



## ABSTRACT

MSPM0Lxx series microcontroller (MCU) portfolio is a subfamily of MSPM0, which offers a wide variety of 32-bit MCUs with ultra-low-power and integrated analog and digital peripherals for sensing, measurement and control applications. Reducing power consumption while performing complex real-time applications presents a major challenge for the recent embedded applications. This article was created to build a simple framework to help developers understand the MSPM0Lxx series low-power features, how power can be optimized to meet the specific needs based on MSPM0 and how to evaluate and measure it.

---

## Table of Contents

<b>1 Overview</b> .....	<b>2</b>
<b>2 Low-Power Features in PMCU</b> .....	<b>4</b>
2.1 Overview .....	4
2.2 Power Management (PMU).....	5
2.3 Clock Module (CKM).....	7
<b>3 Low-Power Optimization</b> .....	<b>9</b>
3.1 Low-Power Basics.....	9
3.2 MSPM0 Low-Power Feature Use.....	10
3.3 Software Coding Strategies.....	12
3.4 Hardware Design Strategies.....	12
<b>4 Power Consumption Measurement and Evaluation</b> .....	<b>14</b>
4.1 Current Evaluation.....	14
4.2 Current Measurement.....	14

## Trademarks

EnergyTrace™ and LaunchPad™ are trademarks of Texas Instruments.

Arm® and Cortex® are registered trademarks of Arm Limited.

All trademarks are the property of their respective owners.

# 1 Overview

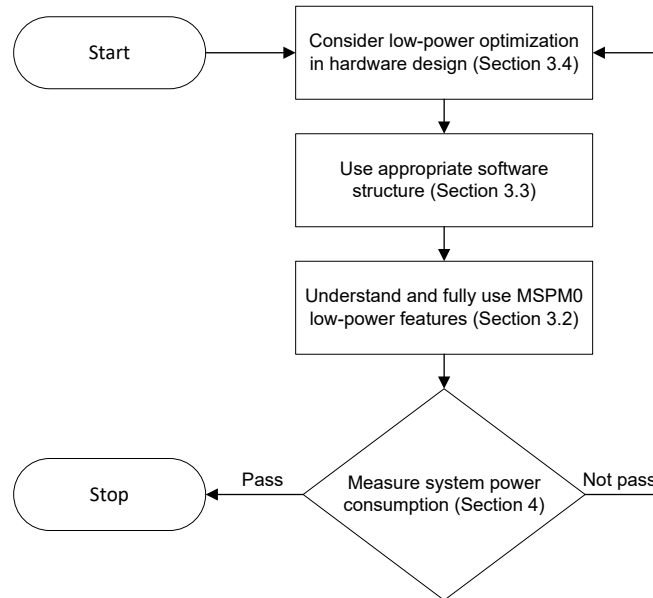
TI's scalable MSPM0Lxx MCUs are based on Arm® Cortex®-M0+ core, with a maximum CPU speed of 32 MHz, which provides the basic general-purpose functions with low-power features. The MCUs are available with up to 256KB of on-chip flash and up to 32KB on-chip SRAM with extended scalable analog Integration. They also integrate an efficient power supply architecture and various power modes that helps the power consumption reduction and simplify application design. Its overall low-power performance is show in [Table 1-1](#). For more details, refer to the device-specific data sheet.

**Table 1-1. MSPM0Lxx Series Low-Power Performance**

Low-Power Mode	MSPM0Lxx
Run <sup>(1) (5)</sup>	85 µA/MHz
Sleep <sup>(2) (5)</sup>	200 µA at 4 MHz
Stop <sup>(3) (5)</sup>	50 µA at 32 kHz
Standby <sup>(4) (5)</sup>	1.1 µA
Shutdown <sup>(5)</sup>	50 nA with IO wakeup capability

- (1) MCLK = SYSOSC = 32 MHz, CoreMark, execute from flash
- (2) MCLK = SYSOSC, CPU is halted
- (3) SYSOSC off, DISABLESTOP = 1, ULPCLK = LFCLK, SRAM and flash are in retention
- (4) STOPCLKSTBY = 1, TIMG0 enabled
- (5) Typical value at 25°C and VDD = 3.3 V. All inputs tied to 0 V or VDD. Outputs do not source or sink any current. All peripherals are disabled.

This application note was created to build a simple framework to help developers understand the MSPM0Lxx series low-power features, how power can be optimized to meet the specific needs based on MSPM0, and how to evaluate and measure it. [Figure 1-1](#) shows the design flow for a low-power design.



**Figure 1-1. Low-Power Development Process**

[Table 1-2](#) lists the items to check during development.

**Table 1-2. Low-Power Development Checklist**

Number	Classification	Item	Comment
1	Hardware design	MCU power supply	Reduce MCU power supply no lower than 1.62 V.
2		Resistors	Choose large resistors after meeting system requirement.
3		Capacitors	Choose low leakage capacitors.
4		Power IC	Normally choose a linear regulator.

