

M E A S U R E D

CAN BENCHMARKS USEFULLY COMPARE ENERGY EFFICIENCY?

BENCHMARKS FOR COMPARING the relative performance between processor alternatives are strictly design-productivity tools. They do not, by themselves, make a technical challenge solvable, but they can reduce the amount of time it takes a developer to choose a processor for a project. That is, in an ideal world, standard benchmarks would provide developers with a way to quickly and accurately compare alternatives in an apples-to-apples fashion. The reality of embedded-processor benchmarking is that it is difficult to provide a simple, quick, and comprehensive approach for an apples-to-apples comparison across all types of applications and processor architectures. In addition to processor-performance benchmarks, several organizations are currently working on ways to meaningfully describe and compare processors' power and energy efficiency.

Benchmarks for processor performance have existed for many years, and the focus of these benchmarks have stressed how much work a processor could do in a period of time and have ignored or downplayed the power efficiency and cost to attain that level of performance. It is up to the person using a set of benchmarks to understand what the data can expose about how a system will perform in the design

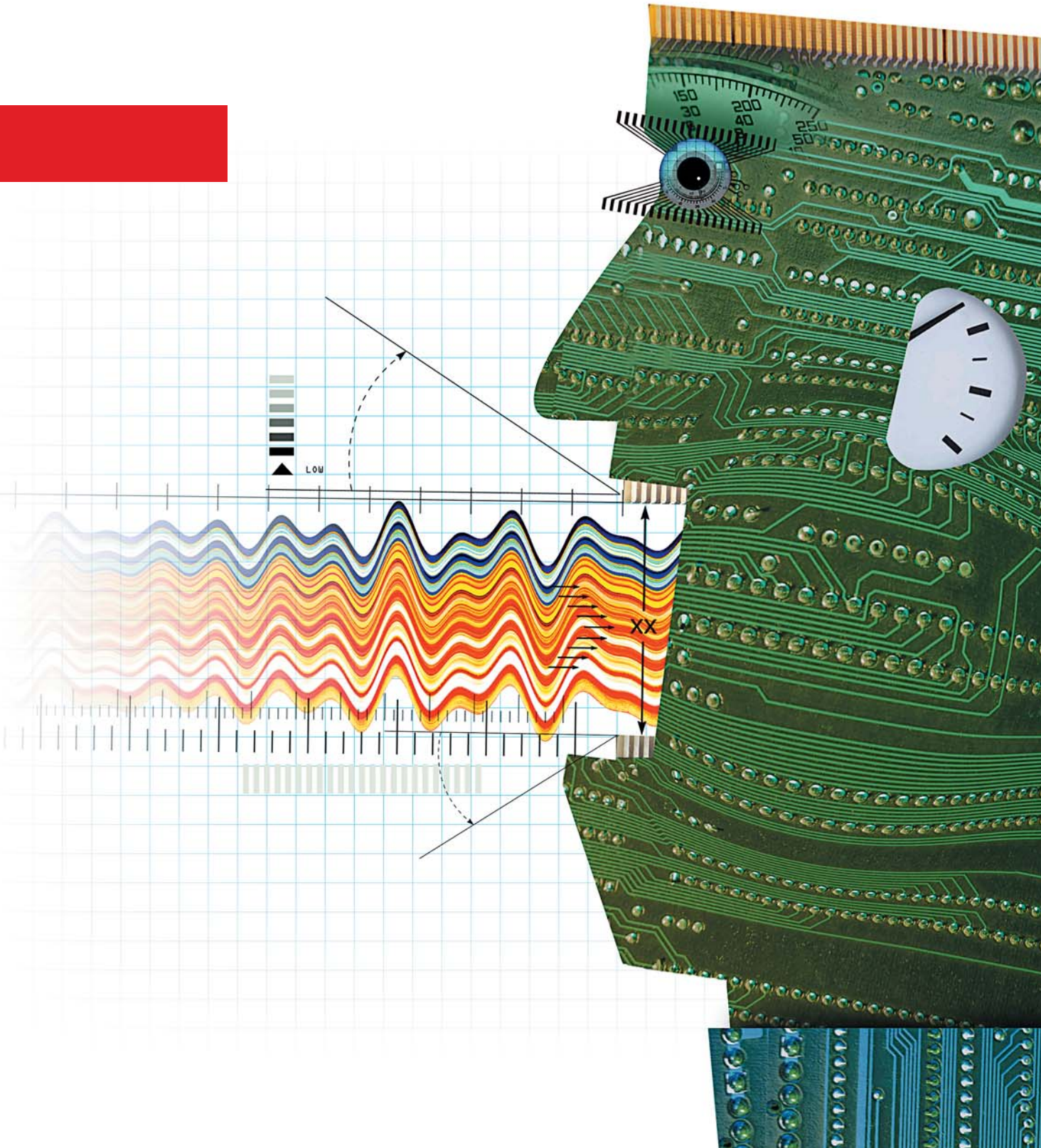
TOOLS AND BENCHMARKS ARE BEGINNING TO APPEAR THAT TRY TO TACKLE HOW TO COMPARE POWER AND ENERGY EFFICIENCY BETWEEN PROCESSORS.

