfig based CLB Tool
  - SysConfig Overview
  - Tool Capabilities, Inputs/Outputs
  - CLB Development flow
  - Roadmap

Where is C2000 Real-time Control?



Energy Delivery

Solar Power

Charging Infrastructure

Wind Power

Motor Control

Appliance

Drones

E-bike

Pumps

Digital Power

Telecom / Server AC/DC Rectifiers

Uninterruptable Power Supplies

DC/DC Converters

C2000™ Real-time Control MCUs

Texas Instruments

Industrial Drives

Robotics

Automation

Servo Drive

AC Drives

Sensors

Automotive & EV/HEV

On-Board Charging (OBC)

EV-HEV Auxiliary Motors

DC-DC

EV Traction

Electric Power Steering

24 GHz Radar Collision Avoidance

C2000™ real-time controllers overview

C2000™ Real-Time Control Microcontrollers Texas Instruments

Scalable, **ultra-low latency**, **real-time controller** platform designed for efficiency in power electronics, such as high power density, high switching frequencies, GaN and SiC technologies

C2000 controllers

Highly accurate sensing

- 12-/16-bit ADCs, up to 24 channels
- ADC post processing, and externally accessible DACs

Sense

Process

High performance processing
Floating-point DSP C28x™ core + parallel multi-core architecture + instructions set optimized for control math, up to 925 MIPS

Highly flexible, High-resolution PWMs:

- Up to 32 outputs
- Integrated comparators, eCAP, eQEP,
- Position Manager, Sigma-Delta Filter Module, trip zones

Control

Interface

CAN, CAN-FD, LIN, UART, SPI, I2C, PMBus, USB, 10/100 Ethernet MAC, EtherCAT®, XEMIF

Expertise and support:
Software libraries, reference designs, and functional safety-compliant devices.

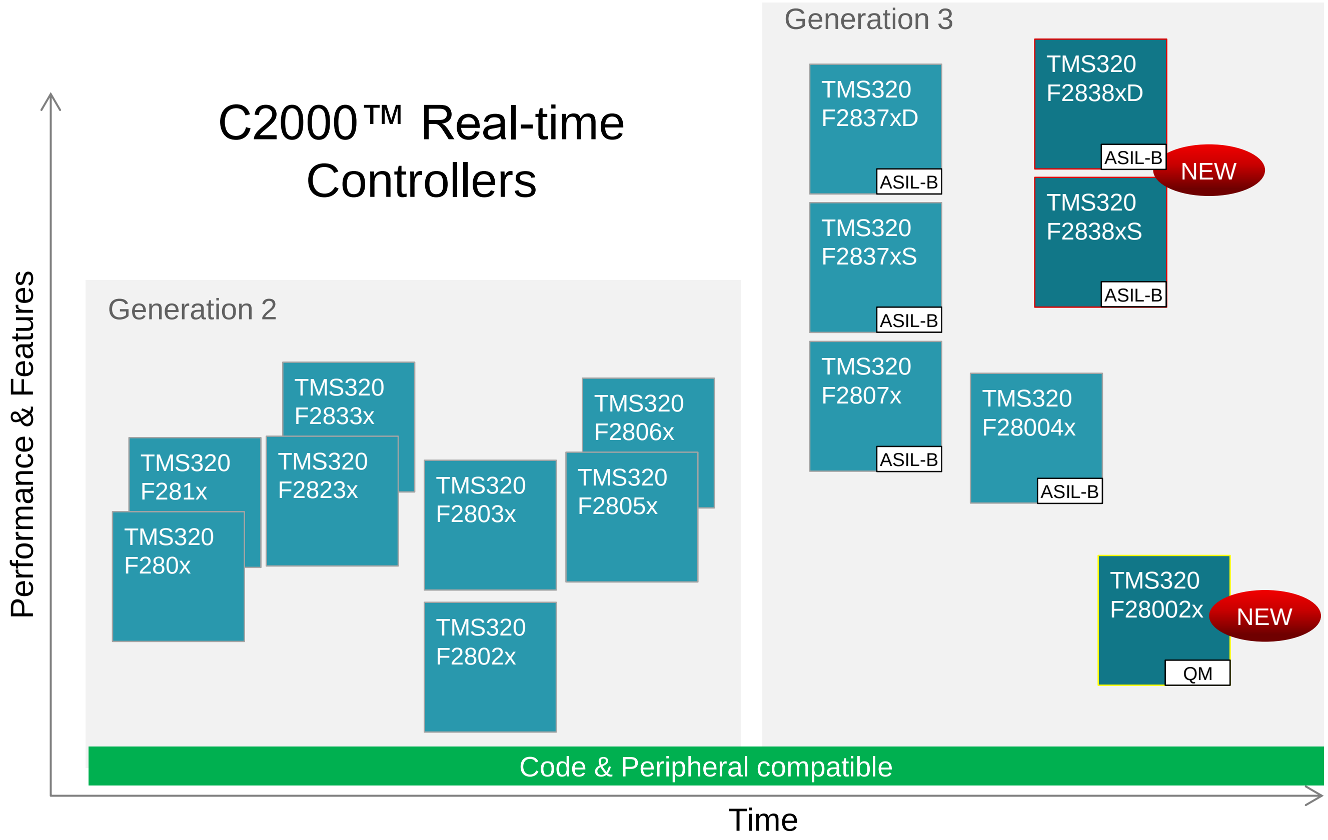
25 years expertise in real-time control systems

Leading innovation:
Configurable Logic Block for peripheral customization, Fast Serial Interface for high-speed communication, ERAD for enhanced diagnostics and profiling

1.2-V core, 3.3-V I/O design	Up to 1.5 MB Flash, 256 kB RAM (ECC protected)	QFN, QFP, BGA packages	-40 to 125C temperature range	Q100 automotive qualified options

Broad range of industrial and automotive applications with compatible software

C2000^{TM} Real-Time Controller Portfolio



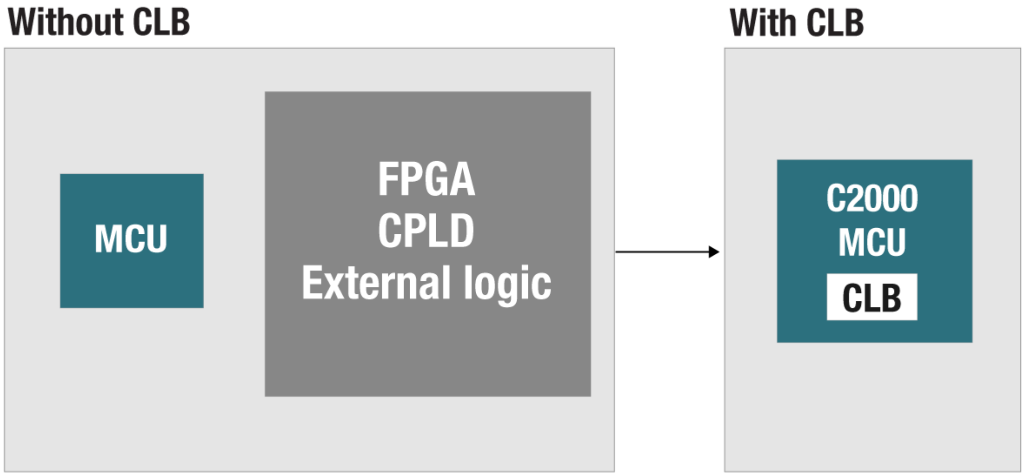
C2000^{TM} families address wide-ranging real-time control applications to sense feedback, process a control response, and actuate the system with <u>minimal latency</u>

- Price points from entry to top performance
- Application-tuned feature-sets
- Integrated flash memory sizes from 16 kB to 1.5 MB
- Temperature ranges from -40°C to 125°C and AEC-Q100 (automotive) qualification
- Security enablers including software IP protection and debug security
- Functional Safety compliant
- Wide-array of package options

CLB Overview

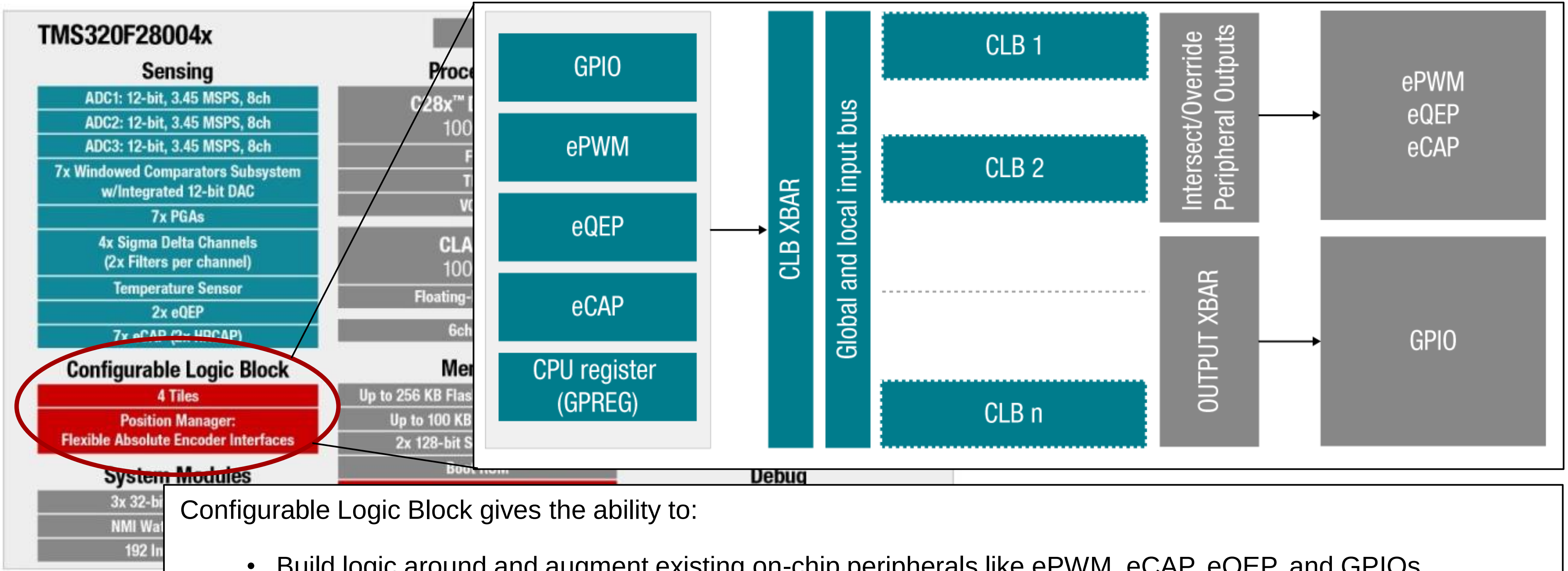
Integrate custom logic and *Augment* peripheral capability in your real-time MCU applications

Customized logic is usually done in a system by adding FPGAs, CPLDs, or external logic. These systems almost always still include a traditional microcontroller as well.