**Table 1-2. Low-Power Development Checklist (continued)**

Number	Classification	Item	Comment
5	Software coding	Conditional code execution	Use a conditional wake-up and code execution structure.
6		Nonblocking programming	Avoid blocking mode by using while loop.
7		Optimize code size	Choose TI Arm Clang, Fully use compiler features and write code with good coding style.
8	MSPM0 low-power feature usage	Use low-power modes	Use different power modes (RUN, SLEEP, STOP, STANDBY, and SHUTDOWN) and three lower mode policy options (XX0, XX1, XX2) according to the application requirement.
9		Reduce system clock and peripheral operation frequency	Only use the needed system clock frequency. Reduce peripherals operation frequency and turn them off when not used.
10		I/O configuration	Leave the unused pin as default high-Z configuration. Reduce the use of internal pull-up or pull-down resistors. Pay attention to the IO-latch in low-power modes.
11		Use event manager	Use event manager to realize peripherals trigger DMA or peripherals trigger peripherals to reduce the CPU usage.
12		Use analog peripherals' low-power features	Compromise between performance and low-power consumption for the ADC, COMP, OPA and GPAMP.
13		Run code from RAM	Move a part of common used code from flash to RAM.

## 2 Low-Power Features in PMCU

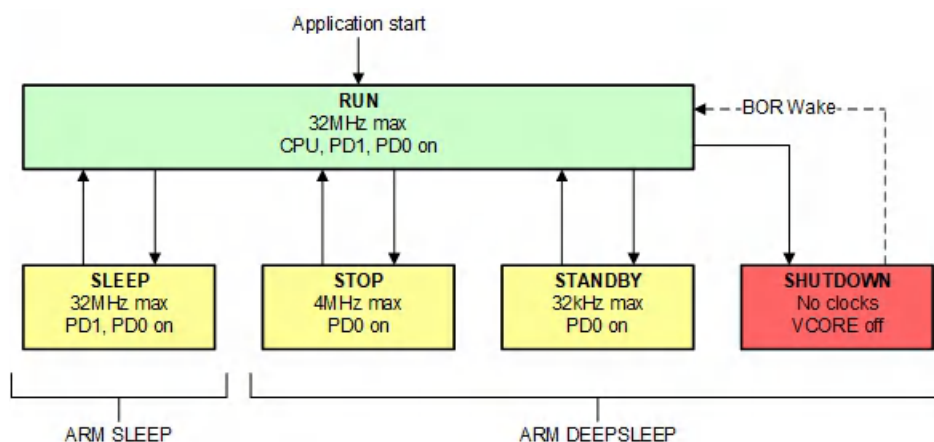
The power management and clock unit (PMCU) provide all power, clocking, reset, and system control services for the device, which is the key peripheral that influences the low-power performance of this device. The following sections help users better understand this peripheral to implement the strategies in [Section 3](#) more easily. The current section describes the low-power related features and attentions in PMCU. For other features, detailed peripheral description, and register control, refer to the related technical reference manual.

### 2.1 Overview

The PMCU contains three submodules to provide this functionality: the power management unit (PMU), the clock module (CKM), and the system controller (SYSCTL).

#### 2.1.1 Power Domains and Power Modes

To realize different power levels, two core power domains are provided on the device: PD1 and PD0 and five operating modes (power modes) are provided to optimize the device power decreasing power: RUN, SLEEP, STOP, STANDBY, and SHUTDOWN. [Figure 2-1](#) indicates what domains are available in each operating mode of the device.



**Figure 2-1. MSPM0Lxx Operating Modes**

A more detailed supported functionality in each operating mode is given in [Table 2-1](#). Users can choose a suitable working condition according to your applications for the limited need to the clock frequency, wake sources, CPU and peripherals requirements.

Abbreviations used in [Table 2-1](#):

EN: The function is enabled in the specified mode.

DIS: The function is disabled in the specified mode, but the function's configuration is retained.

OPT: The function is optional in the specified mode, and remains enabled if configured to be enabled.

NS: The function is not automatically disabled in the specified mode, but its use is not supported.

OFF: The function is powered off in the specified mode, and no configuration information is retained.

