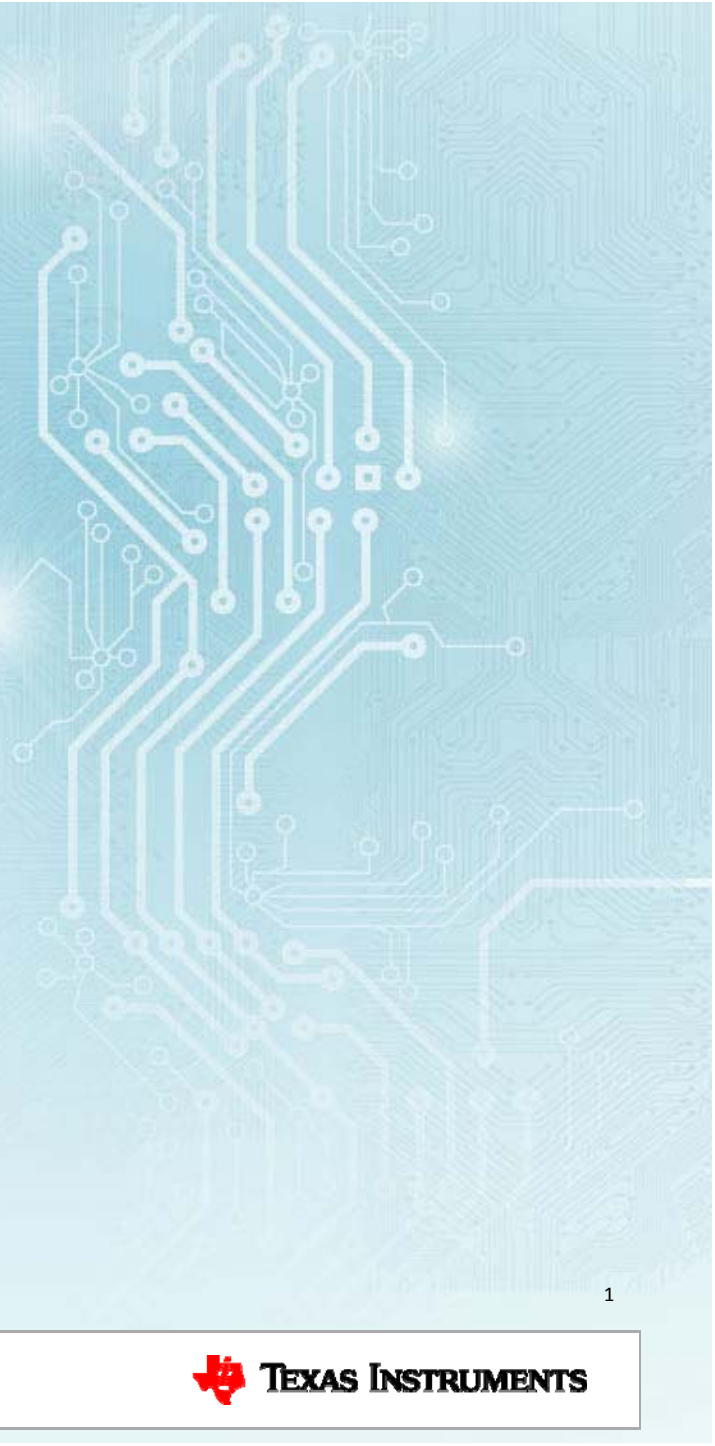
MSP432™ MCUs



Agenda

- MSP432 Overview
- Cortex M4F Core
- Power System
- Clock System & Memory
- Digital Peripherals
- Analog Peripherals
- Software

MSP432 Overview

MSP Offers a Complete Selection of Low-Power Microcontrollers

MSP Low-Power MCUs

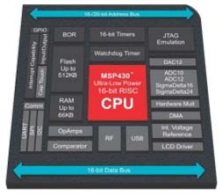
Ultra-Low-Power MCUs	Low-Power + Performance MCUs	Security + Communications MCUs
The World's lowest power MCUs MSP430FRxx	MCUs with expanded MHz and integration MSP430F5x/6x, MSP432	MCUs with integrated RF and more! RF430

MSP430™ 16-bit ULP MCUs

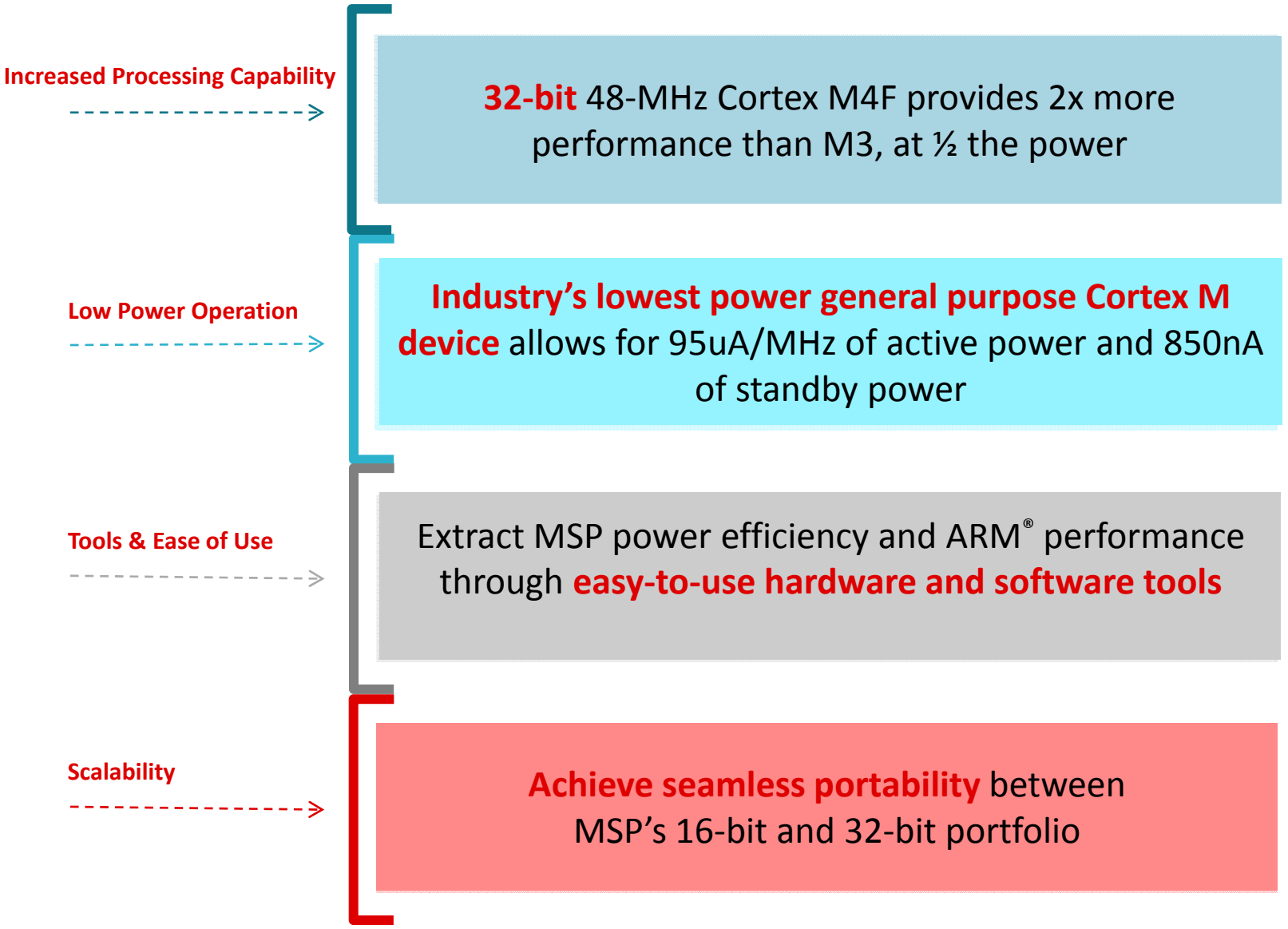
MSP 16/32-bit LP + P MCUs

RF430™ 16/32-bit MCUs

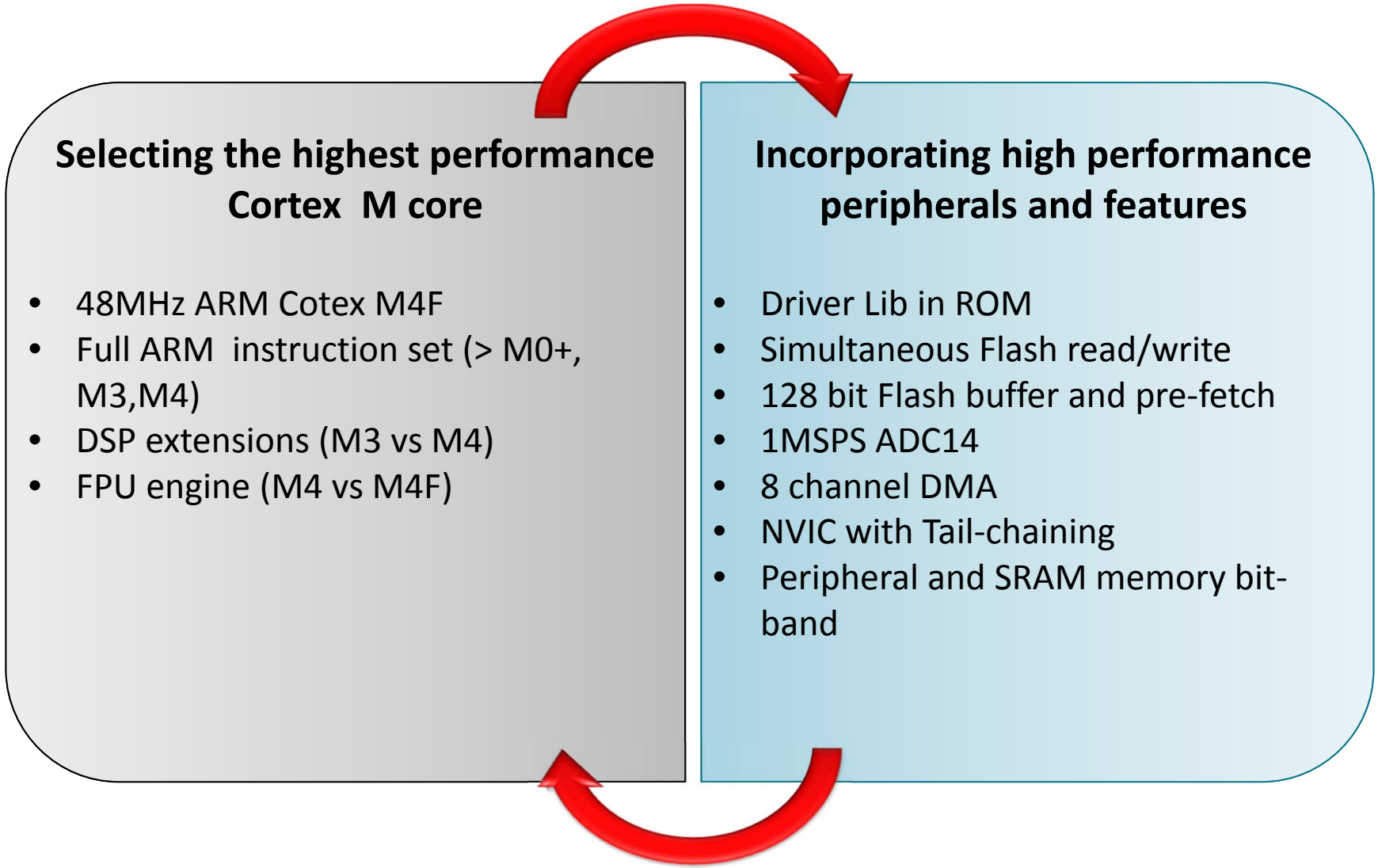
The Lowest Power — **More Integration** — **Embedded RF**



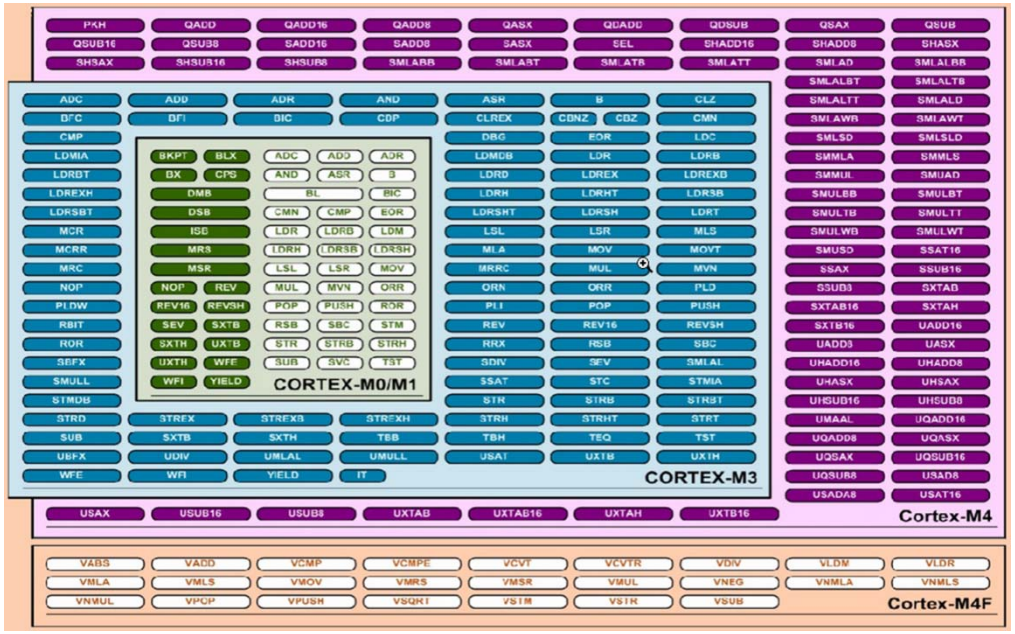
Top Customer Problems… Solved!