C2000 Configurable Logic Block (CLB) enables customization in a microcontroller based real-time control system while eliminating or reducing the size of the FPGA, CPLD, or external logic

Why C2000 Configurable Logic Block Peripheral



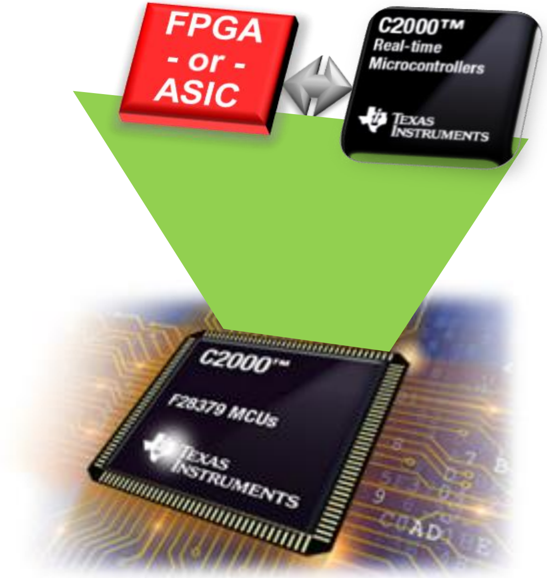
Configurable Logic Block gives the ability to:

- Build logic around and augment existing on-chip peripherals like ePWM, eCAP, eQEP, and GPIOs
- Implement independent custom logic

Typical Applications: Replacing CPLD/FPGA, ASICs

Reduces system cost
Improves system performance

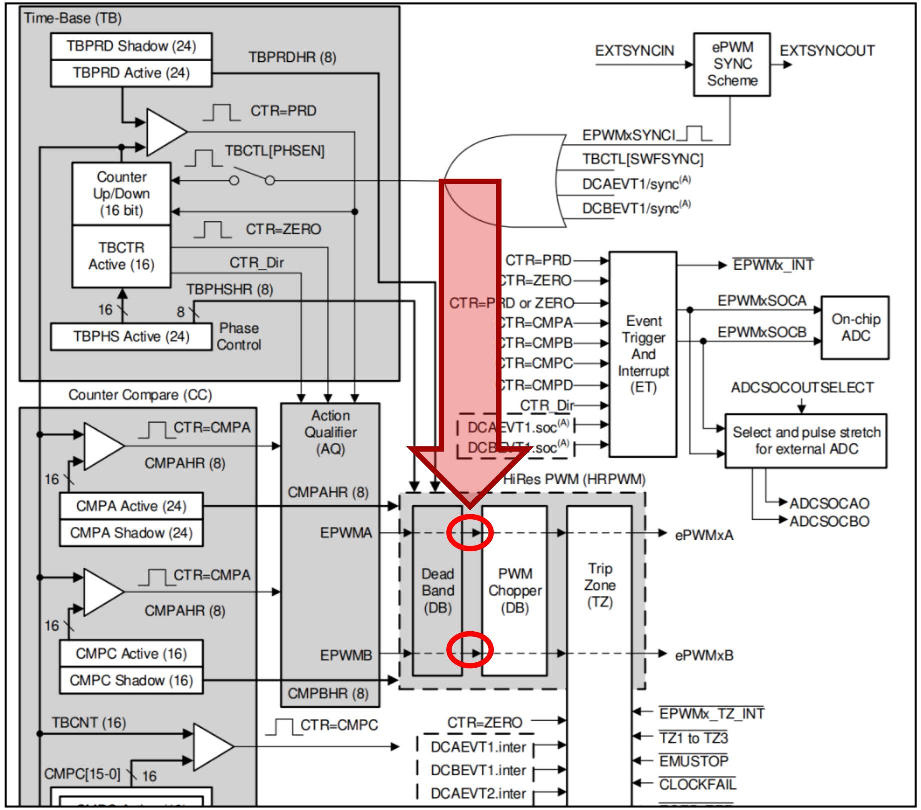
- Applications which require external FPGAs/CPLDs along with C2000™ Real-Time Controller→ Potential replacement with CLB
- Reduces feedback latency thus improving control loop time
- Reduces the cost and board area of adding external devices like FPGAs or ASICs

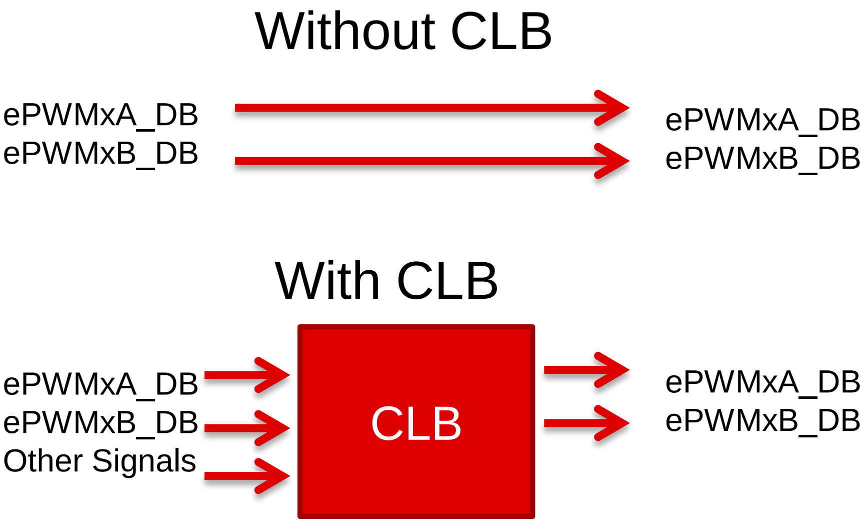


CLB Advantages

- **CLB has certain advantages over CPLDs/FPGAs:**
  - Residing inside the Real-Time Controller, CLB has direct access to key CPU and peripheral signals
  - Internal CPU/peripheral signals can be used to supplement or modify logic inside control peripherals and external glue logic
  - Built-in simple programmable processor (HLC) facilitates data transfers between CLB and MCU memory. Up to four stored programs can be triggered by low-to-high transition of selected internal CLB signals.
  - Timing of CLB signals already designed for the specified frequency – any logic that is created using CLB is guaranteed to meet the timing requirements
  - Fully S/W configurable and can be changed easily

Insert Custom Logic Inside Existing Peripherals





Without CLB

ePWMxA_DB
ePWMxB_DB

ePWMxA_DB
ePWMxB_DB

With CLB

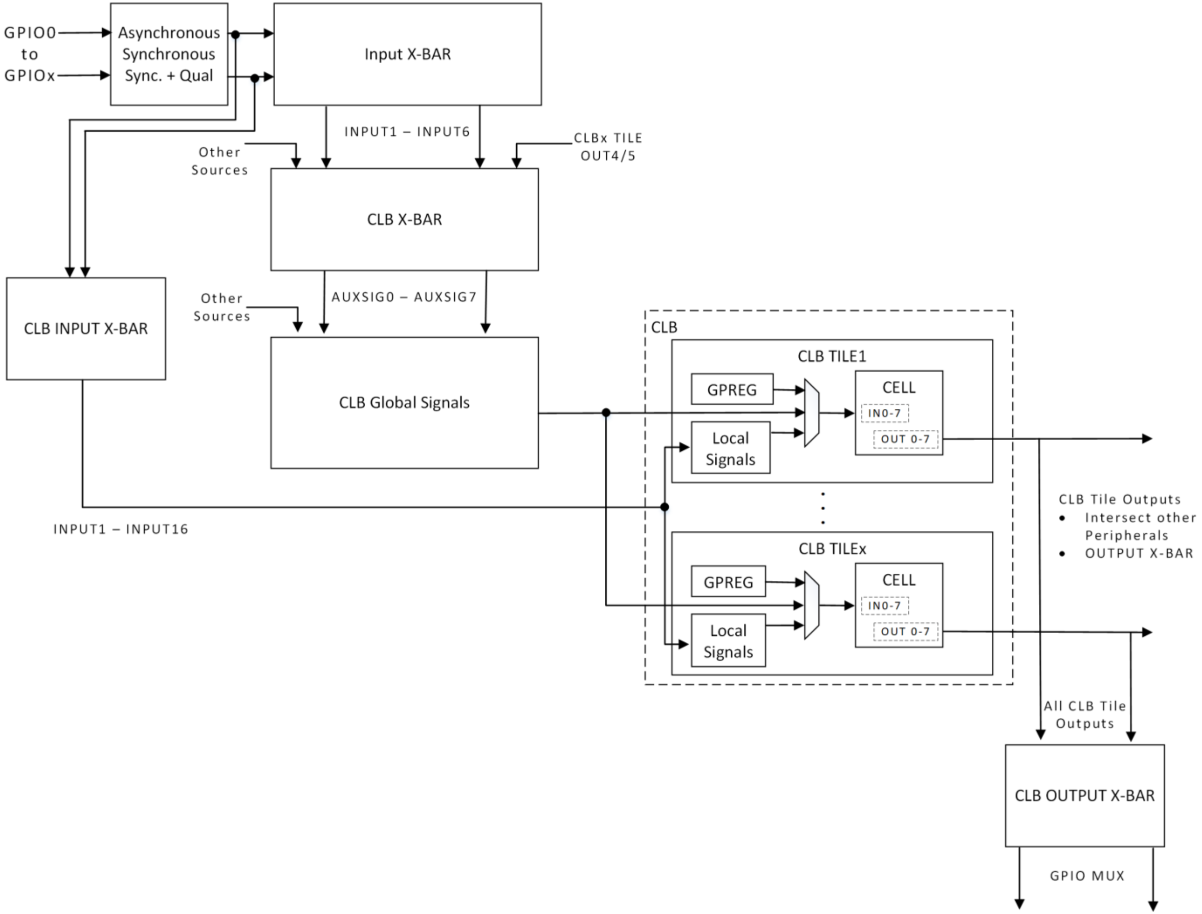
ePWMxA_DB
ePWMxB_DB
Other Signals

CLB

ePWMxA_DB
ePWMxB_DB

CLB Connections

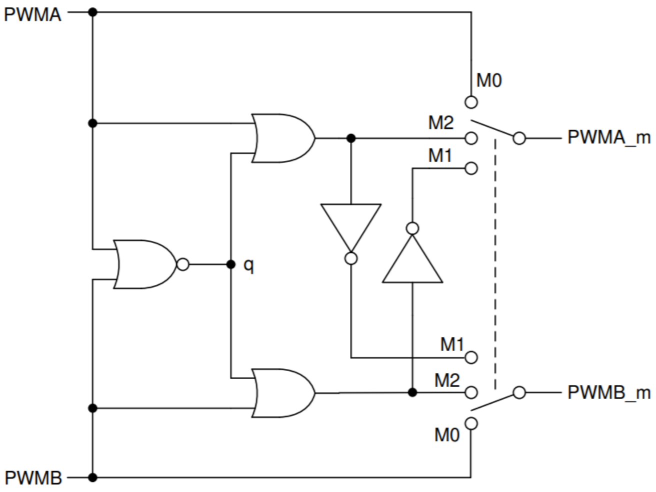
- Insert CLB inside a peripheral
  - Override internal peripheral signals using CLB outputs
- GPIO to CLB to GPIO
  - Design new peripherals inside the Real-Time controller
- Precondition signals before entering a peripheral
- Add logic before sending signals outside of the chip
  - Replace CPLD/FPGA



