

11 Myths About Low Power

[Electronic Design](#)

[Mike Salas](#)

Wed, 2016-03-02 14:07



We've heard about low-power design over a long period now. In fact, power consumption is acknowledged as one of the main figures of merit for any product. But myths and misunderstandings still abound.

1. Power consumption is the only thing that matters (part 1).

Wrong. Even engineers working on power-sensitive products like wearables, smart metering, and IoT devices need ever-increasing processor horsepower so that they can innovate. What really matters is satisfying the need for more processing within

defined power and energy budgets.

There are signs that the wearables market in particular is moving into a new phase. Designers have established the parameters of ultra-low-power design and produced the first generation of wearable products. Now they're looking at processor technologies like sub-threshold switching, which allow them to make much more feature-rich products within the same power budget.

Related

[Q&A: Salas on the SPOT in Low-Power Design](#)

[Subthreshold Cortex-M4F Design Sips Less Power Than Cortex-M3](#)

[11 Myths About the Internet of Things](#)

2. Power consumption is the only thing that matters (part 2).

Instantaneous power consumption is of course important—your product can't ever be allowed to demand more volt-amps than is possible with the energy source. But even more important is energy consumption: The amount of energy required by the product to complete its tasks is a dominant factor in defining battery life. As we know, in consumer markets, time between charges is a key "care-about." It's perhaps even more vital in applications like metering, where a single charge needs to last for years. In areas like this, a few percent on battery life makes a substantial difference to the economics of deployment.

3. Low power is only required in portable and/or battery-powered applications.

Wrong. First, every product is now expected to be energy-efficient. Programs such as Energy Star and increasingly demanding energy performance regulations for buildings have made sure of that.

But the requirement for low-power operation is often driven by more subtle factors. For example, a product like a smart wall switch will commonly derive its power parasitically from the electrical mains. In that case, local

ulations will set very strict limits on how much current it can sink out through the neutral wire. Similarly, if you want to add extra sensors in a multi-drop communications system using a protocol like RS-485, you need to be very careful about keeping within the current budget defined in the standard.

4. The best way to design a low-power MCU is to choose a stripped-down core like M0, which doesn't have much performance.

Wrong. For starters, your application may simply need the capabilities of the higher-featured processor. As we've already said, the focus in a market like wearables is now switching to the feature set. In any case, it's possible that a higher-spec processor will deliver better energy efficiency, because it will "get the job done" in fewer clock cycles than a less-competent alternative.

One such example is [Ambiq's Apollo processor](#)—we chose an ARM Cortex-M4 ahead of the lower-spec M0. A look at its CoreMark benchmarks will help to explain why (*see table*).

ARM CORTEX COREMARK BENCHMARKS			
	Dhrystone (official)	Dhrystone (max opt)	CoreMark *
Cortex-M0	0.87	1.27	2.33
Cortex-M0+	0.95	1.36	2.46
Cortex-M3	1.25	1.89	3.32
Cortex-M4	1.25	1.95	3.40
Cortex-M7	2.14	3.23	5.01

*CoreMark data from ARM website and EEMBC.org website

As you can see, the M4 delivers CoreMark figures that are around 46% better than the M0 running at the same clock frequency. So an M4 device will spend a proportionally smaller amount of time processing, and return to a lower-power sleep state more quickly than the M0. Of course, Ambiq's use of subthreshold voltage technology narrows the power-consumption gap between the M4 and M0 even further.

5. Lower clock rate is better.

Wrong. As with myth #4, getting the job done quicker means a shorter period of active power consumption; therefore, a faster clock might require less energy overall.

The important thing to remember is that there are two components to active power consumption. The most obvious is what we might call the "dynamic component"—the power consumed when the processor performs a task. The second is what we can consider as a "quiescent current"—power that's dissipated while the processor

1 active mode, whether tasks are executing or not. If a task takes a million clock cycles, a 10X increase in clock speed will probably not increase the dynamic component. However, it WILL reduce total execution time (and hence the amount of “quiescent” energy dissipated) by a corresponding factor.

6. What’s the problem? These MCUs spend most of their time in sleep mode. All you need is low sleep mode consumption.

Wrong. You can’t achieve low energy operation without low active power. The fact is that active-mode power consumption is typically three or four orders of magnitude more than the corresponding figure for sleep mode. So the active-passive duty cycle doesn’t have to be high for the active mode to dominate the overall performance figure.

Let’s take a specific example of the Apollo microcontroller. Its active power consumption is 35 $\mu\text{A}/\text{MHz}$. Therefore, at a typical active clock speed of 24 MHz, it draws a little under 1 mA of current. In contrast, its sleep mode draws just 150 nA. So even if the application is asleep 99.9% of the time, the active-mode contribution to the overall energy budget is over five times that of sleep mode.

7. A “good” sleep mode is all about how low you can take the power consumption.

Wrong. You also need to think about how much time and energy you’re going to spend waking the processor up. A very deep sleep mode will require a substantial number of cycles to wake up and restore status. Effectively, these are “wasted cycles.” Consequently, if the application involves frequent wake-ups, it might be better to strike a compromise and choose a “shallower” sleep mode—one that dissipates more power during the sleep period, but requires less energy expenditure on each wake-up.

8. Saving power is all about the processor.

Wrong. Peripherals are increasingly important. You need a low-power peripheral interface, as well as peripherals with a degree of autonomous functionality that can offload tasks from the processor.

In addition, you need to design-in memory structures and architectures that are conducive to low-energy operation. As an example, consider a system in which an application talks to a Bluetooth radio via SPI. To minimize power consumption, we would, of course, want the SPI interface to work without processor intervention. But we would also want to size the incoming message buffer so that the processor only needs to wake up when it has to deal with a complete and valid message. Somewhat surprisingly, you can actually save power by getting the size of a memory buffer right!

9. Low-power design is about continuous evolutionary improvement.

Wrong. Technologies are still coming to market that reduce power consumption by orders of magnitude. Ambiq’s Subthreshold Power Optimized Technology (SPOT) is a case in point. The company developed a microcontroller that, after testing it to the industry-standard [EEMBC ULPBench](#) benchmark, consumes less than half the energy of other comparable MCUs.

10. If you want to build a low-power chip, you use a low-power process.

Not necessarily. Most “low-power processes” are actually designed for low leakage. This can often mean that dynamic power is higher, and as we have seen, dynamic power typically dominates total energy budgets.

11. When you’re working to a power budget, MIPS are always on the “cost” side of the equation.

Another surprising result, until you start thinking about it. However, the fact is that today, MIPS can be

ivered so power-efficiently that they can sometimes be used to save power elsewhere in the system.

As an example, consider a data-logging application, in which data is acquired and written to an external storage device, such as an SSD or memory card. The cost, in terms of power, of the data-acquisition process will typically be microwatts, but writing to the memory may cost milliwatts. So a sensible strategy is to spend some processor cycles on compressing the data and “save big” on the energy cost of reads and writes to and from storage. A 1:10 compression of the data might cost a few extra microwatts in processing. However, if it reduces read/write power consumption by a factor of 10, it’s likely to be more than worth it.

Source URL: <http://electronicdesign.com/power/11-myths-about-low-power>