works with
by SILICON LABS

CMP-103

Maximizing Efficiency and Functionality of MCUs Using **Smart Peripherals**

Ananya Tungaturthi

Associate Product Manager

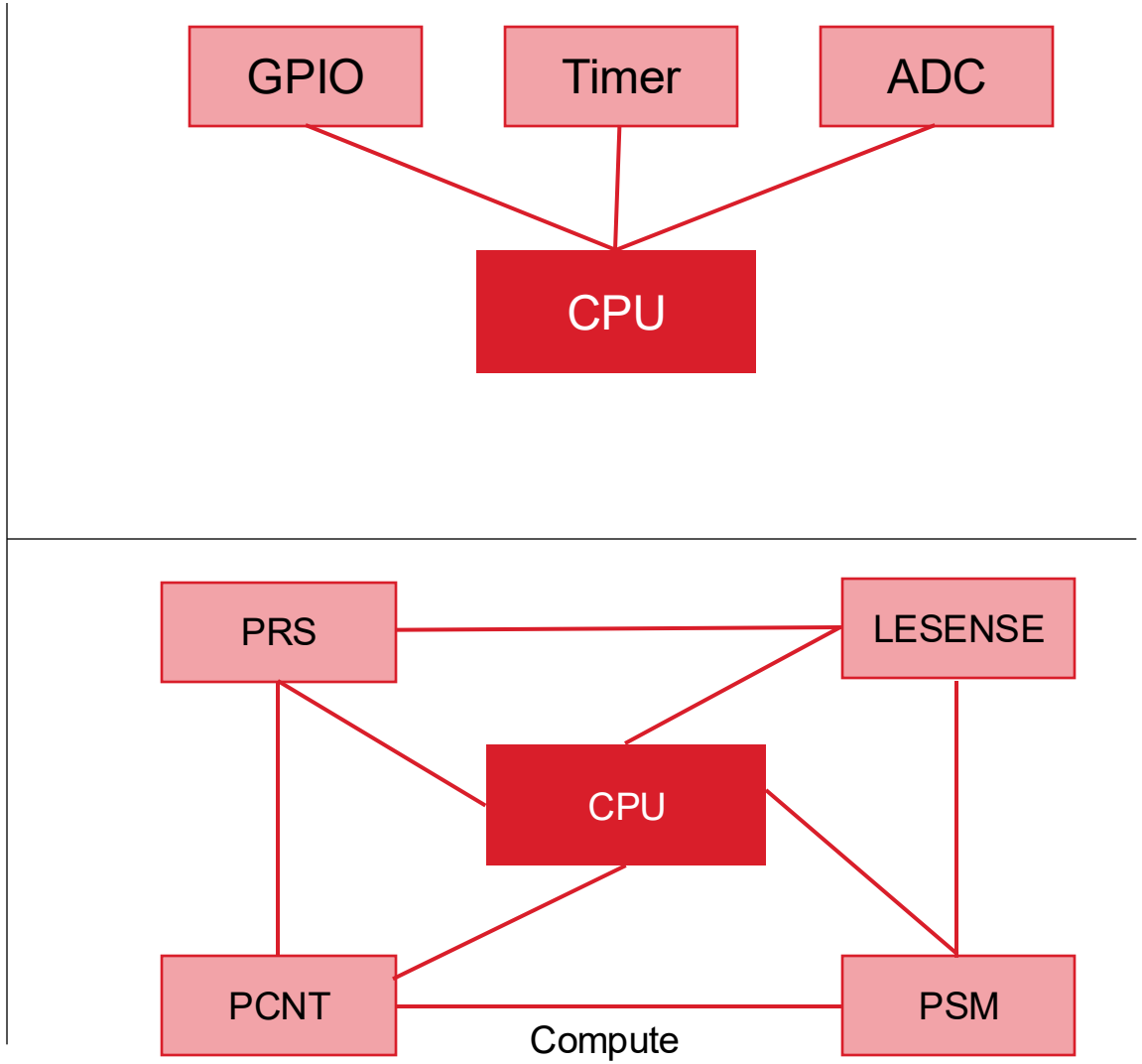
What You'll See in This Session

- **Compute is evolving —** it's now about the intelligence around the CPU, not just inside it.
- **Smart peripherals like PRS, PSM, LESENSE, and PCNT** enable autonomous, low-power, high-function systems.
- **Why this matters to you —** integrating performance, efficiency, and functionality into one platform.