**Table 2-1. MSPM0Lxx Supported Functionality by Operating Mode**

Operating Mode		RUN			SLEEP			STOP			STANDBY		SHUTDOWN	
		RUN0	RUN1	RUN2	SLEEP0	SLEEP1	SLEEP2	STOP0	STOP1	STOP2	STANDBY0	STANDBY1		
Oscillators	SYSOSC	EN	EN	DIS	EN	EN	DIS	OPT	EN	DIS	DIS	DIS	OFF	
	LFOSC	EN											OFF	
Clocks	CPUCLK	32 MHz max	32 kHz	32 kHz	DIS								OFF	
	MCLK to PD1	32 MHz max	32 kHz	32 kHz	32 MHz max	32 kHz	32 kHz	DIS					OFF	
	ULPCLK to PD0	32 MHz max	32 kHz	32 kHz	32 MHz max	32 kHz	32 kHz	4 MHz max	4 MHz	32 kHz	DIS	OFF		
	ULPCLK to TIMG0/1	32 MHz max	32 kHz	32 kHz	32 MHz max	32 kHz	32 kHz	4 MHz max	4 MHz	32 kHz	DIS	OFF		
	MFCLK	OPT	DIS		OPT	DIS		OPT		DIS		OFF		
	LFCLK	32 kHz										DIS	OFF	
	LFCLK to TIMG0/1	32 kHz										OFF		
	MCLK Monitor	OPT										DIS	OFF	
PMU	POR monitor	EN												
	BOR monitor	EN											OFF	
	Core regulator	FULL DRIVE						REDUCED DRIVE			LOW DRIVE		OFF	
Core Functions	CPU	EN			DIS								OFF	
	DMA	OPT						DIS (triggers supported)					OFF	
	Flash	EN						OPT			DIS		OFF	
	SRAM	EN						OPT			DIS		OFF	
Peripherals	PD1 Peripherals	OPT						DIS or OFF (peripheral dependent)					OFF	
	PD0 Peripherals	OPT										OPT	OFF	
Analog	ADC	OPT								NS (triggers supported)				OFF
	OPA	OPT	NS		OPT	NS		OPT		NS			OFF	
	GPAMP	OPT										NS	OFF	
	COMP / 8-bit DAC	OPT	OPT (ULP only)		OPT	OPT (ULP only)		OPT		OPT (ULP only)			OFF	
IOMUX and IO Wakeup		EN											DIS with WAKE	
Wake Sources		N/A			ANY IRQ				PD0 IRQ				IOMUX, NRST	

## 2.2 Power Management (PMU)

The power management unit (PMU) generates the regulated core supplies for the device and provides supervision of the external supply. It also contains a bandgap voltage reference used by the PMU and other analog peripherals.

See from [Figure 2-2](#), the power management unit (PMU) can provides supervision of the external supply, which is a common function in low-power application. Besides, digital and analog peripherals can be dedicated controlled to realize different power levels.

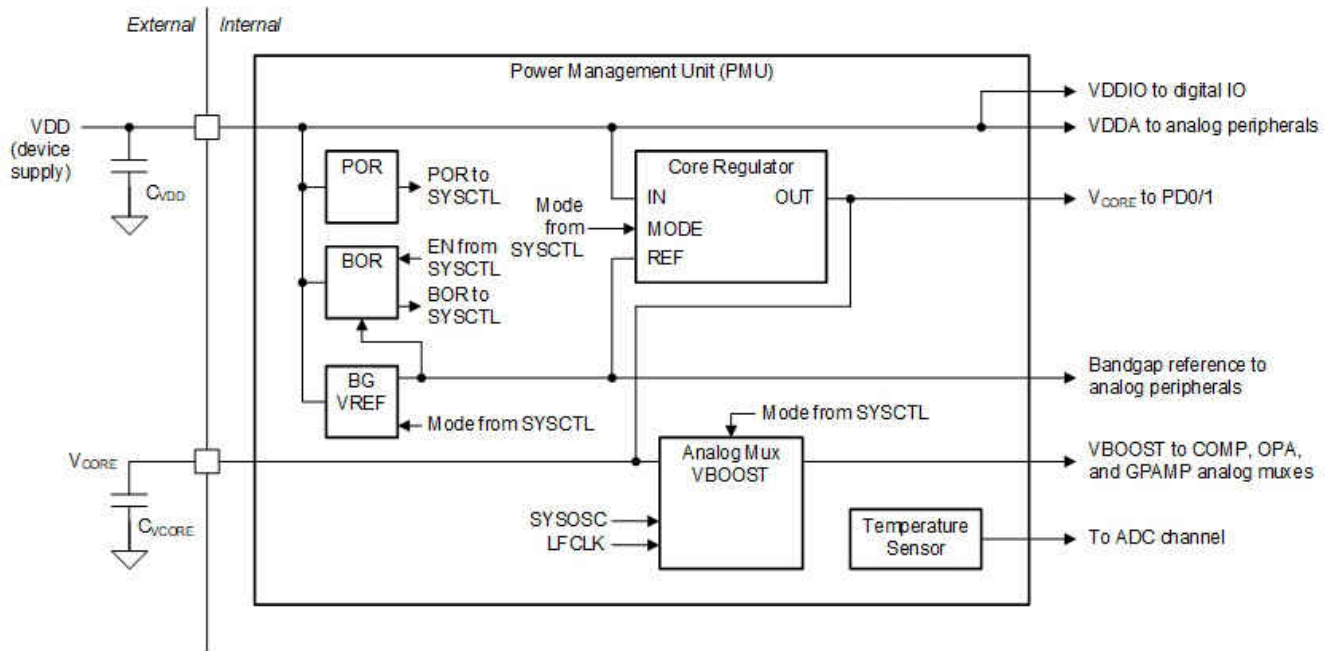


Figure 2-2. MSPM0Lxx PMU Block Diagram

### 2.2.1 Supply Supervisors

Two supply supervisor circuits are available:

The fixed power-on reset (POR) circuit to indicate that the external supply has reached sufficient for the device to run correctly.