Increased processing capability



ARM Cortex M4F Advantages



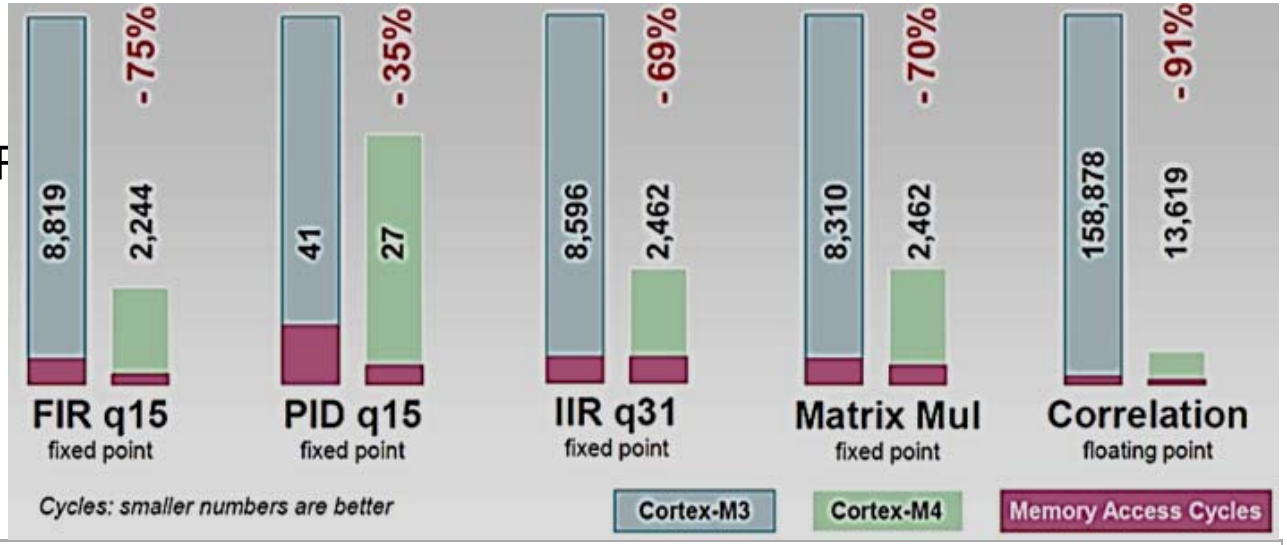
Cortex M3 vs. Cortex M4

(*SIMD + FPU*) DSP Library Benchmark:

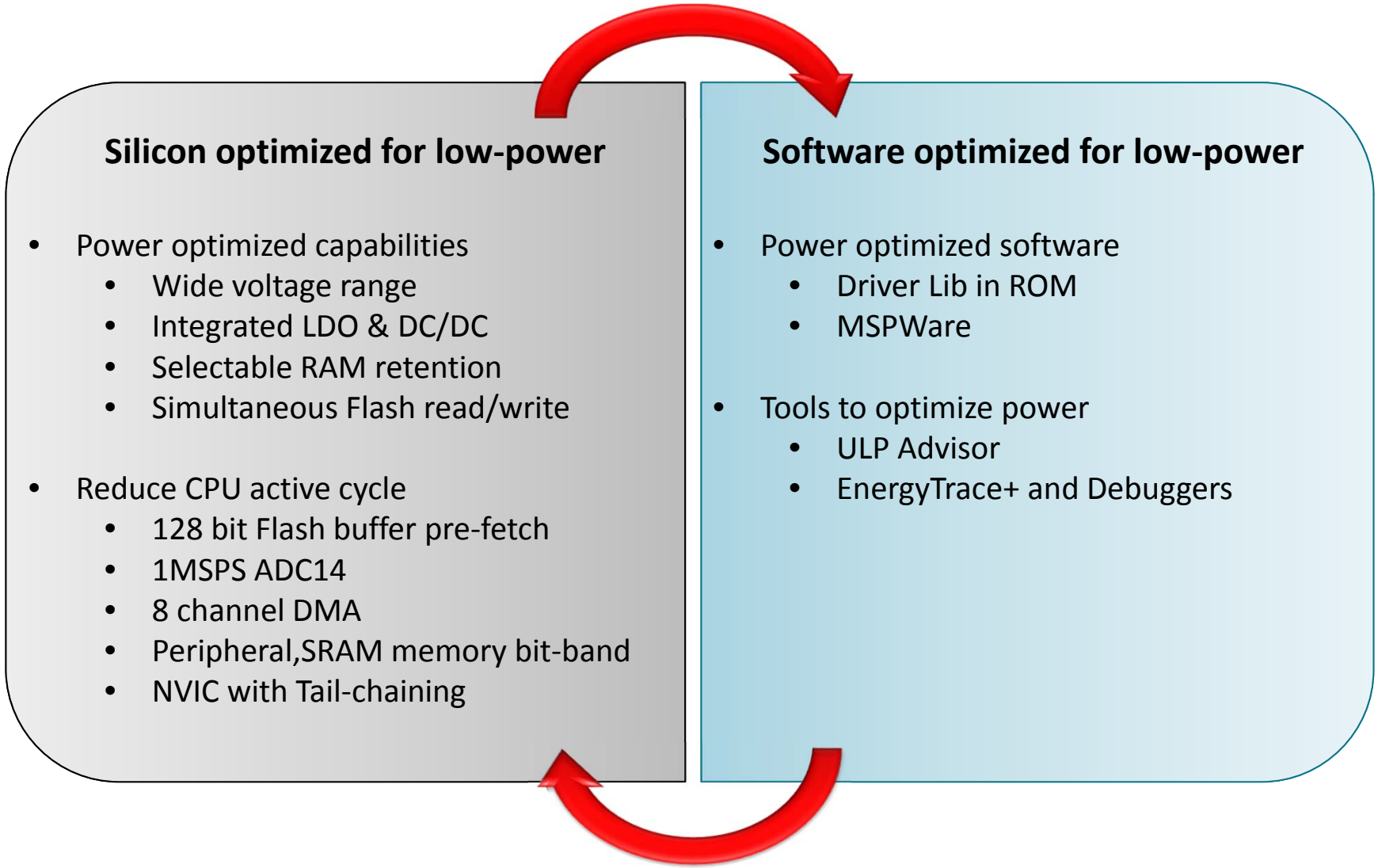
- Fixed-point ~ 2x-4x faster
- Floating-point ~ 10x faster

Source: ARM CMSIS Partner Meeting Embedded World, Reinhard Keil

Utilize the **full ARM instruction set** with M4F

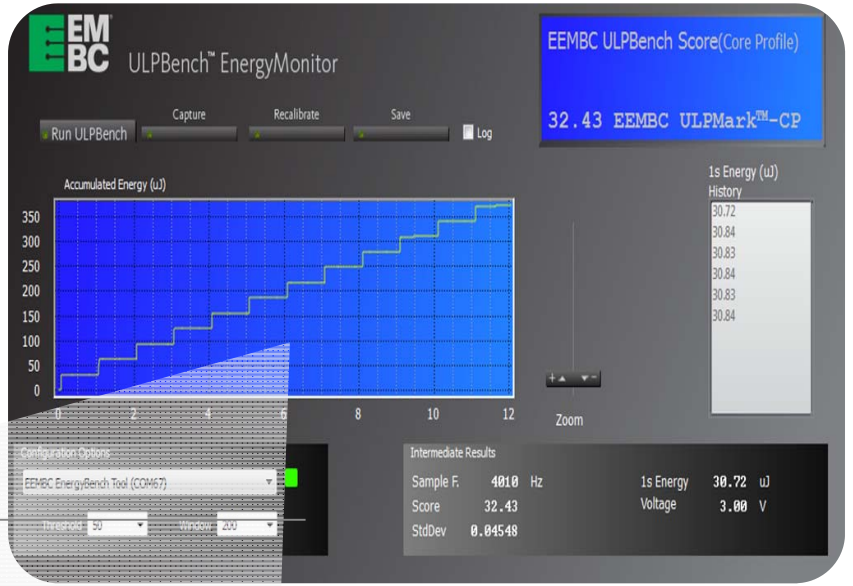


Low power operation

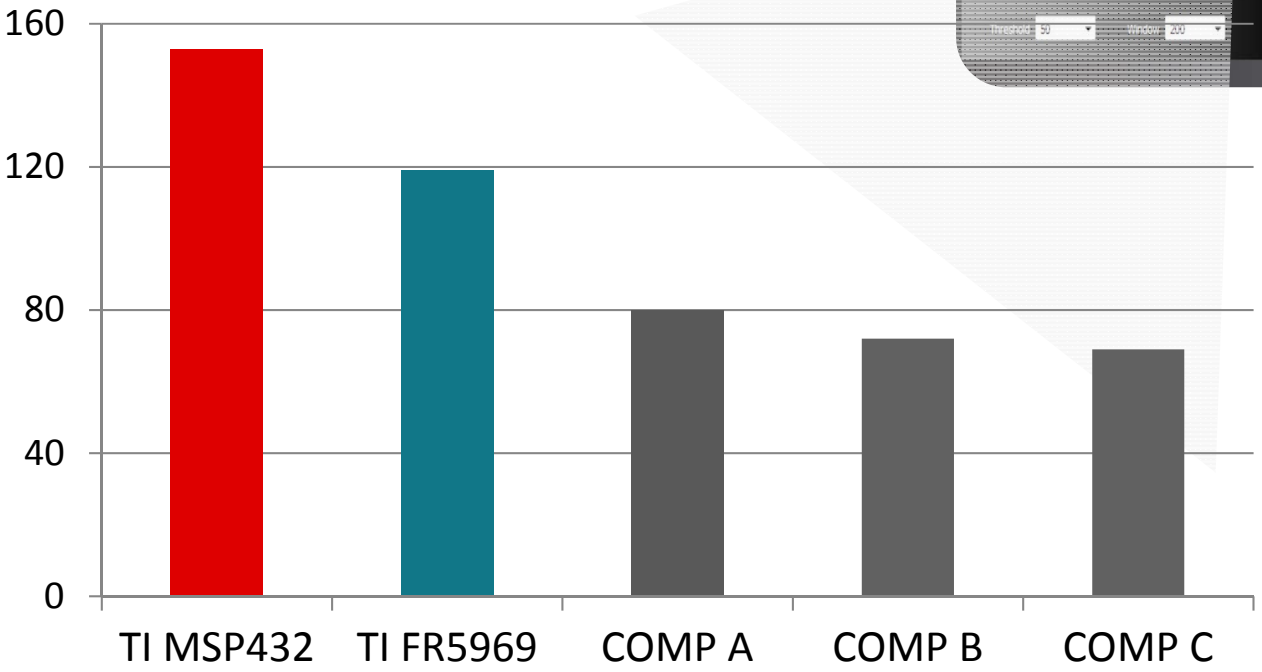


MSP = ULP

EEMBC's ULPBench provides a true comparison of microcontroller current consumption and efficiency



ULPBench Scores



90% better score than our closest competitor!

See for yourself at www.eembc.org/ulpbench

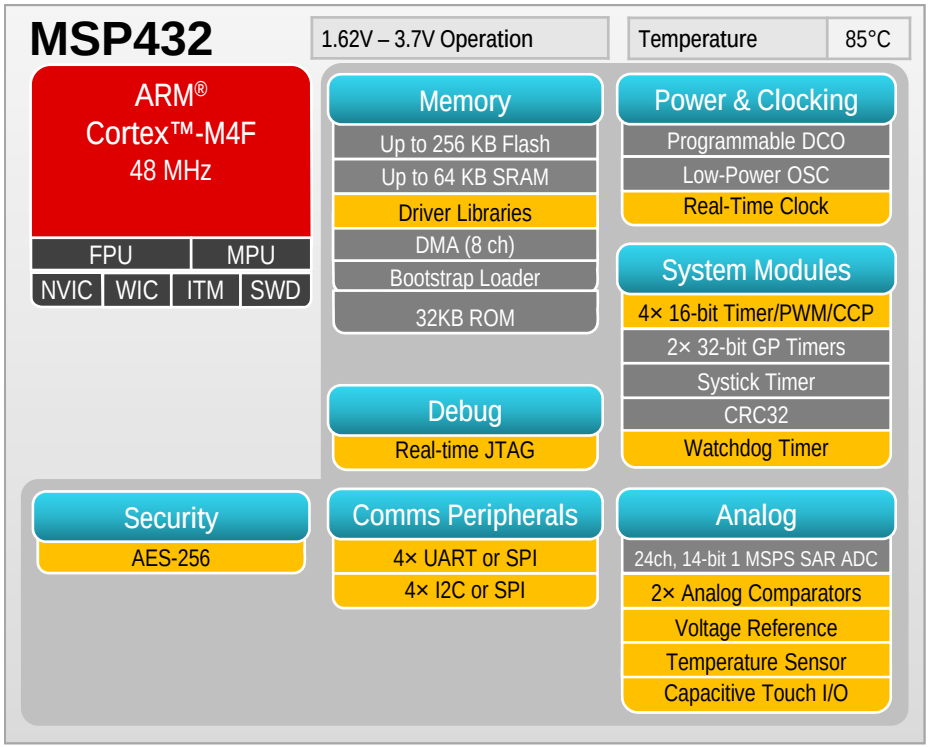
EnergyTrace+™ technology

- Provides a complete ecosystem for real-time power debugging and quickens time to market. Spends less time debugging and more time developing.
- Graphical User Interface in TI's Code Composer Studio integrated development environment (IDE) and IAR Systems' Embedded provides energy profiles of your application
  - Current measurement and CPU states can be tracked over time to help identify power black holes

MSP432™ Microcontrollers

Differentiation

- **Ultra-low standby and active power, and fast wakeup** – 95uA/MHz active, 850nA Standby; Deep sleep to Active: <10us typ
- **Wide supply range** – 1.62-3.7V, including flash operation, enabling multiple battery technologies and eliminating external regulation
- **Integrated high-performance and low-power analog** – Including 1MSPS 14-bit ADC
- **Secure MCU environment** – Flash IP protection & integrated AES-256 encryption
- **Simplified portability from MSP430** - Leverage software & know-how from existing MSP430 designs
  Using 430 Peripherals, Analog & Low Power Modes



Same as MSP430

Kits

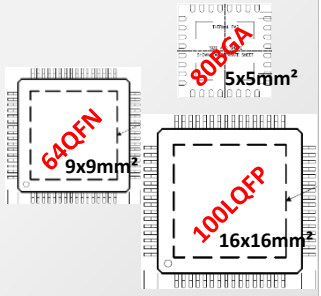
LaunchPad
- Designed for evaluation and initial development
- Includes on-board emulator
- $12.99

Target Board
- Designed for advanced development
- $89

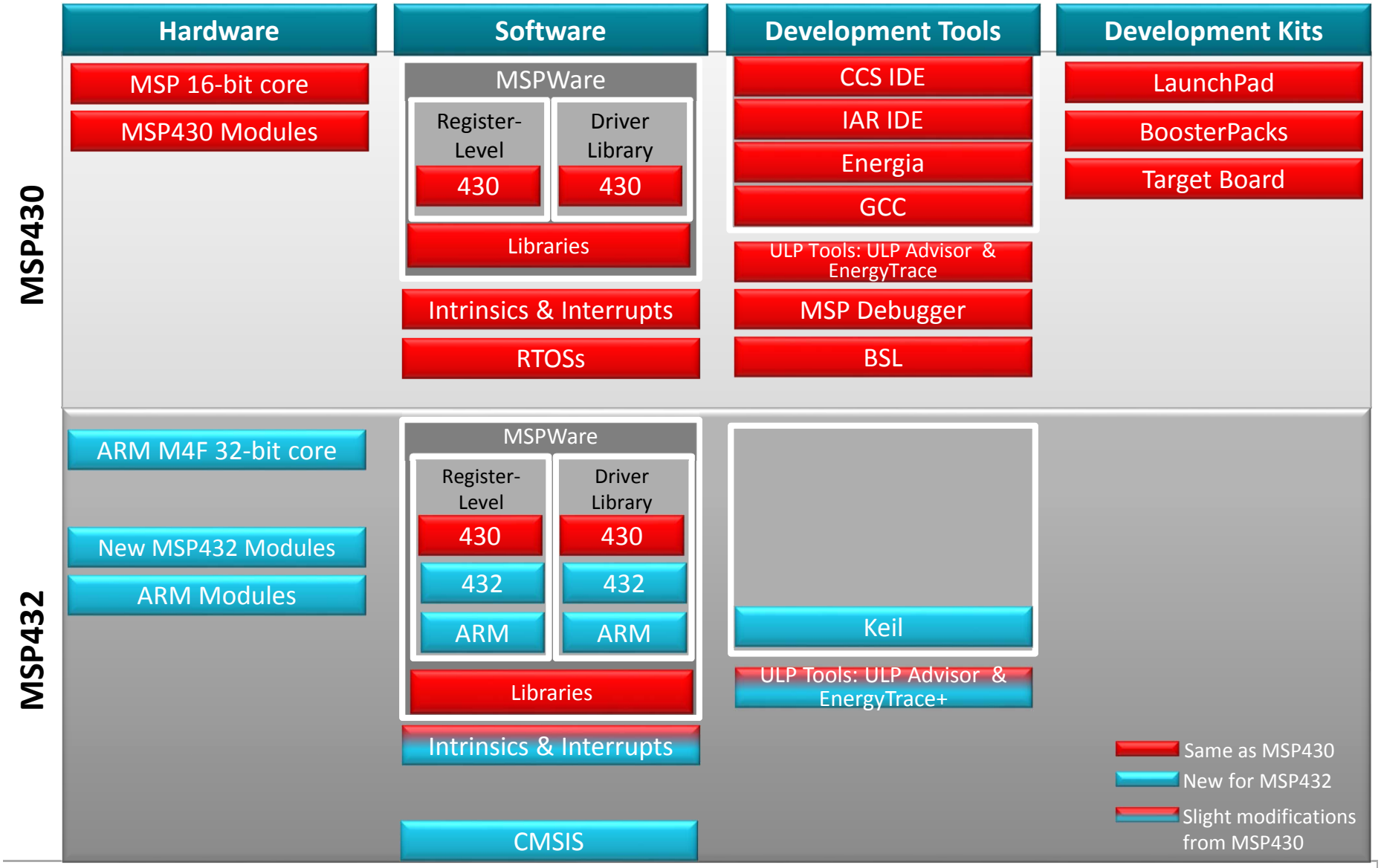
Tools & Software

- **MSPWare –** leverage C-code portable MSP430 peripherals and analog
- **TI RTOS Support**
- **ARM 3rd Party Ecosystem**
- **Code Composer Studio™, IAR, KEIL IDEs, and gcc**