Target Applications

Few areas listed as examples below

- Automotive
  - Advanced PWM Protection schemes for reliability and safety
  - Multi-level inverters
  - Burst Mode PWM
  - PWM periodic blanking of pulses (thinning)
  - Complex PWM generation
- General Purpose
  - Filters
  - Signal conditioning
  - Task profiling / Time threshold monitoring
  - Complex sequence detection on signals (beyond ECAP)
  - Position encoder interfaces
  - PTO (Pulse Train Output)

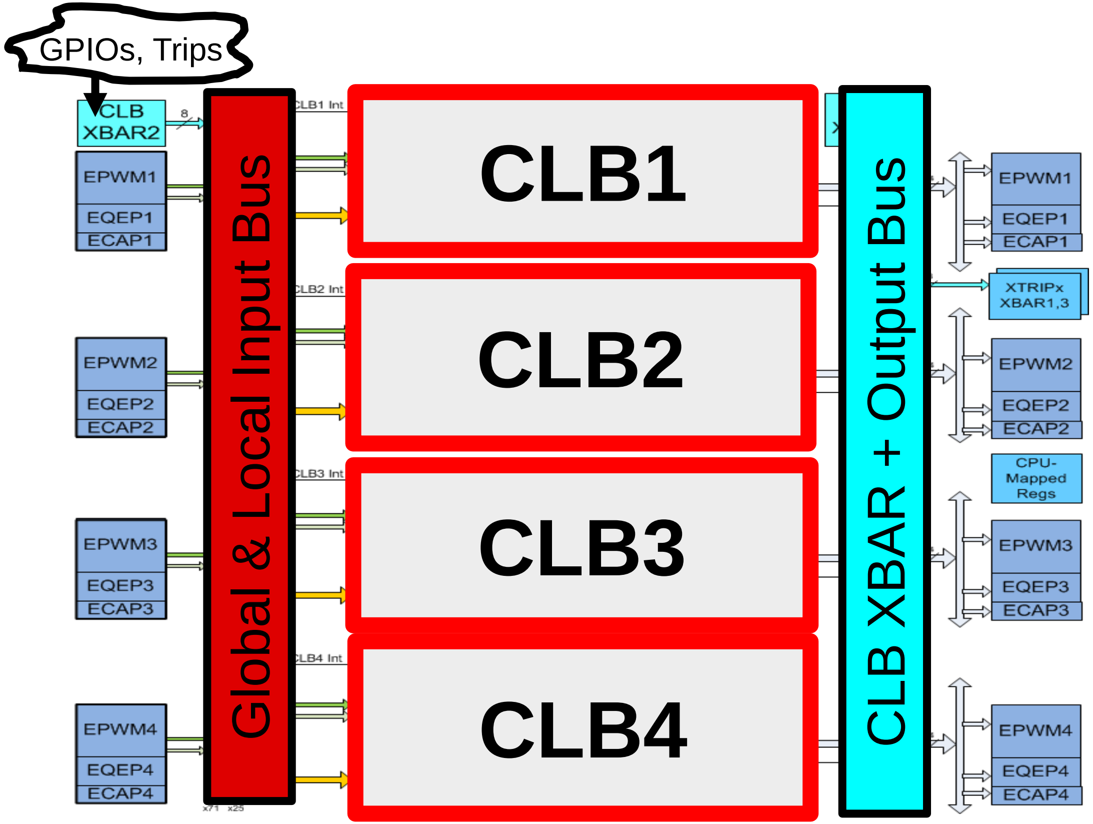


Some Automotive Use Cases

Use Case	Description
Independent external clock monitor	Provide single clock cycle detection of critical clock failure. Provides HW robustness and full coverage vs. SW clock monitoring.
Trip ePWM as a function of eQEP position	Independent control of ePWM trip zones as a function of eQEP position. Creative control of ePWM signals in complex systems and elimination of external circuitry.
Advanced logic ePWM trip	Advanced logic for ePWM trip zones. Solves deep system requirements without complex SW or external logic.
Sync eQEP with external signal	Synchronization of eQEP counts to external signals for downstream synchronous logic. Eliminates CPLD or external circuitry.
Time stamp generation	Use CLB to provide time stamp conditionally. Provides additional flexibility.

CLB Architecture

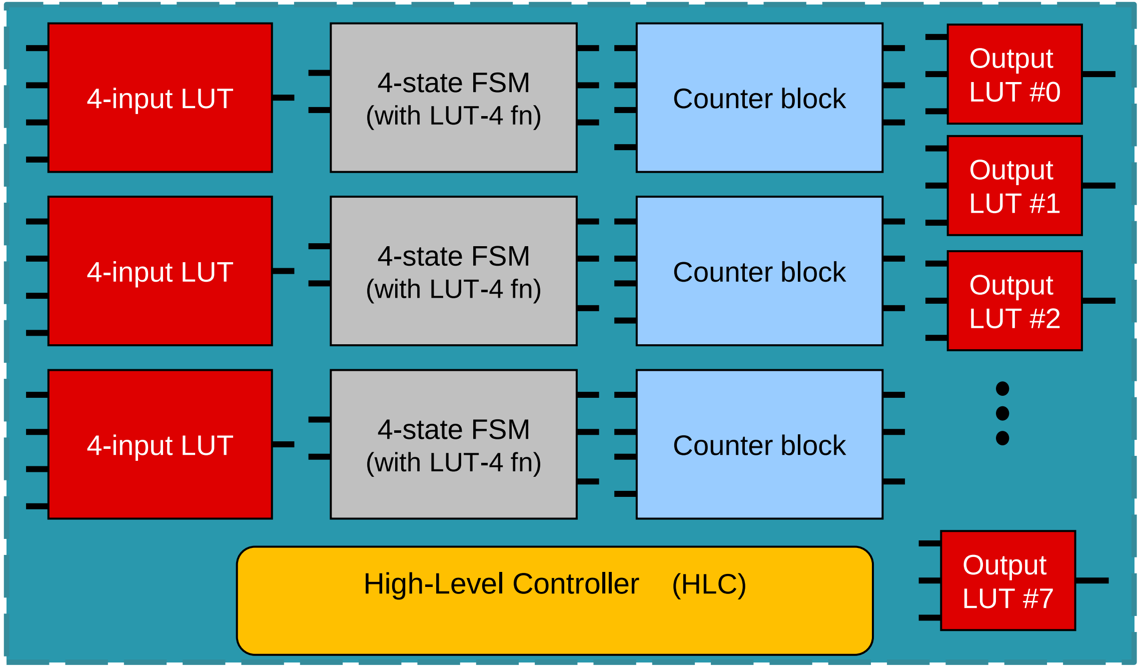
CLB System Architecture



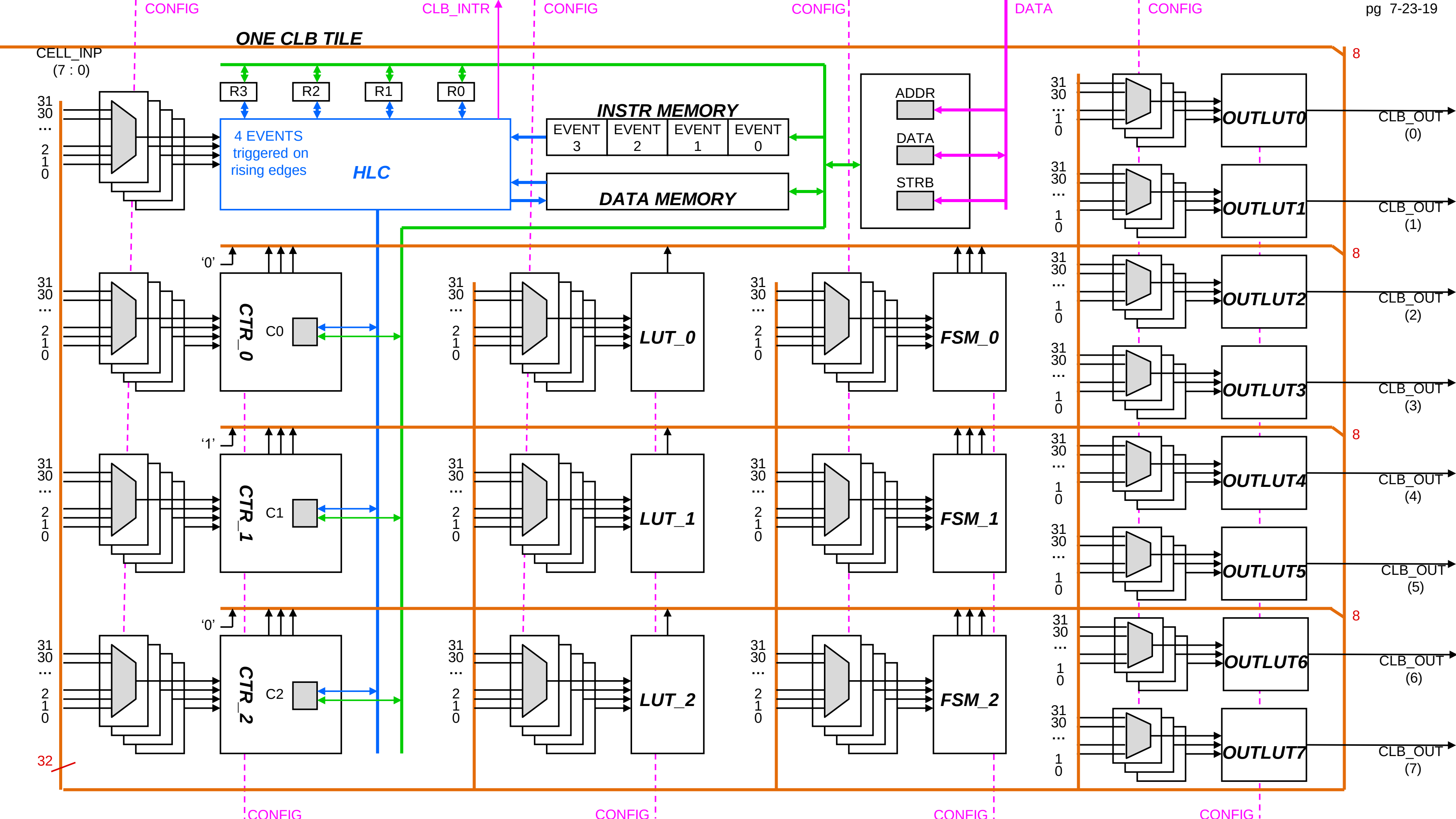
- 8/4/2 independent CLBs depending on the device
- High connectivity to: ePWM, eQEP, eCAP, Trip Events, GPIOs
- 100MHz (150 MHz on F2838x)
- Simplified Timing Closure
- Standard Cell Gate construct (not a hard macro)