and implementation of an application (**Reference 1**). Although marketing material still quotes synthetic processor-performance data, such as instructions per second, this data is mostly useless to designers because it provides little to no relevant insight into how the processor will perform in a specific context.

The BDTI (Berkeley Design Technology Inc), EEMBC (*EDN* Embedded Microprocessor Benchmark Consortium), and SPEC (Standard Performance Evaluation Corp) benchmark organizations support benchmark suites that highlight a processor's performance when performing application-specific tasks. Researchers at BDTI and EEMBC are both working on how to extend their benchmark suites to measure and compare a processor's energy efficiency, as opposed to power consumption, when performing application-specific tasks (see **sidebar** "Energy and power"). The scope of these tasks goes beyond measuring the execution efficiency of a single function, such as an FFT, and address characterizing the performance of higher level tasks, such as at the level of an audio or a video decoder.

At a glance46 Counterintuitive energy efficiency.....48
Energy and power46 For more information50

R E S P O N S E



It is more challenging to define and perform meaningful energy-efficiency benchmarks than to create processor-performance benchmarks. Performance benchmarks currently focus on a processor's maximum processing performance. This information is useful when a developer is trying to determine whether a processor can deliver the performance necessary to implement some set of features, especially when a limited set of processor candidates is available that can meet the project's performance requirements. However, as the number of processors that can meet a given performance threshold continues to increase, power consumption or energy efficiency becomes a more important differentiator between the processor options.

Although processor-power consumption is often a small percentage of the total system-power consumption during active mode, applications are tackling exponentially increasing computational loads, such as when processing larger images, and a greater need exists for better processor-energy efficiency. As an emerging example, in addition to tackling increasing computational loads, medical devices targeting in-home use by patients, such as wearable and implanted health-monitoring units, have size constraints that necessitate the use of tiny batteries that must be able to last for extended periods of time—possibly, many years.

A device's power consumption is the sum of its static and dynamic power consumption. Under normal operating conditions, the dynamic power overwhelms

AT A GLANCE

▶ Power benchmarks measure the energy to perform a task rather than the rate of consumption.

▶ Energy efficiency is becoming a more important differentiator between similar processor options.

▶ Power benchmarks must balance energy consumption with a "good-enough" processing-performance sweet spot.

▶ It is unclear how to fairly apply a consistent approach to measuring energy efficiency.

the contribution of the static power. For applications with long dormant or standby operation, static-power consumption becomes a driving force for battery life. Transistor leakage is the largest component of the static-power component. The transistor leakage increases inversely and exponentially as the gate-dielectric thickness decreases. This fact manifests itself as higher leakage current as transistor geometries continue to shrink. Processor designers can employ a number of techniques to minimize static-power dissipation and to enable application developers to reduce the dynamic power consumption (**Reference 2**). Techniques to minimize static-power dissipation include using slower transistors in noncritical-path circuits, dynamic deactivation of fast and leaky transistors, and dynamically controlling

the body bias of the transistor substrate.