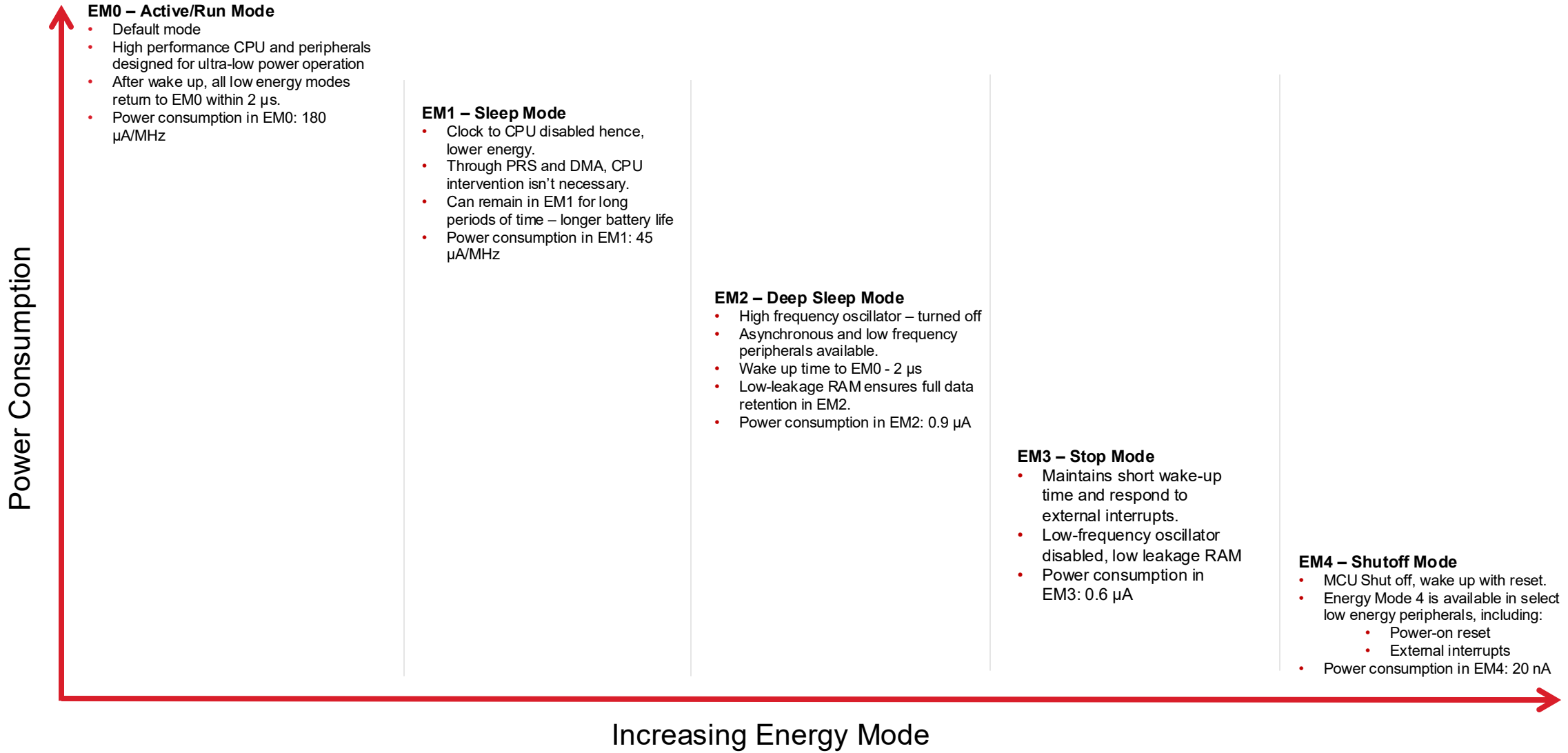
The Big Picture – Compute in the System

- **Legacy View**
  - MCU as a central controller as every event, sensor, or signal wakes the CPU.
- **Today's Reality**
  - CPU becomes an orchestrator when smart peripherals handle real-time tasks autonomously.
- **Why This Shift Matters:**
  - Enables true always-on systems
  - Extends battery life by minimizing active CPU time
  - Simplifies software by offloading low-level interactions

