ABSTRACT

With the growing system complexity in ultra-low-power microcontroller applications, minimizing the overall energy consumption is one of the most difficult problems to solve. Multiple aspects including silicon, other onboard hardware components, and application software must be considered. There are some obvious generic techniques that can be used to reduce energy consumption such as reducing operating voltage or frequency. Many of these generic techniques may not greatly reduce energy consumption independently, but taken as a whole, the results can be significant, as there are many interdependencies across these components.

This application report offers an overview of various low-power features of the industry’s leading ultra-low-power microcontroller and, more importantly, how to exploit optimal combinations of these features to achieve the lowest energy consumption for a given application. This application report uses the industry-standard ULPBench™ benchmark from EEMBC as a case study to map individual and combinations of low-power features on the MSP432™ platform to a relevant application scenario.
1 Power Optimization Vectors on a Microcontroller

The MSP432 microcontrollers integrate a number of power enhancements that include lower active power consumption, low-power (LP) mode current consumption, and more efficient peripherals with lower current consumption. In addition, the MSP432 devices also provide a number of options and power configurations that enable developers to further optimize the power consumption for a specific MSP432 application. Nonetheless, for any given microcontroller, there are a number of vectors that can be used to optimize the power consumption of the device.

[Vector 1] Reduce operating voltage

Power is the product of voltage and current. For a given application system that consumes a certain amount of current, lower supply voltage can help reduce the power consumption, with the constraint that minimum voltage requirement is met. This requirement can sometimes be related to the microcontroller supply voltage level itself (for MSP432P4xx minimum $V_{CC} = 1.62$ V) or the setting the internal voltage regulator to appropriate voltage level. It could be dependent on the CPU operating frequency or a minimum voltage required by a certain peripheral to be fully functional. For example, if a 2.5-V reference voltage is required by the ADC in the application, supply voltage must be greater than or equal to 2.5 V.

[Vector 2] Reduce the operating frequency

As can be deduced from any microcontroller data sheet, power and current consumption are directly proportional to the operation frequency. In many scenarios, higher operating frequency means that the CPU can execute code and complete the task faster. However, in certain real-time scenarios, many application activities are timing-dependent or event-driven. Having the CPU running faster in an idle loop waiting for a certain event to trigger or waiting for serial data to come in at a lower baud rate can consume additional power that can be saved. For these scenarios, either putting the device into a low-power mode or reducing the operating frequency might be worth investigating.

[Vector 3] Maximize sleep time

A typical low-power embedded application spends most of its lifetime in two modes: active mode executing meaningful tasks and low-power mode with minimal activity other than time-keeping or waiting for an interrupt or event to wake up to active mode. Low-power mode can consume much less current than active mode—on the MSP432P4xx family, LPM3 current can be as low as approximately 750 nA while active mode current can be up to several milliamps. Minimizing active time and maximizing time spent in low-power modes can significantly reduce the overall current consumption. This can usually be achieved by minimizing the active task, optimizing the active code, or increasing the operating speed in active mode.

[Vector 4] Minimize transition time

In addition to time allocated for active and low-power modes, some applications might unknowingly spend a considerable amount of time transitioning between these various power modes. If the transitions times go unnoticed, they might contribute to the total energy consumption increase. One of the power optimization activities is to identify all of the system transitions and determine if any of them can be reduced or removed.

[Vector 5] Solving the intermodule dependencies

The previous vectors are all possible options to individually optimize. However, on a given microcontroller platforms, these vectors might have some interdependencies, or optimizing on a particular vector might negatively affect another. For example, reducing operating frequency could potentially increase active duty cycle. Therefore, it is even more important to take into account the intermodule dependencies and determine optimal combinations of settings for a given platform.