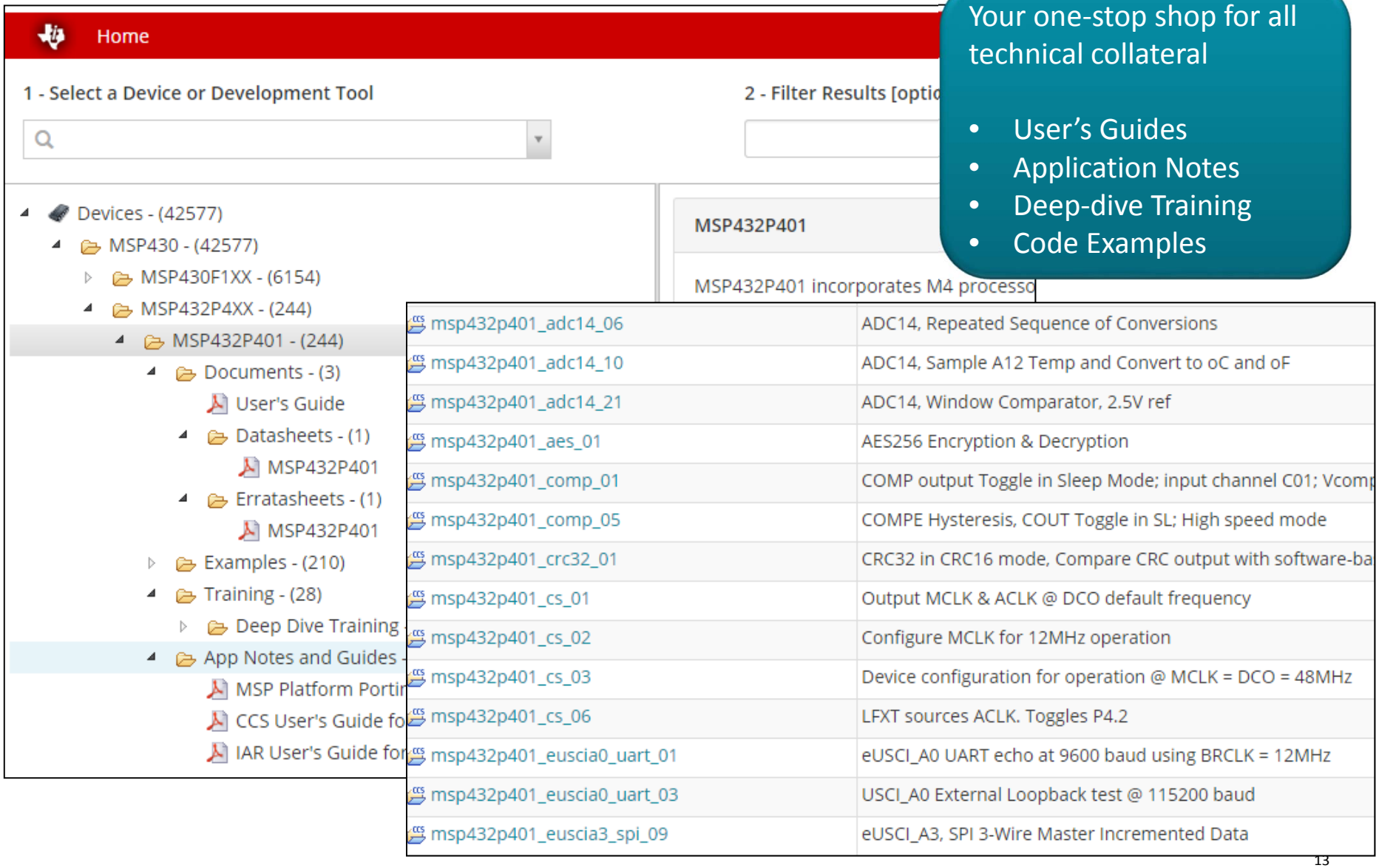
Packages



MSP Platform Portability



It’s all in MSPWare

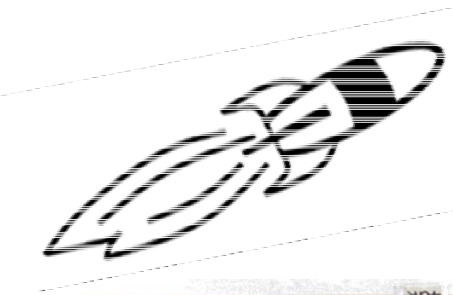


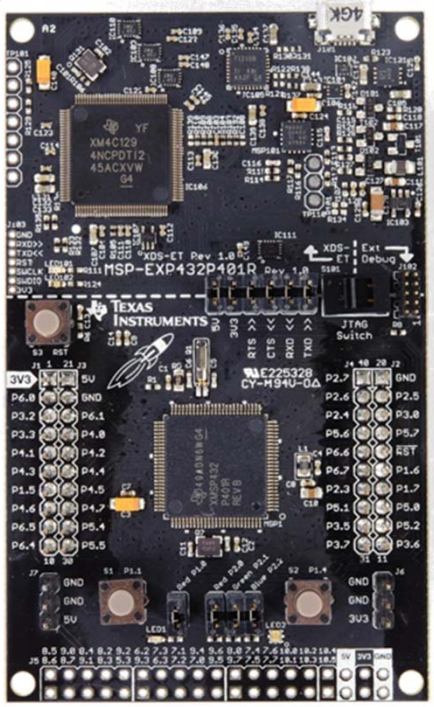
Your one-stop shop for all technical collateral

- User’s Guides
- Application Notes
- Deep-dive Training
- Code Examples

Introducing the MSP432 LaunchPad

Develop high performance applications that benefit from low power operation





MSP-EXP432P401R
$12.99

Features

- Low-power, high performance MSP432P401R MCU
- 40 pin BoosterPack Connector, and support for 20 pin BoosterPacks
- Onboard XDS-110ET emulator featuring EnergyTrace+ Technology
- 2 buttons and 2 LEDs for User Interaction
- Back-channel UART via USB to PC

Kit Includes

- Development board with demo application
- USB cable
- Quick start guide

Software

- MSPWare featuring example code, User's Guides, Application notes, training, and more
- Out-of-box LaunchPad GUI

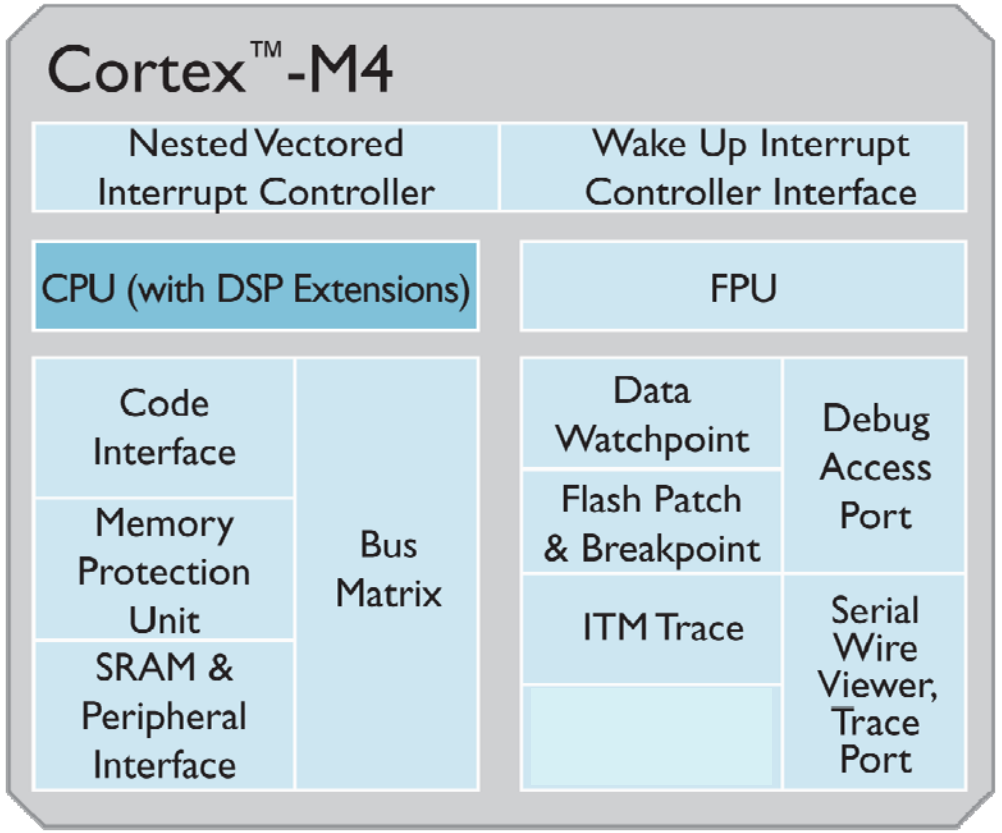
MSP432 device options

Part Number	Flash (KB)	SRAM (KB)	ADC14 Chan	Comp-0 Chan	Comp-1 Chan	Timer A	eUSCI		20mA Drive I/O	Total I/O	Package Type
							Chan A: UART/ IrDA/SPI	Chan B: SPI/I2C			
MSP432P401RIPZ	256	64	24/ext 2/int	8	8	5,5,5,5	4	4	4	84	100 LQFP 16x16mm
MSP432P401MIPZ	128	32	24/ext 2/int	8	8	5,5,5,5	4	4	4	84	100 LQFP 16x16mm
MSP432P401RIZXH	256	64	16/ext 2/int	6	8	5,5,5	3	4	4	64	80 BGA 5x5mm
MSP432P401MIZXH	128	32	16/ext 2/int	6	8	5,5,5	3	4	4	64	80 BGA 5x5mm
MSP432P401RIRGC	256	64	12/ext 2/int	2	4	5,5,5	3	3	4	48	64 QFN 9x9mm
MSP432P401MIRGC	128	32	12/ext 2/int	2	4	5,5,5	3	3	4	48	64 QFN 9x9mm

Cortex M4F Core

MSP432 | 32-bit Cortex-M4F

- 32-bit pipeline architecture
- Cortex-M4 with DSP extension instruction set
- Floating Point Unit
- Standard Cortex-M Debugger Module, Serial Wire Debug, ITM Trace support
- Core modules including DMA, SysTick, & Interrupt (NVIC)

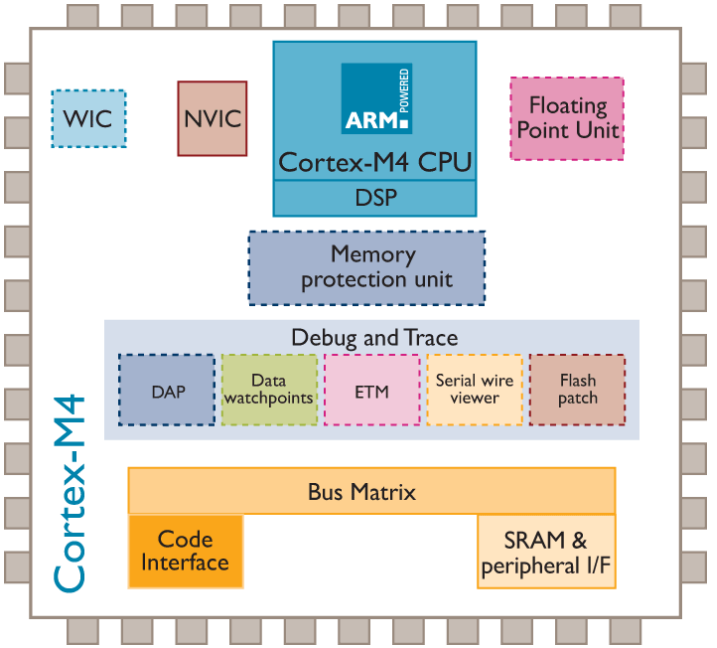


Cortex-M | Core Comparison

Cortex-M	Thumb	Thumb-2	HW MPY	HW DIV	Saturated math	DSP-extensions	FPU	ARM architecture
Cortex-M0	Most	Subset	1 or 32 cycle	No	No	No	No	ARMv6-M Von Neumann
Cortex-M0+	Most	Subset	1 or 32 cycle	No	No	No	No	ARMv6-M Von Neumann
Cortex-M1	Most	Subset	3 or 33 cycle	No	No	No	No	ARMv6-M Von Neumann
Cortex-M3	Entire	Entire	1 cycle	2-12 cycles	Yes	No	No	ARMv7-M Harvard
Cortex-M4	Entire	Entire	1 cycle	2-12 cycles	Yes	Yes	Optional **Yes for MSP432**	ARMv7E-M Harvard

FPU | Floating-Point Unit

- The FPU provides floating-point computation functionality that is compliant with the IEEE 754 standard
- Enables conversions between fixed-point and floating-point data formats, and floating-point constant instructions
- The Cortex-M4F FPU fully supports single-precision:
  - Add
  - Subtract
  - Multiply
  - Divide
  - Single cycle multiply and accumulate (MAC)
  - Square root



MSP432 | CoreMark = 3.41CM/MHz

Core	Device	CoreMark
Cortex-M4	MSP432	3.40.
	Competitor A	3.34 [168MHz-Ohs, --no_size_constraints]
Cortex-M3	Competitor A	2.98
	Competitor B	2.15
Cortex-M0+	Competitor C	2.28
	Competitor B	2.07