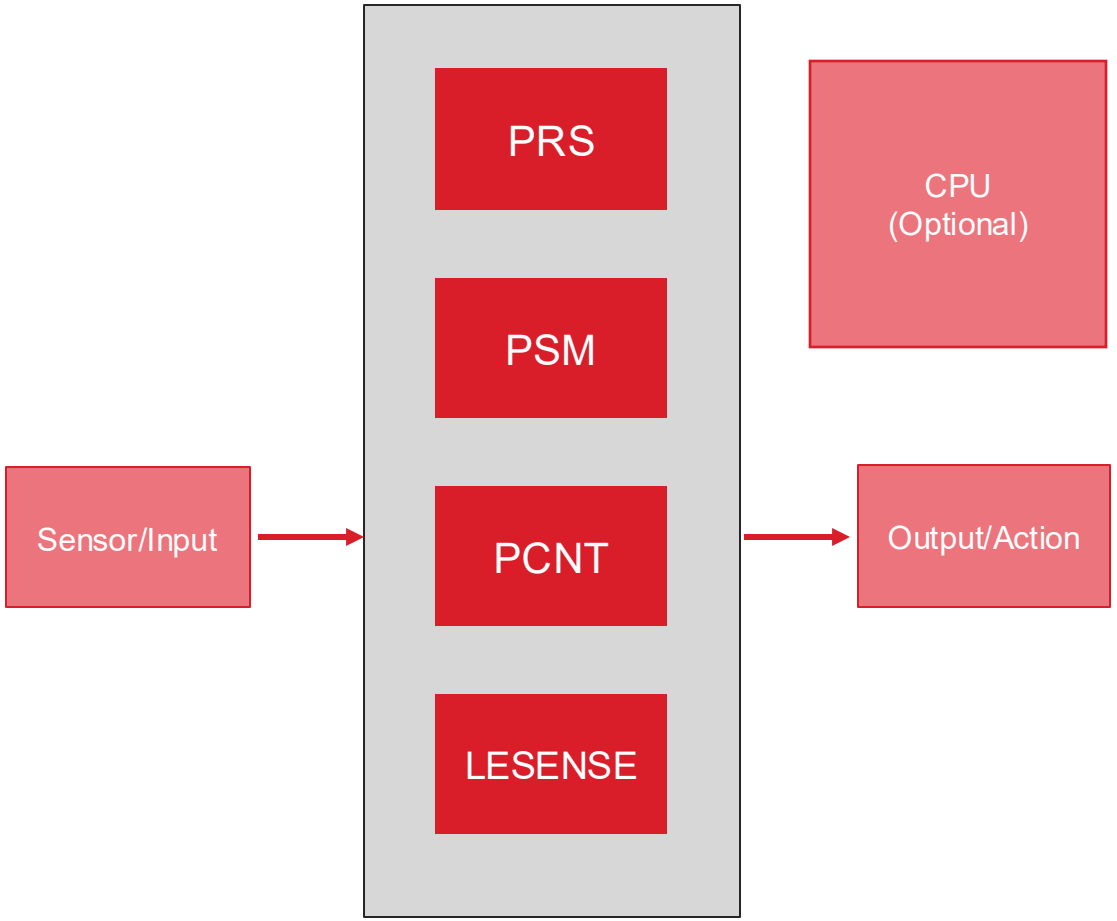
- **Think beyond CPU cycles. It's time to design around capability distribution across subsystems.**