The remaining sections of this application report describe the system peripherals of MSP432 MCUs, identify critical features and options, correlate them back to one or more of the five vectors above, and determine how they can help to optimize power consumption. The key core peripherals on MSP432P4xx devices include: power system [Power Controller Module (PCM)], clock system, and memory system (flash, SRAM, and ROM). More importantly, the intermodule dependencies specific to MSP432 are discussed to help show the combinations that makes sense for an MSP432 application.
2 Power System

Many MCUs scale performance with voltage; as the voltage drops, the MCU core operates at a reduced frequency. This can increase energy consumption if the MCU must stay awake longer to perform an equivalent amount of work. With MSP432 MCUs, however, the core is able to operate at relatively good speed at the lowest voltage. Thus, operating at a lower voltage gives a true and full improvement in energy efficiency. [Vector 1]

2.1 CPU and Digital Logic Core Voltage

The MSP432P4xx family of devices requires a secondary core voltage ($V_{\text{CORE}}$) for its internal digital operation in addition to the primary one applied to the device ($V_{\text{CC}}$). In general, $V_{\text{CORE}}$ supplies the CPU, memories (flash and RAM), and other digital modules, while $V_{\text{CC}}$ (connected to the DVCC and AVCC pins of the device) supplies I/Os and all analog modules (including the oscillators). The $V_{\text{CORE}}$ output is maintained using a dedicated voltage regulator. $V_{\text{CORE}}$ is programmable with two predefined voltage levels, and each level is restricted to particular maximum operating frequency. This provides programmable dynamic voltage frequency scaling (DVFS) operation.

This directly correlates to [Vector 1], because the digital logic of the system is now supplied with a lower voltage, it consumes less power than on a microcontroller where the entire system operates out of the higher $V_{\text{CC}}$ supply rail.

Furthermore, if the application operating frequency does not require the higher $V_{\text{CORE}}$ voltage, reducing the $V_{\text{CORE}}$ voltage level further reduces the power consumption of the device. For example, specifically for MSP432P4xx family, if maximum operating clock frequency is ≤24 MHz, the VCORE0 option can be chosen, which is 1.2 V (typical) compared to VCORE1 which is 1.4 V (typical). VCORE0 provides better power consumption at a given frequency (≤24 MHz) than VCORE1.

2.2 Regulators: LDO and DCDC

To generate the secondary core voltage ($V_{\text{CORE}}$) from the primary one ($V_{\text{CC}}$), the MSP432P4xx family uses two regulators: a low-dropout voltage regulator (LDO) as the default regulator, and an inductor-based DC-to-DC step-down switching regulator (DC-DC) as a secondary or optional one. The LDO and DC-DC are connected in parallel. The two modules share a common output (the digital supply $V_{\text{CORE}}$ rail) and a common reference or target voltage.

For the majority of microcontrollers with multiple voltage rails ($V_{\text{CC}}$ and $V_{\text{CORE}}$), the LDO regulator is typically used due to a number of advantages such as cost-effective, relatively noise-free output due to no switching component, faster to ramp up and down from low-power modes, among others. These advantages might be critical for certain application with stringent requirements related analog or RF performance or cost-sensitivity. However, the LDO regulator might not be ideal for power savings, because current drawn by supply is same as that of the current drawn from LDO: $I_n = I_{\text{load}} - V_{\text{core}}$.

On the other hand, the DC-DC regulator's main advantage is the significant power saving compared to the LDO. Unlike a linear regulator, which generates a voltage rail from another by dropping the extra voltage across a linear, effectively passive, element (thereby wasting power equivalent to the voltage dropped multiplied by the load current drawn), the DC-DC efficiently generates one voltage rail from another using active (inductor and capacitor) elements. In an ideal DC-DC, power drawn from the output rail ($V_{\text{CORE}} \times I_{\text{load}}$) equals the power drawn from the input rail ($V_{\text{CC}} \times I_n$). $I_n$ is therefore smaller than $I_{\text{load}}$ by the fraction $V_{\text{CORE}} / V_{\text{CC}}$. After accounting for efficiency losses, the DC-DC typically has efficiency in the range of 75% to 90% for moderate to maximum loads. This allows the DC-DC regulator to provide significant power saving of up to 45% compared to the LDO regulator. Table 1 shows the current consumption comparison between the DC-DC and LDO when the system operates at 24 MHz and 48 MHz.

<table>
<thead>
<tr>
<th>Regulator</th>
<th>CPU Frequency = 24 MHz</th>
<th>CPU Frequency = 48 MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDO</td>
<td>3950 µA</td>
<td>7600 µA</td>
</tr>
<tr>
<td>DC-DC</td>
<td>2200 µA</td>
<td>4600 µA</td>
</tr>
</tbody>
</table>
While the DCDC power saving for active mode is apparent, the LDO's advantages and flexibility might be required by the application in certain scenarios, as explained previously. The LDO is also the default regulator that is used upon device startup or after returning to active mode from low-power mode. So any DC-DC regulator usage should also take into account the transition time from LDO to DC-DC as well as DC-DC back to LDO prior to going back to low-power mode.

