

eXtreme Energy Conservation: Advanced Power-Saving Software for Wireless Devices

White Paper

Freescale Semiconductor, Inc.

Document Number # XTMENRGYCNSVWP
Rev #0
02/2006



OVERVIEW

Gradual improvements in battery technology have not kept pace with the power consumption demands of the latest handheld electronic devices. Next-generation rich multimedia mobile phones that use the high-bandwidth 3G cellular radio network consume more power than ever. Hardware designers are using advanced power-saving techniques to help minimize integrated circuit and system power consumption. Yet these techniques will not yield significant power savings without intelligent energy conservation software to exploit them effectively.

Freescale Semiconductor's eXtreme Energy Conservation (XEC) software is a multi-phase, multi-generational project to develop advanced power-saving software technology for Freescale wireless mobile platforms. Based on advanced runtime performance algorithms, XEC uses a standardized software framework that supports multiple concurrent predictors, policies, power cost rules, etc. It runs as system software with commercial operating systems.

CONTENTS

What Is XEC?	2	Software Techniques and "Smart"	
Manufacturer Benefits	2	Algorithms	6
Power Management Technologies	2	Role of the Operating System.....	7
Battery Technology: The Energy Gap	2	Typical DVFS Algorithm	8
Sources of Power Consumption	3	Scope Within Wireless Applications	8
Power-Saving Technologies.....	3	Freescale's System Approach to Power	
Solutions to Power Waste	4	Management.....	9
Dynamic Frequency Scaling.....	5	Technology-Specific Smart Algorithms	10
Dynamic Voltage Scaling	5	XEC Framework	12
Dynamic Process and Temperature		Multi-Generational XEC for Multiple	
Compensation	6	Platforms	13
Handling Idle Modes.....	6		

What Is XEC?

Freescalé's eXtreme Energy Conservation (XEC) software is a comprehensive ready-to-use energy saving solution with the system intelligence and connections to exploit on-chip power-saving features effectively. XEC targets more areas for energy savings—such as IC-specific low-power modes and LCD panel hardware—so it can provide potentially better energy savings than competitor solutions.

XEC is optimized for Freescalé's wireless platforms and is delivered as part of the customized operating system (OS) software provided with a chipset's OS board support package. Because XEC will run on many of these platforms and is designed to take advantage of each chipset's particular features, energy savings will vary by use case.

XEC is transparent to application programs and middleware, unless that software is optimized as a "power aware" application; in that case, the application could communicate its power needs to XEC. Using advanced techniques, XEC dynamically discovers each piece of software's required performance, sets hardware power management features (dynamic voltage and frequency scaling, multiple low-power idle modes, etc.) for just-enough performance and minimal power waste, and achieves greater energy savings than traditional techniques. Adapting leading-edge university research, XEC has performance predictors, real-time cost-benefit analyzers, policies and a policy manager, and many other advanced algorithms.

Manufacturer Benefits

Power reduction is an increasingly important consideration for wireless device manufacturers. Consumers are demanding more and more features in ever-slimmer devices. Putting more integrated circuits and applications into these shrinking devices increases manufacturers' costs. Because XEC helps reduce power consumption, manufacturers can use smaller batteries to get the same power, and this helps reduce costs. Design effort can also become faster and simpler, because designers don't have to engineer so aggressively for reduced power consumption—XEC is an out-of-the-box solution that is designed to help with that complex step. Designers also don't need to modify applications to run with XEC, because XEC is designed to work with the operating system to manage the performance needs of those programs.

The XEC production software will be provided with those operating systems supported by Freescalé for its wireless chipsets including Symbian™, Linux® and Microsoft® CE. In general, XEC is transparent to the product end user, except for the XEC Policy Manager which manufacturers can optionally use to expose policies to end users for limited high-level control of power settings. Manufacturers can also customize software modules within XEC or tune factory-set systems parameters for optimal performance in their environments.

Power Management Technologies

There are many approaches that designers and manufacturers can take to energy conservation. Some are more appropriate than others for wireless applications. This section examines some of those software and hardware technologies and analyzes how appropriate they may be for miniaturized wireless devices.

Battery Technology: The Energy Gap

Of all consumer products, mobile phone designs place one of the highest premiums on minimal physical size and weight. The battery is already the largest and heaviest component of a phone, so increased energy capacity should come through improved battery efficiency, not greater size or weight. Batteries for early mobile devices used lead acid technology, then progressed to nickel cadmium (Ni-Cd) and nickel-metal hydride (Ni-MH). Now lithium ion (Li-Ion) is the primary choice. A Li-Ion battery

has good energy density in terms of watt-hours per kilogram, holds its charge for a long time and has none of the “memory” charging effects of Ni-Cd. Lithium ion technology has shown steady—but not dramatic—efficiency improvements in recent years. However, this improvement has not been fast enough to offset the power needs of recently developed handheld devices such as multimedia mobile phones and PDAs with color screens and multimedia capability.