Energy Modes

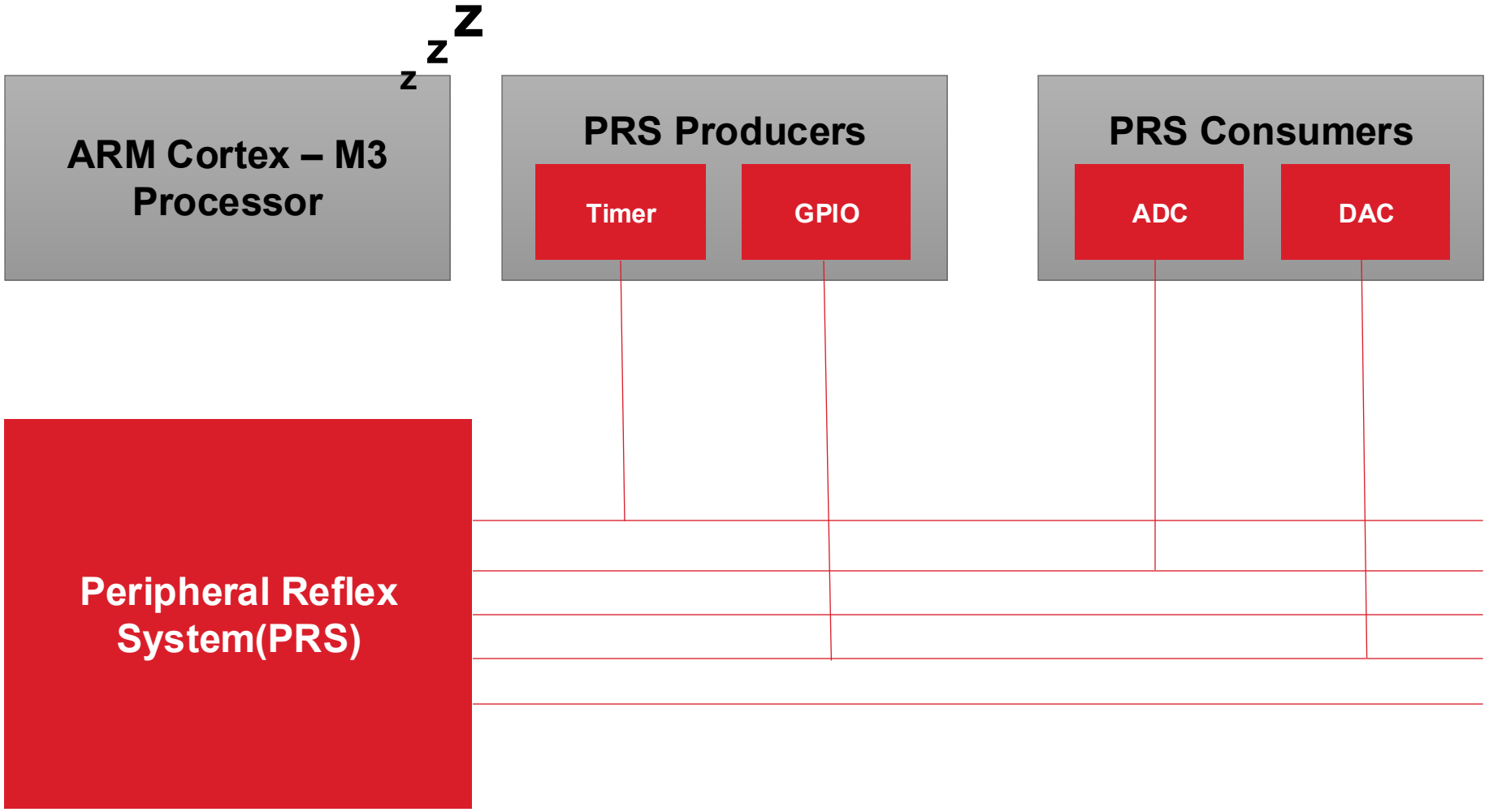


Smart Peripherals: Enable Compute and Efficiency



- **Smart Peripherals**
  - Hardware modules capable of autonomous, context-aware and critical time operations without CPU involvement.
- **Key Characteristics**
  - Event-driven and deterministic
  - Bypass CPU for faster and more efficient operation
  - Operate in low energy modes – EM2, EM3
- **What we'll cover:**
  - Peripheral Reflex system(PRS)
  - Preamble Sense Mode(PSM)
  - Pulse counter(PCNT)
  - Low energy sensor interface(LESENSE)

Peripheral Reflex System (PRS)



PRS – Event Routing Without the CPU

WHAT IS PRS?

- The PRS is a signal routing network allowing direct communication between different peripheral modules without involving the CPU.

- Key Components:
  - Producers
  - Channels
  - Consumers

COMPARISON WITH TRADITIONAL SYSTEMS

Scenario	CPU Wakeups	Avg. Current	Latency
Traditional ISR	High	250 µA	~10 µs
PRS-Based Workflow	Near zero	40 µA	<1 µs

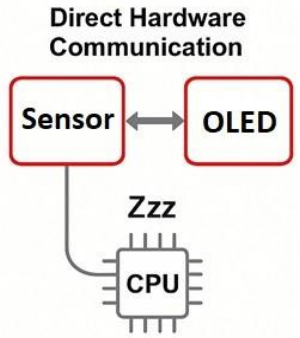
USE CASE

A temperature sensor needs to sample and store values periodically.