The user-programmable brownout reset (BOR) circuit, which ensures that the external supply is maintained at a sufficient voltage to support correct operation of the device, which can be configured to other three BOR levels after startup.

### 2.2.2 Peripheral Power Control

PD0 and PD1 power supply is automatically controlled by SYSCCTL according to the power mode setting. Users do not need to control the power supply manually.

When PD1 peripherals are forced to a disabled state by SYSCCTL upon entry into a STOP or STANDBY mode, most PD1 peripheral configuration settings are retained. See the peripheral-specific chapter in the TRM for details on which peripheral registers are retained.

If a PD1 peripheral was multiplexed to an IO pin (through the IOMUX) in an output configuration, the last valid IO output state is latched upon entry to STOP or STANDBY mode. This feature can be a leakage current source.

### 2.2.3 VBOOST for Analog Muxes

The VBOOST circuit in the PMU generates an internal VBOOST supply that is used by the analog peripherals. It is automatically managed by SYSCCTL module as well.

As the VBOOST circuit has a startup time requirement (12 μs typical), users need to choose between analog peripherals' power-on speed and low static current. Table 2-2 gives the behavior of the VBOOST control.

Table 2-2. VBOOST Control

Mode	VBOOST Enable
ONDEMAND	VBOOST is automatically enabled by SYSCCTL only when a COMP, OPA, or the GPAMP is enabled.
ONACTIVE	VBOOST is forced to be enabled when the device is in RUN or SLEEP mode. VBOOST is also kept enabled in STOP or STANDBY mode if a COMP, OPA, or the GPAMP is enabled.
ONALWAYS	VBOOST is forced to be enabled in all operating modes except SHUTDOWN.

The VBOOST circuit requires a functional clock to operate, which can be the SYSOSC (4 MHz output) or the LFCLK (32 kHz) based on the currently active MCLK/ULPCLK tree source, which is selected automatically by SYSCTL. However, certain VBOOST operating conditions require it to be 4 MHz (sourced from SYSOSC) and not 32 kHz (sourced from LFCLK). Such conditions include:

1. OPA operation or fast mode COMP operation
2. VBOOST is starting up (transitioning from disabled to enabled)

### 2.3 Clock Module (CKM)

The clock module contains the internal oscillators, the clock monitors, and the clock selection and control logic. A frequency clock counter is also provided for checking and/or calibrating the frequency of high-speed clocks against either the LFCLK or a reference period/pulse provided on an IO pin.

Normally, the clock source and clock frequency highly affects the system power consumption. As the SYSCTL controls the clock source switching between different power modes, users only need to select suitable oscillator clock sources for clocks, and use clocks with limited frequency range according to their typical application.

Figure 2-3 shows a simplified block diagram.

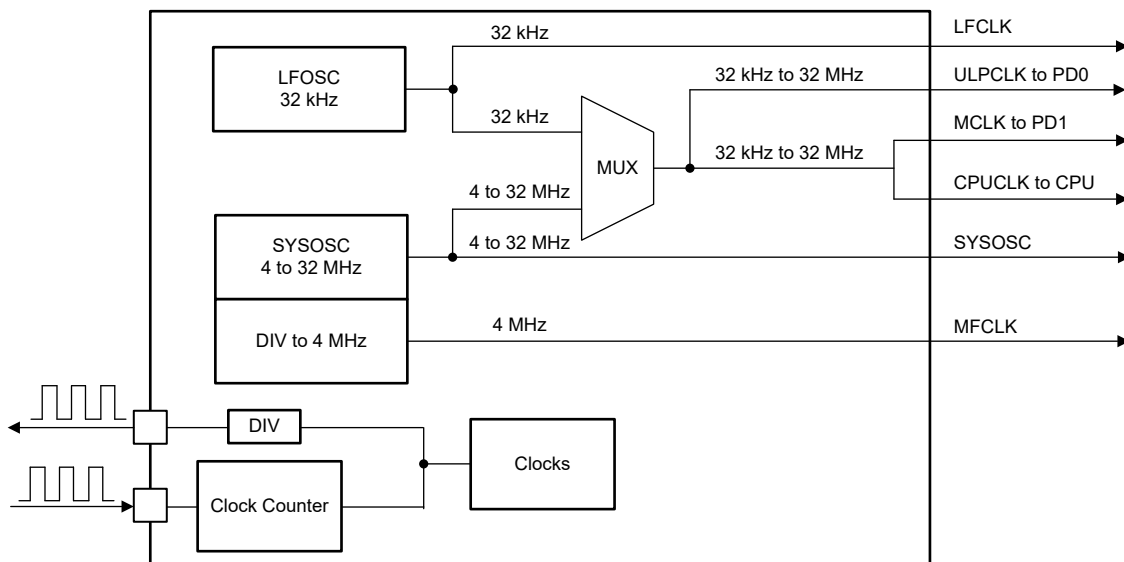


Figure 2-3. MSPM0Lxx CKM Block Diagram

#### 2.3.1 Oscillators

There are only Internal oscillators in the system and no external oscillators to reach a low cost. For the oscillator with low frequency, it is mostly used to help the system reach low power.