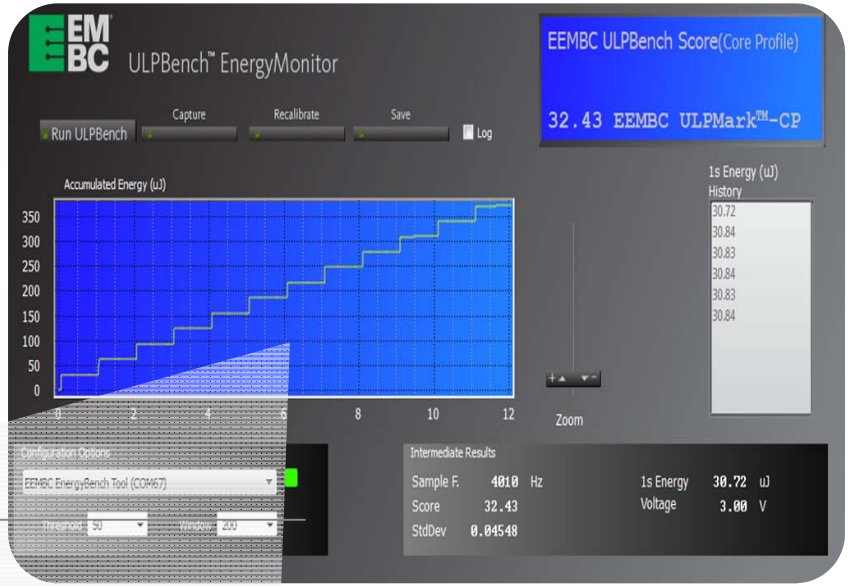
Test Conditions

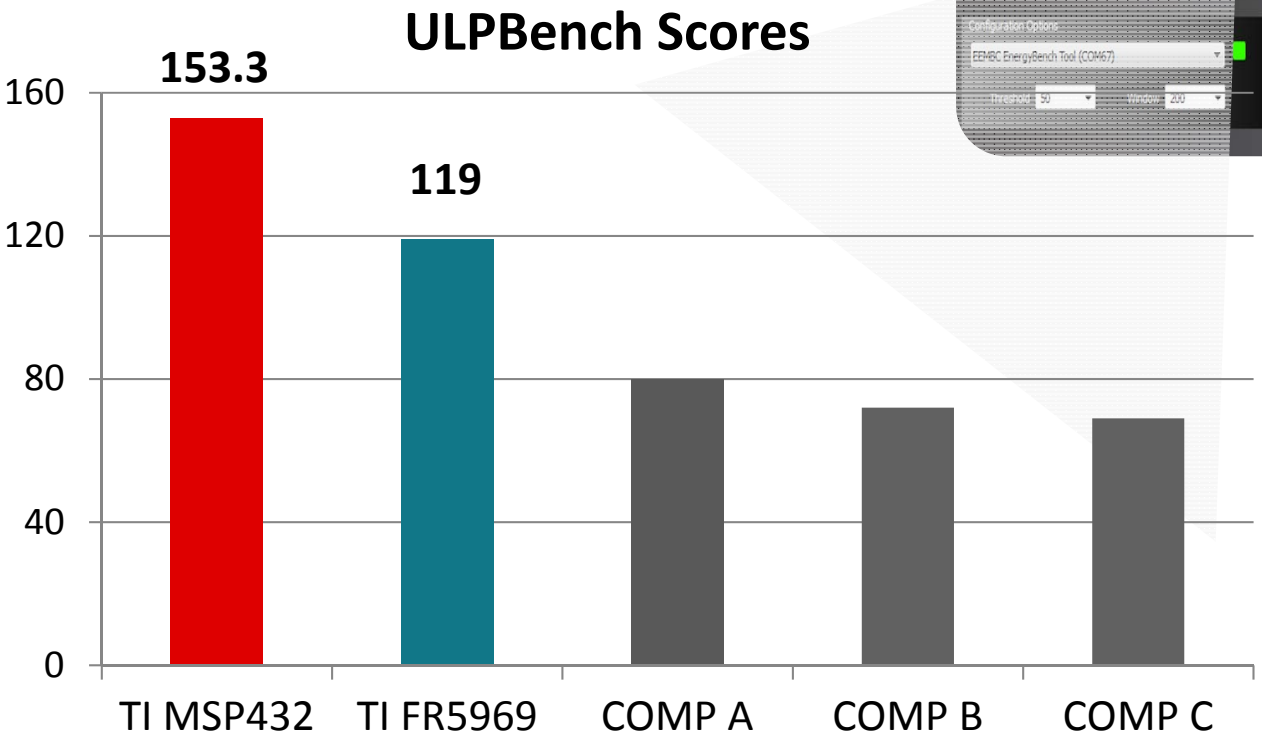
- Number measured with IAR
- 3MHz, Active Mode with VCORE=0

*Source: MSP-Benchmark twiki

ULPBench

EEMBC's ULPBench provides a true comparison of microcontroller current consumption and efficiency

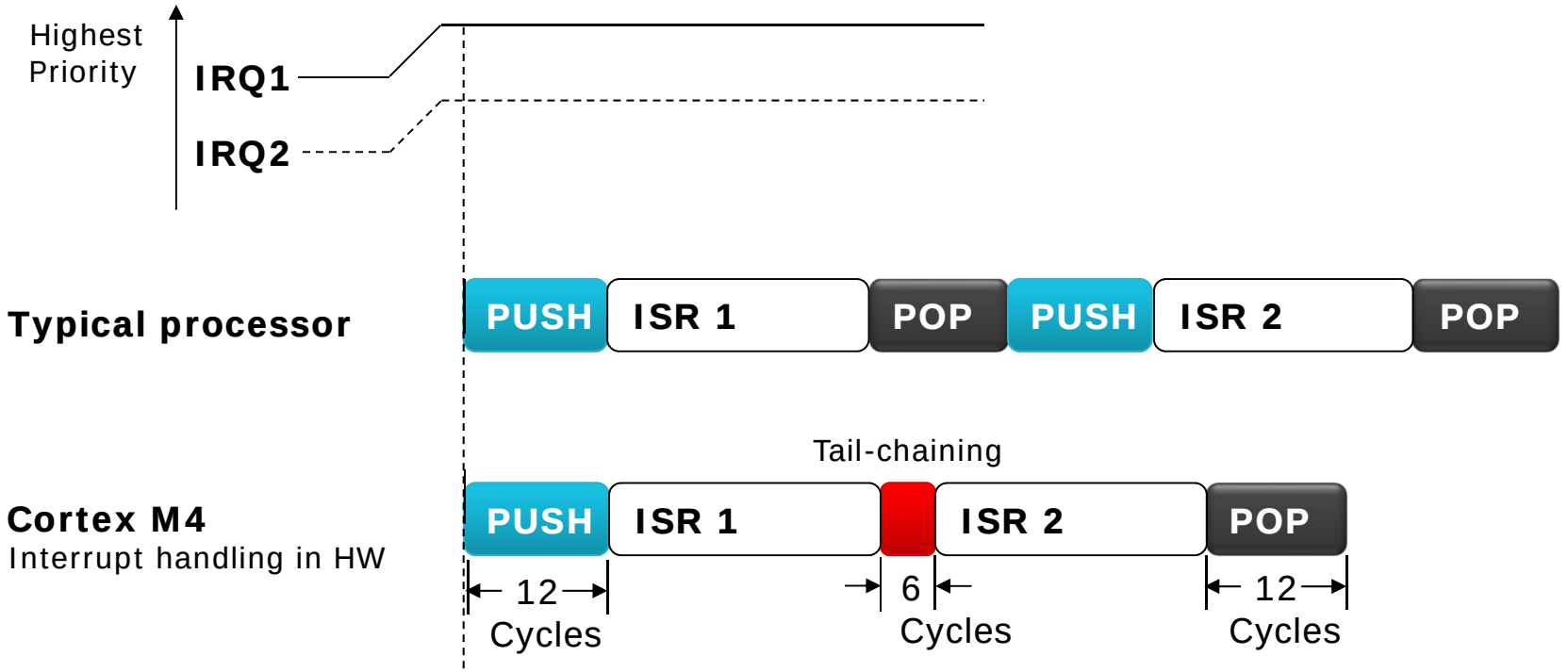




90% better score than our closest competitor!

www.eembc.org/ulpbench

Interrupts | Latency: Tail Chaining



Saving 18 cycles:

24 cycles (POP + PUSH) → 6 cycles (Tail-chaining)

Interrupts | Declaration on MSP432

Option 1: Declare the entire Interrupt Vector table

msp432_startup_ccs.c

```
#pragma DATA_SECTION(interruptVectors, ".intvecs")
void (* const interruptVectors[])(void) =
{
    (void (*)(void))((unsigned long)&__STACK_END),
                                            /* The initial stack pointer */
    resetISR,                               /* The reset handler         */
    nmiISR,                                 /* The NMI handler           */
    faultISR,                               /* The hard fault handler    */
    intDefaultHandler,                      /* The MPU fault handler     */
    intDefaultHandler,                      /* The bus fault handler     */
    intDefaultHandler,                      /* The usage fault handler   */
    0,                                      /* Reserved                  */
    0,                                      /* Reserved                  */
    0,                                      /* Reserved                  */
    0,                                      /* Reserved                  */
    intDefaultHandler,                      /* SVCall handler            */
    intDefaultHandler,                      /* Debug monitor handler     */
    0,                                      /* Reserved                  */
    intDefaultHandler,                      /* The PendSV handler        */
    SysTick_ISR,                            /* The SysTick handler       */
    intDefaultHandler,                      /* PSS ISR                   */
    CS_ISR,                                 /* CS ISR                    */
    PCM_ISR,                                /* PCM ISR                   */
    intDefaultHandler,                      /* WDT ISR                   */
    intDefaultHandler,                      /* FPU ISR                   */
    intDefaultHandler,                      /* FLCTL ISR                 */
    COMP0_ISR,                              /* COMP0 ISR                 */
    intDefaultHandler,                      /* COMP1 ISR                 */
    TA0_0_ISR,                              /* TA0_0 ISR                 */
    intDefaultHandler,                      /* TA0_N ISR                 */
    intDefaultHandler,                      /* TA1_0 ISR                 */
    intDefaultHandler,                      /* TA1_N ISR                 */
    intDefaultHandler,                      /* TA2_0 ISR                 */
    intDefaultHandler,                      /* TA2_N ISR                 */
    intDefaultHandler,                      /* TA3_0 ISR                 */
    intDefaultHandler,                      /* TA3_N ISR                 */
    UART0_ISR,                              /* EUSCIA0 ISR               */
    SPI1_ISR,                               /* EUSCIA1 ISR               */
```

ISR Handlers: stubs defined in table.
Treated as regular function, code in user's application

Option 2: MSP430 method
Use `#pragma vector`

```
#pragma vector = USCI_B0_VECTOR
__interrupt void USCI_B0_ISR(void)
{
  switch(__even_in_range(UCB0IV,12))
  {
  case  0: break;
  ……
}

//All unused interrupts trapped
#pragma vector = unused_interrupts
__interrupt void intDefaultHandler(void)
{
   //trap
}
```

1. MSP430 code re-use
2. Interrupt vector & handler function **defined together**

Power System

Power | Feature Overview

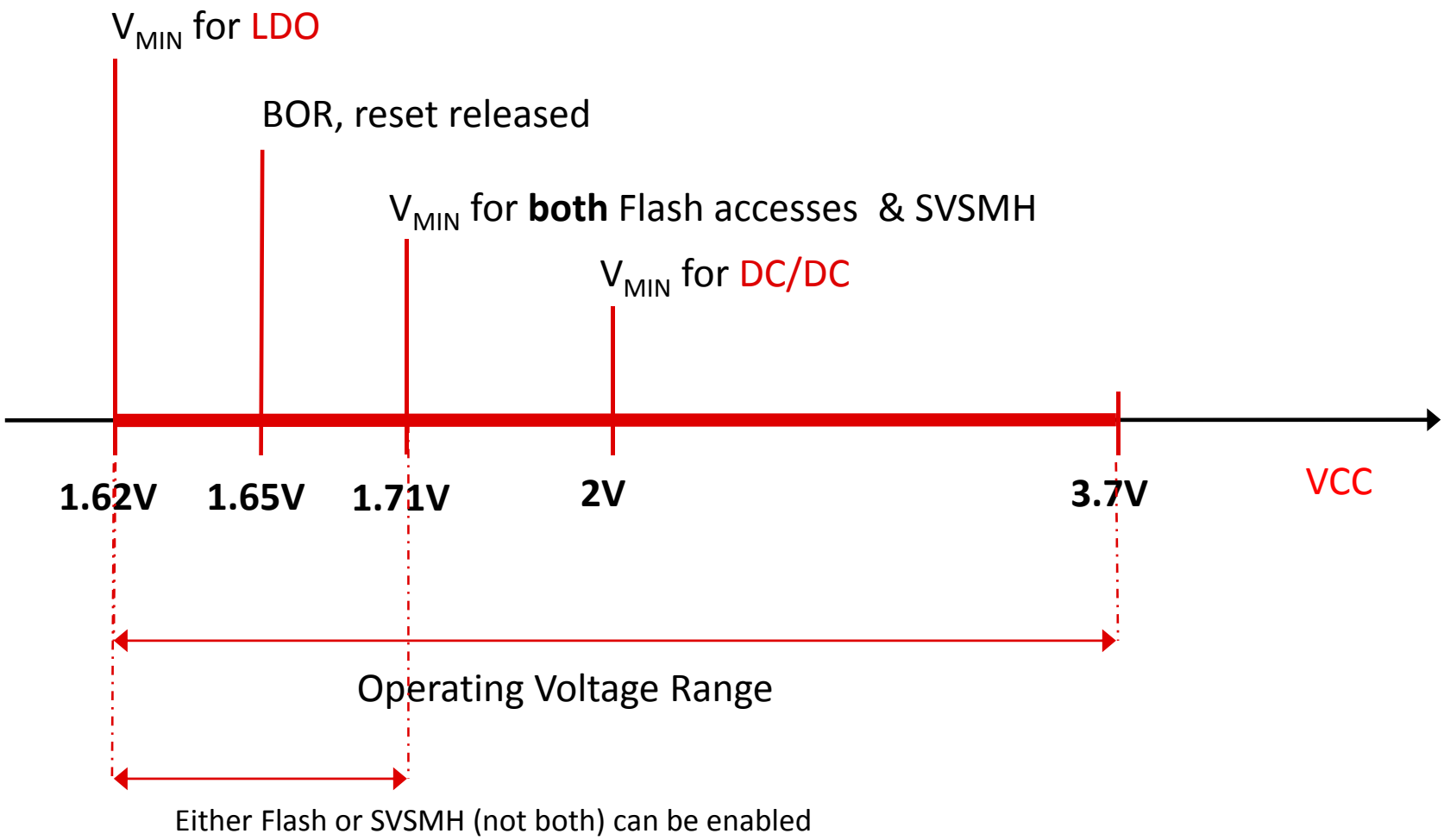
- **Wide supply range with true 1.8V+/-10% operation: 1.62V-3.7V**

- **Two internal core voltages for system frequency power-scaling**
  - 1.2V: 1-24MHz operation
  - 1.4V: 1-48MHz operation
- **Two internal voltage regulators to adapt for power requirements/profiles**
  - LDO: default regulator
  - DC/DC: additional regulator for better efficiency @ higher frequency