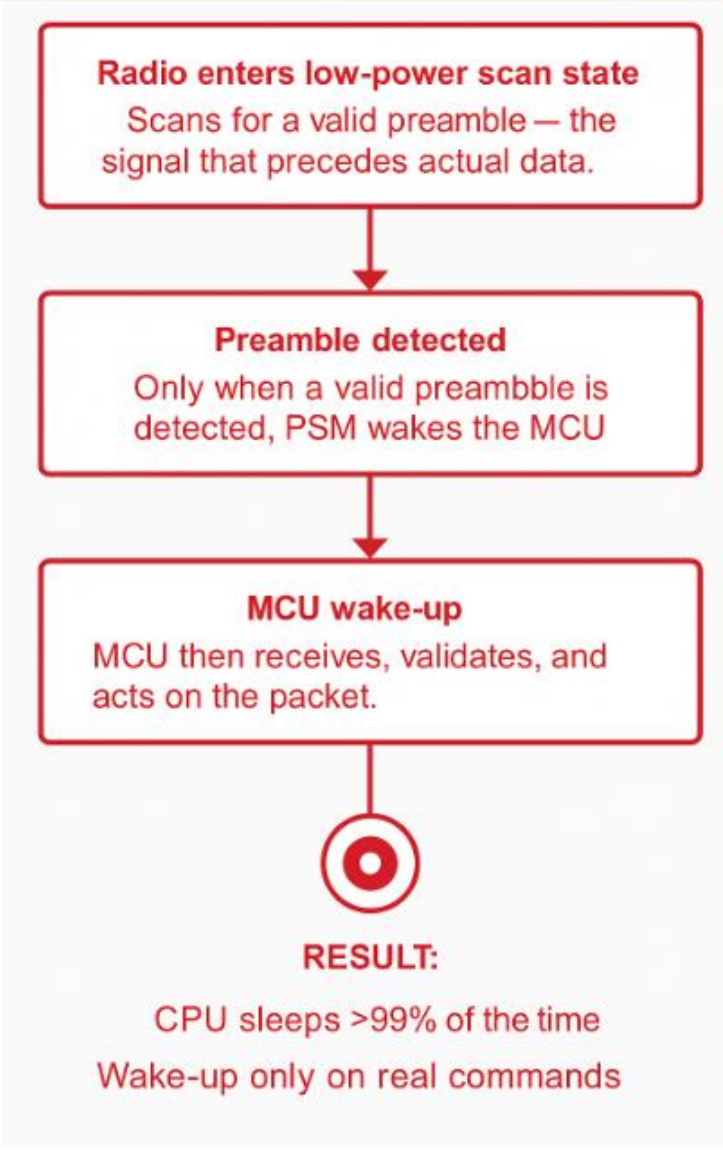
- PRS triggers ADC conversion every second from a Timer
- ADC finishes sampling
- DMA writes the sample to memory
- After several samples, CPU wakes up to process the batch.
- PCNT counts, CPU wake up at threshold.

Benefits:

- CPU sleeps >95% of the time
- No interrupt storm
- Super responsive and energy-efficient



PSM – Wireless Event Triggering Without CPU



WHAT IS PSM?

- **Preamble Sense Mode (PSM)** is a low-power radio mode that listens for incoming wireless activity and detects a valid preamble without waking up the CPU.
- Helps devices stay in deep sleep until they detect *relevant* wireless activity.

- **Why it matters:**
  - Reduces idle energy consumption.
  - Avoids waking up on irrelevant RF noise.
  - Ideal for event-based wireless systems.

SMART HOME USE CASE

Your smart door lock should only wake up when a wireless unlock command is sent — not every time there's RF noise in the air.

Benefits:

- Ultra-low standby power
- No polling or manual wake-ups
- Fast and selective wake-up on meaningful wireless activity

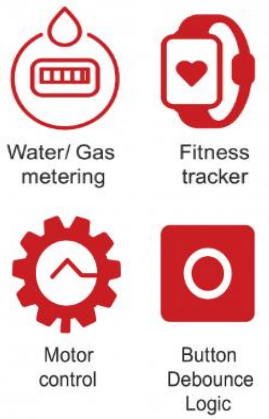
PCNT – Event Counting Without the CPU

WHAT IS PCNT?