There are several additional advantages/drawbacks for both regulators. Figure 1 summarizes the advantages and disadvantages of the LDO and DC-DC to help developers determine how and when it is possible to use the DC-DC regulator to maximize power saving.

Typically, developers must choose between using either an LDO or DC/DC convertor for all use cases. By providing two options for power regulation, developers can dynamically optimize regulation based on the current mode of operation. When the system is in a standby mode of operation, for example, the LDO can be used to minimize wake time. For operating modes or use cases where active current plays a larger role in power consumption, the DC/DC convertor can be used. For more details on DC-DC usage as well in-depth analysis on its advantages and tradeoffs, refer to Maximizing MSP432P4xx Voltage Regulator Efficiency: DC-DC and LDO Features and Tradeoffs (SLAA640).

![DC-DC and LDO Regulator: Features, Advantages, and Tradeoffs](image)

**Figure 1. DC-DC and LDO Regulator: Features, Advantages, and Tradeoffs**

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

### 2.3 Low-Power Modes

The MSP432P4xx family provides a number of low-power operating modes that originated in the MSP430™ devices including LPM3, LPM4, LPM3.5, and LPM4.5. Similar to earlier MSP devices, these low-power modes on MSP432P4xx consume extremely little power, putting the MSP432P4xx in a great position for power saving.

In addition to the traditional MSP low-power modes, the MSP432P4xx family introduce another class of low-power modes that can provide additional power saving: low-frequency power modes. The low-frequency modes are special low-power low-frequency options that can be used in conjunction with both active and LPM0 modes. In these low-frequency modes, memory and peripherals can execute at maximum speed of 128 kHz. Because the regulator can reduce the drive strength to supply minimal current to drive the entire low-frequency system, the device's total current consumption can be as little as <80 µA when using these modes.

Refer to Leveraging Low-Frequency Power Modes on MSP432P4xx Microcontrollers for more details on these special power modes (SLAA657).
Clock System

As described in Section 1, power and current consumption is directly proportional to the frequency of operation (\(f\)). Naturally if \(f\) is reduced, current consumption also reduces linearly. But, reducing frequency may reduce overall system throughput or may increase the overall energy consumption. The reason is that the device may have to be in an active operating mode (AM_VCORE0 or AM_VCORE1) for more time and in low-power mode for less time. Always operating at high frequency to complete an active mode task also may not help, because of the peak current requirements during such short bursts, which in turn can affect the battery capacity and hence battery life. Therefore, the optimal selection of clock frequency for system operation is critical to optimize the energy consumption of the system.

The Clock System (CS) provides options such as choices of different oscillators for different clock sources, dynamic selection, and clock dividers to enable optimal power consumption.

The digitally controlled oscillator (DCO) provides options such as changing frequency on the fly and fine tuning to an intermediate frequency of 0.98 MHz to 52 MHz, minimal current consumption of <160 µA at 24 MHz and <220 µA at 48 MHz. The maximum DCO clock frequency drift with temperature in external resistor mode is ±35 ppm/°C, and the drift for internal resistor mode is ±250 ppm/°C. An application can program the DCO for the best combination of optimal power consumption and required accuracy. If higher accuracy is needed, the application needs to use the on-chip high-frequency crystal oscillator (HFXT) with the trade-off being higher current consumption.

The CS also provides flexible clock source and frequency selection for different on-chip clock sources (MCLK, SMCLK, and ACLK). It can also divide the clocks dynamically, and all of this can be done with one control register (CSCTL1).

For example, DCO can be used as the main clock source for MCLK and SMCLK, HSMCLK. LFXT can be used as the clock source for RTC. For slower peripherals such as I²C, the DIVS and DIVHS register bits can be configured to select slower source clock frequency. For ADC, MODOSC can be used as a quick ON/OFF oscillator that remains enabled only when ADC conversion operation is ongoing.

The ULPBench example in Section 5 uses the DCO as the main clock source at 24 MHz instead of at 48 MHz for optimal current consumption.

Refer to the CS section in the MSP432P401xx microcontroller technical reference manual, data sheet, and the application report Multi-Frequency Range and Tunable DCO on MSP432P4xx Microcontrollers (SLAA658) for further details.