Fuel cell technology appears to have great potential. Like a battery, a fuel cell uses an oxidation/reduction reaction, but the reaction takes place in the fuel, not on the electrodes. This makes fuel cells clean and efficient. Unfortunately, there are several practical drawbacks that make fuel cells impractical for mobile devices. Most micro fuel cells use methanol, which is toxic and flammable, and currently a prohibited substance for passengers on commercial airplanes. Ethanol and hydrogen are lower-efficiency alternate fuels, but they too have their drawbacks. Fuel cells also generate thermal and chemical by-products. With operating temperatures of 50 degrees C to 100 degrees C, they are unsuitable for small handheld devices (although newer versions run at lower temperatures). Some fuel cell types will operate only in the correct orientation. Replacement (refueling) lifetimes, currently about one year, need to be improved to compare with Li-Ion equivalents. Despite the current issues, commercialized fuel cells for mobile phones are expected later this decade. Such phones could run two to three times longer than a Li-Ion phone before recharging is necessary.

Will fuel cells make the need for complex power management redundant? The energy capacity improvement for commercialized cells is useful, but not dramatic. Almost certainly the increased capacity will be used, not to accommodate power inefficiencies, but rather to provide the user with more power-hungry services and longer operating times. Intelligent energy conservation still has a bright future in handheld appliances.

Sources of Power Consumption

Power consumption in complementary metal-oxide semiconductor (CMOS) integrated circuits (ICs) is broadly classified as dynamic power in a circuit while it is operating (e.g. switching), and static power while it is not operating but still powered (e.g. non-switching steady state or transistor-off state). Static or leakage power dissipation also occurs when a circuit is operating, although for current CMOS processes, this is tiny compared to the dynamic power dissipation. However, as CMOS geometries continue to shrink, the static power will become ever more significant. Therefore, power-saving technologies must address both forms of power consumption to improve phone talk time, standby time and other power metrics of the appliance.

Power-Saving Technologies

Various on-chip and off-chip power-saving technologies have been developed to address sources of power waste. Many are all-hardware solutions such as smaller silicon process geometries, active well biasing and auto-idle detection circuits. Other technologies require software. Different software techniques must be used to exploit each of these different hardware technologies effectively. Dynamic power management (DPM) describes a system that sets the power states of its hardware modules in real time to minimize power waste, yet still meets performance needs. DPM includes techniques such as dynamic voltage and frequency scaling (DVFS), dynamic process and temperature compensation (DPTC), and idle time prediction for controlling low-power idle modes (doze, sleep, etc.).

Some of those software techniques are shown in Figure 1. Application programs and other system software are monitored during execution by one or more of the software techniques illustrated. Some of these applications know their performance-power needs (“power aware” software) and others—probably most—do not. These techniques can be used to control a power manager that drives the hardware power-saving mechanisms using software drivers and power handlers in the operating system.