- **PCNT(Pulse Counter)** is a hardware peripheral that counts pulses(rising/falling edges) without waking the CPU.
- Can be used for counting incoming pulses on a single input or to decode quadrature encoded inputs.
- Can track GPIO transitions, encoder signals, sensor events or mechanical movement.
- **Key Components:**
  - Input Source: GPIO / Timer / Sensor
  - Counter Logic: Up/down counting, thresholds
  - Wake-up Trigger at threshold: CPU or PRS

NOTABLE FEATURES

Mode Name	What It Does
OVSSINGLE	Counts when changes are noticed by the chip's timer
EXTCLKSINGLE	Counts only when something taps it externally
EXTCLKQUAD	Counts movement AND direction (left/right)



SMART METERING USE CASE

- **Problem:**
  - You want to count every rotation of a water meter wheel, but waking the CPU for each tick wastes power.
- **PCNT-Based Solution:**
  - Sensor sends pulses to PCNT
  - PCNT counts in hardware, even during sleep
  - CPU wakes only when total crosses a threshold (e.g. 100 rotations)
- **Benefits:**
  - CPU sleeps most of the time
  - Highly energy-efficient
  - Accurate pulse capture at high frequency

LESENSE – Autonomous Sensor Monitoring Without CPU

WHAT IS LESENSE?

- **LESENSE is a hardware engine that can monitor capacitive, inductive, or resistive sensors — all while the CPU stays asleep.**
- **Utilizes other on-chip peripherals to perform measurement of a configurable set of sensors.**
- **It works entirely in low-energy modes (EM2/EM3).**
- **Why It Matters:**
  - Keeps power consumption ultra-low
  - Monitors environmental changes or user interaction
  - Only wakes the CPU when something meaningful happens (like a touch, proximity, or light change)

KEY FEATURES

- Operates in EM2/EM3: All sensor monitoring happens while the MCU is in deep sleep
- Supports up to 16 input channels with multiplexed scanning
- Uses on-chip DACs, comparators, and timers — no external components required
- Threshold-based detection logic triggers PRS signals or CPU interrupts
- Fully configurable scan period, duty cycle, and evaluation criteria
- Seamless integration with GPIO, ADC, and other low-energy peripherals

USE CASE

You want your device to wake up when someone touches it but you don't want the CPU to keep checking the touch input every second.

LESENSE-Based Solution:

- LESENSE periodically scans the touch pad
- Compares readings to a stored threshold
- If touch is detected, triggers PRS or wakes the CPU
- Otherwise, system stays asleep

Benefits:

- No polling or timer-driven wake-ups
- CPU sleeps 99%+ of the time
- Super efficient standby power