The following **equation** describes a processor's dynamic-power consumption: $P=CFV^2$, where C is the device's dynamic capacitance, F is the clock rate or switching frequency, and V is the supply voltage. Processor designers use power-management techniques to enable application developers to minimize dynamic-power consumption, including low-power modes of operation, frequency scaling, and voltage scaling.

COMPLICATIONS ARISE

A system's application-specific software and how it uses a processor's resources can dramatically affect a device's energy efficiency, and this fact greatly complicates how to define, measure, and benchmark a device's energy efficiency when performing application-specific tasks (see **sidebar** "Counterintuitive energy efficiency"). It is unrealistic to benchmark with publicly available reference code. First, a developer most likely never uses the reference code as is but performs some optimizations on the code based on the target processor's architecture, resources, and any available application-specific accelerators.

The unreasonableness of using nonoptimized reference code is no different for performance benchmarks. However, optimizations for performance benchmarks, such as with the "full-fury" benchmarks from EEMBC, can involve unrealistic use of memory and other processor resources to obtain the highest performance. Power benchmarks

ENERGY AND POWER

Even though people often use the terms "energy" and "power" interchangeably, they have distinct definitions. "Energy" is the amount of charge available or necessary to perform work, whereas "power" is the rate at which a device consumes energy to perform work. It is usually more useful to compare energy efficiency, or the energy required to complete a task, when comparing devices for a battery-powered design. Energy consumption better describes battery life than does power consumption because energy

directly correlates to a battery's energy capacity. Both BDTI (Berkeley Design Technology Inc) and EEMBC (*EDN* Embedded Microprocessor Benchmark Consortium) power benchmarks are energy-efficiency measures that compare the amount of energy a device consumes to complete a task.

You can obtain a device's energy consumption to perform a task by integrating the device's power consumption over time: $\int Pd t$. A developer can lower the device's energy consumption and increase the energy efficien-

cy by minimizing the power consumption, time, or both required for completing the task processing. The **equation** $P=CFV^2$ describes a processor's dynamic power consumption, where C is the device's dynamic capacitance, F is the clock rate or switching frequency, and V is the supply voltage. Some processors allow developers to scale the clock frequency and supply voltage. Scaling the clock frequency affects power linearly, but scaling voltage in conjunction with the clock-frequency scales affects the power exponentially.

Although benchmarking a device's energy consumption to perform the processing for a task may be the appropriate measure for comparing energy efficiency between processors, a designer may have to consider a device's power consumption from a system perspective. For example, if a battery, such as a lithium-ion battery, discharges too fast because a device's power consumption is too high, the battery may be less efficient and deliver less energy.

must account for the fact that, when a developer optimizes the reference code, a balance must exist between energy consumption and obtaining “good-enough” performance from the code. Performance benchmarks currently ignore this need.

Obtaining useful power benchmarks across a range of performance points for a single processor device, therefore, may ideally involve using different algorithms, software code, and on-chip resources. This requirement drastically impacts a designer’s ability to perform an apples-to-apples comparison with the same device at different performance points, never mind what it means for comparing disparate processor architectures. This scenario, however, is unrealistic, especially because benchmarks are supposed to be a development-productivity tool to help designers quickly explore alternative architectures and configurations and more quickly make trade-off decisions.

A compromise is to execute the same optimized code across multiple performance points, clock rates, and voltages, meaning that, when a processor vendor applies power benchmarks to a processor, the vendor’s optimized code should target a performance sweet spot. The benchmark data is most relevant at and near the sweet spot and may lose validity the further a performance point is from the sweet spot. The processor vendor thus needs to identify the targeted performance sweet spot; however, this