Here are the Internal oscillators:

- LFOSC: low frequency oscillator (32 kHz typical frequency)
- SYSOSC: system oscillator (4 or 32 MHz factory-trimmed frequencies, 16 or 24 MHz user-trimmed frequencies) with one clock cycle gear shift ability.

#### 2.3.2 Clocks

The CKM takes oscillator outputs and generates a variety of functional clocks for use by the device. Different oscillator clock source paired with clocks help meet different low-power requirement (see Table 2-3). Users can select clock with suitable clock range for different peripherals.

Table 2-3. Clocks

Clock Name	Frequency Range	Source	Direction
CPUCLK	32 kHz to 32 MHz	LFCLK, SYSOSC	CPU
MCLK			PD1
ULPCLK	32 kHz to 32 MHz		PD0

**Table 2-3. Clocks (continued)**

Clock Name	Frequency Range	Source	Direction
SYSOSC	4 to 32 MHz	SYSOSC	PD1, PD0
MFCLK	4 MHz	SYSOSC	PD1, PD0
LFCLK	32 kHz	LFCLK	PD1, PD0
MFCLK	4 MHz	SYSOSC	PD1, PD0
LFCLK	32 kHz	LFCLK	PD1, PD0
RTCCLK			RTC

### 2.3.3 Asynchronous Fast Clock Requests

Peripherals can be configured to asynchronously assert a hardware request to the SYSCTL for a fast clock source (32 MHz) from SYSOSC, even if the device is operating in STOP or STANDBY mode. This mechanism is ideal for applications where the MCLK/ULPCLK tree is normally sourced from either LFCLK (at 32 kHz) or SYSOSC (at 4 MHz), but a faster clock is temporarily needed to quickly handle a peripheral event peripheral activity. The peripheral support information can get from [Table 2-4](#).

**Table 2-4. Peripheral Support for Asynchronous Fast Clock Requests**

Peripheral	Purpose	Request Source
RTC	Fast CPU wake from RTC event	RTC IRQ to CPU
TIMG0 and TIMG1	Fast CPU wake from TIMG0/TIMG1 event	TIMG0 or TIMG1 IRQ to CPU
GPIO	Fast CPU wake from GPIO event	GPIO activity
Comparator	Fast wake from a comparator event	Comparator event
SPI	Temporarily use fast clock for bit clock generation	SPI activity
I2C	Temporarily use fast clock for bit clock generation	I2C activity
UART	Temporarily use a fast clock for baud rate generation	UART activity
ADC	Temporarily run the SYSOSC to support timer-triggered ADC operation from a low-power mode	ADC

The SYSCTL can be configured to generate an asynchronous fast clock request upon any IRQ request to the CPU with 32-MHz clock rate. This provides the lowest latency interrupt handling when the system is running at the LFCLK rate (32 kHz).

### 2.3.4 Shutdown Mode Handling

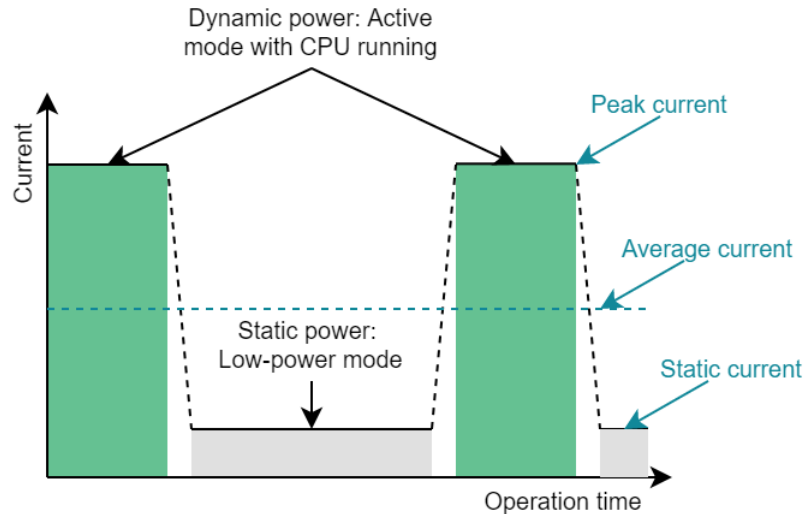
When the device is configured to enter SHUTDOWN mode, the core regulator is powered down and the device register contents and SRAM contents are lost. An exit from SHUTDOWN mode generates a BOR level reset. Two mechanisms are provided to preserve the device state when entering SHUTDOWN mode: IO latching and a small shutdown memory.