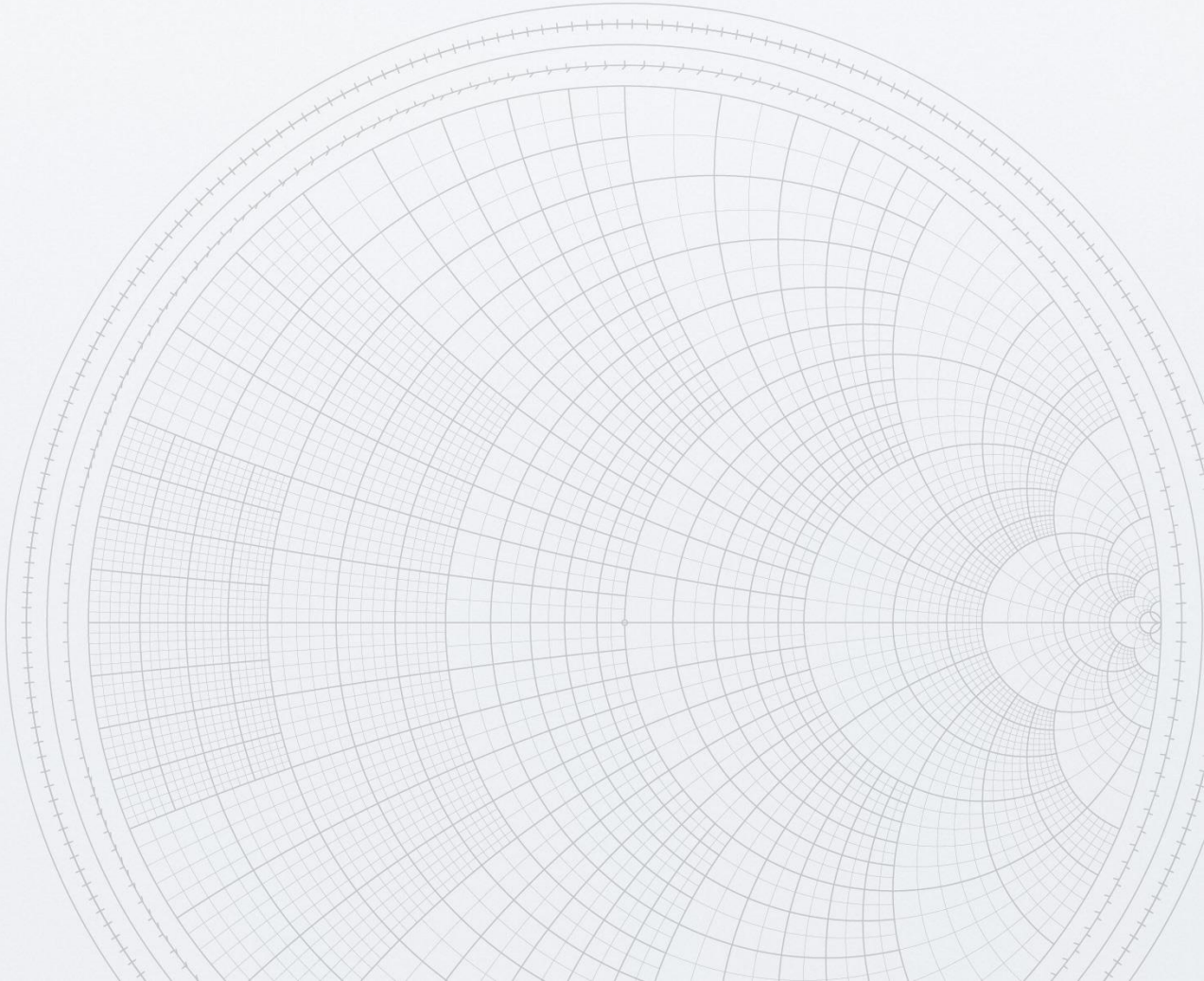
Why This Matters to You

- **Up to 90% CPU offload**
  - CPU stays in sleep modes(EM2/EM3)
  - PRS routes events directly between peripherals — no CPU wake needed for sensor-to-peripheral routing
- **2-3x improvement in battery life**
  - **Sub-µA standby current** with **LESENSE + PCNT** tracking events in sleep
- **Sub-ms wake-up latency**
  - **<1 µs event response** via hardware-triggered action (e.g. PRS → TIMER, LESENSE → GPIO toggle)
- **Leaner firmware – less RTOS, fewer polling loops**
  - **Hardware-driven logic reduces code size** and RTOS needs — simplifies certification & testing
- **Enables scalable ultra-low power context aware design**
  - Reusable peripheral configurations (PRS/LESENSE tuning via Simplicity Studio) enable fast reuse across SKUs

Tools That Support This:

- **Simplicity Studio** – configure PRS, LESENSE, PCNT visually
- **Energy Profiler** – quantify system-level savings in real time
- **Secure Vault** – ensures that all compute + peripheral interactions are protected by hardware-rooted security

PRS Demo



SILICON LABS
CONNECTED INTELLIGENCE

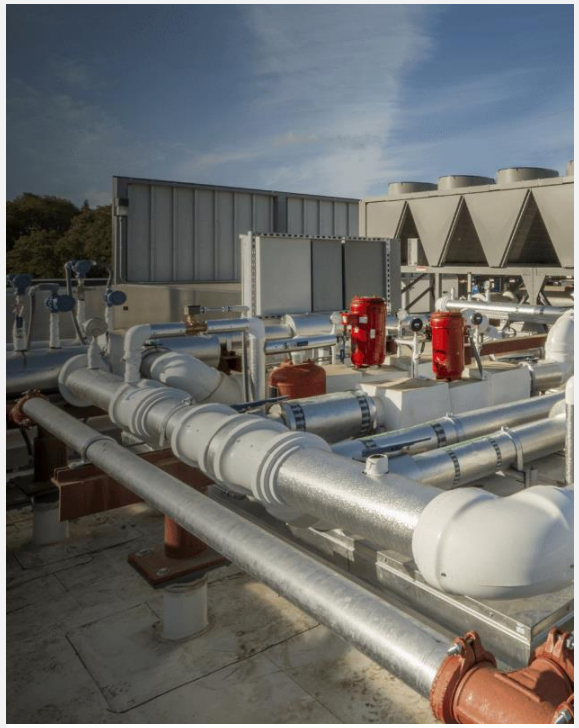
Applications Where Compute Meets Context