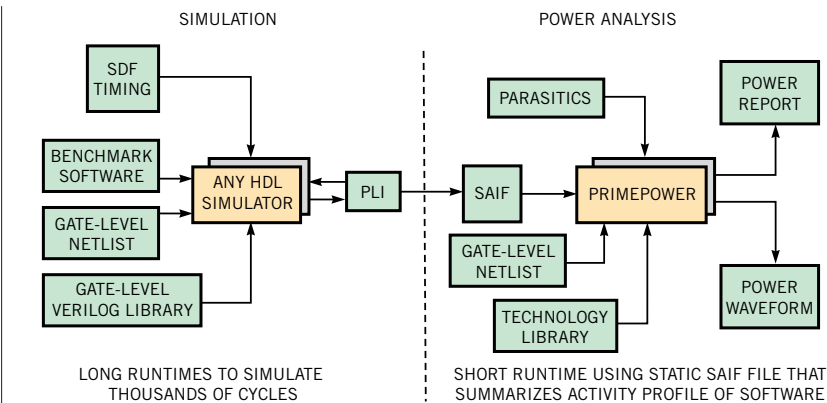


Figure 1 The benchmark flow for an IP core includes executing a gate-level netlist over many days. The resulting SAIF file from the simulation, along with the netlist, tech library, and parasitics model, feeds into Synopsys’ PrimePower tool (courtesy EEMBC).

problem may be insignificant for application-specific benchmarking with standardized performance thresholds.

MORE CHALLENGES

Markus Levy, president of EEMBC, says, “It is easy to run the benchmark suite on hardware, but it is hard to measure the energy consumption. In contrast, it is easy to measure the energy consumption of IP (intellectual-property) cores and hard to run the benchmark suite on them.” This statement alludes to the fact that EEMBC benchmarks can execute at the processor speed on hardware but are simulated, over periods of days, using gate-level netlists for IP cores. Likewise, it is sometimes unapparent where

to attach measuring devices to the hardware, which system subcomponents you should include in the measurement, or when the measurement should begin and end. When simulating IP cores, it is a relatively simple matter of instrumenting the simulation to capture data anywhere.

It is unclear how to define and apply a consistent approach to measuring energy efficiency and correlating it with a performance point. Both BDTI and EEMBC now propose that measuring the core and local memories to a workload is sufficient, provided that proper disclosure of the testing configuration exists. The configuration disclosure is important because a device with a large on-chip memory or a built-in accelerator may look less

COUNTERINTUITIVE ENERGY EFFICIENCY

A general rule of thumb for the energy efficiency of an algorithm is that the fewer instructions necessary to complete the algorithm, the lower the energy consumption. As with any rule, there are exceptions. For example, when performing motion detection on a video stream, a spiral search may be the fastest algorithm with an answer discovered in the shortest number of executed instructions, but the algorithm can cause loading and reloading of image data into the cache. Because the processor’s memory subsystem can perform the cache loading in parallel with the image processing, the per-

formance hit is minimal for loading the same data in the cache multiple times. However, reloading the same data in the cache consumes energy without producing useful work. With a proper memory architecture, designers can employ a slower search algorithm, such as a linear search, that more efficiently works with the cache policies and uses less energy overall for the same net amount of useful work.

PowerEscape offers power-optimization tools, which it partially based on the Interuniversity MicroElectronics Center’s Atomium technology, which tar-

gets software and hardware designers and can reveal when this rule of thumb is inaccurate. The tools work by exposing how the system’s memory architecture affects performance and power consumption. These tools use power-aware platform models and can correlate software and memory interactions. They can help a developer to explore how varying the processor-memory architecture and resources, including registers, caches, and on- and off-chip memories, can improve power consumption.

Jeff Bier, general manager at BDTI, identifies another example

of a counterintuitive energy-consumption test. This test involves executing a stream of NOP (no-operation) instructions on an unspecified processor. Bier expected that, because the NOP command performed no work, it would consume little power; however, total power cost was high. This scenario happened because the NOP command executed every cycle, which is more frequent than typical instructions, and because fetching the next NOP instruction and pushing it through the CPU every cycle incurred a cost.