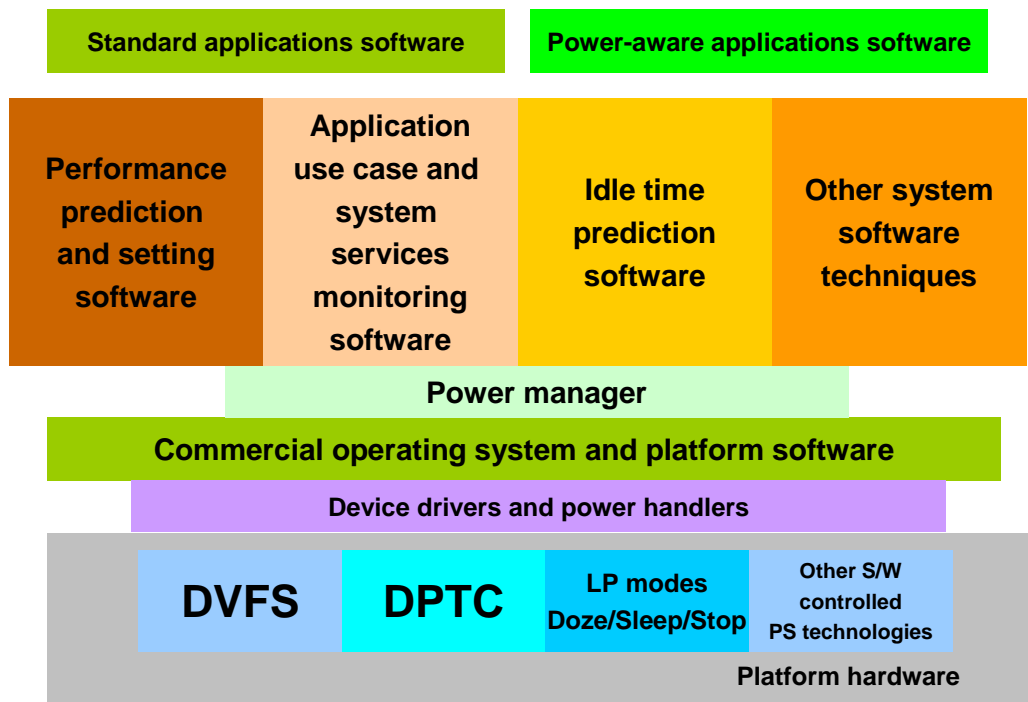


Figure 1. Power-Saving Software Techniques

Solutions to Power Waste

One dynamic power saving technique is to disable the clock to a logic circuit when the circuit is idling. Clock gating or clock freezing saves power not just in the registers whose clock is gated off, but also in combinational logic circuits connected to them, as the register signals are no longer propagated. Clock gating is very quick to turn on and off, so software that uses such circuits should not be affected if it is timed correctly.

Static or leakage power dissipation needs more drastic measures. One solution, called power gating, is to power off the device or subcomponent. Power gating reduces both dynamic and leakage power, and can be implemented either locally on-chip or externally at the power supply unit.

Another power-saving technique is to vary the supply voltage to a circuit either when no performance is required (idling) or when variable performance is required. During idling mode the hardware could switch automatically from a higher to a lower voltage when the device or subcomponent transitions from an active state to a low-performance state. An example would be a processor core design where operating voltage is reduced automatically when it enters a sleep or stop mode. Although the core is not clocked in this mode, it still suffers steady-state current leakage. Because the core does not need to execute instructions or other functions, the operating voltage could be lowered just enough to ensure that internal state data is retained correctly. This is sometimes called stop mode voltage scaling.

The non-idling situation, where variable performance is needed, is addressed by varying the operating frequency, the operating voltage, or both.

Dynamic Frequency Scaling

IC dynamic power consumption is roughly proportional to operating frequency. It makes sense, therefore, to lower the clock frequency of a processor to the lowest value that still meets the required processing performance. This means that although the software runs more slowly, it still meets its real-time deadlines with acceptable margins. This has to be done dynamically and needs special power management software to figure out which frequency setting is acceptable. It may seem intuitive that although this lowers the instantaneous power consumption, it might not reduce the overall energy requirement, but instead spreads it over a longer period. In fact, some memory systems such as level-2 on-chip caches and SDRAMs will incur fewer accesses at the reduced operation frequency, so their circuits incur reduced switching and therefore reduced power consumption. However, the benefits of frequency scaling alone to total energy conservation are marginal.

Dynamic Voltage Scaling

Better power savings can be achieved if the operating voltage is scaled. Since power varies with the square of voltage, square-law power savings potentially are possible with voltage scaling. If voltage scaling and frequency scaling are both used, the combination, called dynamic voltage and frequency scaling, can yield power savings roughly proportional to the cube of operating voltage. These square-law and cube-law power savings depend not only on the configuration and efficiency of the voltage control circuits, but also (as we shall see later) on the efficiency of the prediction software used to set the voltage/frequency settings.

For a given IC design, the operating voltage determines the maximum usable operating frequency. The voltage (and hence frequency) are scaled to trade required performance against minimal power waste. When scaling the voltage up or down (thereby consuming more or less power), one must accept that the operating frequency must be scaled—and with it the available performance of the device—to remain within operational tolerance of the design.