## 3 Low-Power Optimization

### 3.1 Low-Power Basics

Before describing how to optimize the power consumption, the following helps the reader to know some basic concept about low power. In MCUs, the power consumption mostly comes from 2 parts: dynamic power and static power (see [Figure 3-1](#)).



**Figure 3-1. Power Consumption Demonstration**

#### Dynamic power

Dynamic power refers to the current when the CPU and most peripherals are running. This is the highest current during the whole operation time, which is called peak current. Power sources can be CPU, flash, RAM, power management unit (PMU), digital peripherals, and analog peripherals. Dynamic power can generally be divided into the current when switching CMOS circuits and the bias current of analog circuits.

Due to the structure of CMOS inverters, the power consumption mainly occurs when the input voltage is switched. This power can be described by the following equation.  $U$  is the system voltage,  $f$  is the switching frequency, and  $C$  is the load capacitance.

$$P = U^2 * f * C \quad (1)$$

From the equation, there are two methods to help improve the dynamic power, considering that the load capacitance cannot be changed. The first is to reduce the system voltage, which is very helpful, but it is limited by the working voltage range of the MSPM0. The designer also needs to check the working voltage of peripherals, which may be higher than the minimum working voltage of the MCU. In addition, some voltage margin should be left for the power supply.

Second, reduce the system frequency. However, this is usually not a good method because a high working frequency requires less working time but the power caused from bias current increases. Usually, the best choice is to complete the task quickly and return to a low-power mode.

#### Static power

Static power consumption mainly refers to the bias current and leakage current of analog circuit and digital circuit when CPU and most peripheral devices are turned off. This will be the lowest current during the whole operation time, which is called quiescent current.

In this mode, analog circuits, such as system controller, PMU and IOs, as well as digital circuits, such as real-time clock (RTC) and watchdog timer, need a certain amount of bias current to maintain the basic functions and wake-up possibility. The best strategy is to turn off unused modules and pair them with the suitable operation frequency and appropriate MCU operation mode.

Leakage current is caused by the non-ideal operation of the MOSFETs used in CMOS devices, especially when the process technology shrinks. Leakage current increases with higher power supply and operating temperature. This part of power consumption depends on the MCU you choose.

### Optimization directions

For some applications, the most cared part is the peak current. It may lie on the limited current supply, like 4-20mA current loop application or the designer wants to fully use the battery capacity, because large current will reach the terminated voltage faster. The focus optimization direction should be the dynamic power range and decrease the peak current.

For most applications, especially for battery implement, the most cared current is the average current, which defines how long the system can operate. The optimization direction should reduce the dynamic power range, peak current and static current. The equation is shown below. The current in that equation means the average current.

$$\text{Average current} \approx \frac{I_{\text{Active}}t_{\text{Active}} + I_{\text{Power-down}}t_{\text{Power-down}}}{t_{\text{Active}} + t_{\text{Power-down}}} \quad (2)$$

As it is not easy to get the average current in different modes, for the more accurate low-cost method, refer to [Section 4.2](#).

## 3.2 MSPM0 Low-Power Feature Use

This section gives a brief description of the device level power-saving features based on MSPM0. For the peripheral configuration, refer to the related TRM. For some detailed load current value, see the device-specific data sheet.

### 3.2.1 Low-Power Modes

Using low-power modes is the most powerful solution to reduce the overall average consumption by keeping the device functions as much as possible in the run time. The key method is to supply different clock sources, clock frequencies and power domain to realize different power level.

The common used approach consists of switching between different power modes (RUN, SLEEP, STOP, STANDBY, and SHUTDOWN) and three lower mode policy options (XX0, XX1, XX2). Remember to leverage between of the application requirements in terms of power consumption, wakeup sources/time and the peripherals simultaneously. For the low-power mode basics, see Section 2.

### 3.2.2 System Clock and Peripheral Operation Frequency

Several clock dividers are used to configure the system clock and peripheral clocks. It is possible to further reduce the power consumption by programming the registers clock dividers to the highest values to provide the minimum required clock frequency.

Reducing the peripheral operation frequency can also help. For example, reduce the ADC sampling and conversion frequency and the UART, SPI, or I2C transmit frequency.

### 3.2.3 I/O Configuration

I/O setting is the most common problem causing unexpected high-power consumption.

For unused pins, the IOMUX configuration should remain in the default (high-Z) state.

For the digital input pin used, please pay attention to the internal pull-up or pull-down resistor. If the input voltage is incompatible, leakage current will be caused. When the input voltage is VCC or VSS, the digital input pin consumes the least power. If the voltage is between VCC or VSS, the internal MOSFET will be biased in the linear region and will consume a lot of power. This usually occurs when unused digital input pins float.

For the digital output pin used, when entering low-power mode, please pay attention that the last valid IO output state may be latched, if no further inversion control. It may cause the additional power consumption.

### 3.2.4 Event Manager

The event manager is a module to transfer digital events from one entity to another. It implements event transfer through a defined set of event publishers (generators) and subscribers (receivers) which are interconnected through an event fabric containing a combination of static and programmable routes.