Memory Execution Optimization

Although typical code execution involves opcode and literal fetches from memory, it may also involve continuous and intermittent data writes to and reads from stack (SRAM), function calls from software library provided in flash or ROM, data writes to or reads from peripherals, DMA data transfers, and other operations. A significant portion of the power consumption in a typical MCU originates from code execution from a nonvolatile memory such as flash. Hence, it is beneficial to optimize memory accesses to reduce as much power consumption as possible during code execution.

Flash

MSP432P401xx has flash as the main nonvolatile memory for application program code and data.

Wait State and Clock Dependency

Low-power flash memory is typically slower than the high-frequency clocks used at system level such as CPU or DMA. Therefore, at higher operating frequency, flash memory requires wait-stated accesses, which means that the CPU is halted for certain number of clock cycles depending on the CPU-to-flash frequency ratio. Refer to the device data sheet for details.
4.1.2 Data and Instruction Buffers

To offer optimal power consumption and performance across predominantly contiguous memory accesses, the flash controller offers a read buffering feature. If read buffering is enabled, the flash memory is always read in entire 128-bit chunks, even though the read access may only be 8, 16, or 32 bits wide. The 128-bit data and its associated address is buffered by the flash controller, so that subsequent accesses (expected to be contiguous in nature) within the same 128-bit address boundary are serviced by the buffer. Using this scheme, the flash accesses have wait states only when the 128-bit boundary is crossed, while read accesses within the buffer’s range are serviced without any bus stalls. If read buffering is disabled, accesses to the flash bypasses the buffer and the data read from the flash is limited to the width of the access (8, 16, or 32 bits). Each bank has independent settings for the read buffer. In addition, within each bank, the application has the flexibility of enabling read buffering either for instruction fetches only, for data fetches only, or for both.

4.1.3 Flash Power Consumption

Flash power consumption is dependent on the number of bits read. If 128 bits are read, power consumed is higher than if 32 bits are read but not four times the power of 32-bit read power. So for contiguous or linear code execution or data fetches, TI recommends enabling the buffer to enhance performance and reduce power consumption.

4.1.4 What Options to Choose for Code Execution From Flash

- Enabling buffer definitely gives performance advantage but whether it can help for getting better power consumption or not depends on the linear or nonlinear execution of the code or data reads. If an application needs higher throughput always at the cost of current consumption, buffers can always be enabled. With wait-states >0 configured for flash, buffers are recommended to be always enabled for higher throughput.
- In general, TI recommends enabling instruction and data buffers if the code execution or data reads is in linear fashion to prevent the controller from constantly accessing flash and ultimately reduce to reduce overall current consumption (refer to Section 4.1.2).
  - Specific example is when the code is running with 0-wait states, there is no performance or throughput advantage. In that case, the code profile really determines whether to enable the buffer (with a power advantage for linear code execution) or to disable the buffer.
- Because the buffer configuration is dynamic, it can be enabled or disabled for specific code sections or function calls. If there is a function with significant size with mostly linear code execution, TI recommends enabling and disabling the buffer dynamically before and after the function call respectively.
- It is also recommended to check the current consumption for your specific application and determine specific combinations of settings that work best for that scenario. MSP432P4xx family of devices support EnergyTrace™ technology, which can enable developers to construct a detailed energy profile of the system and correlate back to areas in the code with high contribution to energy consumption. Based on the findings, developers can leverage capabilities provided by MSP432P401xx to dynamically change wait-states, enabling/disabling buffers which can be chosen to cater to specific application needs.
- Refer to ULPBench results for an example analysis on how and when to use flash buffer.

4.1.5 Flash Access Benchmark Registers (*BMRK* Registers in Flash Controller)

The flash controller offers two counters for application benchmarking purposes. This feature is extremely useful for monitoring the number of flash accesses. Using these counts, the application can manage the optimal wait-states and the buffer enable or disable settings as discussed in previous sections.
- Instruction fetch benchmark counter (32 bits) – Increments on each instruction fetch to the flash
- Data fetch benchmark counter (32 bits) – Increments on each data fetch to the flash
4.2 **SRAM**

SRAMs are standard read-write memories on MCUs. Particularly on MSP432P401xx, SRAMs work at the same speed as the CPU clock frequency so code execution from SRAM clearly gives a performance and throughput advantage compared to code execution from flash. There is a power consumption advantage for SRAM compared to flash as well. Power consumption (commonly measured in μA/MHz) for SRAM is significantly lower than the μA/MHz consumption of flash memory. Therefore, TI recommends executing small loops or functions that are frequently used from SRAM instead of flash for performance and power benefits.