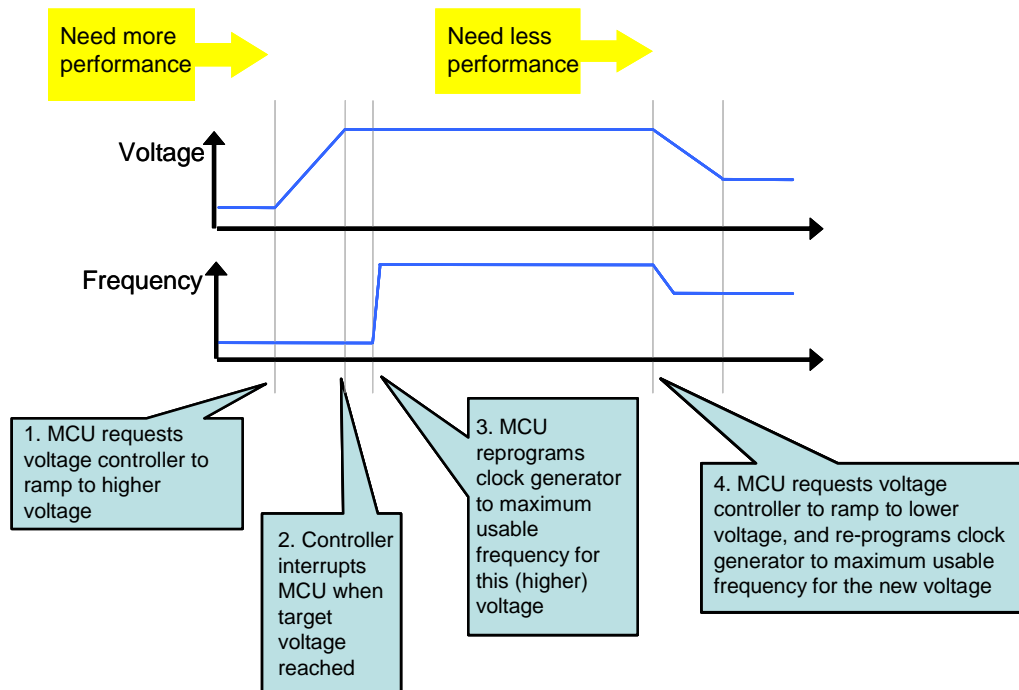


Figure 2. Dynamic Voltage Scaling

Figure 2 shows the required sequence for changing:

1. First, to a higher voltage and frequency, and then
2. Later, back to a lower voltage and frequency.

At point 1 a request, typically from the MCU software, is made to the voltage controller to ramp the MCU voltage up to the new higher value. The voltage will increase at a defined slew rate, to avoid excessive power surges, and the controller will notify the requester (e.g. via MCU interrupt) when the target voltage has been reached at point 2. The MCU must continue at the old (lower) frequency until then, at which point software then signals its clock generator (often a programmable PLL) to change the MCU frequency to a higher one, usually the highest usable frequency for the new voltage, which occurs at point 3. The frequency change is quick compared to the slow slew rate that must be adopted.

When going to lower voltages/frequencies, the MCU changes its frequency and requests the voltage change at point 4. The MCU need not wait until the voltage ramp down has completed before it continues normal software execution, since the new (lower) frequency is already within the operating range of the voltage profile.

Dynamic Process and Temperature Compensation

The operating voltage setting chosen with DVFS to achieve a specific operating frequency is a worst-case value. It includes a voltage margin for variations in process and temperature. This margin represents power waste, because the IC is operating at a slightly higher voltage than it needs for the operating frequency. By monitoring the IC using an on-chip process and temperature-dependent structure, it is possible to calculate a lower operating voltage that is very close to process limits, which thereby minimizes power waste. If the IC is manufactured in two or more silicon processes, that information could also be fed into the calculations.

Handling Idle Modes

DVFS technology addresses varying but continuous software workloads—but what happens when the processor or other power-hungry device has periods of no activity? Most hardware devices have low-power idle modes that software can exploit to save power. Some devices have more than one idle mode. For example, an ARM11™ processor core could have doze, sleep and stop modes. The deeper modes yield more power savings, but usually come with a greater “cost” penalty in time and transition power when entering and exiting the mode. For example, an idle mode based on power gating would require the hardware’s internal memory state to be saved and restored, either by hardware or software.

In a real-time system, the device’s run state must be re-entered in time-to-service events that otherwise would degrade the user’s quality of service. Smart software must either predict or have advanced knowledge of an event and the time taken to restore the hardware from idle to run mode. Based on the predicted start and end of an idle period, it controls the shut-down and wake-up mechanisms associated with the idle modes. The prediction software has to determine which (if any) of the idle modes will yield a net power saving for each idle period it encounters.

Software Techniques and “Smart” Algorithms

A simple DVFS software driver attached to the OS is used to interface to the on-chip DVFS control logic to increase or decrease the processor’s operating frequency and voltage. Controlling and monitoring the DVFS mechanism is the easy part. Knowing in real time which frequency/voltage setting to use, and when, is harder. For that, “intelligent” software is needed to figure out dynamically how much processor performance the application programs and other system software need at any given time.

Just how smart that software is will determine how close you can get to the theoretical hardware limit on power saving. Higher-performance software is more complex and more costly to develop. There is a performance vs. complexity/cost trade-off which will vary for different applications and products. The trick is to develop smart energy conservation technology that is both flexible and scalable to give superior performance at acceptable cost across a wide range of platforms. That includes the ability to support one or more algorithms for many hardware power-saving technologies.