	FREQUENCY	IDLE STATUS	% USAGE	WRITES (%)	BITS	SWITCH (%)	TRACE (IN.)	CAPACITANCE (pF)	OTHER
CLKGEN	72	ACTIVE							
CPU	72	ACTIVE	75						
CLKOUT	72	DISABLED					1	10	
EMIF	72	IDLE	0	100	16	75	2	6	SYNC
HPI	72	IDLE	0	100	16	0	1	10	
DMA	72	ACTIVE							
CHANNEL 0		ENABLED	0.06		16	100			
CHANNEL 1		ENABLED	0.005		16	100			
CHANNEL 2		DISABLED	0		16	0			
CHANNEL 3		DISABLED	0		16	0			
CHANNEL 4		DISABLED	0		16	0			
CHANNEL 5		DISABLED	0		16	0			
MCBSP0	1.41	ACTIVE	100		32	100	1	10	EXTERNAL
MCBSP1	36	IDLE	0		32	100	1	10	EXTERNAL
MCBSP2	36	IDLE	0		32	100	1	10	INTERNAL
MMC1	36	ACTIVE	5.3	0		100	2	3	
SD1	9	IDLE	0	0		100	1	10	
MMC2	9	IDLE	0	0		100	1	10	
SD2	9	IDLE	0	0		0	1	10	
I ² C	12	ACTIVE	0			100	1	10	
TIMER 0	5	IDLE	100				1	10	T _{OUT} ENABLED
TIMER 1	20	IDLE	100						
WATCHDOG TIMER									
ADC		IDLE	0			100	6 FEET		
USB		IDLE	25						
REAL-TIME CONTROL									
GPIO/XF	0		0				0	1	10

Figure 2 Spreadsheets help developers plan and analyze how their design decisions will affect the system's power and energy consumption (courtesy Texas Instruments).

energy-efficient than a processor that a vendor benchmarked with off-chip memories and off-chip accelerators, even though the device with a large on-chip memory of built-in accelerator enables lower overall system-energy consumption. It is difficult to meaningfully compare the energy efficiency of IP cores if the benchmarking targets different process geometries or libraries. Both BDTI and EEMBC have adopted a compromise position: to apply a consistent

process and set of libraries. Both organizations expect to evolve and refine their power-benchmark process over time. Another challenge is that processor vendors' evaluation boards sometimes don't accommodate power measurements. Companies that provide evaluation boards for devices that target low-power applications usually provide access points and separate power lines for each chip and subsystem on the processor. Useful separation of power and ac-

cess points include one line for the CPU core and L1 cache, another for the I/O pins, and another for everything else on the device. Boards that do not separate power complicate the measuring and benchmarking process. According to the EEMBC process for measuring energy during a benchmarking suite, each tested device must operate under identical conditions, including ambient temperature and measuring equipment. The designer should run the

FOR MORE INFORMATION...

For more information on products such as those discussed in this article, contact any of the following manufacturers directly, and please let them know you read about their products in *EDN*.

- Altera**
1-408-544-7000
www.altera.com
- ARC**
1-408-437-3400
www.arc.com
- Arm**
1-408-579-2200
www.arm.com

- BDTI (Berkeley Design Technology Inc)**
1-510-665-1600
www.bdti.com
- EEMBC (EDN Embedded Microprocessor Benchmark Consortium)**
1-530-672-9113
www.eembc.com

- Interuniversity MicroElectronics Center**
+32-16-281-509
www.imec.be/design/atomium/
- Microchip**
1-480-792-7200
www.microchip.com

- NEC Electronics**
1-408-588-6000
www.necel.com
- PowerEscape**
1-408-348-1848
www.powerescape.com
- QuickLogic**
1-408-990-4000
www.quicklogic.com

- SPEC (Standard Performance Evaluation Corp)**
1-540-349-7878
www.spec.org
- Synopsys**
1-650-584-5000
www.synopsys.com
- Texas Instruments**
1-800-336-5236
www.ti.com

test multiple times to ensure that the processor reaches a stable operating level. The power sampling uses a smart sampling method that relies on random and multiple intervals to avoid aliasing. The testing does not consider the contribution of static power, but it does require that the vendor discloses the cooling method and clock rate as part of the configuration. Measurement results are the average energy per testing iteration, and designers correlate them with the performance benchmarks.