Inside the CLB

CLB 1

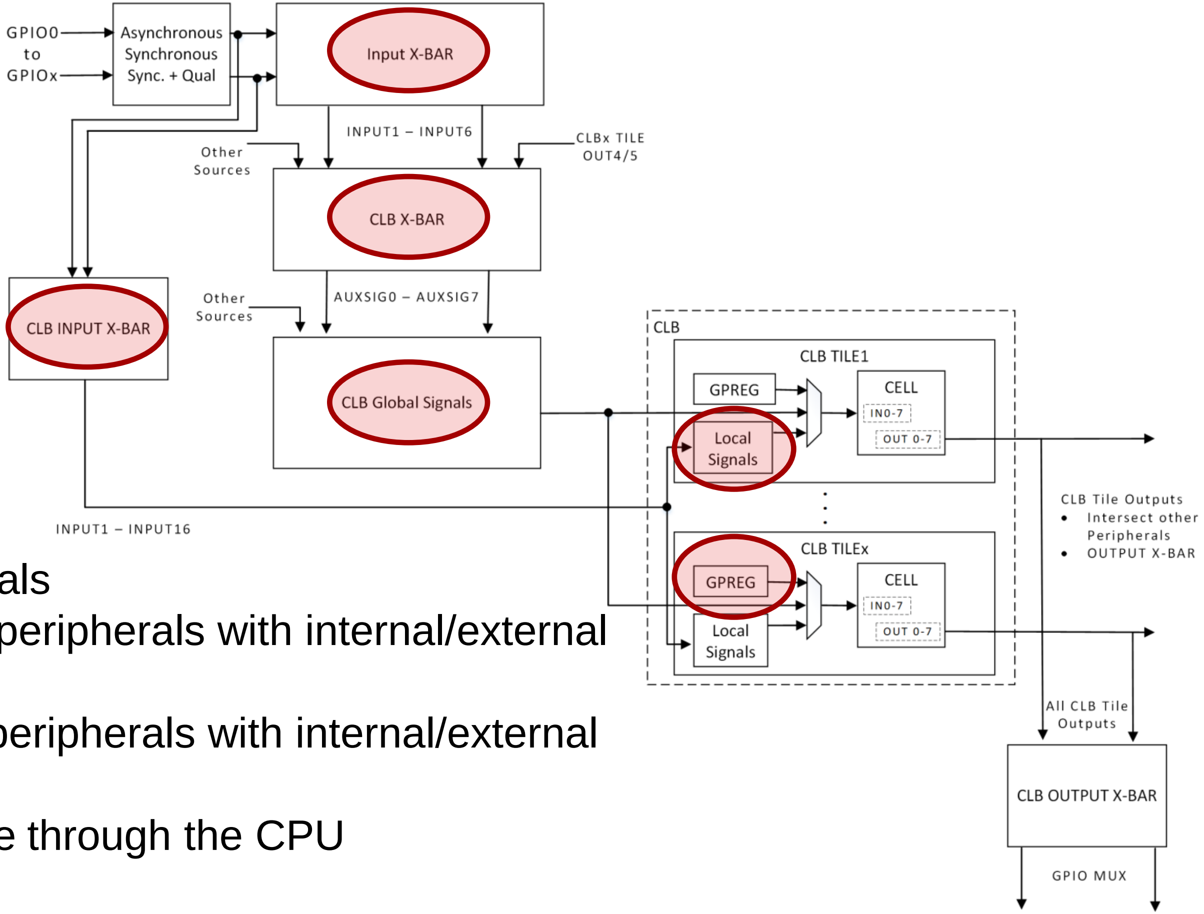


LUT = Look Up Table
FSM = Finite State Machine


ONE CLB TILE
CONFIG
CLB_INTR
DATA
CELL_INP
(7 : 0)
31
30
...
2
1
0
R3
R2
R1
R0
4 EVENTS
triggered on
rising edges
HLC
INSTR MEMORY
EVENT 3
EVENT 2
EVENT 1
EVENT 0
DATA MEMORY
ADDR
DATA
STRB
OUTLUT0
OUTLUT1
OUTLUT2
OUTLUT3
OUTLUT4
OUTLUT5
OUTLUT6
OUTLUT7
CLB_OUT (0)
CLB_OUT (1)
CLB_OUT (2)
CLB_OUT (3)
CLB_OUT (4)
CLB_OUT (5)
CLB_OUT (6)
CLB_OUT (7)
8
'0'
'1'
'0'
CTR_0
CTR_1
CTR_2
C0
C1
C2
LUT_0
LUT_1
LUT_2
FSM_0
FSM_1
FSM_2
32

CLB Connectivity

CLB Input Signals



The inputs to the CLB can be:

Input X-BAR – GPIO

CLB Input X-BAR – GPIO

CLB X-BAR – Other Peripherals

CLB Global Signals – Many peripherals with internal/external signals

CLB Local Signals – Many peripherals with internal/external signals

GPREG – Software accessible through the CPU

CLB X-BAR, Local and Global Mux Signals

- Some of the many, CLB X-BAR, Local and Global Mux signals

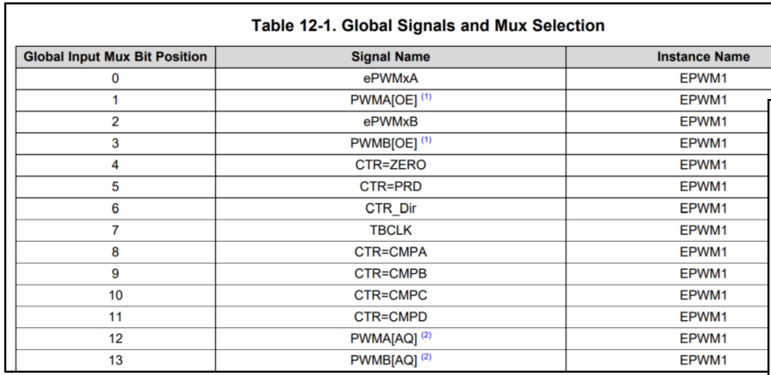


Table 12-1. Global Signals and Mux Selection

Global Input Mux Bit Position	Signal Name	Instance Name
0	ePWMxA	EPWM1
1	PWMA[OE] [(1)]	EPWM1
2	ePWMxB	EPWM1
3	PWMB[OE] [(1)]	EPWM1
4	CTR=ZERO	EPWM1
5	CTR=PRD	EPWM1
6	CTR_Dir	EPWM1
7	TBCLK	EPWM1
8	CTR=CMPA	EPWM1
9	CTR=CMPB	EPWM1
10	CTR=CMPC	EPWM1
11	CTR=CMPD	EPWM1
12	PWMA[AQ] [(2)]	EPWM1
13	PWMB[AQ] [(2)]	EPWM1

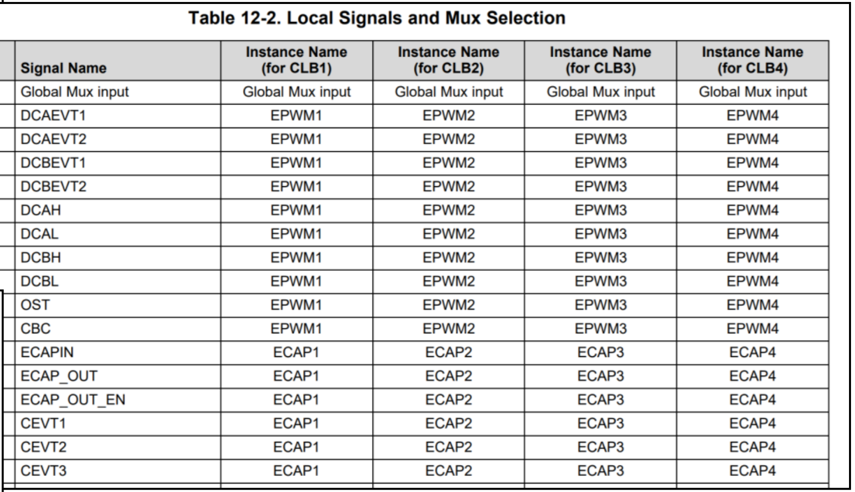


Table 12-2. Local Signals and Mux Selection

Bit Position	Signal Name	Instance Name (for CLB1)	Instance Name (for CLB2)	Instance Name (for CLB3)	Instance Name (for CLB4)
0	Global Mux input	Global Mux input	Global Mux input	Global Mux input	Global Mux input
1	DCAEVT1	EPWM1	EPWM2	EPWM3	EPWM4
2	DCAEVT2	EPWM1	EPWM2	EPWM3	EPWM4
3	DCBEVT1	EPWM1	EPWM2	EPWM3	EPWM4
4	DCBEVT2	EPWM1	EPWM2	EPWM3	EPWM4
5	DCAH	EPWM1	EPWM2	EPWM3	EPWM4
6	DCAL	EPWM1	EPWM2	EPWM3	EPWM4
7	DCBH	EPWM1	EPWM2	EPWM3	EPWM4
8	DCBL	EPWM1	EPWM2	EPWM3	EPWM4
	OST	EPWM1	EPWM2	EPWM3	EPWM4
	CBC	EPWM1	EPWM2	EPWM3	EPWM4
	ECAPIN	ECAP1	ECAP2	ECAP3	ECAP4
	ECAP_OUT	ECAP1	ECAP2	ECAP3	ECAP4
	ECAP_OUT_EN	ECAP1	ECAP2	ECAP3	ECAP4
	CEVT1	ECAP1	ECAP2	ECAP3	ECAP4
	CEVT2	ECAP1	ECAP2	ECAP3	ECAP4
	CEVT3	ECAP1	ECAP2	ECAP3	ECAP4