Role of the Operating System

Typically, each microprocessor and digital signal processor in wireless applications uses operating system software to manage the large number of hardware and software resources. A DPM approach should treat the OS as a state machine in which each state may require its own power management techniques. For example, DVFS could be used using normal software execution (tasks and OS), idle mode predictors used during software idling, and full-speed execution (i.e. no DVFS) during interrupt handling.

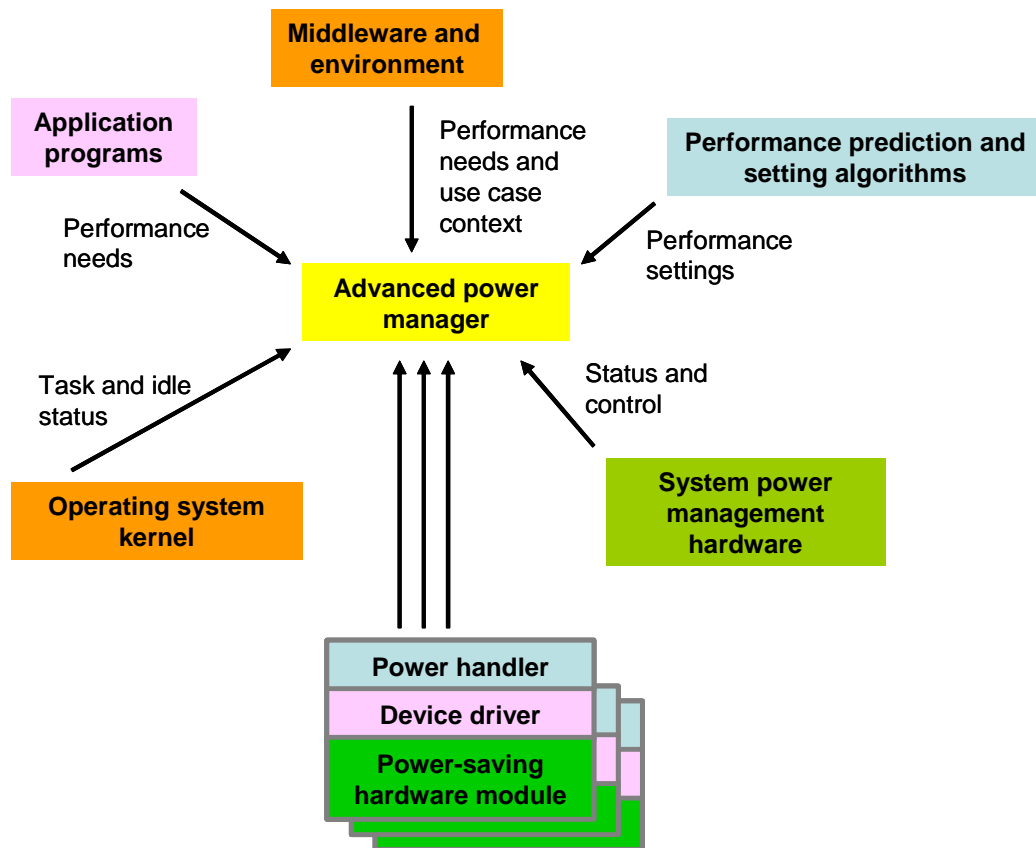



Figure 3. Power Management in the Operating System

Figure 3 shows a possible architecture for the power-management software for a modern applications-domain operating system. The hardware modules may each have several power states (i.e. fully powered, plus one or more low-power idle states), and the number and type will vary from module to module. A DVFS-controlled processor, for example, will have multiple active states and may also have several idle states.



The instantaneous values of the collective power states could be related to the phone use modes in many ways. The main criterion for judging an algorithm is the average power saving obtained over a range of typical use cases compared to a system which has no software power minimization (i.e. one in which all hardware modules are fully powered at all times).

Performance-prediction and performance-setting algorithms are used to control the performance-power states of the system hardware (e.g. the processor speed-voltage levels) dynamically. These algorithms are very sophisticated and may need to be adapted or tuned to suit different system designs.

Typical DVFS Algorithm

Many algorithms exist for use with DVFS-based processors that set the processor's operating frequency (and hence voltage) based on predicting the short-term software workload on the processor. An example algorithm in this class tracks the recent software workload (or utilization history) of each task running in the OS. This technique assumes a reasonable correlation between the recent past workload of a task and that of the near future. The task status information must be supplied by the OS kernel.