SMART AGRICULTURE

- **Uses LESENSE and PRS + ADC**

WEARABLES

- **Uses PCNT and LESENSE**



INDUSTRIAL MONITORING

- **Uses PCNT and PRS**

SMART HOME

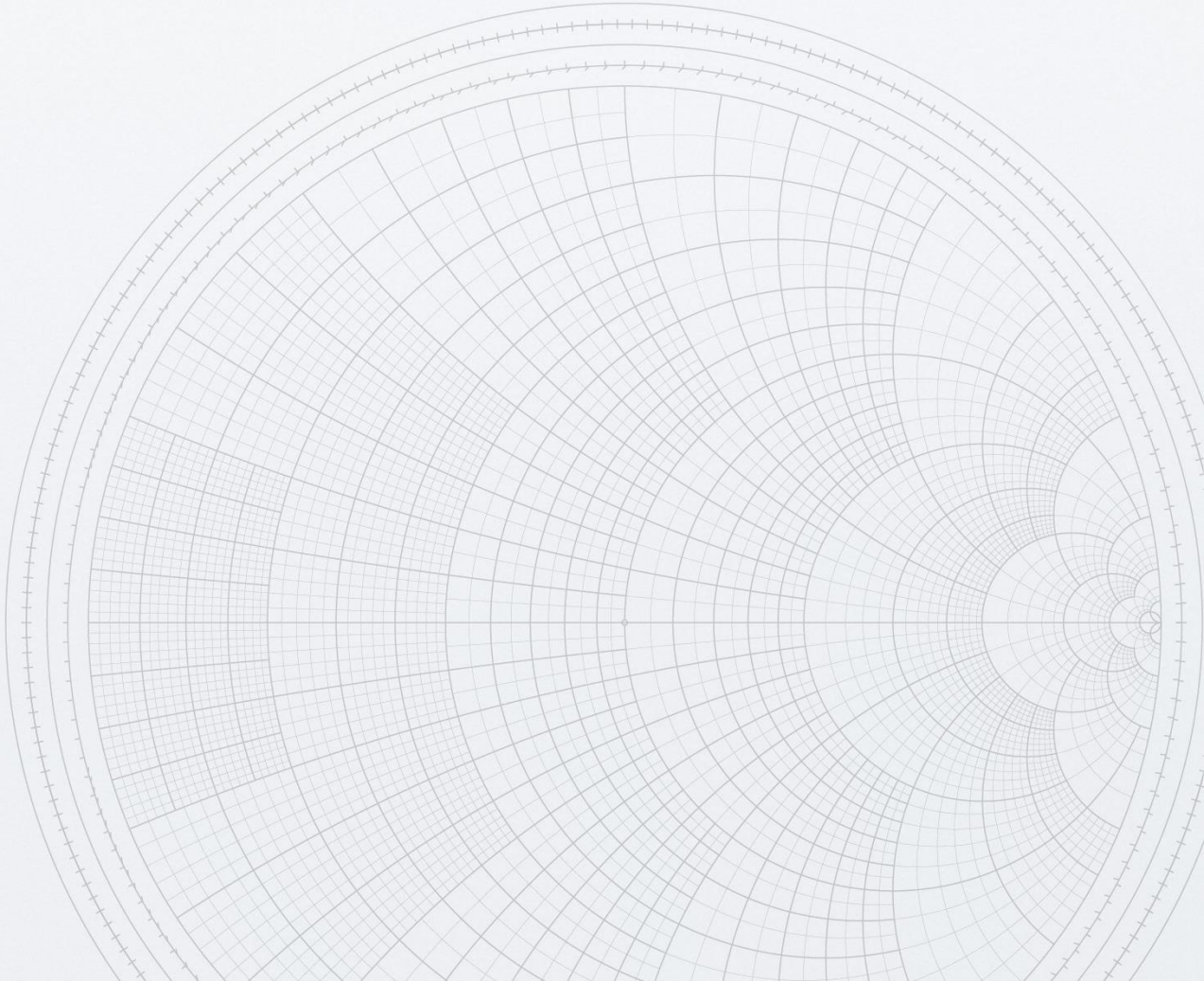
- **Uses PSM and PRS**

works with by SILICON LABS

Key Takeaways

- Offload to peripherals. Let the CPU sleep.
- PRS, LESENSE, PCNT, PSM = smarter event handling.
- Lower power, faster response, simpler code.
- Scalable architecture for better products.
- Compute isn't just the CPU — it's the system.

Q&A



SILICON LABS
CONNECTED INTELLIGENCE