This may be handled by using a smart function to copy a function or subset of code from flash to SRAM and execute from SRAM whenever needed. This SRAM bank can be kept enabled during LPM3 mode so that the contents are retained and copying the same function over is not needed after LPM3 wakeup.

**SRAM retention**

- **SRAM_BANKRET** in the SYS module controls the retention of SRAM contents in LPM3 mode.

  Depending on the application, either ALL or a FEW banks are programmed to be retained in LPM3 mode which can affect the current consumption in LPM3.

4.3 **ROM**

In addition to flash, the MSP432P4xx also provides read-only memory (ROM) as a second nonvolatile memory option. As the name indicates, the ROM content is programmed when the device is manufactured, and after that memory contents can only be read or executed but not modified.

MSP432P4xx provides 32Kbytes of on-chip ROM including the built-in peripheral driver library (DriverLib) application program interfaces (APIs) to provide developers with a highly-abstracted set of APIs to exercise various functionalities of the MSP432P4xx peripherals. The easy-to-understand ROM DriverLib APIs allow developers to speed up the process of learning how to configure peripheral registers, change power modes, and other standard tasks.

There are a number of benefits when using ROM DriverLib. First, the DriverLib in ROM is thoroughly tested and optimized, and it represents the TI-recommended procedures to exercise the peripherals. In the context of this application report, executing out of ROM yields both higher performance (0 wait state access to ROM) and better power consumption (much lower than flash execution, and even better than SRAM). Both of these performance increases, faster execution time and lower power consumption, directly result in minimizing the energy consumption of the device. [Vector 3]

Maximizing the driver library use helps to reduce flash memory accesses and enable better power consumption.

Other than the power benefit, there is an indirect benefit of using software driver library. Leveraging DriverLib in ROM also frees valuable flash memory space for application code. Shifting this additional approximately 31Kbytes of memory towards the high-level application code for intelligent software to uniquely differentiate the application is a much better allocation than taking up the memory for routine peripheral configuration tasks that, if implemented perfectly, would offer only marginal improvement over the TI-provided ROM solution.
5 Optimizing Power of an Entire MCU System

Now that we have examined several modules within the MSP432 microcontroller system and how they can be configured to optimize and reduce power consumption, let’s take a look at how these components can be used together and any potential dependencies or constraints they might have on each other. ULPBench by EEMBC is used as a case study to understand possible trade-offs.

5.1 Is Faster Always Better?

The MSP432 microcontroller as a whole is a heavily clock-gated system, as the CPU and operating clock frequency can ultimately have a significant effect not just on the clock power consumption but also how much power required to sustain other peripherals. As a matter of fact, multiple peripherals’ power consumption can be throttled or boosted depending on the CPU frequency requirement.

Furthermore, the flash memory operation is limited to 12 MHz. Running the CPU at any frequency faster than 12 MHz requires an appropriate amount of flash wait-states to stall the CPU until the flash memory content can be completely fetched.

Combining these dependencies, Table 2 summarizes the effects of increasing the system operating frequency.

Table 2. Effects of Increasing System Operating Frequency

<table>
<thead>
<tr>
<th>System Frequency</th>
<th>VCORE</th>
<th>LDO/DC-DC</th>
<th>Flash Wait States</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 kHz to 128 kHz</td>
<td>0</td>
<td>LDO (LF mode)</td>
<td>0</td>
</tr>
<tr>
<td>128 kHz to 12 MHz</td>
<td>0</td>
<td>LDO/DC-DC</td>
<td>0</td>
</tr>
<tr>
<td>12 MHz to 16 MHz</td>
<td>0</td>
<td>LDO/DC-DC</td>
<td>1</td>
</tr>
<tr>
<td>16 MHz to 24 MHz</td>
<td>0</td>
<td>LDO/DC-DC</td>
<td>1</td>
</tr>
<tr>
<td>24 MHz to 32 MHz</td>
<td>1</td>
<td>LDO/DC-DC</td>
<td>1</td>
</tr>
<tr>
<td>32 MHz to 48 MHz</td>
<td>1</td>
<td>LDO/DC-DC</td>
<td>2</td>
</tr>
</tbody>
</table>

5.2 Speed Does Not Always Improve Energy Efficiency

Running the CPU at 48 MHz in can result in a lower ULPMark score than running it at 24 MHz. At 48 MHz, workload is definitely completed faster than at 24 MHz, but core voltage must be increased and there are mode transition latencies involved that can result in lower ULPMark scores. This is described in the following sections as part of the effect due to transition latencies. In Figure 2, 3-, 12-, and 24-MHz scores are with 0 wait states, and 48-MHz scores are with 2 wait states.