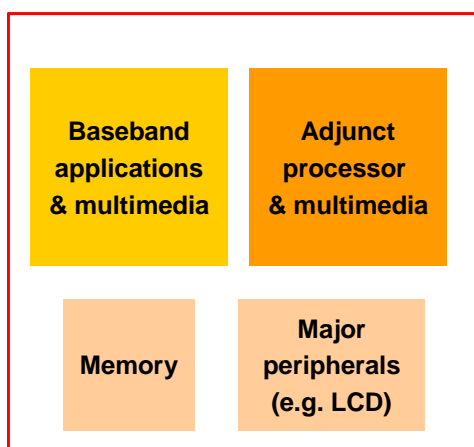
The algorithm maintains estimates of workload and unused idling time to predict the aggregate workload (for all tasks). This normalized MCU processing level is translated by associated software into the relevant frequency and voltage settings required for the specific DVFS mechanism used. The algorithm continuously re-calculates and supplies new predictions in response to changing software workloads.

Ideally, the algorithm correctly predicts the required processor performance that meets individual deadlines for each OS task. The algorithm works well for OS tasks whose workloads don't change very rapidly.

Scope Within Wireless Applications

Some may question whether such complex prediction techniques are really necessary in a mobile application. After all, most of the "engine" software that runs on the radio modem DSP and the Bluetooth™ and WLAN processors, although it involves complex protocols, is fairly deterministic and can be well characterized and optimized for energy conservation at design time. The main and largely unexploited opportunity for large power savings exists in the "applications domain", i.e. the processor where the user applications run and the real-time multimedia processing is performed.

Current scope



Applications domain –
either on adjunct processor
or integrated processor on
baseband IC

Future potential scope

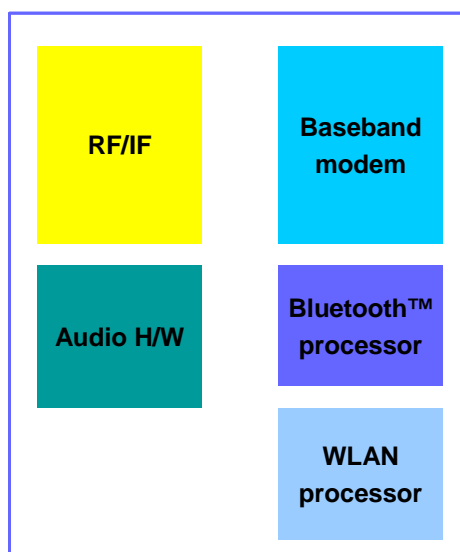


Figure 4. Power Savings in the Applications Domain

The applications processor is usually a general-purpose MCU implemented either as a separate adjunct processor or integrated into the same IC as the radio modem. In either case, the issue is the seemingly unpredictable performance variations caused by the flexibility that new high-speed multifunction wireless devices need. This can involve one or more concurrent executing applications programs, some of them downloaded over the air (and hence not pre-characterized) plus the associated middleware service software, some of which is highly real-time (e.g. video telephony).

Collectively, this makes the applications domain almost non-deterministic due to its software complexity and highly dynamic behavior. Traditional system software has not been smart enough to know how much performance it needs and when. This has led to sub-optimal exploitation of platform IC power-saving features. Software is in effect playing catch-up with platform hardware.

Freescalé's System Approach to Power Management

Superior energy conservation requires a systematic solution to advanced power management. Freescalé is involved in many approaches to power management, including:

- Advanced silicon processes
- Design methodologies
- Power-optimized IC features
- Power-optimizing software tools
- Power-aware application programs

- Power-optimized software
- Intelligent runtime software

For our power-managed platforms, Freescale is taking an architectural approach, and is creating a generic software/hardware XEC framework with APIs and hardware interfaces. Its purpose is to help maximize the portability of XEC technology across Freescale platforms and support scalability across customers' product tiers.

Prediction-based software control of power-saving techniques is an immature technology and is just starting to emerge within the mobile wireless industry. As yet there are no existing, widely accepted power management standards among mobile device manufacturers, operating system vendors, semiconductor companies and others. Freescale wants to work with industry players and standards bodies to help create open standards in this area—standards that help drive down costs but yet are flexible enough to allow individuals to differentiate.

Technology-Specific Smart Algorithms

Algorithmic software, known as performance predictors, is used to predict runtime workloads of specific modules or power-saving technologies (e.g. DVFS). A predictor module can be included or removed from the software build depending on whether or not its corresponding hardware technology is present. An advanced energy-saving solution such as XEC uses two or more performance predictors for certain technologies such as DVFS, to ensure high performance under a wide range of operating conditions.