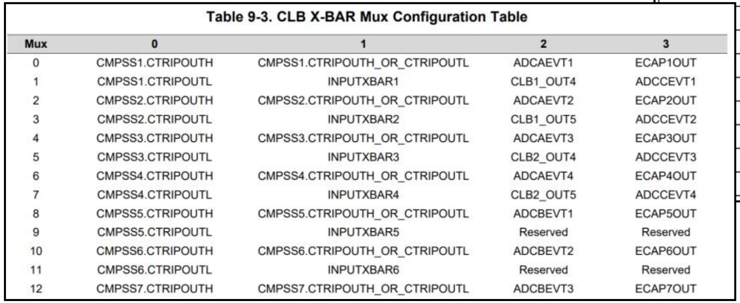


Table 9-3. CLB X-BAR Mux Configuration Table

Mux	0	1	2	3
0	CMPSS1.CTRIPOUTH	CMPSS1.CTRIPOUTH_OR_CTRIPOUTL	ADCAEVT1	ECAP1OUT
1	CMPSS1.CTRIPOUTL	INPUTXBAR1	CLB1_OUT4	ADCCEVT1
2	CMPSS2.CTRIPOUTH	CMPSS2.CTRIPOUTH_OR_CTRIPOUTL	ADCAEVT2	ECAP2OUT
3	CMPSS2.CTRIPOUTL	INPUTXBAR2	CLB1_OUT5	ADCCEVT2
4	CMPSS3.CTRIPOUTH	CMPSS3.CTRIPOUTH_OR_CTRIPOUTL	ADCAEVT3	ECAP3OUT
5	CMPSS3.CTRIPOUTL	INPUTXBAR3	CLB2_OUT4	ADCCEVT3
6	CMPSS4.CTRIPOUTH	CMPSS4.CTRIPOUTH_OR_CTRIPOUTL	ADCAEVT4	ECAP4OUT
7	CMPSS4.CTRIPOUTL	INPUTXBAR4	CLB2_OUT5	ADCCEVT4
8	CMPSS5.CTRIPOUTH	CMPSS5.CTRIPOUTH_OR_CTRIPOUTL	ADCBEVT1	ECAP5OUT
9	CMPSS5.CTRIPOUTL	INPUTXBAR5	Reserved	Reserved
10	CMPSS6.CTRIPOUTH	CMPSS6.CTRIPOUTH_OR_CTRIPOUTL	ADCBEVT2	ECAP6OUT
11	CMPSS6.CTRIPOUTL	INPUTXBAR6	Reserved	Reserved
12	CMPSS7.CTRIPOUTH	CMPSS7.CTRIPOUTH_OR_CTRIPOUTL	ADCBEVT3	ECAP7OUT

CLB Inputs

- CLB also has access to communication peripheral signals such as:
  - FSI
  - SPI
  - SCI
  - ETHERCAT
- CLB also has access to debug signals:
  - ERAD
  - CPU HALT signals

CLB Outputs

- CLB outputs can be connected to:
  - GPIOs (Output X-BAR, CLB Output X-BAR)
  - Or override internal signals inside the device/peripherals

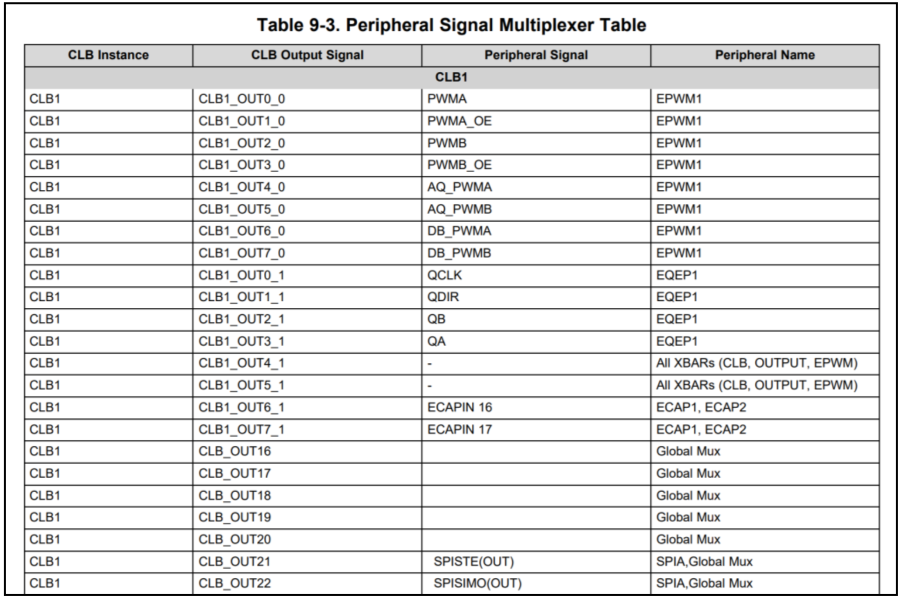


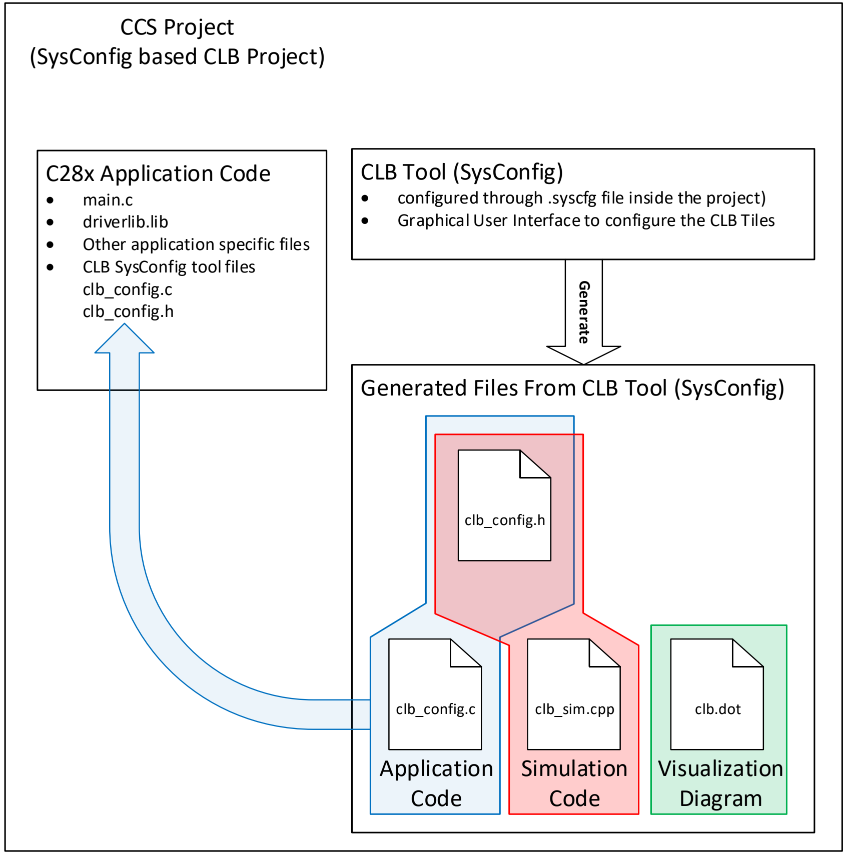
Table 9-3. Peripheral Signal Multiplexer Table

CLB Instance	CLB Output Signal	Peripheral Signal	Peripheral Name
CLB1			
CLB1	CLB1_OUT0_0	PWMA	EPWM1
CLB1	CLB1_OUT1_0	PWMA_OE	EPWM1
CLB1	CLB1_OUT2_0	PWMB	EPWM1
CLB1	CLB1_OUT3_0	PWMB_OE	EPWM1
CLB1	CLB1_OUT4_0	AQ_PWMA	EPWM1
CLB1	CLB1_OUT5_0	AQ_PWMB	EPWM1
CLB1	CLB1_OUT6_0	DB_PWMA	EPWM1
CLB1	CLB1_OUT7_0	DB_PWMB	EPWM1
CLB1	CLB1_OUT0_1	QCLK	EQEP1
CLB1	CLB1_OUT1_1	QDIR	EQEP1
CLB1	CLB1_OUT2_1	QB	EQEP1
CLB1	CLB1_OUT3_1	QA	EQEP1
CLB1	CLB1_OUT4_1	-	All XBARs (CLB, OUTPUT, EPWM)
CLB1	CLB1_OUT5_1	-	All XBARs (CLB, OUTPUT, EPWM)
CLB1	CLB1_OUT6_1	ECAPIN 16	ECAP1, ECAP2
CLB1	CLB1_OUT7_1	ECAPIN 17	ECAP1, ECAP2
CLB1	CLB_OUT16		Global Mux
CLB1	CLB_OUT17		Global Mux
CLB1	CLB_OUT18		Global Mux
CLB1	CLB_OUT19		Global Mux
CLB1	CLB_OUT20		Global Mux
CLB1	CLB_OUT21	SPISTE(OUT)	SPIA,Global Mux
CLB1	CLB_OUT22	SPISIMO(OUT)	SPIA,Global Mux

CLB Tool

How do you configure the CLB?

- CLB Tool is a Code Composer Studio (CCS) SysConfig plug-in
- GUI based tool to configure and program each CLB tile
- Simulation and Visualization tool to verify logic
- Feature Examples in C2000Ware and System examples in application Software Development Kits



CCS Project
(SysConfig based CLB Project)

C28x Application Code

- main.c
- driverlib.lib
- Other application specific files
- CLB SysConfig tool files
  clb_config.c
  clb_config.h

CLB Tool (SysConfig)

- configured through .syscfg file inside the project)
- Graphical User Interface to configure the CLB Tiles

Generate

Generated Files From CLB Tool (SysConfig)

clb_config.h

clb_config.c

clb_sim.cpp

clb.dot

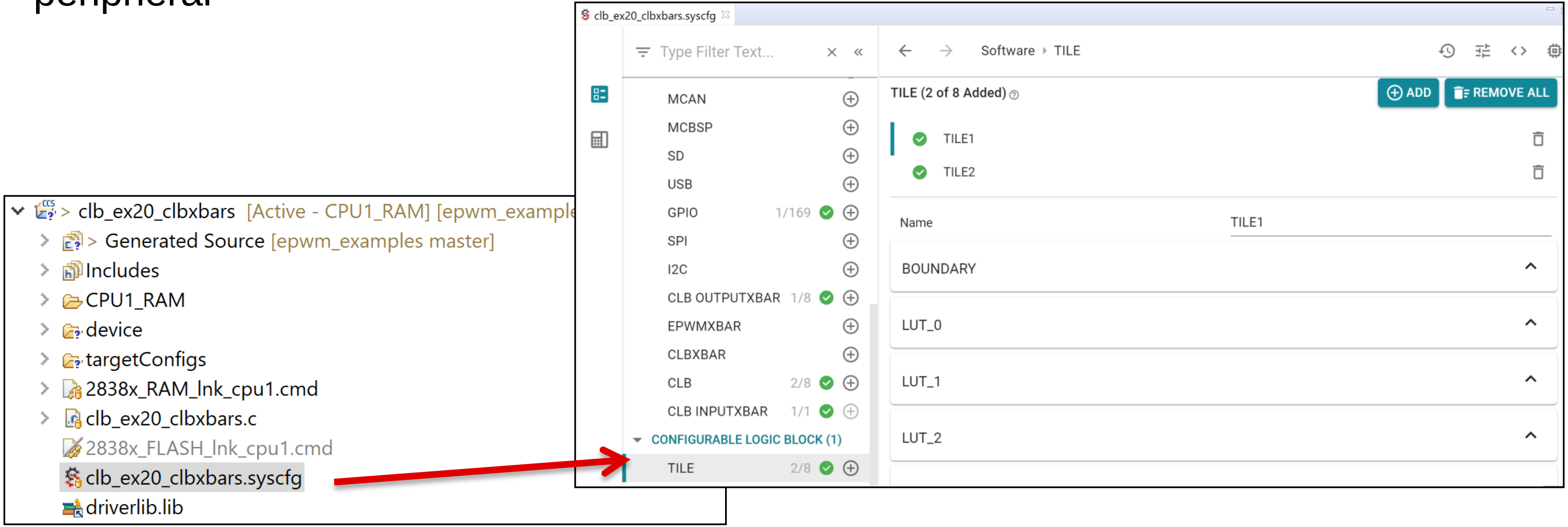
Application Code

Simulation Code

Visualization Diagram

CLB Tool

GUI tool to configure the CLB peripheral



clb_ex20_clbxbars [Active - CPU1_RAM] [epwm_example

- Generated Source [epwm_examples master]
- Includes
- CPU1_RAM
- device
- targetConfigs
- 2838x_RAM_lnk_cpu1.cmd
- clb_ex20_clbxbars.c
- 2838x_FLASH_lnk_cpu1.cmd
- clb_ex20_clbxbars.syscfg
- driverlib.lib

clb_ex20_clbxbars.syscfg

Type Filter Text...

