ISSN: 2582-8118 ***Volume 3, Issue 2; March 2023***

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYSIS IN SCIENCE AND ENGINEERING

Web: https://www.iarj.in/index.php/ijrase/index

2. Management of Power Behavior for Embedded System

Anitha M. T.

ABSTRACT

Power management and power efficiency go hand in hand in embedded designs and are critical for ensuring a viable end product that is also environmentally friendly. This is true for a wide variety of embedded products today, ranging from industrial applications to medical devices and other mission critical use cases. The factors that contribute to optimal power efficiency are complex, and this is an area in which OEMs frequently struggle in order to develop solutions that sell and meet the needs of their customers while also doing their part for the environment. Knowing the pitfalls and choosing a development platform designed to assist developers in designing power-efficient products are critical issues. This paper examines techniques and tools for designing power-efficient embedded systems while taking hardware into account.

KEYWORDS

Power Behaviour, embedded System, power-efficiency, Power Development, Power Management.

Introduction:

Anyone who works with electronics, whether as a designer or in another capacity, is aware that electronic components produce a lot of heat. They can also consume a lot of power while running. With such an emphasis on energy efficiency in today's world, there is always a focus on reducing power consumption in any electronic system.

For a variety of reasons, embedded systems are prime targets for reduced power consumption and increased energy efficiency. Some of these systems must run on batteries and have long lifetimes with specific form factor requirements. Other systems may draw power from the grid or a generator, but they should not waste power in order to reduce total energy consumption and prevent excess heat generation. [1]

Some basic power management techniques in embedded systems can help reduce heat generation, excess power consumption while the system is idle, and much more. Today's components, highly efficient regulator designs, and advanced power management algorithms can all be very useful in ensuring the energy efficiency of a new embedded system. If you're designing a new embedded system, keep these tips in mind to keep your system running at peak efficiency. [2-3]

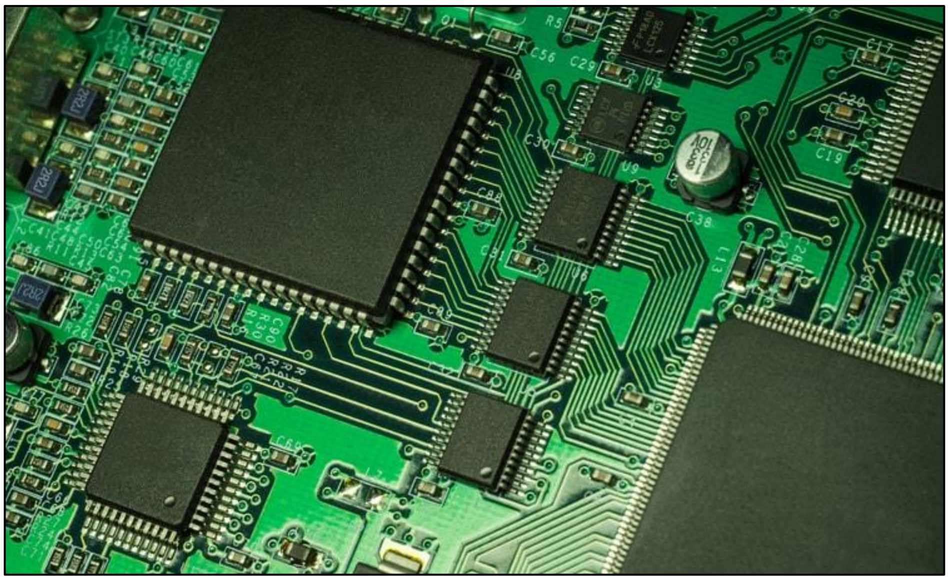

Figure 1: The ICs in this image can have lower power sleep modes, where they do not consume power until they receive a wake/alarm signal

Result and Discussion:

Power Reduction Using Power Management Techniques in Embedded Systems:

Every embedded system provides computing power for a specific purpose, but they contain much more than just a CPU. Many embedded system areas can be targeted for energy reduction using power management strategies:

- Processor: The processor is the first target for power consumption reduction. Many processing units include a variety of onboard energy-saving features. Similarly, systems with multiple processing blocks can be throttled on and off to consume power only when necessary.
- Power regulation: Choosing a power regulation strategy is critical for ensuring that the system's power supply is highly efficient while still supplying the necessary amount of power to various system blocks.
- Peripherals: Just like processors, various peripheral units can consume power even when not in use.
- Signalling protocol selection: Different signalling protocols and digital interfaces consume and require different amounts of power and current.
- Wireless communications: Analogue front-ends and wireless blocks in embedded systems are one area where power consumption can be reduced significantly.
- Embedded firmware and software: Expertise in developing efficient algorithms for carrying out processing instructions is required in these areas.