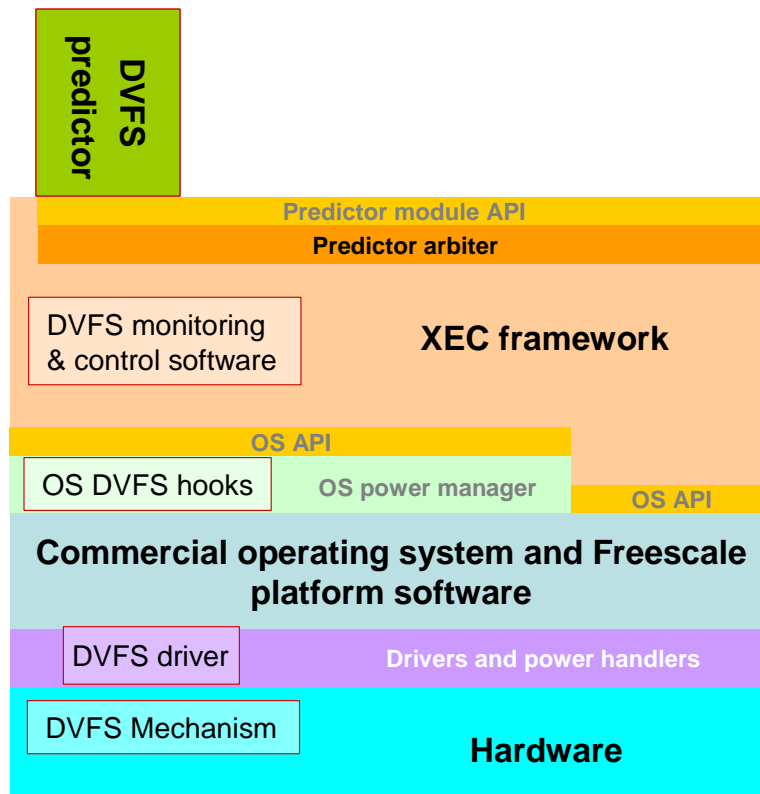


Figure 5. DVFS Predictor

Typically, a performance predictor operates abstractly. For example, a DVFS predictor only tries to predict required processing power as a normalized fraction of the maximum possible. It does not deal with the MCU frequencies, operating voltages, MIPS, etc. These device- or platform-specific details of the DVFS hardware technology are handled by the XEC Framework and OS driver software. Ideally, performance predictors are designed to be independent of the specific OS and hardware, to enable easy porting from one platform to another.

Depending on the software activity, the processor and other hardware modules will have periods of idling where they could be put into low-power idle modes. The XEC technology uses an idle-time predictor. It contains prediction software with connections to the OS and other event-monitoring software. The monitoring and control software specific to the platform's idle mode hardware is located in the XEC framework and in the OS device drivers for the hardware.

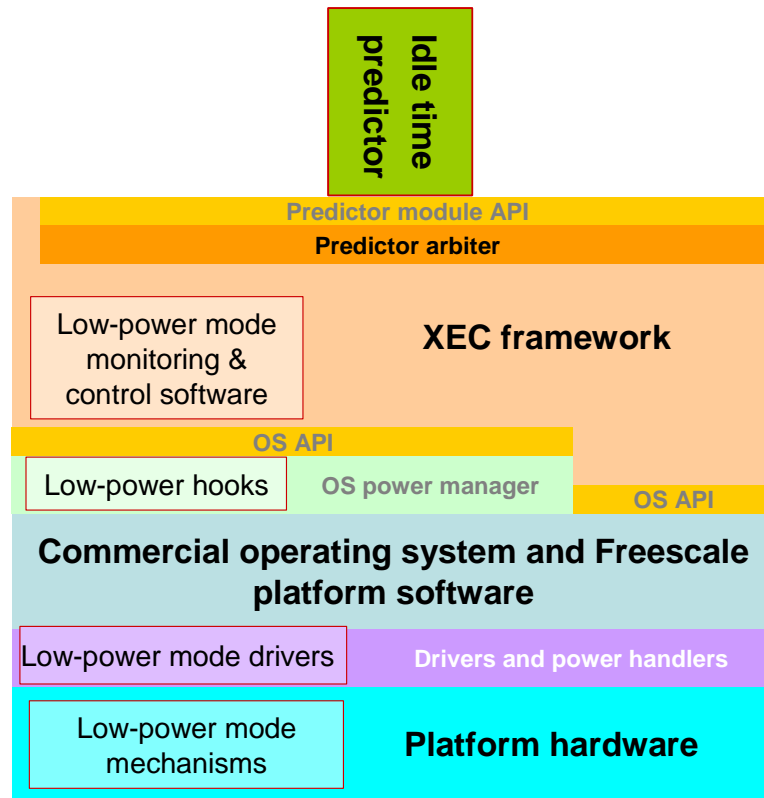


Figure 6. Idle Time Predictor

In principle, any hardware module with software-settable power-saving modes can be managed by the idle time policy software for minimum power waste.