Figure 2. MSP432P4xx ULPBench Results

---

**Designing an Ultra-Low-Power (ULP) Application With MSP432™ Microcontrollers**

www.ti.com

Copyright © 2015, Texas Instruments Incorporated
5.3 Efficiency of Flash Instruction and Data Buffers and Correlation to Wait States

128-bit instruction and data buffers for flash reads can be used for better throughput and power consumption if code execution is linear. If execution is not linear, there may be overhead due to reading 128 bits by using more power and discarding some of the words. MSP432P4xx provides flexibility to enable or disable the buffers dynamically, and software can optimally use this feature depending on the linear or nonlinear nature of the code execution (see Figure 3).

The 158.6 score with the buffer disabled is higher, which implies nonlinearity of the code at 0-wait state condition. But with higher operating frequency and with 1-wait state, buffer disabled gives a better score of 155.8 at 24 MHz.

![Figure 3. MSP432P4xx ULPBench Results](chart)

5.4 $V_{\text{CORE}}$ Switching Latency Diminishes Energy Saving

Code execution of ULPBench workload function during active mode of MSP432P4xx takes insignificant number of cycles (<0.06%) with high optimization of IAR compiler at 24 MHz, 1-wait state condition compared the time spent in LPM3 mode (approximately 99.94%). So any additional time it spends in active mode can severely hamper the ULPBench score.

For example, if the lowest power consumption is needed in LPM3 mode with VCORE0 setting, and the highest performance is needed at 48 MHz, then there can be many mode transitions involved as per the guidelines in the MSP432P4xx Technical Reference Manual (SLAU356).

AM_LDO_VCORE0 → AM_LDO_VCORE1 → AM_DCDC_VCORE1 → AM_LDO_VCORE1 → AM_LDO_VCORE0 → LPM3_VCORE0

For example, additional transitions can be avoided if higher performance is needed as soon as the device wakes up from LPM3 mode.

AM_LDO_VCORE0 → AM_LDO_VCORE1 → AM_DCDC_VCORE1 → AM_LDO_VCORE1 → AM_LDO_VCORE1 → LPM3_VCORE1

In both the cases, transition time can be significant and can affect the duration for which the device is in active mode, which affects the ULPBench score. So it is important to understand four aspects from the application's standpoint.

- Duration spent in active operational mode by an application
- Duration spent in low-power mode by an application
- Duration spent in transitioning between different active modes
- Duration spent in transitioning between active and low-power modes
Another reason for results not being so attractive for configurations involving these mode transitions is that the transition functions are written in such a way that the CPU waits for the mode transition to be complete before the actual code execution starts. For example, in the condition of 24 MHz, 1 wait state, buffer enabled, the ULPBench score is 155.8. If the LPM3\textsubscript{VCORE0} \rightarrow AM LDO\textsubscript{VCORE0} transition function is changed to initiate the mode transition and continue with the rest of the code execution without waiting for PMR\_BUSY flag to be set, the score can go as high as 167.4.

Figure 4. MSP432P4xx ULPBench Results

6 Summary

Section 1 described generic vectors to reduce power consumption on a generic microcontroller design, and the rest of the sections described how these can be easily achieved on the MSP432 platform:

- Choose the right combination of CPU frequency and flash wait states for optimal performance. Refer to the data sheet for additional information.
- Use features that allow 128-bit instruction and data buffer in flash controller to be enabled and disabled dynamically according to linear or nonlinear nature of code execution from flash memory. This can affect performance and power consumption.
- Choose optimal operating voltage point to balance performance and power consumption
- Optimize different mode transition latencies by minimizing the transitions depending on the time duration spent in each mode