To benchmark IP cores, the EEMBC benchmark flow executes a gate-level netlist in any HDL simulator (Figure 1). An output from the simulation, which takes days to complete, is a SAIF (Switching Activity Interchange Format) file that enters, along with the netlist, technical library, and parasitics model, into Synopsys' PrimePower tool. The result of the flow is a power waveform and report. This process is currently core-centric and includes only those caches that took part in the performance benchmarks. The energy benchmarks are not absolute meas-

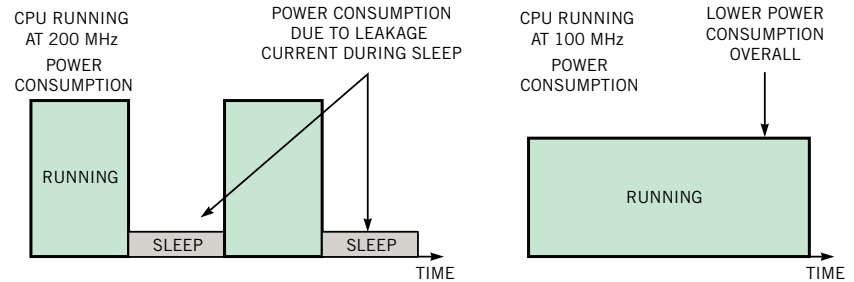


Figure 3

Depending on the application requirements, a designer can choose to implement an algorithm in a full-speed duty cycle or in a continuous lower speed manner. It is unclear, until you analyze each approach, which will better meet your requirements.

ures of the energy, but act as relative indications of energy efficiency when comparing benchmark scores.

OTHER TOOLS

Every designer must address how to validate and map what benchmark data is available to the specifics of a project—an important task when performing a benchmark. Processor vendors until recently provided spreadsheets as the only tools to assist in analyzing power and en-

ergy consumption (Figure 2). These spreadsheets vary in complexity, depending on the power-management features, such as low-power modes and frequency and voltage scaling, that the target processor supports. Unfortunately, using low-power modes and scaling features can be complex. These spreadsheets can help a designer more confidently explore the impact on energy consumption of using different hardware and software configurations.

An example of a power-consumption design trade-off is whether to execute an algorithm at full speed on a duty cycle so that the device can drop into a low-power mode or to continuously execute the same or an equivalent algorithm at a lower clock rate (Figure 3). The spreadsheets can help identify nonobvious contributors, such as static power during low-power modes, the system latencies and energy cost to switch between modes, and the total energy consumption of each approach. The discreteness and fixed format of these spreadsheets does not simplify the analysis for systems that support and can take advantage of power efficiencies from dynamic frequency and voltage scaling. For now, these spreadsheets force designers to estimate a static average for this dynamic behavior.

Power planning, analysis, and optimization tools are beginning to go beyond the spreadsheets. For example, Texas Instruments offers a power manager that resides within the

DSP/BIOS to address boot-time power savings and assists developers in implementing more intelligent scheduling of processor resources and supervising and controlling power scaling. Altera's Quartus II tools now include power-analysis and -optimization tools. Arm's IEM (Intelligent Energy Manager) technology comprises hardware and software components to balance the processor's workload and energy consumption.

PowerEscape's tools can help developers identify how efficient their software works with the processor-memory architecture to perform data movement; this feature includes focusing the developer's attention on the power efficiency of the memory resources from such events as context switching and cache misses. This type of tool can assist a developer in structuring and arranging data accesses in contiguous blocks that take better advantage of burst-oriented memories. Other cache-analysis tools currently force developers to exam-

ine the cache behavior from a processing-performance perspective rather than from an energy-efficiency perspective.

Standard-power and energy-efficiency benchmarks are coming to fruition, and a lot of opportunity exists for people to refine them. The importance of power benchmarks will continue to grow, especially because a growing number of processors have similar or identical core architectures. However, just like performance benchmarks, power benchmarks require developers to practice due diligence when mapping the benchmark data and testing configuration to their project's requirements. □

REFERENCES

1. Cravotta, Robert, "Uncover the truth in benchmarks" *EDN*, Oct 2, 2003, pg 57.
2. Cravotta, Robert, "Squeeze play: Wring the power out of your design" *EDN*, Feb 19, 2004, pg 36.

TALK TO US

Post comments via TalkBack at the online version of this article at www.edn.com.