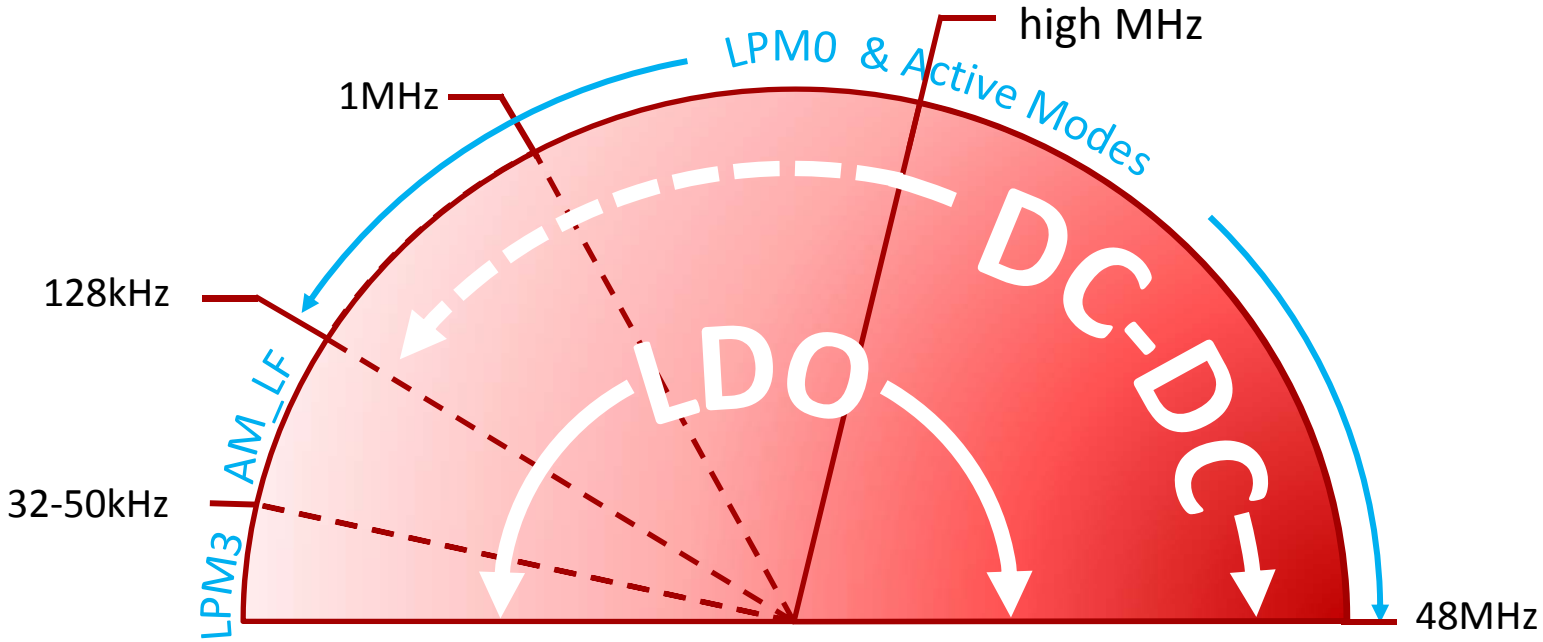
- **Supply Voltage Monitor & Supervisor**
  - Low performance modes for extremely low power in LPM3/4/x.5

- **DriverLib-assisted power state transitions & configurations**

Power | Operating Conditions



Power | Regulators: LDO & DC-DC



LDO	DC-DC
Default regulator at startup	Secondary, requires external inductor
VCC = **[1.62V-3.7V]**	VCC = [2.0V-3.7V]
Available in **all power modes**	Available in LPM0 & Active Modes
Flexible with **scalable output loads** for low power modes	**Efficient, optimized for high-speed/high-load** operations
Fast on/off switching operations	Slow on/off/failsafe switching from/to LDO

Power | MSP432 Flexible Operating Modes

MSP432	
Current	
CPU	
FLASH	
SRAM	
LDO	
DC-DC	
PSS Bandgap	
Clocks	
Core Domain Logic	
Backup Domain Logic (RTC/WDT)	
I/O State	

LPMx.5 modes
Ultra low leakage modes, with no state retention (as low as 670nA typ with RTC active)

LPM3.5: allows wake up from RTC & WDT events

LPM4.5: allow reset wake up through reset

: 0 – 24MHz
gh LDO (for maximum

: 0 – 48MHz
gh LDO or DCDC (for

Power | Software & Intrinsic Support

- Blends with Cortex-M sleep & interrupt mechanisms

- Go to LPM0/Sleep using Cortex-M CMSIS instructions
  - __wfi()

- Go to LPM3/Deep-Sleep using CMSIS & MSP-DNA intrinsic
  - __deep_sleep()

- DriverLib calls:
  - PCM_setPowerState() to transitions between all states.
  - PCM_shutdownDevice(PCM_LPM45);
  - PCM_gotoLPM0();
  - PCM_gotoLPM3();
  - PSS_enableHighSide();
  - PSS_setHighSidePerformanceMode();

Clock System & Memory

CS | High-level Features

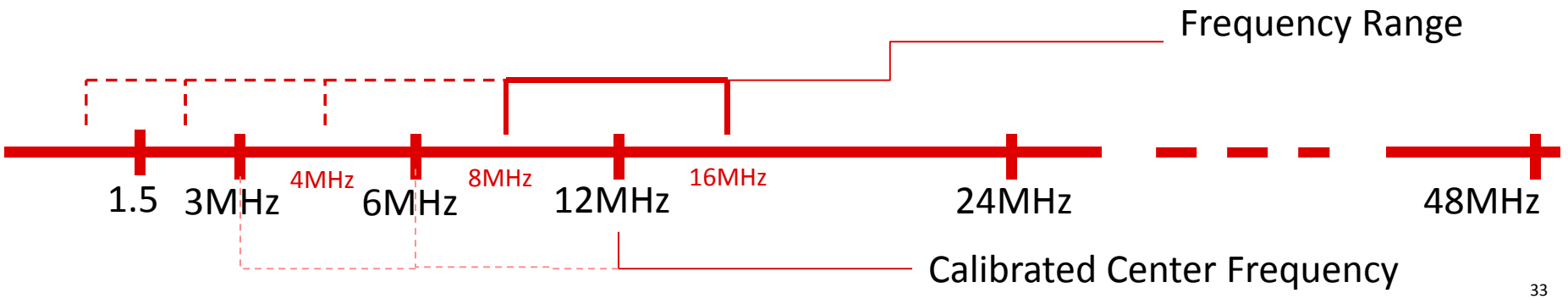
- **Flexible clock sources & distribution:**
  - 5 clocks from 7 sources (2 external, 5 internal)
  - Selections suitable for high-speed & low-power operations
- **Wide range of operating frequency**
  - 10kHz to 48 MHz
  - Fine intermediate steps with dividers & tuning
- **Configurable & robust system:**
  - Run-time lockable configuration
  - Failsafe mechanism with interrupts for external sources

CS | HF & LF Oscillators

Frequency		Oscillators	MCLK	SMCLK	HSMCLK	ACLK	BCLK	Comments
HF	1-48 MHz	DCO	✓	✓	✓			Internal integrated digitally controlled oscillator.
	1-48 MHz	HFXT	✓	✓	✓			High frequency crystal. Frequency range is SW configurable.
	24MHz	MODOSC	✓	✓	✓			Internal osc. option for peripherals such as ADC
	5MHz	SYSOSC						Internal, direct clock for ADC failsafe for HFXT
LF	32kHz	LFXT	✓	✓	✓	✓	✓	Low-frequency oscillator
	32kHz 128kHz	REFO	✓	✓	✓	✓	✓	Internal low-frequency oscillator. Failsafe* (32kHz) for LFXT
	10kHz	VLO	✓	✓	✓	✓		Internal ULP LF oscillator Clock selection for WDT

CS | High-accuracy tune-able DCO

- **6 tune-able frequency ranges**
  - Each range has calibrated center frequency
  - Example: [8-16MHz] range has a calibrated 12MHz center frequency
- **Tune-able within each frequency range**
  - Center Frequency +/- 2^{12} steps → DCOTUNE register
- **DCO accuracy:**
  - Internal resistor: ± 2.65 % [Calibrated]
  - External resistor : ± 0.4 % [91kΩ ± 0.1%]
    - Failsafe for internal resistor mode



Memory | Overview

Memory	Size	Speed	Features	
Flash	256kB + 4kB Sector: 4kB	16MHz	Speed boost with 128-bit buffer & pre-fetch	Powerful security features
SRAM	64kB Bank: 8kB	48MHz	Dynamic bank power-down & retention options for low power	
ROM	32kB	48MHz	Robust DriverLib APIs integrated to save application space	Lower power execution
BSL	8kB	16MHz	UART/I2C/SPI Boot-Strap Loader provided	

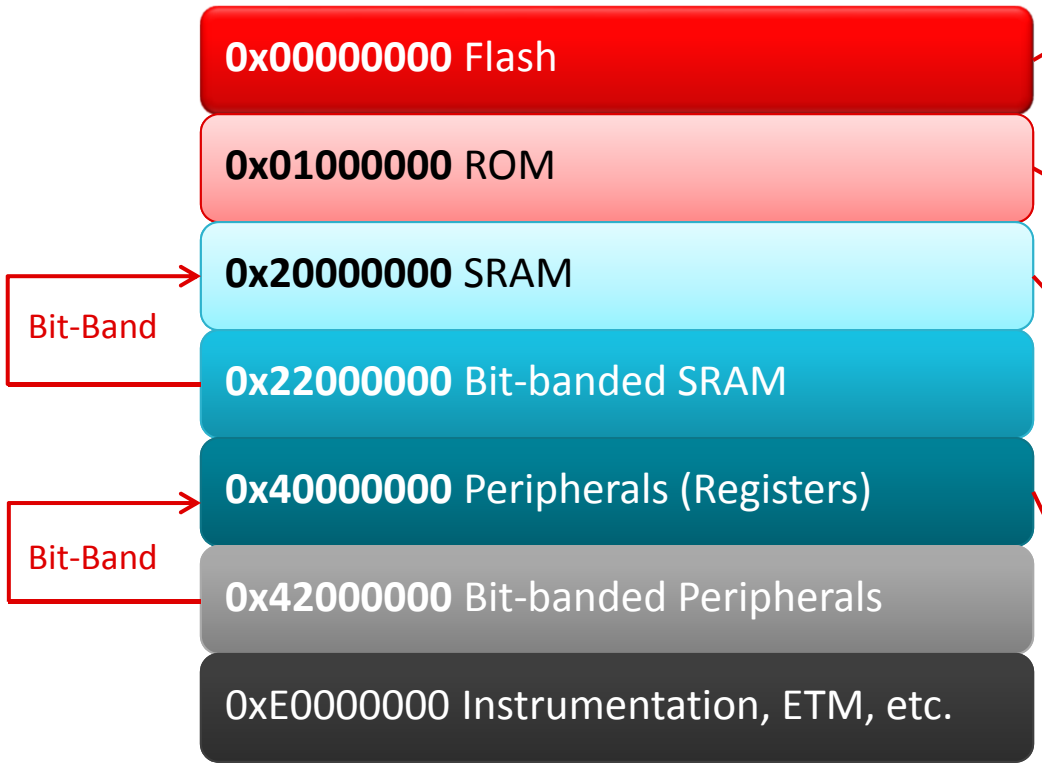
*Possible change: programmable BSL in next devices/revisions

Memory | RAM

- **Up to 64KB of banked SRAM architecture**
- **8 dynamically configurable banks:**
  - Enable/disable banks to optimize **active mode** power consumption
  - Retain/not retain content in LPM3 to minimize **SRAM leakage** power consumption

SRAM banks	Memory size
Bank 0 enable/retention (always enabled)	8KB
Bank 1 enable/retention	16KB
Bank 2 enable/retention	24KB
. . .	…
Bank 6 enable/retention	56KB
Bank 7 enable/retention	64KB

Memory | Memory Map



256kB + 4kB
- Interrupt Vector Table
- Application Code

Peripheral Driver Library

Ultra-low-leakage SRAM
- 64kB = 8 x 8kB banks
- Bit-banded

Peripheral Space
- Register directly accessible
- Bit-banded

Digital Peripherals

Timers: TimerA, Timer32, RTC, WDT
eUSCI: UART, SPI, I2C
GPIO
DMA

Timers | Overview

+ Cortex

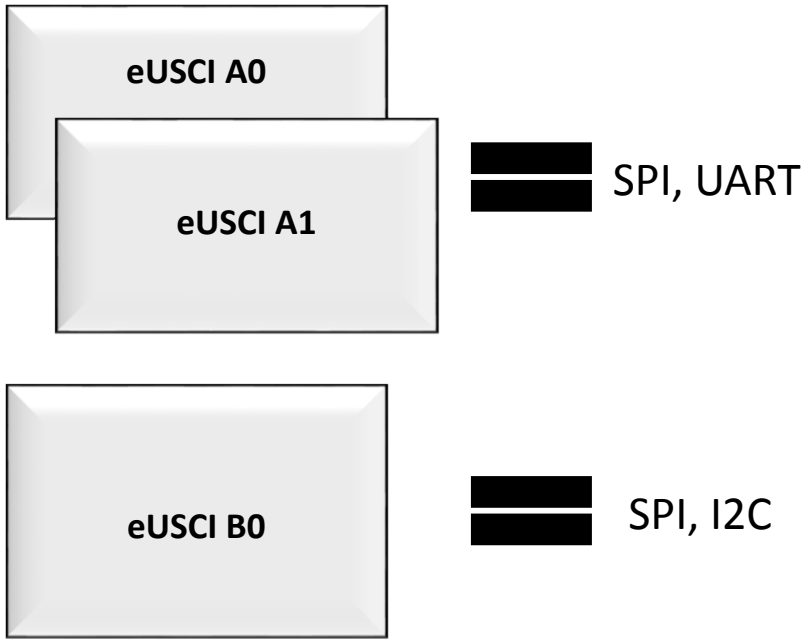
Timer	Instances	# of Bits	Lowest Operating mode	Interval	PWM	Main/Other Function
Timer_A	4	16	LPM0, LPM0_LF	✓✓	✓	Capture, Compare
Timer32	2	32	LPM0, LPM0_LF	✓✓		
SysTick	1	24	LPM0, LPM0_LF	✓✓		
RTC	1	32*	LPM3, LPM3.5	✓		RTC
WDT	1		LPM3, LPM3.5	✓		WDT

✓ RTC & WDT: only have a few selection of pre-set intervals

✓✓ Timer_A, Timer32, SysTick: configurable intervals with 16/32/24-bit granularity

eUSCI | Serial Communication

- **I²C**
  - 400kHz & up to 1MHz
  - 4 hardware slave addresses supported with dedicated TX/RX interrupts
  - Byte counter for autonomous operation
- **UART**
  - Enhanced baud rate generation
  - Auto baud rate detection
- **SPI**
  - 3 & 4-wire modes
  - Up to 14MBps



Digital I/O | Overview

- Ultra low-leakage IOs (± 20 nA Max)
- Capacitive touch function available on all IOs
- Pull-up/Pull-down
- Interrupt/Wake-up capability available on 6 ports (48 pins)
- Port Map: 30 digital functions available to each of 24 port-mappable IOs
- Up to 4 pins with 20mA max output, suitable for optional high-drive functionality (P2.0-P2.3)
- 8 IOs with glitch filtering capability

DMA | PL230 MicroDMA

- Single 32-bit AHB-Lite master interface for transferring data
- Supports multiple DMA transfer data widths and address increment values
- 1-1024 transfers in a single DMA access
- Each DMA channel has 2 channel control data structures (primary/alternate) stored in RAM
- Channel control data is stored in RAM
- Multiple DMA transfer types:
  - Basic
  - Auto-Request
  - Ping-Pong
  - Memory Scatter-Gather
  - Peripheral Scatter-Gather
- Each DMA channel has a programmable priority level.
- Error indication in case any transfer comes back with a bus error response.