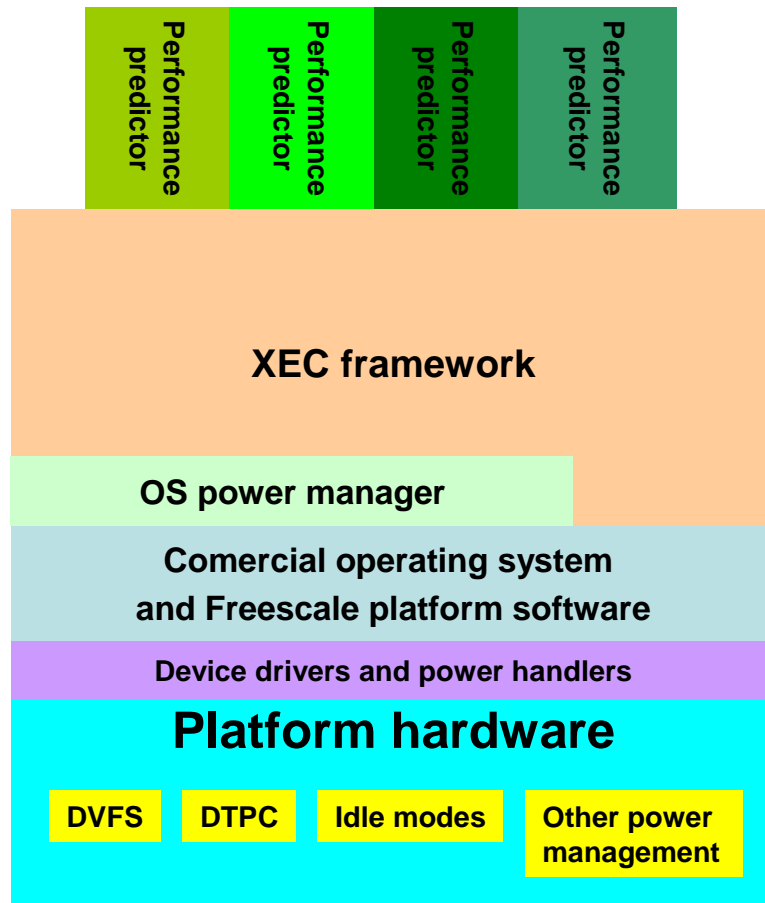


Figure 7. Performance Predictors

XEC Framework

The XEC framework isolates the algorithmic components (e.g. the performance predictors) from the details of the specific platform OS and hardware. It contains platform-specific mechanisms for controlling and monitoring hardware power-saving technologies (e.g. DVFS). It also provides common API interfaces for power-aware application programs and replaceable XEC modules like the performance predictors.

Because performance predictors may operate concurrently, a predictor arbiter arbitrates between multiple predictors to resolve which of their performance-power recommendations to select at any given time. The arbiter is programmable, so that the priority of any one performance predictor may be changed during normal operation depending on the runtime circumstances. Another framework component called the policy manager, acting as an agent for the product user, dynamically selects a policy (from a group of policies) that sets the criteria for trading off performance, power and energy in general or for specific situations or programs.

The XEC framework uses platform-specific power-cost rules to determine when and if to transition a particular hardware module from its current power state. Finally, the framework has an OS adaptation

layer containing OS-specific code. Its purpose is to minimize the porting effort and configuration management problems for migrating the XEC software between the operating systems on which it runs.

Multi-Generational XEC for Multiple Platforms

Energy conservation technology is not a one-time solution. For the foreseeable future, the energy demands of new and exciting software programs on mobile devices could outstrip the capability of battery technology to match those demands in a small and lightweight form. To meet user expectation of battery recharge cycle times, system designers must continue to find innovative ways of reducing power waste while meeting instantaneous performance needs. Freescale's XEC technology is architected to be extensible for future enhanced generations and easily deployable across multiple hardware platforms and operating system software.

How to Reach Us:

Home Page:
www.freescale.com

e-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
1-800-521-6274
480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064, Japan
0120 191014
+81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate,
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor
Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447
303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright license granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. ARM is the registered trademark of ARM Limited.

© Freescale Semiconductor, Inc. 2006.

Document Number: XTMENRGYCVSWP