The table below illustrates some opportunities for energy savings in each of the previously mentioned areas. Some of these topics are more prevalent in embedded system development and deserve more attention. These specific areas are discussed further below. [4-6]

Processor	Various processors may enter low power modes automatically, or logic may be implemented in code to activate low power modes.
Power regulation	To maximise efficiency for a given power draw, power regulation should be used. To ensure efficient power conversion, use a PFC circuit in AC-DC converters.
Peripherals	Power to peripherals can be cut off as needed by placing them in a layout with their own regulation section.
Signaling protocol selection	Because many components can communicate via multiple interfaces, the designer and firmware/software developer have some leeway in selecting the lowest power interfaces.
Wireless communications	Wireless protocols may not necessitate a constant connection, allowing RFFEs and modems to be turned off as needed.
Embedded firmware and software	Firmware/software can be optimised, or power management strategies can be implemented to turn on/off peripherals.

Software and Firmware:

Code execution strategies, in a nutshell, involve writing code in such a way that the system completes required tasks while minimising the number of logical operations. The use of digital logic to implement a computational task can increase the algorithmic complexity of the task, so firmware developers should think about the logic they use to implement their algorithms.

For systems such as single-board computers, the operating system should be optimised to reduce the number of background tasks. Because they can be customised with as much or as little peripheral application support as needed, Linux kernels are the industry standard for embedded computers. By removing background processes and services, the processor will perform fewer tasks while idle, consuming less power. [7]

Component and Processor Selection:

As shown in the table above, component selection in each of the above areas has a significant impact on power consumption. Many processor units (MCUs, FPGAs, MPUs, and so on) are specifically marketed as low-power components, allowing the following power management techniques to be used in embedded systems:

- Run instructions with different signalling protocols so that the lowest power protocol can be selected if necessary.
- Enter a low-power hibernation or idle mode, in which the core voltage and frequency are reduced to conserve power.

- Lowering the sampling rate of ADCs or other receivers reduces total power consumption when interacting with analogue sensors.
- Dynamically turn on and off various peripherals.
- Use peripherals with a disable/enable switch that can be triggered by a standard low-speed digital protocol (I2C, SPI, GPIO, and so on).

The final point necessitates a power management strategy in which power is distributed to various functional blocks, each of which has its own power regulator. This is depicted in the block diagram below, which can be realised using a sensing interface and a low-speed digital interface. [8-10]