DMA | Triggers mapping

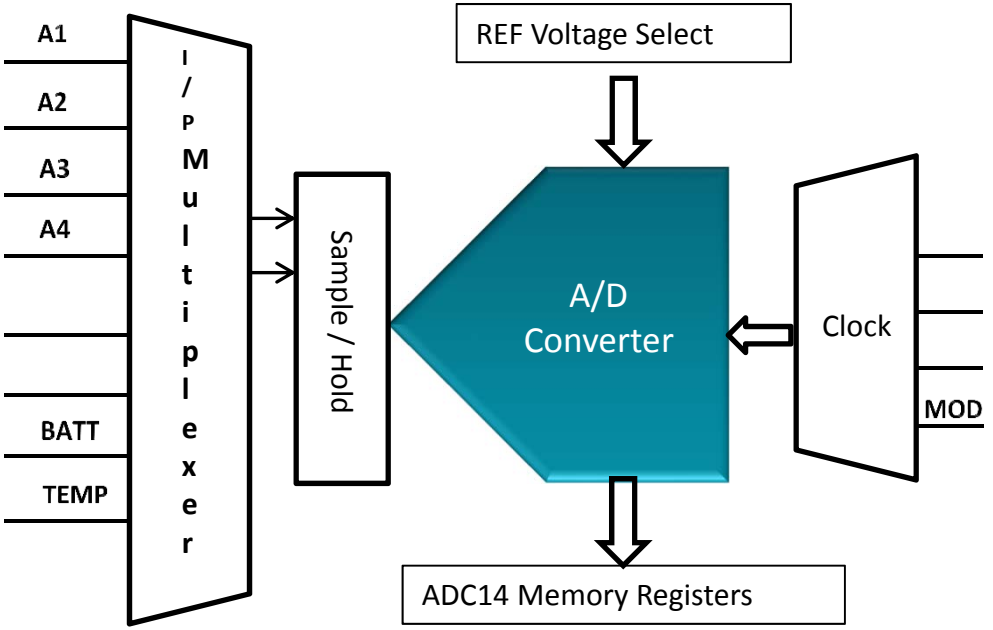
Channel	SRCCFG=0	SRCCFG=1	SRCCFG=2	SRCCFG=3	SRCCFG=4	SRCCFG=5	SRCCFG=6	SRCCFG =7
0	Reserved	eUSCI_A0 TX	eUSCI_B0 TX0	eUSCI_B3 TX1	eUSCI_B2 TX2	eUSCI_B1 TX3	TA0CCR0	AES256_Trig gen0
1	Reserved	eUSCI_A0 RX	eUSCI_B0 RX0	eUSCI_B3 RX1	eUSCI_B2 RX2	eUSCI_B1 RX3	TA0CCR2	AES256_Trig gen1
2	Reserved	eUSCI_A1 TX	eUSCI_B1 TX0	eUSCI_B0 TX1	eUSCI_B3 TX2	eUSCI_B2 TX3	TA1CCR0	AES256_Trig gen2
3	Reserved	eUSCI_A1 RX	eUSCI_B1 RX0	eUSCI_B0 RX1	eUSCI_B3 RX2	eUSCI_B2 RX3	TA1CCR2	Reserved
4	Reserved	eUSCI_A2 TX	eUSCI_B2 TX0	eUSCI_B1 TX1	eUSCI_B0 TX2	eUSCI_B3 TX3	TA2CCR0	Reserved
5	Reserved	eUSCI_A2 RX	eUSCI_B2 RX0	eUSCI_B1 RX1	eUSCI_B0 RX2	eUSCI_B3 RX3	TA2CCR2	Reserved
6	Reserved	eUSCI_A3 TX	eUSCI_B3 TX0	eUSCI_B2 TX1	eUSCI_B1 TX2	eUSCI_B0 TX3	TA3CCR0	External pin
7	Reserved	eUSCI_A3 RX	eUSCI_B3 RX0	eUSCI_B2 RX1	eUSCI_B1 RX2	eUSCI_B0 RX3	TA3CCR2	ADC14

Analog Peripherals

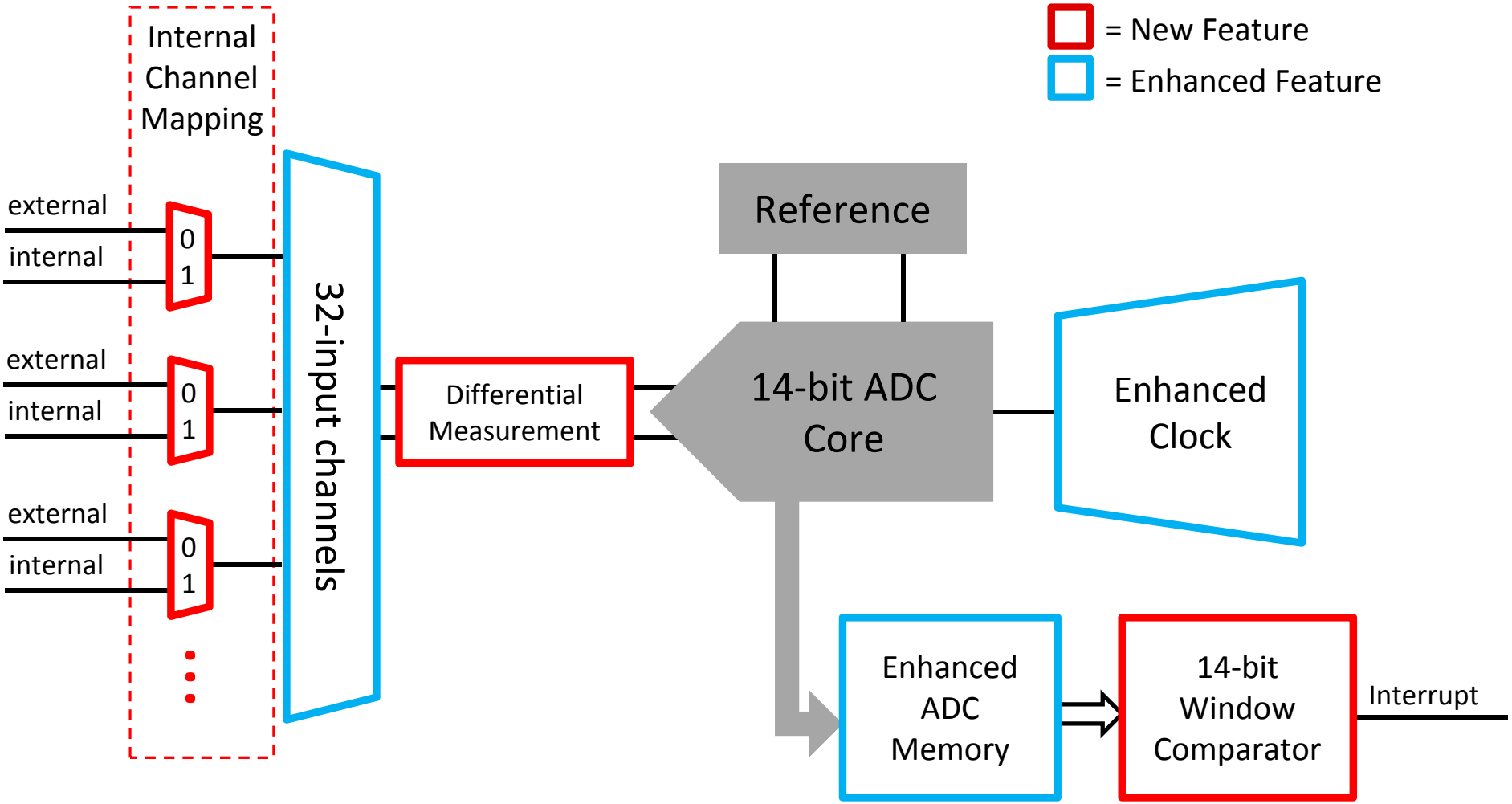
ADC14
COMP_E
REF

ADC14 | Overview

- 14-bit Accuracy
  - INL <= +/- 2 LSB;
  - DNL <= +/- 1 LSB
  - ENOB 13-bit
- 32-input channels
- Single-ended & differential Inputs
- 2 Window comparators
  - High interrupt
  - Low interrupt
  - In [Between] interrupt
- Separate internal channels for AVcc and TempSensor
- Ultra Low current consumption
  - Single ended 210uA @ 1.8V, 1Msps
  - Differential 260uA @ 1.8V, 1Msps

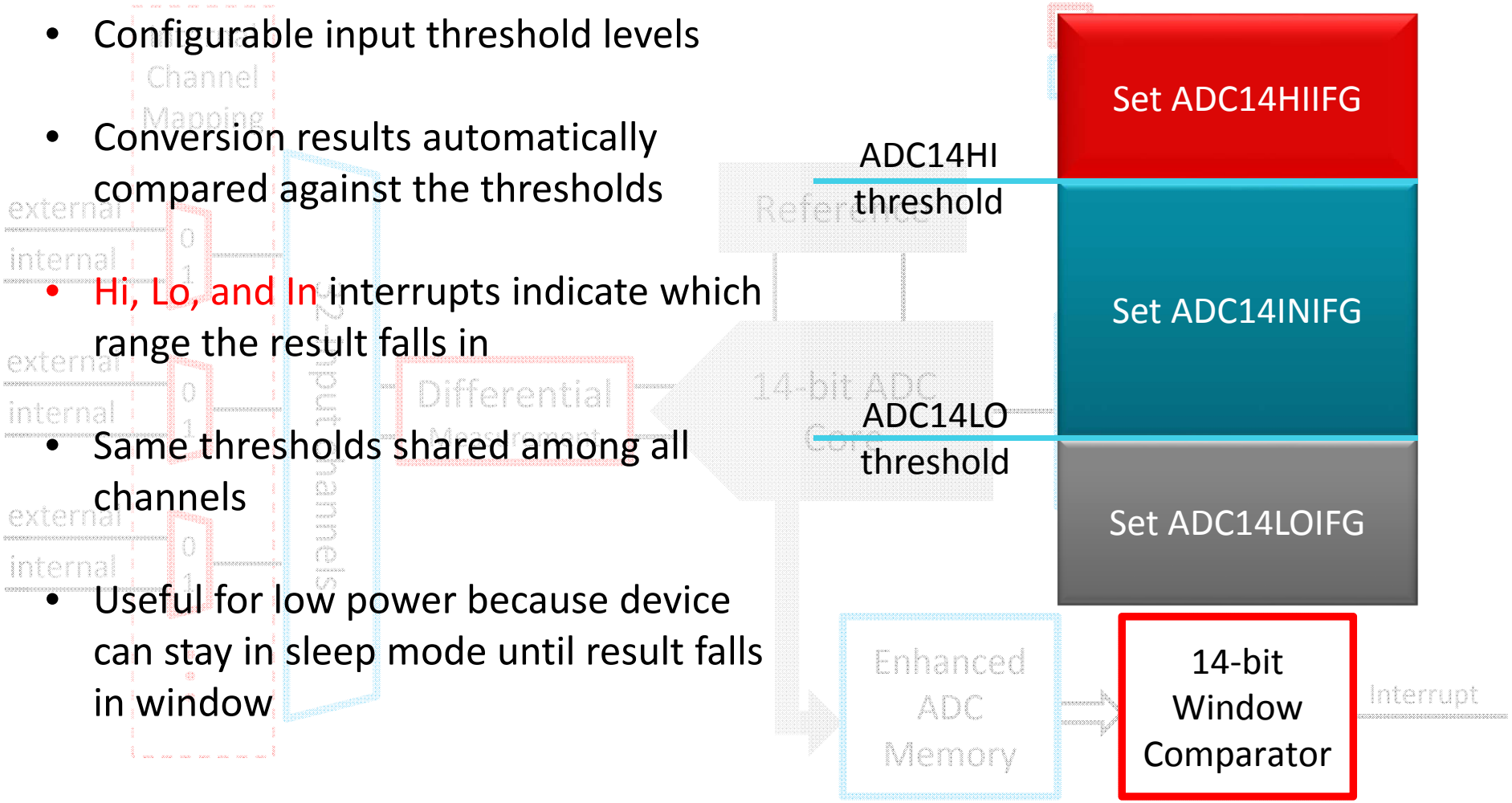


ADC14 | New Features



ADC14 | Window Comparator

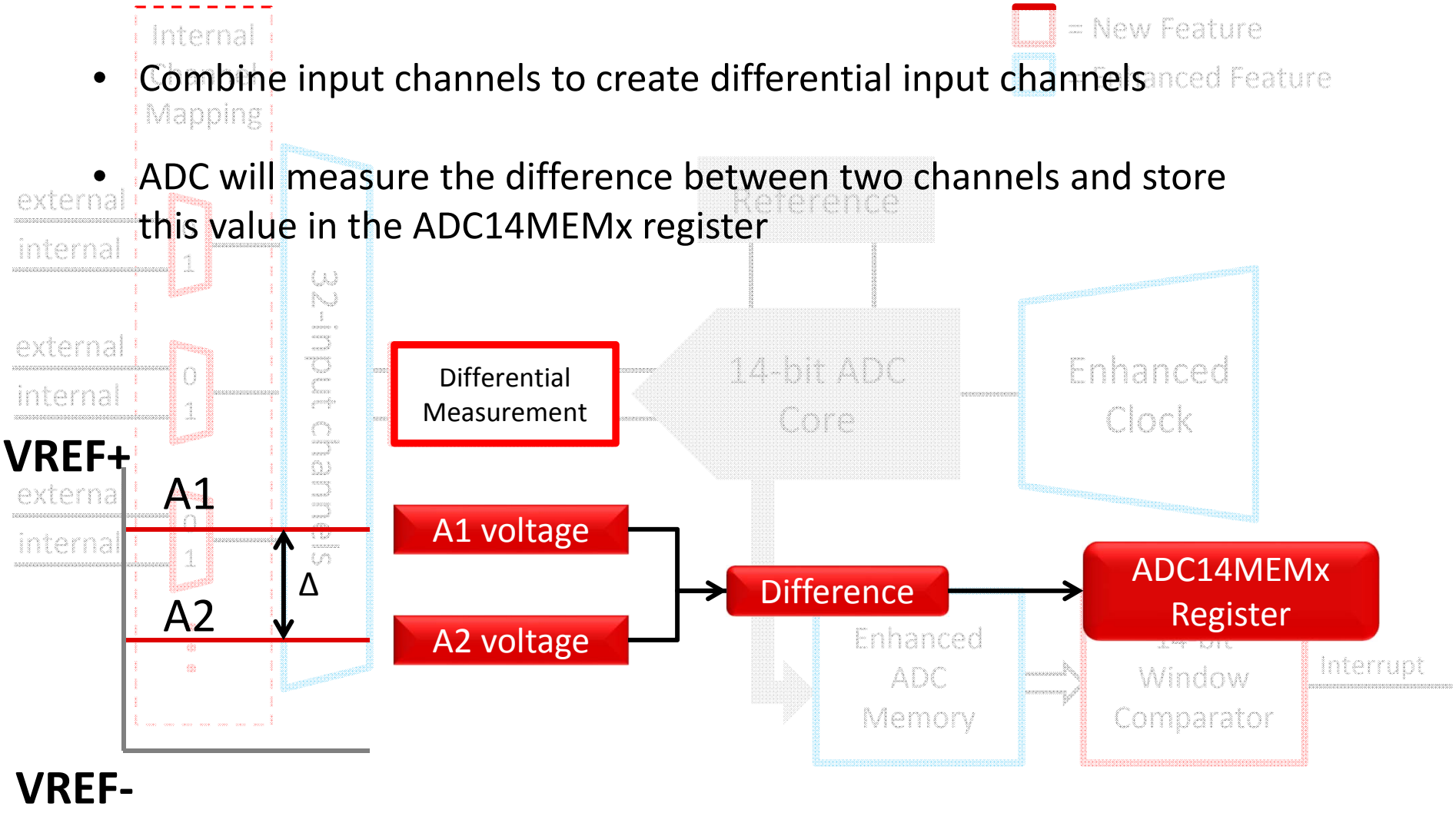
- Configurable input threshold levels
- Conversion results automatically compared against the thresholds
- Hi, Lo, and In interrupts indicate which range the result falls in
- Same thresholds shared among all channels
- Useful for low power because device can stay in sleep mode until result falls in window