The event transfer type is shown in [Table 3-1](#). Users can fully use the event peripheral features to reduce the access of CPU.

**Table 3-1. Event Transfer Type**

Event Type	Publisher	Subscriber
IRQ	Peripheral	CPU
DMA trigger	Peripheral	DMA
General trigger	Peripheral	Peripheral

**Using event manager paired with DMA:** It can increase the data processing speed and CPU can go to Sleep mode until the completion of DMA transfer.

**Using event manager paired with peripherals:** It can reduce the CPU injection and realize fast reaction. You can set customized hardware trigger routing according to your application, like timer triggers ADC to sampling or comparator triggers GPIO to turn off.

### 3.2.5 Analog Peripheral Low-Power Features

MSPM0 devices have peripherals with particular power features that allow to design and develop low-power applications while maintaining a high flexibility. For the detailed parameters, refer to the data sheet.

#### ADC

The ADC has 12-, 10-, and 8-bit analog-to-digital conversion mode. Choose the low conversion mode can use fewer conversion cycles and save the power.

The ADC has two power down modes, configured by PWRDN. The first is that the ADC automatically power down at the end of a conversion and when the next sample signal is not required to be asserted immediately. The second mode is that the ADC stays powered on when the peripheral is enabled. Using the default mode uses less current.

#### COMP

The comparator on MSPM0 has two power modes. Fast mode can get highest reaction to signal change. Lower-power mode can get a balance between functionality and power consumption. Please pay attention that the default configuration of comparator is in the fast mode.

#### OPA

The OPA is a zero-drift chopper-stabilized operational amplifier with a programmable gain stage. In addition to output current, there are two configurations for high performance that also affect current consumption as shown in [Table 3-2](#). With the default low-performance setting, the OPA consumes less power.

**Table 3-2. OPA Operation Modes**

Parameter	Configuration Options
Rail-to-rail input (RRI)	Enabled / disabled
Gain bandwidth product (GBW)	6 MHz (STD mode) / 1 MHz (LP mode)

#### GPAMP

The GPAMP is a chopper-stabilized general-purpose operational amplifier with rail-to-rail input and output. The rail-to-rail input range influences the current consumption. With the default low-performance setting, the GPAMP consumes less power.

### 3.2.6 Run Code From RAM

Move a part of code from flash to RAM can also help save the power. First, the code can run faster in RAM. For MSPM0, the RAM clock is the same as CPU clock, up to 80 MHz. However, flash wait states are normally needed. Second, the code run in RAM can require less  $\mu\text{A}/\text{MHz}$ . Find more details in the device-specific data sheet.

## 3.3 Software Coding Strategies

### Conditional code execution

One commonly used software structure is to use a conditional wake-up and code execution structure. A common wake-up source is RTC. If the MCU does not have RTC, the application can use TIMG as a replacement. Find the static current information in the data sheet.

### Non-blocking programming

In low-power applications, it is necessary to avoid blocking mode by using while loop. Using non-blocking mode paired with interrupts and polling to wake up the CPU to process tasks is more acceptable. Although it will sacrifice time and resources to wake up and respond to interrupt sources, push and pop the stack, it can save more power.

### Optimize code size

Small code size means that CPU needs less time to finish tasks in active mode. The easiest way is to choose a highly optimized compiler such as TI Arm Clang Compiler and leverage the supplied code optimization levels. You can also spend some time or coding skills to realize this aim.

## 3.4 Hardware Design Strategies

### MCU power supply

The MSPM0 supports operating across 1.62 V to 3.6V. To reach lower power consumption, Users can supply a voltage no lower than 1.62 V to MCU. However, attention need to be paid to some peripheral working limit, such as internal reference. When power lower than 2.7 V, only 1.4-V reference can be used and 2.5-V reference cannot be used.

### Resistors

Resistors are common used in the circuit, which is for current limitation or voltage divider. Please ensure the selected resistor value is large enough, after considering the drive strength and voltage setting time. More attention needs to be paid for the voltage divider, which may have constant leakage current. For some low-cost application, you can use the controllable GPIO as the voltage source.

### Capacitors

More types and sizes of capacitors will be used in the circuit than resistors. All capacitors will have a small leakage current loss. Aluminum electrolytic and tantalum capacitors can provide a large capacitance with a large leakage current, which typical value is  $\mu\text{A}$  level and closed to the standby current of MSPM0. While ceramic, foil, capacitors have small capacity and leakage currents, which typical value is  $\text{nA}$  level.

Generally, capacitors with high capacitance values tend to have higher leakage current. capacitors with higher voltage rating have lower leakage but also have less capacitance in relation to package size. Besides, applying high voltage and high working temperature will also increase the leakage current. Remember to refer to the capacitor data sheet, in the hardware design.

However, there is still some cases to typical use large capacitors. If the users care more about the peak current, large capacitors can be used to cut down peak current and spread it to static current. In this condition, users need to trade off the peak current against the static current.