7 Other Methods to Optimize Device Power Consumption

7.1 Optimize Device Power Consumption

- Terminate and configure I/Os as soon as possible after application begins. By default, all of the I/Os are in input mode. Depending on the application, relevant I/Os should be driven externally, should be internally or externally pulled up or down, or should be programmed in output mode with value driven as 0 or 1.
- Use smart module enable and disable features to consume power only when module is needed to be active. For example, use the UCSWRST bit in eUSCIA and eUSCIB modules to keep the module in reset or active.
- Use an optimal clock source. MODOSC or SYSOSC are also automatically enabled to provide MODCLK or SYSCLK to ADC14 when needed and are disabled when not needed for ADC14 or for rest of the device. Use these as ADC14 clock sources for optimal ADC operation.
7.2 Leverage Other MSP432 ULP Tools

The MSP432 microcontroller ecosystem also provides a unique and powerful set of ultra-low-power tools and software that developers can use to further optimize the application power consumption. Starting with a well-designed embedded systems using silicon and hardware knowledge such as from this application report, embedded developers can then take advantage of static code analyzer tool such as ULP Advisor to further identify areas in the code to optimize for execution time and to reduce power consumption. Next, during debug phase, developers can use EnergyTrace technology to construct an energy profile of the application and correlate critical energy consumption spikes to areas in the code. This allows developers to identify high-energy components in the configuration or software for further power improvement.

8 Disclaimer

- ULPBench measurements, current consumption measurements were done on limited number of units which were from a specific manufacturing lot. Results measured on a LaunchPad™ development kit or with different units may vary.
- External board components can also affect current consumption. For example, LFXT current consumption can change significantly depending on the external capacitors.
- Refer to the electrical specification in the data sheet for various parameters and their MIN, TYP, and MAX values.

9 References

1. Multi-Frequency Range and Tunable DCO on MSP432P401xx Microcontrollers (SLAA658)
2. Maximizing MSP432P4xx Voltage Regulator Efficiency (SLAA640)
3. Getting Started With EEMBC ULPBench™ on MSP-EXP432P401R (SLAA667)
5. EnergyTrace Technology (www.ti.com/energytrace)
6. ULP Advisor (www.ti.com/ulpadvisor)
IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products. Buyer is required to ensure that its products are designed and manufactured in compliance with appropriate safety standards and any other regulatory requirements. Buyer acknowledges and agrees that any military or aerospace use of TI components is at Buyer’s sole risk. Buyer acknowledges and agrees that any military or aerospace use of TI components is at Buyer’s risk
to comply with requirements. As used herein, “military/aerospace applications or environments” means applications or environments requiring particularly high-operating or environmental standards or longevity requirements above those applicable to all other general industrial applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have not been so designated is solely at the Buyer’s risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products

Audio: [www.ti.com/audio](http://www.ti.com/audio)
Amplifiers: [amplifier.ti.com](http://amplifier.ti.com)
Data Converters: [dataconverter.ti.com](http://dataconverter.ti.com)
DLP® Products: [www.dlp.com](http://www.dlp.com)
DSP: [dsp.ti.com](http://dsp.ti.com)
Clocks and Timers: [www.ti.com/clocks](http://www.ti.com/clocks)
Interface: [interface.ti.com](http://interface.ti.com)
Logic: [logic.ti.com](http://logic.ti.com)
Power Mgmt: [power.ti.com](http://power.ti.com)
Microcontrollers: [microcontroller.ti.com](http://microcontroller.ti.com)
RFID: [www.ti-rfid.com](http://www.ti-rfid.com)
OMAP Applications Processors: [www.ti.com/omap](http://www.ti.com/omap)
Wireless Connectivity: [www.ti.com/wirelessconnectivity](http://www.ti.com/wirelessconnectivity)

Applications

Automotive and Transportation: [www.ti.com/automotive](http://www.ti.com/automotive)
Communications and Telecom: [www.ti.com/communications](http://www.ti.com/communications)
Computers and Peripherals: [www.ti.com/computers](http://www.ti.com/computers)
Energy and Lighting: [www.ti.com/energy](http://www.ti.com/energy)
Industrial: [www.ti.com/industrial](http://www.ti.com/industrial)
Medical: [www.ti.com/medical](http://www.ti.com/medical)
Video and Imaging: [www.ti.com/video](http://www.ti.com/video)

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2015, Texas Instruments Incorporated