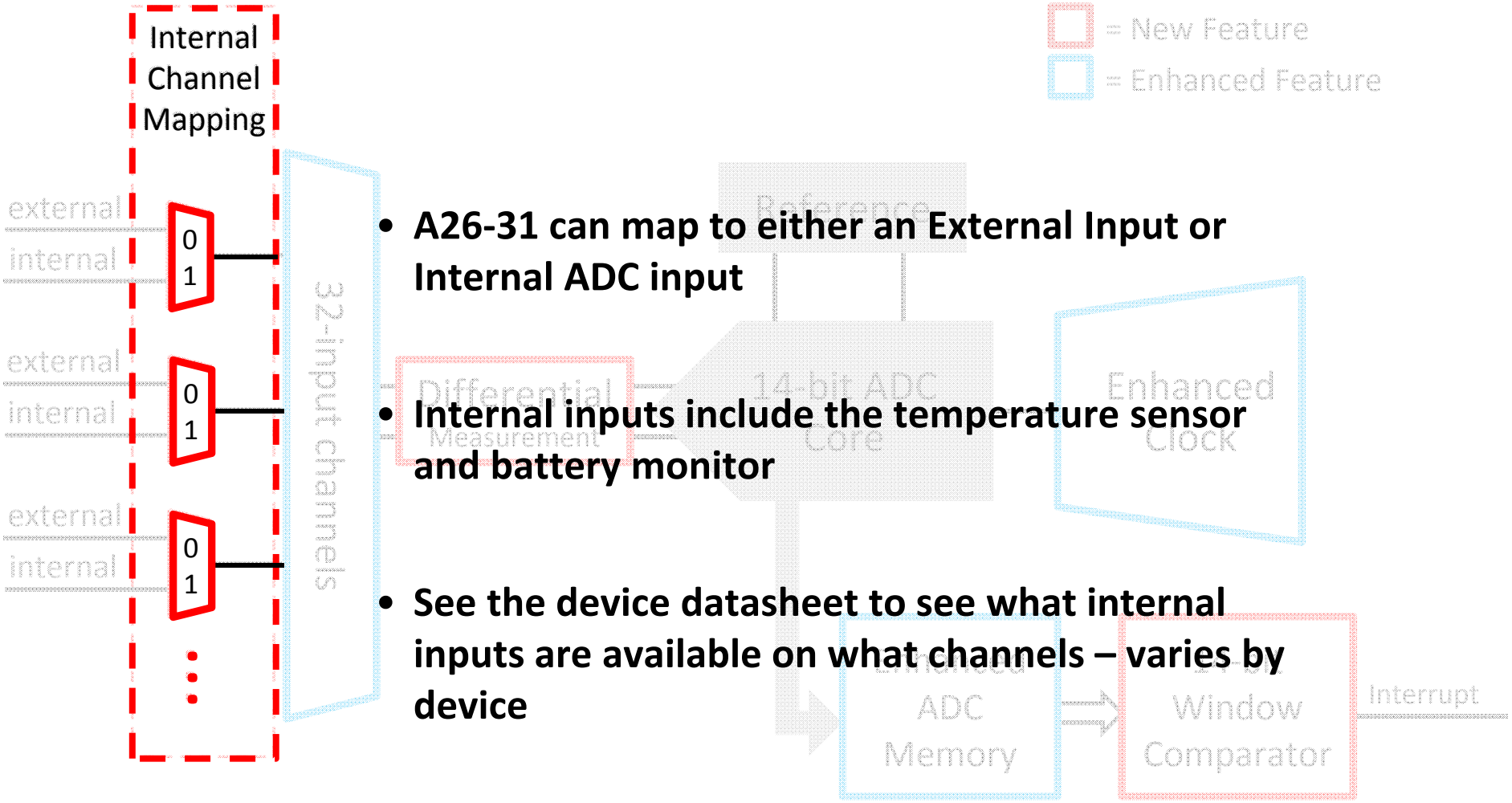
ADC14 | Window Comparator Example

ADC14 | Differential Measurement

- Combine input channels to create differential input channels
- ADC will measure the difference between two channels and store this value in the ADC14MEMx register

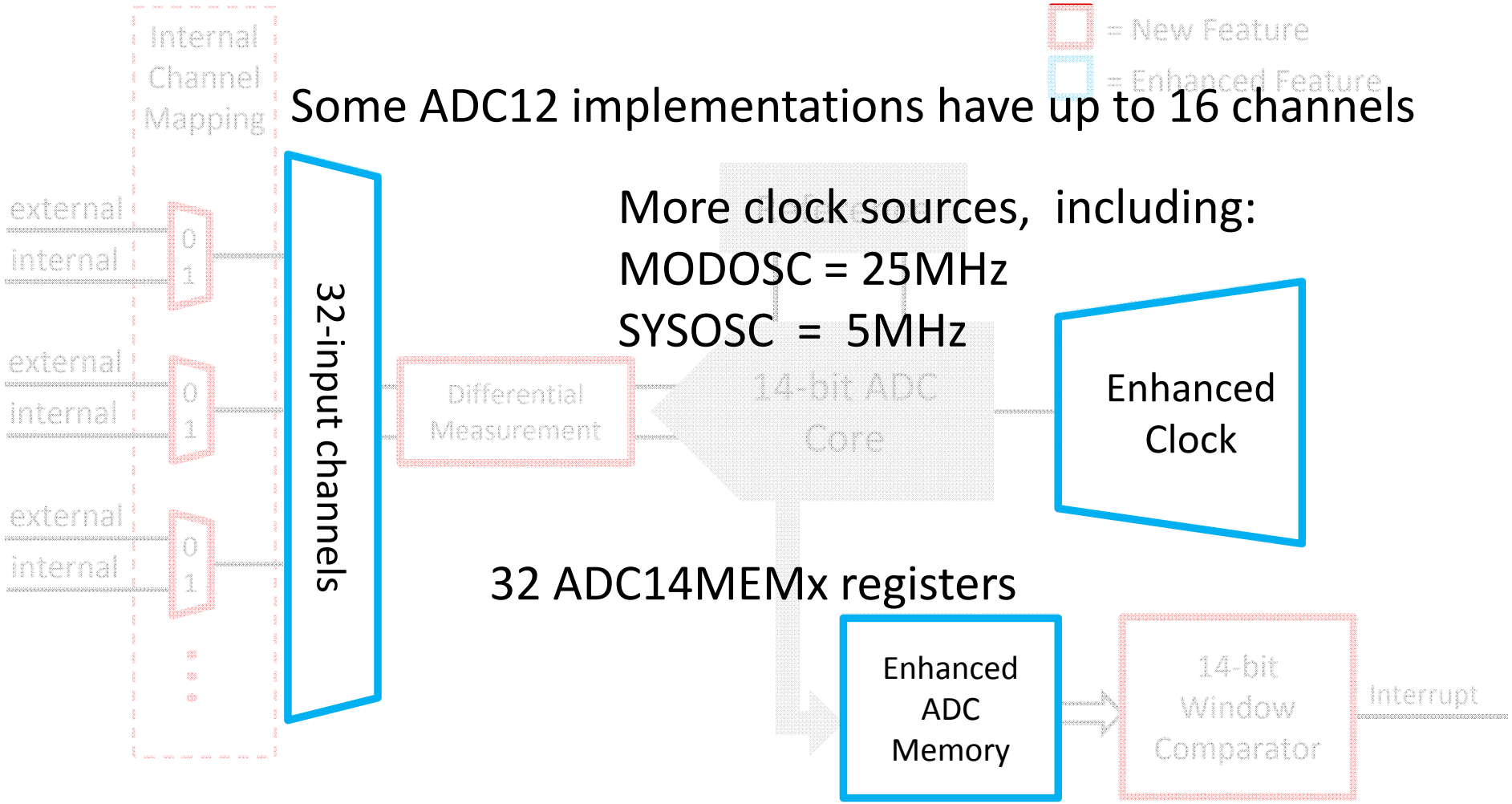


ADC14 | Internal Channel Mapping



- **A26-31 can map to either an External Input or Internal ADC input**
- **Internal inputs include the temperature sensor and battery monitor**
- **See the device datasheet to see what internal inputs are available on what channels – varies by device**

ADC14 | Other enhancements



Comp_E | Overview

- Interrupt driven for low power
- Uses the REF module like ADC14
- Up to 15 external input channels
- Ultra-low-power comparator mode
- Software selectable RC filter
- Selectable reference voltage generator
- Voltage Hysteresis generator
- Output internally connected to Timer A capture input (for event capture & capacitive touch implementations)

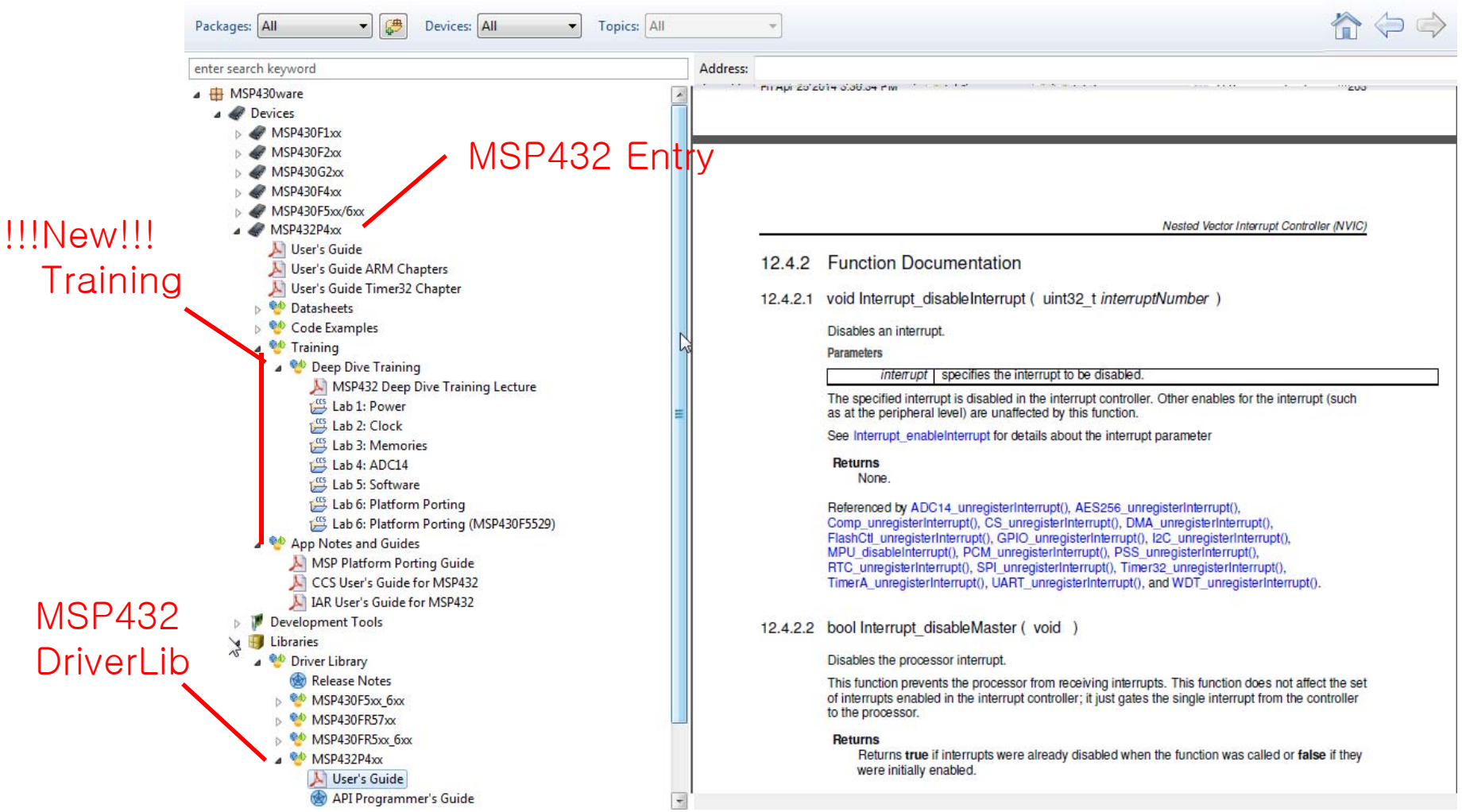
REF | Reference Module

- **Generates voltage references for analog modules**
  - 1.2V, 1.45V, 2.5V
- **Available on output pin**
- **Scale-able power options**
  - Static mode for high precision outputs
  - Sampled mode for low-power operations
- **Two operation modes:**
  - Continuous Mode: reference output continuously on
  - Burst Mode: reference output on only during ADC conversions
- **Two buffers to throttle power consumption:**
  - Large buffer for continuous mode or output VREF to pin
  - Small buffer for burst mode
- **Provides temp sensor channel for ADC**
  - Option to disable to minimize power

Software

MSPWare
DriverLib
CMSIS

It's all in MSPWare



Software | MSP Register-Level

- Traditional MSP register-level access code fully supported
- Header files provide complete register & bit definitions
- Complete portability for common peripherals across 16 & 32-bit platforms
- 100+ code examples for MSP430-shared & new MSP432 peripherals

c code example

```
WDTCTL = WDTPW | WDTHOLD;           // Stop WDT
P5SEL1 |= BIT4;                     // Configure P5.4 for ADC
P5SEL0 |= BIT4;
__enable_interrupt();               // MSP432: Enable master interrupt
SCS_NVIC_ISER0 = INT_ADC14_BIT;     // MSP432: Enable ADC14 interrupt
ADC14CTL0 = ADC14SHT0_2 | ADC14SHP | ADC14ON
ADC14CTL1 = ADC14RES_2
ADC14MCTL0 |= ADC14INCH_1;          // A1 ADC input select;
ADC14IER0 |= ADC14IE0;              // Enable conv. interrupt
SCS_SCR &= ~SCS_SCR_SLEEPONEXIT;    // MSP432: Wake up on exit from ISR
```

Software | Driver Library

```
GPIO_setAsPeripheralModuleFunctionOutputPin(PARAMETERS);
Timer_generatePWM(PARAMETERS)
```

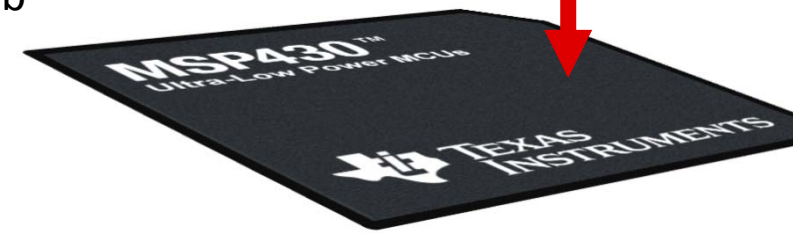
Driver Library

Traditional C code

```
P2DIR |= 0x04;
TA1CCTL1 = OUTMOD_7;
P2SEL |= 0x04;
TA1CCR1 = 384;
TA1CCR0 = 511;
TA1CTL = TASSEL_1 | MC_1 | TACLR;
```

Low level programming

```
00101010
10010010
01010100
10010010
11001010
```



- Driver Library offers easy-to-understand functions
- No more cryptic registers to configure
- MSP430/432 shared peripherals also share DriverLib APIs → reduce porting effort