### Power IC

Switching regulators can reach high efficiency at heavy load. However, they have a low efficiency and normally with high power noise at weak load. For linear regulators, its efficiency relies on the input and output voltage setting and it has less power noise and cost. For some battery applications, if the battery input voltage all

covered by the MCUs, removing power IC could be a good choice. Users need to do the selection according to their application.

### **Crystals**

Normally, using an external low frequency 32-kHz crystal can reach lower power consumption than using the internal one in lower power modes. When using external crystals, remember to refer to the crystal data sheet for appropriate load capacitor values and layout rules. Improper capacitors will cause crystal frequency shift and even vibration initiation failure.

For using external high-frequency crystal, as it is not possible to disable SYSOSC when using a different high frequency clock to source MCLK (such as HFCLK or the PLL), the power consumption will even be higher than using internal high frequency crystal. This is because SYSOSC is used by SYSCTL logic when MCLK is sourced from HFCLK or the PLL.

## 4 Power Consumption Measurement and Evaluation

### 4.1 Current Evaluation

Current evaluation is a common requirement in low-power application, which can be used to evaluate the system running time and select the suitable power IC. There are two ways to evaluate the current.

First, use the configuration from SysConfig and estimate the current consumption. This feature will be included for MSPM0 in the future. You need to input the ADC conversion frequency, op-amp power mode, clock frequency, and so on. It outputs the following values:

- Average current, power
- Peak current, power
- Estimated battery life

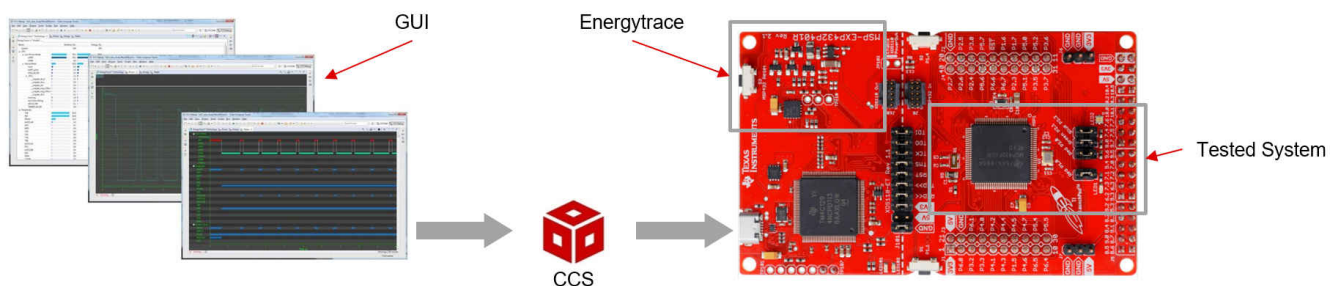
Second, you can calculate the current according to the parameters in the data sheet. First, refer to the *Supply Current Characteristics* section in the data sheet to know the basic MCU current consumption under certain power modes. Second, refer to the specifications for all peripherals that are used to find the current consumption for each. Third, add these values together to calculate the total current consumption.

### 4.2 Current Measurement

The common used tools are the multimeter and oscilloscope. For normal multimeter, it may be an easiest tool to evaluate the power consumption, but it is limited by its low sampling speed and no time stamps. For the oscilloscope, it is a good tool to detect the maximum power consumption combined with the current clamp or simple resistors. But it is hard to detect  $\mu\text{A}$  level current, because of its resolution and the common noise across the space.

#### 4.2.1 Current Measurement

EnergyTrace™ technology is a power analyzer hardware tool to detect average power consumption paired with Code Composer Studio IDE (CCS). It is exported through a USB bulk endpoint in the XDS110 Debug Probe on every MSPM0 LaunchPad™ development kit. The work system is shown as [Figure 4-1](#).



**Figure 4-1. EnergyTrace Technology Work Flow**

EnergyTrace implements a new method for measuring current consumption for ultra-low power currents. A software-controlled DC-DC converter generates the target power supply and keeps it regulated via a train of pulses. The density of the DC/DC converter charge pulses is proportional to the energy consumption of the target microcontroller. Because the width of each charge pulse remains constant, the Debug Probe simply counts each charge pulse and then sums them over time to calculate the average current. The accuracy of measurements is controlled by a built-in calibration circuit in the debug probe, which quantifies the energy equivalent of a single charge pulse. Using this approach, even the shortest device activity that consumes energy contributes to the overall recorded energy. The [XDS110 user's guide](#) and [EnergyTrace technology](#) website contain additional details.

Table 4-1 lists the basic electrical specifications.

**Table 4-1. EnergyTrace Technology Electrical Specifications**

Range	Accuracy	Observations
I < 25 mA	±2% ±500 nA	VBUS = 5 V constant during and after calibration
I > 25 mA and I < 75 mA	±5% ±500 nA	VBUS = 5 V constant during and after calibration

## 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](http://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 © 2022, Texas Instruments Incorporated