MCAN	
MCBSP	
SD	
USB	
GPIO	1/169 ✓
SPI	
I2C	
CLB OUTPUTXBAR	1/8 ✓
EPWMXBAR	
CLBXBAR	
CLB	2/8 ✓
CLB INPUTXBAR	1/1 ✓

CONFIGURABLE LOGIC BLOCK (1)

TILE	2/8 ✓

Software ▸ TILE

TILE (2 of 8 Added)

ADD   REMOVE ALL

- ✓ TILE1
- ✓ TILE2

Name: TILE1

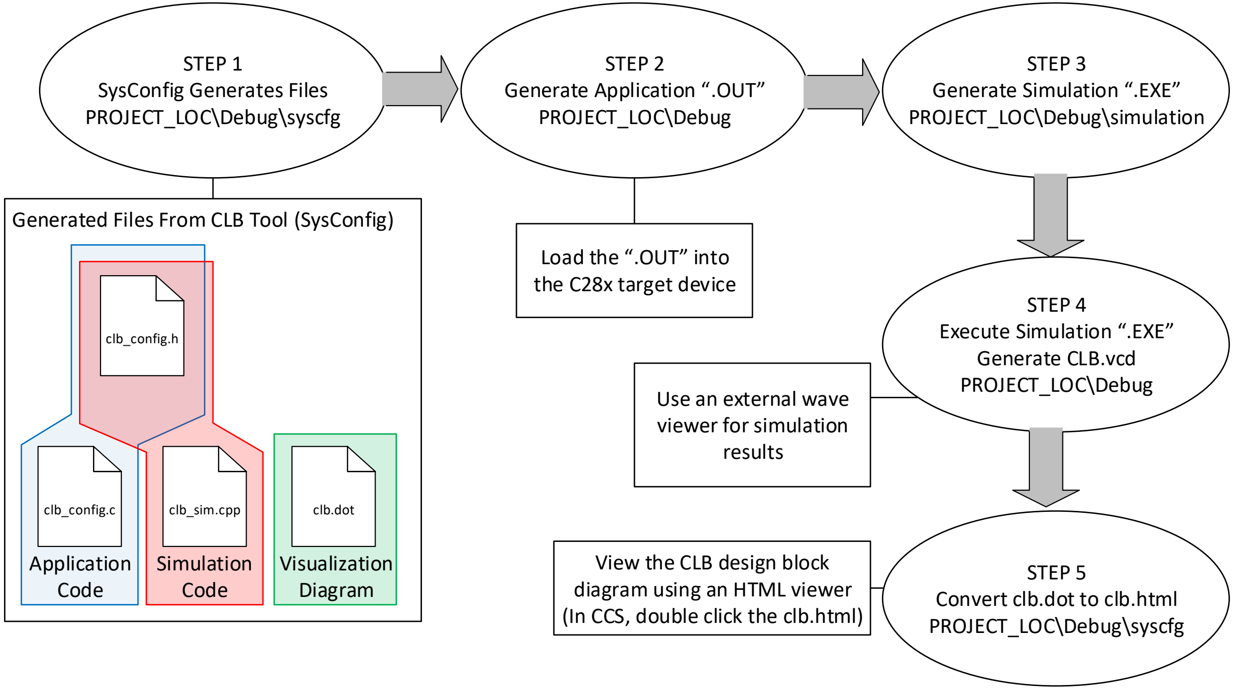
BOUNDARY

LUT_0

LUT_1

LUT_2

CLB Tool Development Flow



STEP 1
SysConfig Generates Files
PROJECT_LOC\Debug\syscfg

STEP 2
Generate Application “.OUT”
PROJECT_LOC\Debug

STEP 3
Generate Simulation “.EXE”
PROJECT_LOC\Debug\simulation

Generated Files From CLB Tool (SysConfig)

Load the “.OUT” into the C28x target device

STEP 4
Execute Simulation “.EXE”
Generate CLB.vcd
PROJECT_LOC\Debug

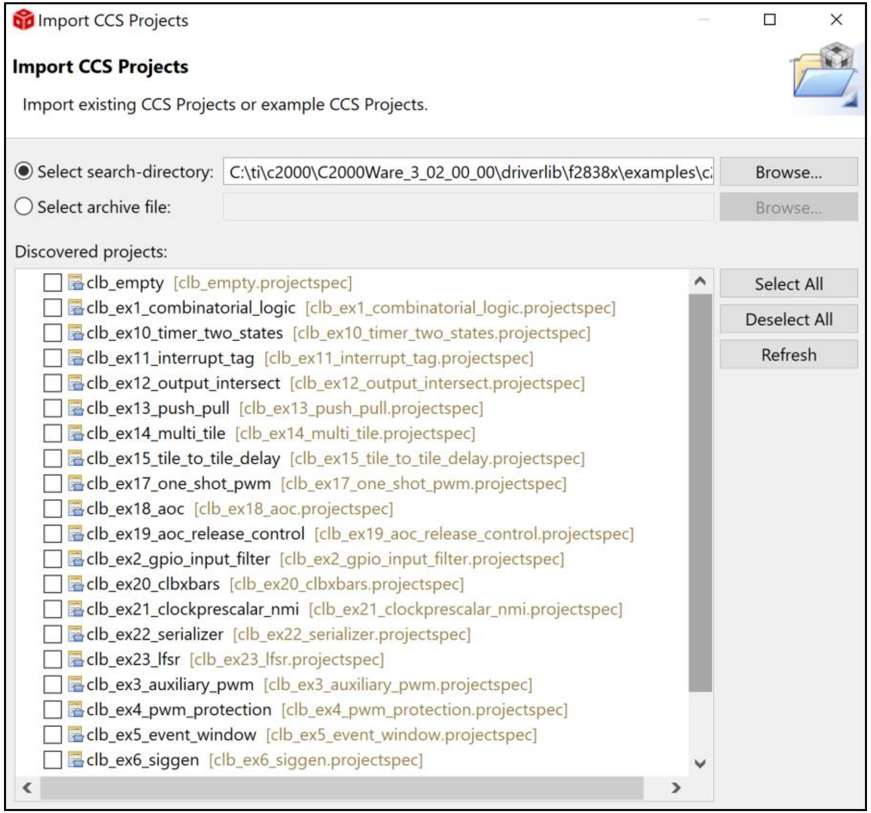
Use an external wave viewer for simulation results

View the CLB design block diagram using an HTML viewer (In CCS, double click the clb.html)

STEP 5
Convert clb.dot to clb.html
PROJECT_LOC\Debug\syscfg

CLB Examples

- Most examples for a peripheral
  - Simple building block examples
  - Complex system level examples
  - New feature examples



Import CCS Projects

Import existing CCS Projects or example CCS Projects.

● Select search-directory: C:\ti\c2000\C2000Ware_3_02_00_00\driverlib\f2838x\examples\c:

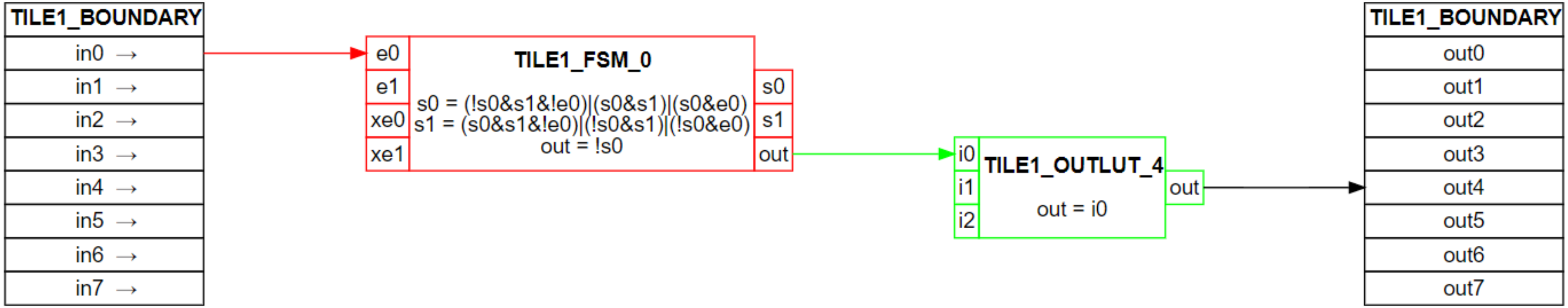
○ Select archive file:

Discovered projects:

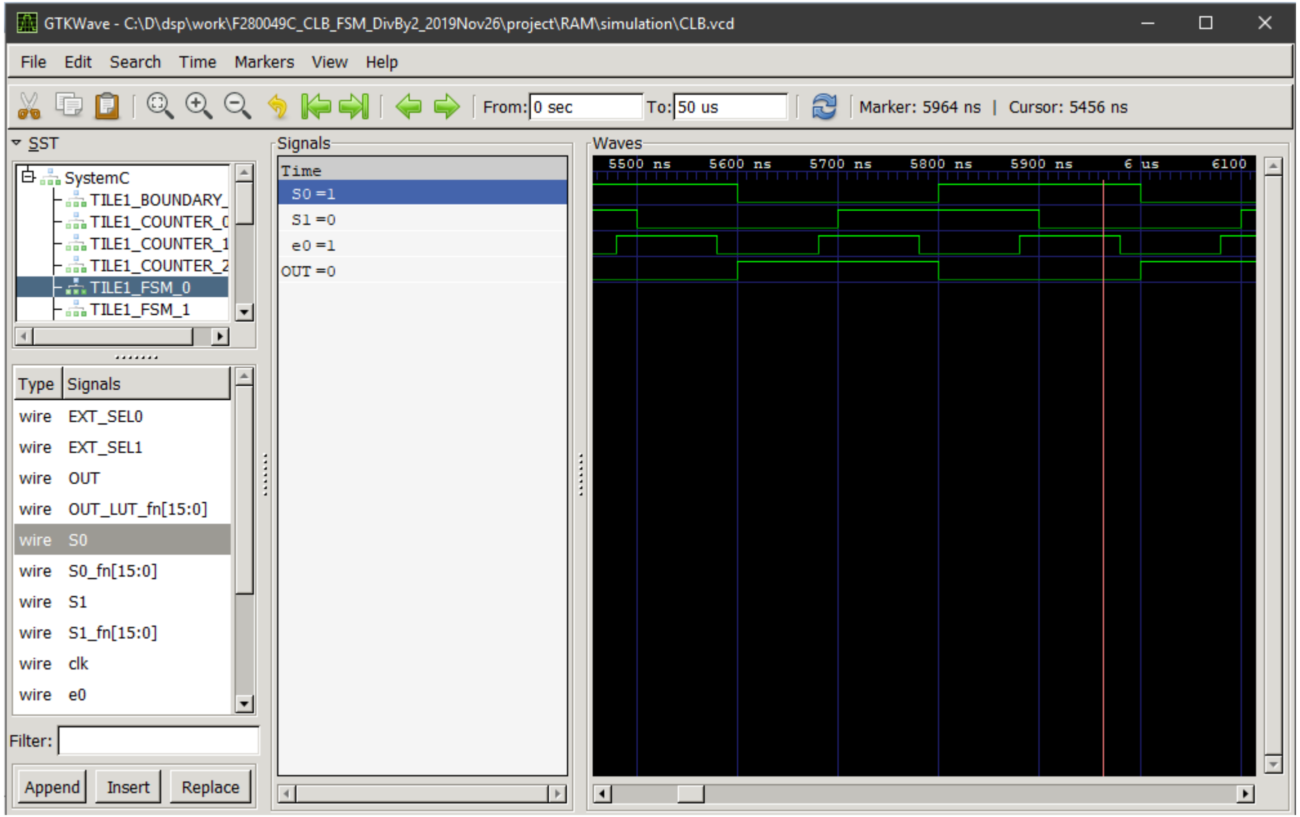
- □ clb_empty [clb_empty.projectspec]
- □ clb_ex1_combinatorial_logic [clb_ex1_combinatorial_logic.projectspec]
- □ clb_ex10_timer_two_states [clb_ex10_timer_two_states.projectspec]
- □ clb_ex11_interrupt_tag [clb_ex11_interrupt_tag.projectspec]
- □ clb_ex12_output_intersect [clb_ex12_output_intersect.projectspec]
- □ clb_ex13_push_pull [clb_ex13_push_pull.projectspec]
- □ clb_ex14_multi_tile [clb_ex14_multi_tile.projectspec]
- □ clb_ex15_tile_to_tile_delay [clb_ex15_tile_to_tile_delay.projectspec]
- □ clb_ex17_one_shot_pwm [clb_ex17_one_shot_pwm.projectspec]
- □ clb_ex18_aoc [clb_ex18_aoc.projectspec]
- □ clb_ex19_aoc_release_control [clb_ex19_aoc_release_control.projectspec]
- □ clb_ex2_gpio_input_filter [clb_ex2_gpio_input_filter.projectspec]
- □ clb_ex20_clbxbars [clb_ex20_clbxbars.projectspec]
- □ clb_ex21_clockprescalar_nmi [clb_ex21_clockprescalar_nmi.projectspec]
- □ clb_ex22_serializer [clb_ex22_serializer.projectspec]
- □ clb_ex23_lfsr [clb_ex23_lfsr.projectspec]
- □ clb_ex3_auxiliary_pwm [clb_ex3_auxiliary_pwm.projectspec]
- □ clb_ex4_pwm_protection [clb_ex4_pwm_protection.projectspec]
- □ clb_ex5_event_window [clb_ex5_event_window.projectspec]
- □ clb_ex6_siggen [clb_ex6_siggen.projectspec]

FSM Example: CCS Implementation (2/3)

CLB Tile Configuration



FSM Example: CLB Simulation



GTKWave - C:\D\dsp\work\F280049C_CLB_FSM_DivBy2_2019Nov26\project\RAM\simulation\CLB.vcd

From: 0 sec To: 50 us | Marker: 5964 ns | Cursor: 5456 ns

SST: SystemC, TILE1_BOUNDARY_, TILE1_COUNTER_0, TILE1_COUNTER_1, TILE1_COUNTER_2, TILE1_FSM_0, TILE1_FSM_1

Type	Signals
wire	EXT_SEL0
wire	EXT_SEL1
wire	OUT
wire	OUT_LUT_fn[15:0]
wire	S0
wire	S0_fn[15:0]
wire	S1
wire	S1_fn[15:0]
wire	clk
wire	e0

Signals:

```
Time
 S0 =1
 S1 =0
 e0 =1
OUT =0
```

Roadmap: CLB Type 2 and Type 3 Additional Features

CLB TYPE 2	CLB TYPE3
Asynchronous Output Conditioning Block (AOC) • Inverting, Gating, Set/Reset, Delay capabilities • Used to pre-condition signals COUNTER serializer mode • Operate COUNTER as serializer/shift register • Operate COUNTER as linear feedback shift register • Ideal for communication peripheral design HLC • Trigger events on both rising/falling edge of signals • Use CLB outputs as the source of the event triggers Clock Prescalar TILE to TILE connections through input muxes	Faster speed of operation at 150 MHz Pipeline mode • COUNTER uses pipelined version of the active counter register • HLC uses pipelined of the CLB output signals Continuous export of data from CLB • Uses SPI RX buffer to continuously export 16-bit data out of the CLB to the CPU.
3Q20 Release	1Q21 Release

CLB in C2000 Devices

Device Family	Number of CLB Instances	CLB Type	CLB XBARS
F2807x F2837xS F2838xD	4	Type 1	CLB X-BAR
F28004x	4	Type 2	CLB X-BAR
F2838x	8	Type 3	CLB X-BAR CLB Input X-BAR CLB Output X-BAR
F28002x	2	Type 3	CLB X-BAR CLB Input X-BAR CLB Output X-BAR

Software

- Code Composer Studio (CCS) version 9.3 or later
  - http://www.ti.com/tool/ccstudio-c2000
- C2000Ware latest version
  - http://www.ti.com/tool/C2000WARE
- Other 3rd party software (refer to CLB tool user guide section 2.2)
  - GCC compiler
    - Download “tdm-gcc” from the following link:http://sourceforge.net/projects/tdm-gcc/files/TDM-GCC%20Installer/tdm-gcc-webdl.exe/download
  - Simulation Viewer
    - Download the waveform viewer GTKwave from this link: https://sourceforge.net/projects/gtkwave/files/

Resources

- www.ti.com/c2000clb
- C2000Ware including the CLB Tool User's Guide **[SPRUIR8]**
- Application Note: How to Design with the CLB **[SPRACL3]**
- Application Note: How to Migrate from FPGA/CPLD to CLB **[SPRACO2]**
- Video Training Series: https://training.ti.com/clbtooloverview

***Integrate* custom logic and *Augment* peripheral capability in your real-time Controller applications**

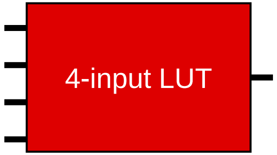
Video Training

- Configurable Logic Block (CLB) introduction
  - https://training.ti.com/c2000-configurable-logic-block-clb-introduction
- CLB architecture
  - https://training.ti.com/c2000-configurable-logic-block-clb-architecture
- CLB Programming Tool
  - https://training.ti.com/c2000-configurable-logic-block-clb-programming-tool

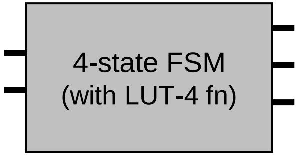
Appendix

Additional CLB Architecture Information

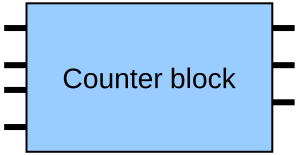
CLB Sub-Modules



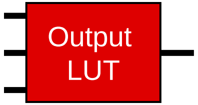
Capable of realizing any combinatorial Boolean equation of up to four inputs.



Configurable either as a single four-state finite state machine, or as two independent two-state finite state machines. The FSM accepts two external inputs, and generates two state outputs and one combinational output. When not used as a state machine, the FSM submodule can accept two external inputs and function as a 4-input LUT.



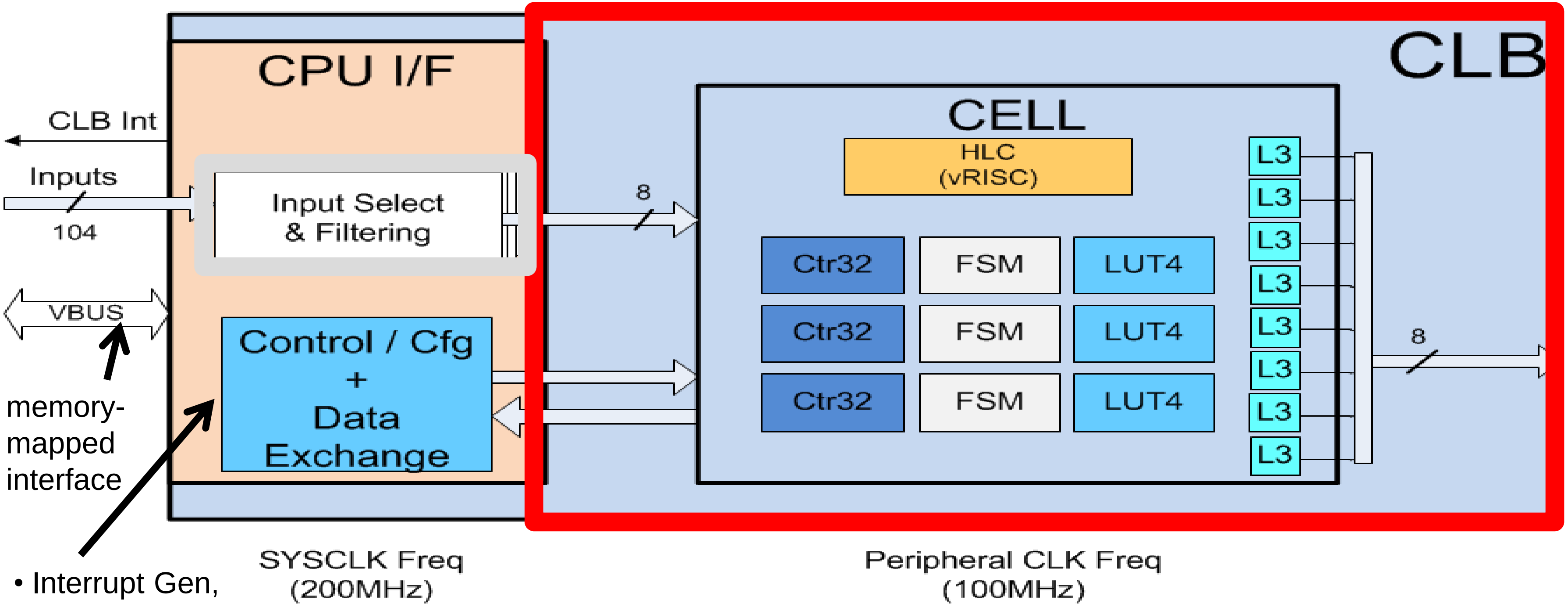
Configurable either as an adder, a counter, or a shifter. As an adder, it can either add or subtract. As a counter, it can count up or count down. As a shifter, it can shift left or right. The counter event inputs, as well as the reset input, may be freely connected to any of the other submodules in the same tile.