MSP432 DriverLib | ROM & Source

MSP432 DriverLib available as source

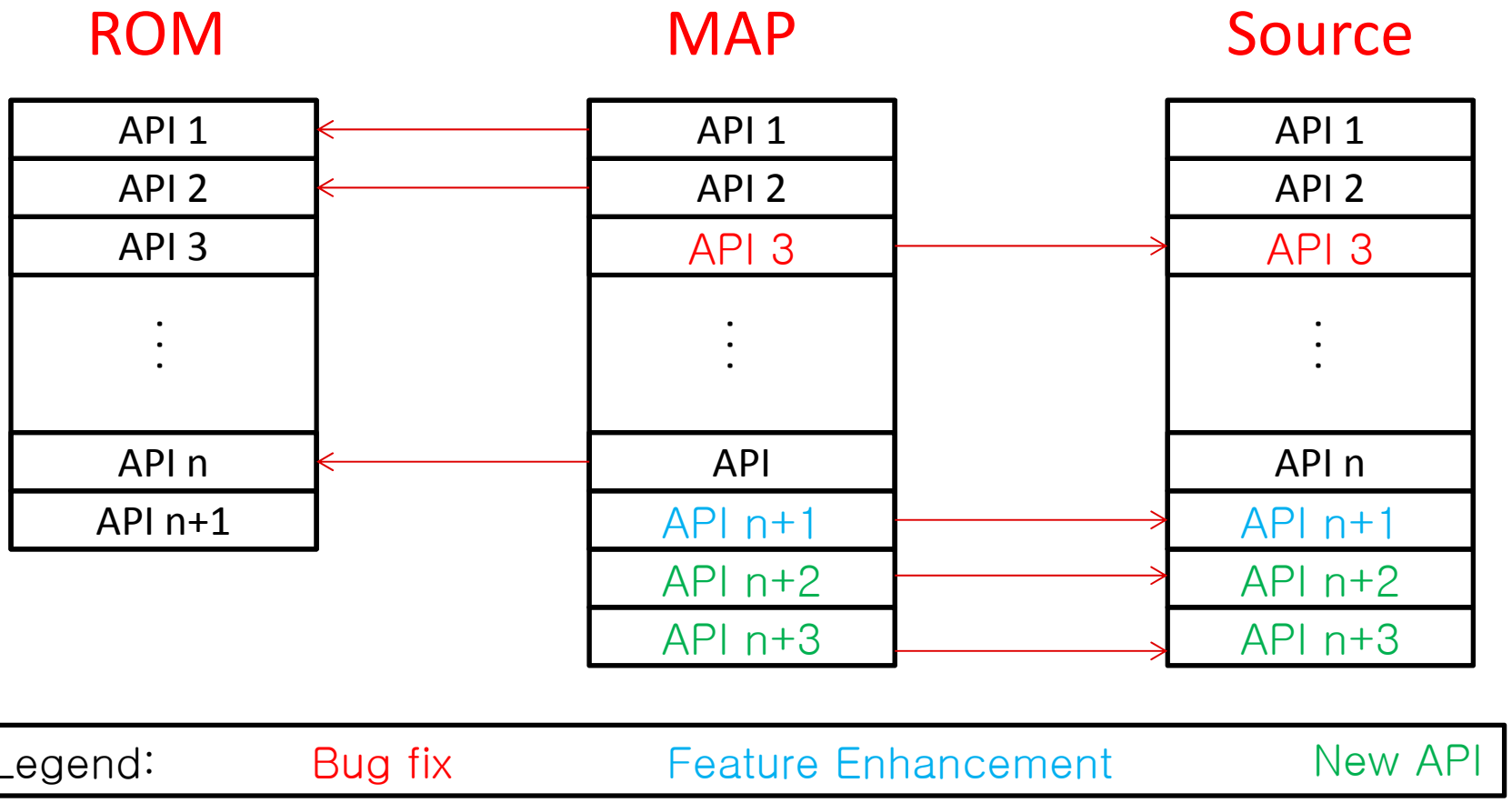
- Always the latest, most updated version
- Library available as BSD source = industry standard
- Flexibility for users to further customize/enhance

MSP432 DriverLib implemented & tested in ROM

- ROM has lower current consumption
- No wait-state [even @ max speed = 48MHz]
- Frees up flash space (25kB) for application code
- Fully validated & robust ROM code: no future major changes

MSP432 DriverLib | ROM & Source

MAP file: method to ensure ROM API is always used
UNLESS there's an update (fix/enhancement) or APIs only available in Source



MSP432 DriverLib | Calling Convention

MSP432 DriverLib source (Flash)

- Include *driverlib* source folder in your project
- API Call: TimerA_generatePWM(param1, param2, etc.);

MSP432 DriverLib ROM

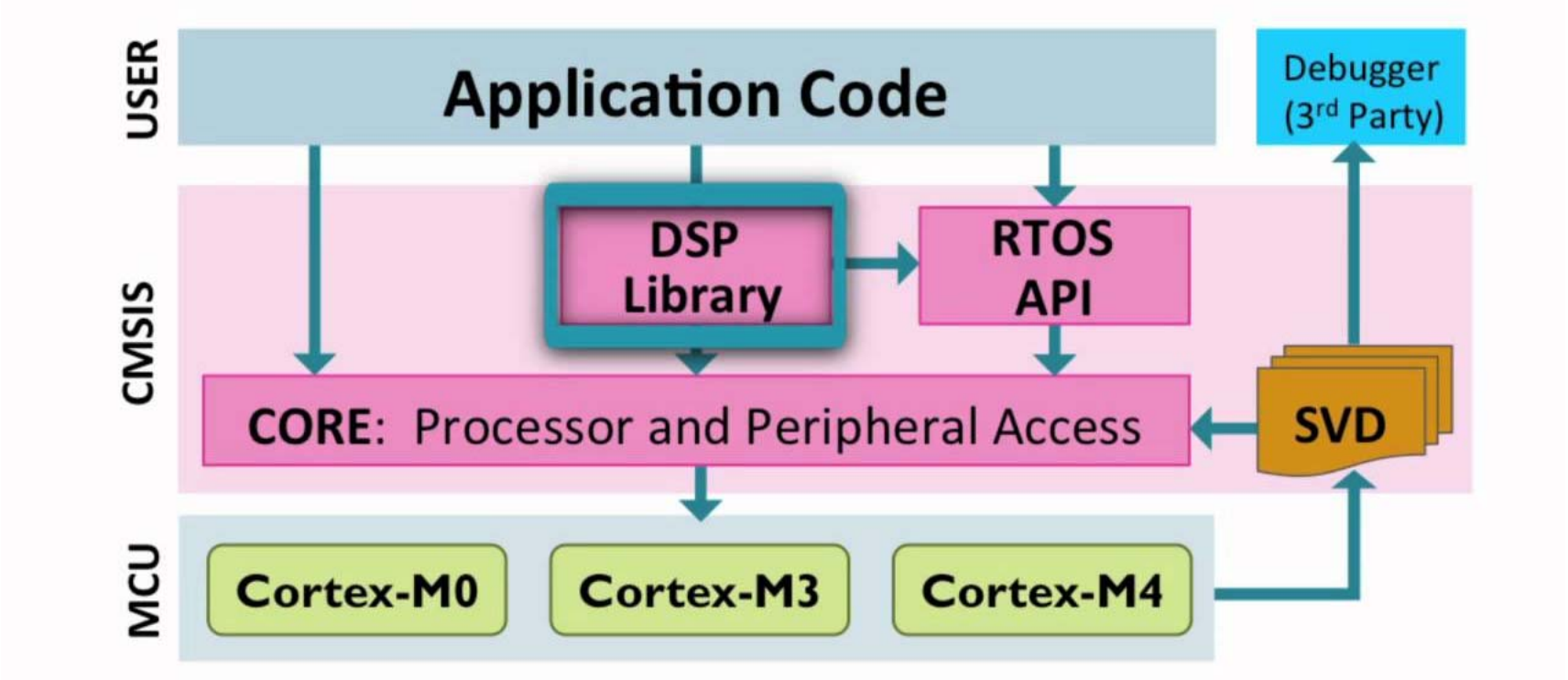
- Use *"rom.h"* header file
- API Call: **ROM_**TimerA_generatePWM(param1, param2, etc.);

Patches new features or bug fixes:

- Use *"rom_map.h"* header file
- API Call: **MAP_**TimerA_generatePWM(param1, param2, etc.);

Software | CMSIS

- Cortex Microcontroller Software Interface Standard (CMSIS)
- Standardized hardware abstraction layer for the Cortex-M4 processor series



Software | CMSIS

MSP definition

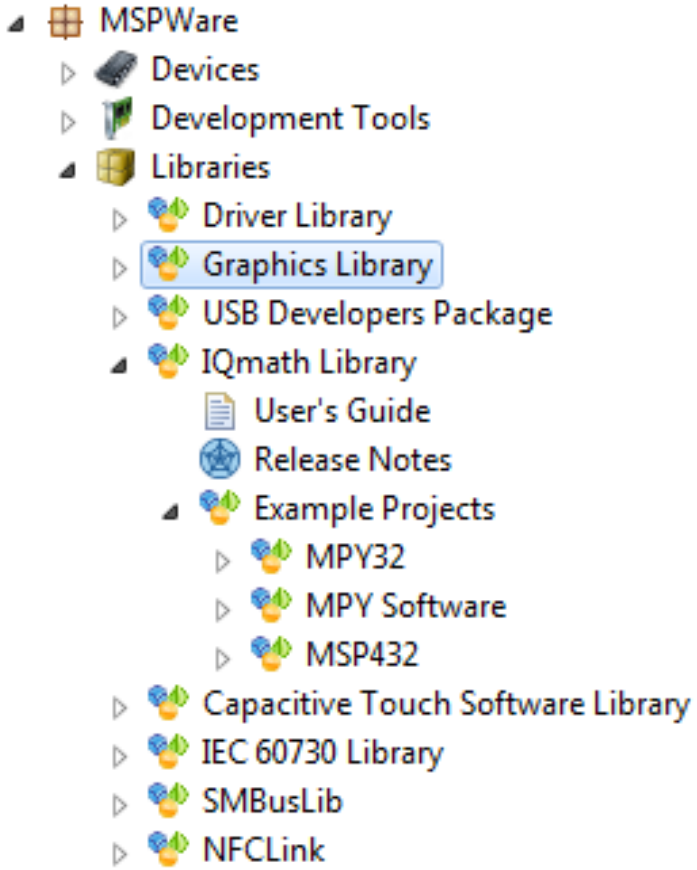
```
#define WDTCTL \
        (HWREG16(0x4000480C))

/* WDTCTL Control Bits */
#define WDTIS0      (0x0001)
#define WDTIS1      (0x0002)
#define WDTIS2      (0x0004)
#define WDTCNTCL    (0x0008)
#define WDTTMSEL    (0x0010)
#define WDTSSEL0    (0x0020)
#define WDTSSEL1    (0x0040)
#define WDTHOLD     (0x0080)
#define WDTPW       (WDTPW_VAL)

#define __WDT_BASE__ (0x40004800)
```

CMSIS definition

```
typedef struct {
  uint8_t RESERVED0[12];
  union {     /* WDT_CTL Register */
    __IO uint16_t reg;
    struct { /* WDT_CTL Bits */
      __IO uint16_t IS    :      3;
      __O  uint16_t CNTCL :      1;
      __IO uint16_t TMSEL :      1;
      __IO uint16_t SSEL  :      2;
      __IO uint16_t HOLD  :      1;
      __IO uint16_t PW    :      8;
    } bit;
  } CTL;
} WDT_Type;
#define WDT ((WDT_Type *) __WDT_BASE__)
```

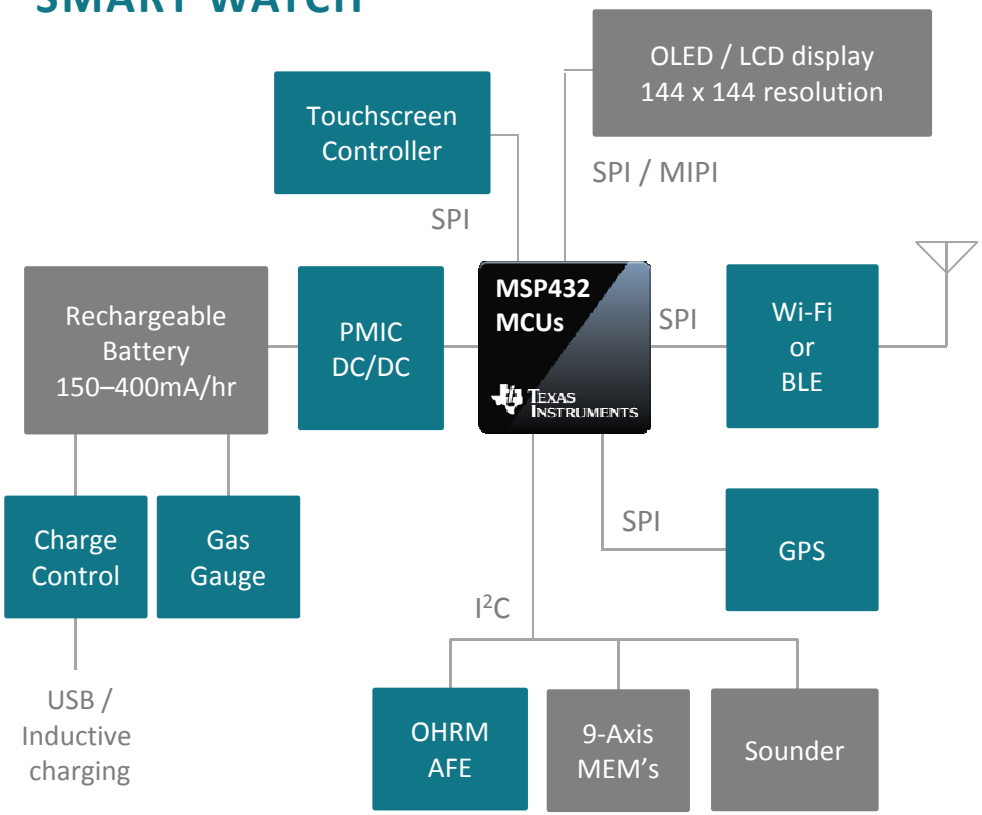
Software | Libraries & RTOS

- **High-level libraries**
  - Graphics Library
  - Capacitive Touch Library
  - IQMath
- **Application-specific libraries**
  - NFCLink & TRF7970A NFC Library
  - CC3100 WiFi SimpleLink
  - BLE & Bluetooth
  - And more …
- **RTOS**
  - TI-RTOS
  - FreeRTOS
  - Micrium µC/OS
  - And more….



MSP432™ MCUs: OPTIMIZED FOR Personal Electronics

SMART WATCH



WHY MSP432 MCUs?

High performance (48MHz)

- Quickly process changes in distance, heart rate, calories
- 48 MHz single-cycle RAM access
- Leverage extensive ARM ecosystem

High RAM size (64KB)

- Update LCD or dot matrix array displays in real time

Ultra-low power

- Ultra-low-power ADC and peripherals provide days/weeks of power

Flexible flash

- Independent banks allow simultaneous program read & erase

Wide voltage range 1.62-3.7V

- Enables 1.8V±10% operation for application processors and sensors
- Extends support for higher voltage batteries (e.g. 3.7V Li-ion)

INDUSTRIAL | PERSONAL ELECTRONICS | INTERNET OF THINGS

MORE ECOSYSTEM WITH EASY-TO-USE TOOLS AND SOFTWARE

1 Get started here

MSP432™ LaunchPad

Easy to use, low-cost evaluation kit with integrated emulator and EnergyTrace+ technology

2 Connect to your computer

BoosterPacks

Expand MSP432 LaunchPad evaluation with easy to use, low-cost BoosterPack add-on daughter boards

CONNECTIVITY | DISPLAY | SENSORS

Software optimized for low-power

3 Choose from your favorite IDEs

- CCS
- IAR
- Keil

4 Develop or access code and collateral online, instantly

TI Cloud IDE:
- Resource Explorer
- Code Composer Studio™
- PinMux

5 Ease code development with easy to use APIs and examples

MSPWare™:
- Driver library
- App notes & user's guides
- Example code
- Tutorials

6 Optimize your code and system for ultra-low-power operation

Optimization tools:
- ULP Advisor
- EnergyTrace+™