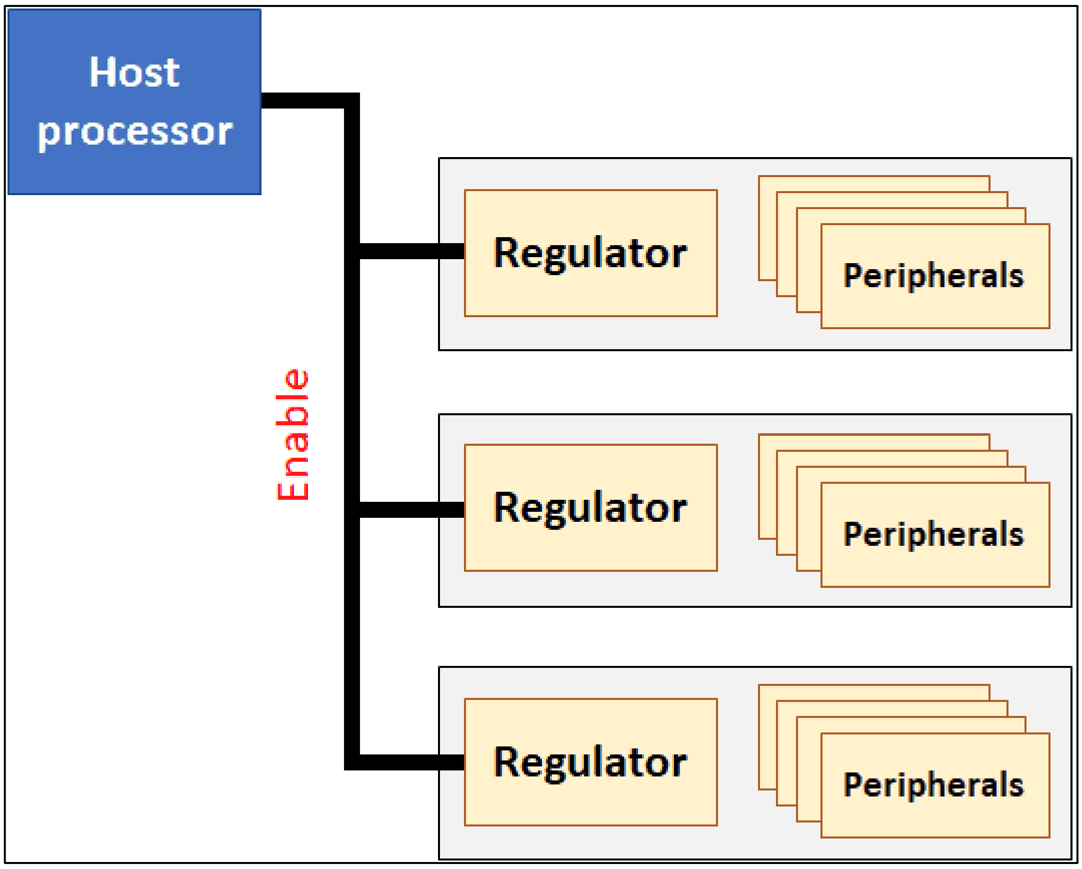

Figure 2: Block diagram showing peripheral power management techniques in embedded systems

Use the best set of system design and analysis tools you can find when implementing any power management techniques in embedded systems. Through an integrated set of field solvers, Cadence provides powerful software that automates many important tasks in systems analysis, such as power integrity simulations and power management analysis. [11-12]

Power Management and Power Efficiency: The Key Concepts

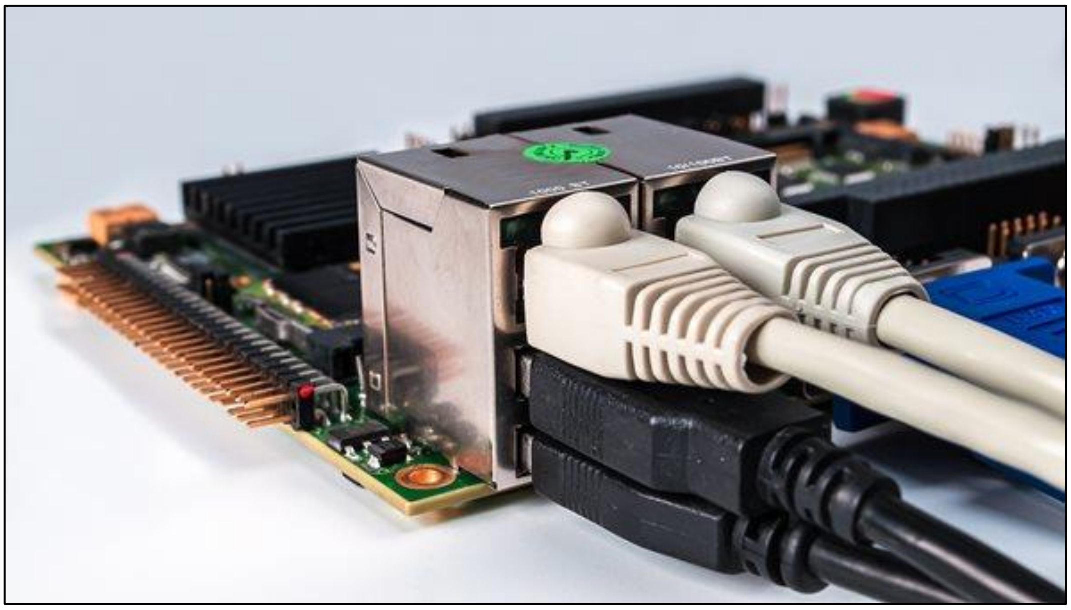

Figure 3: Power Efficiency: The Key Concepts

Power management and power efficiency can be differentiators when designing products for industrial, medical, or other demanding applications.

- Power management is the process of controlling power consumption in a system using hardware or software. You can, for example, actively or automatically disable an unused system component in order to save power.
- On the other hand, power efficiency is officially defined as the ratio of power output to total power input provided to a system.

Putting Power Management to Use:

Furthermore, I would like to broaden the definition of power efficiency to include the ratio of useful power output to total power input to a system. In the context of an IoT or computing system, useful power output can be defined as the feature set that a product can support with a given power budget. Connecting a display, a camera, and other internal or external components to a system, for example. [13]

This is a critical parameter in any system design and integration, particularly for mobile or space-constrained devices. One common goal in product development is to optimise the power budget in order to support a required feature set and/or to use less energy to perform the same task, thereby eliminating energy waste.