Capable of realizing any combinatorial Boolean equation of up to three inputs.

An event-driven block which can handle up to four concurrent events. The event can be an activity on any of the other block outputs. A predefined set of operations is executed when each event occurs. The HLC also provides a data exchange and interrupt mechanism to the CPU subsystem. There are four working registers (R0, R1, R2, and R3) which can be used for basic operations, and to modify or set up values for the three counter blocks.

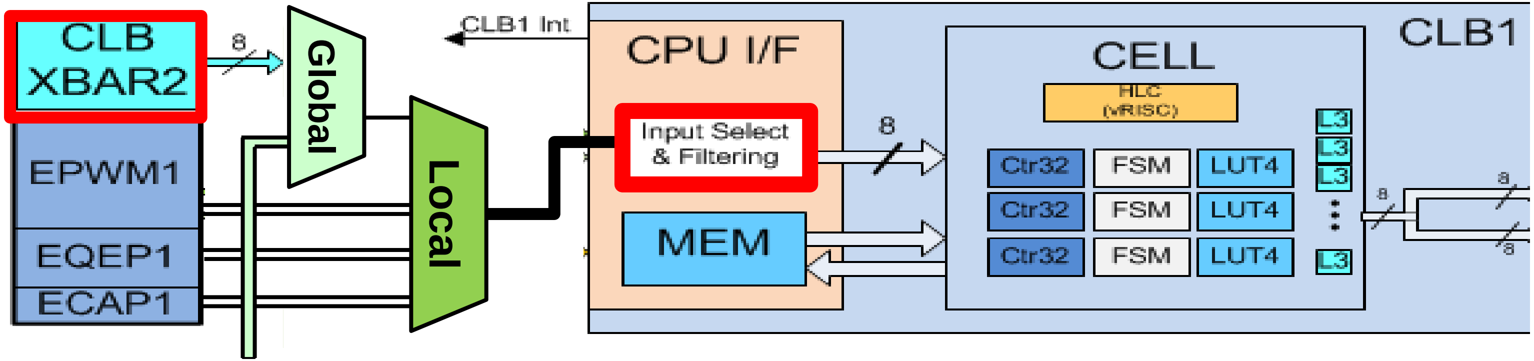
CLB Structure



- Interrupt Gen,
- Data Exchange
- Programmation

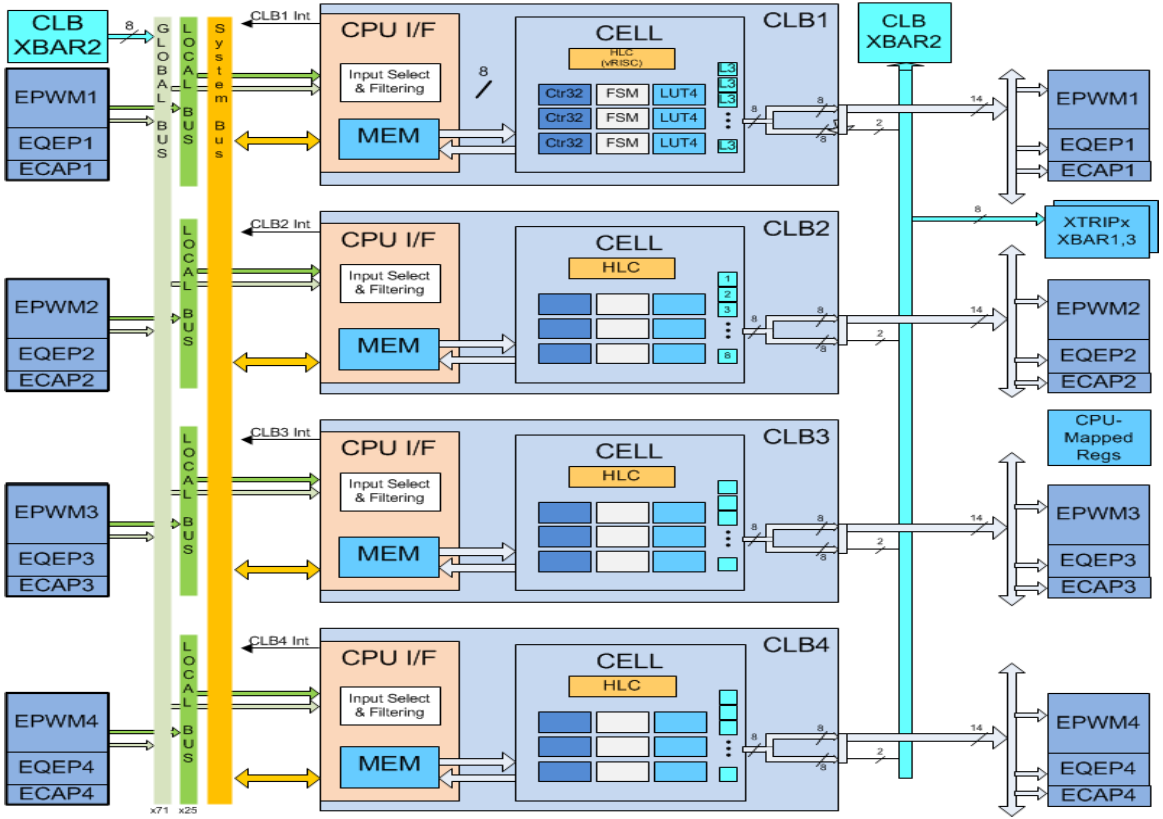
L3: OUTLUT(0-7) (3-inputs, 1-output)
LUT4: LUT(0-2) (4-inputs, 1-output)
FSM: FSM(0-2) Finite state machine
Ctr32: CNT(0-2) Counter module
HLC: High level controller

CLB Input Connections



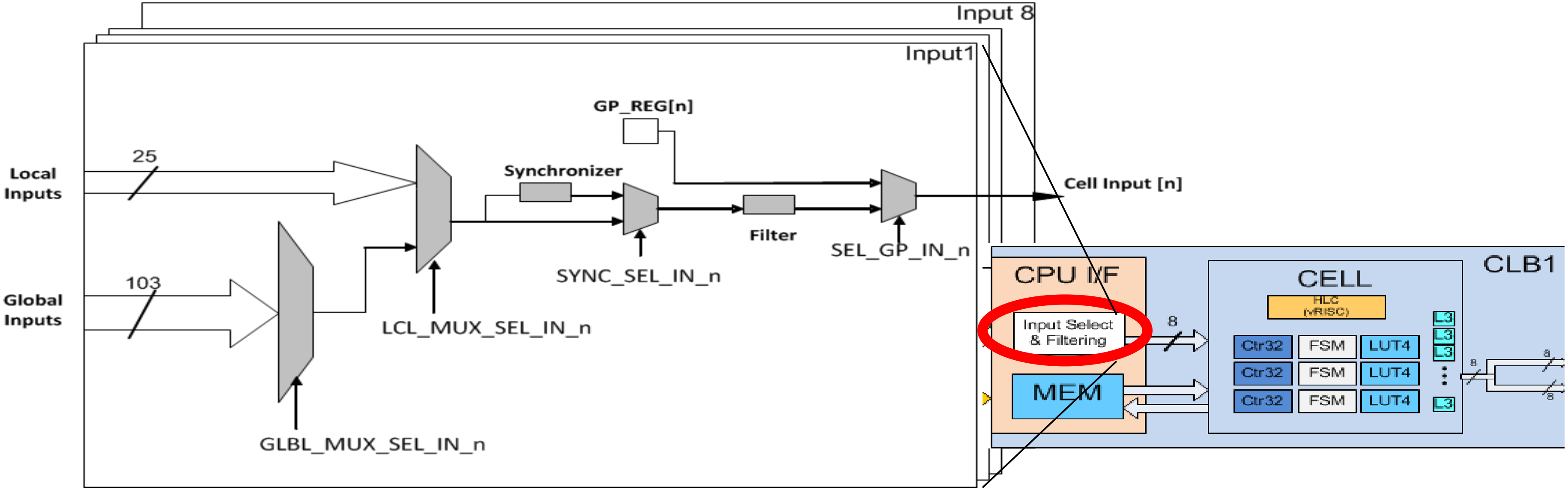
- **The Local Bus** connects 1 CLB to multiple signals within one instance of the ePWM, eQEP, and eCAP blocks.
  - Each CLB is connected to a **unique** peripheral instance.
  - Examples: PWM Digital Compare events, eCAP Capture event inputs, QEP A/B and Index/Strobe outputs.
- **The GLOBAL Bus** connects to ALL CLBs and passes various signals of PWMs 1-4 and the CLB Xbar.
- The Muxes for both buses are contained in the Input Select block, which also contains input latching/filtering functions

CLB Connections



- **Local (input) B**us has dedicated connections to specific instances of EPWM,EQEP,ECAP
- **Output bus** (duplicated from 8 to 16) has direct connections to these same peripherals, plus CLB (system) crossbar
- **Global (input) Bus** has connections to many peripherals, including system crossbars

CLB Inputs: Input Selection & filtering



- Input Multiplexers for Global and Local Buses
- Input synchronizer (Synchronization is required in most cases)
- Input Filtering (Rising edge, Falling edge, Both edge detect logic)
- Software-writable input (GP_Reg[n])

TEXAS INSTRUMENTS

The material is provided strictly "as-is" for informational purposes only and without any warranty.
Use of this material is subject to TI's **Terms of Use**, viewable at TI.com

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), 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, 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 (www.ti.com/legal/termsofsale.html) 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.

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