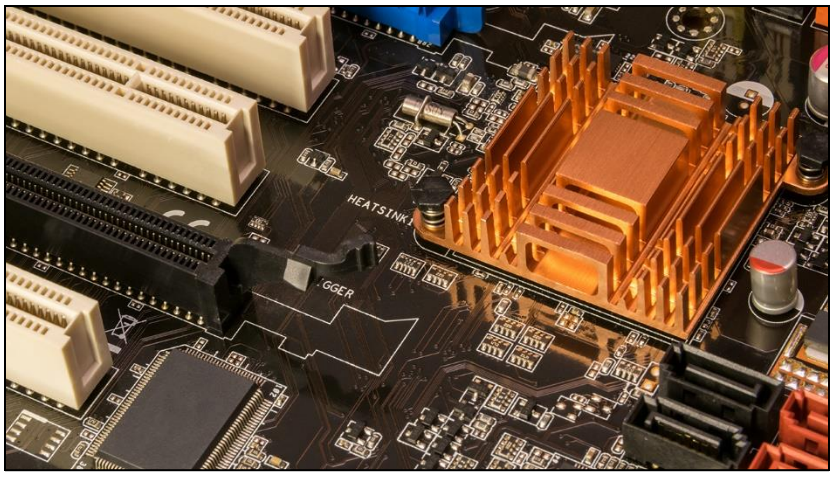

Figure 4: Power Management to Use

There are some overarching reasons for power efficiency that are unrelated to specific application requirements. For example, more efficient power use can help to reduce greenhouse gas emissions and thus have a lower environmental impact. It can also help to reduce product operating costs due to lower power consumption.

In addition to lowering costs and environmental impact, many specific applications require power efficiency. Consider some examples. [14-15]

- The OEM may have a fixed power budget in an industrial application such as agricultural irrigation or feedbin monitoring. By increasing the system's efficiency, more compute power can be integrated and additional features can be deployed, but only when necessary.
- Size may be the most limiting factor in a medical or transportation application. As a result of producing less heat, running the system more efficiently allows you to reduce the number of heat sinks and fans. Simultaneously, a smaller power subsystem would be required. This can even improve the device's safety and reliability.
- In an untethered medical application, such as a cart-based product, system mobility may be required, necessitating the use of a smaller, more agile platform. If the system is lighter in weight or designed for extended battery operation, this decision is much easier.

Power Management Techniques and Strategies:

It is best to use the component in the system with the lowest power consumption for power management tasks whenever possible. Here are some strategies to think about: [16-17]

- To "outsource" power management, many developers use external very low power microcontrollers.
- Digi includes a low-power controller called "microcontroller assist," or MCA, on ConnectCore SOMs that is separate from the processing cores and can be used for power management. The MCA, for example, includes GPIOs, tampering pins, and a

serial port as an additional wakeup source. The system can be set to wake up when a magic word is received on the serial interface of this MCA.

- The Power key is a useful feature that we discussed earlier, as it allows you to use different boot modes.
- Wake on LAN — an option to remotely wake the system over a network, similar to the serial interface — is available on many systems, using a magic word, pattern, or arbitrary data to resume normal operation.
- Finally, waking up from a low power state via wireless is still uncommon in many systems because it requires the system's RF component to be powered in some way. Digi has recently added Wake on Bluetooth to its devices, which is a welcome addition to the list of wakeup sources, providing additional options for applications to interact with low-power devices.

Most of these features necessitate lower-level system access via a command line, or at the very least, configuration at the operating system level.

A user-friendly IoT system should include a mechanism for developers to use power management features directly from their application code.

Digi has added a programming interface called APIX to its operating system offerings, allowing programmers to access many of the system's embedded hardware interfaces, as well as high level access to power management capabilities. [18]

This can be used to control CPU and GPU frequencies, disable and enable CPU cores, and set up temperature management templates, among other things.

Conclusion:

Embedded systems are devices that perform specific tasks despite having limited resources such as memory, processing power, and battery life. Power management is an important aspect of embedded system design because it can affect the device's performance, reliability, and lifetime. In this article, we will look at how to debug and test your embedded system's power management functionality, as well as what tools and techniques you can use to optimise it.

References:

1. V. Tiwari, S. Malik, and A. Wolfe, “Power analysis of embedded software: A first step towards software power minimization,” Proc. Int. Conf. Computer-Aided Design, Nov. 1994.
2. T. Sato, Y. Ootaguro, M. Nagamatsu, and H.Tago,“Evaluation of architecture-level power estimation for CMOS RISC processors,” Proc. Int. Symp. Low Power Electronics, pages 44-45, Oct. 1995.
3. C.-T. Hsieh, M. Pedram, G. Mehta, and F.Rastgar, “Profile-driven program synthesis for evaluation of system power dissipation,” Proc. Design Automation Conf., pages 576-581, June 1997.

4. Li and J.Henkel, “A framework for estimating and minimizing energy dissipation of embedded HW/SW systems,” Proc. Design Automation Conf., pages 188-193, June 1998.
5. C-T. Hsieh and M. Pedram, “Micro-processor power estimation using profile-driven program synthesis," IEEE Trans. on Computer Aided Design, Vol. 17, No. 11, pages 1080-1089, Nov. 1998.
6. T. Simunic, L. Benini, and G. De Micheli, “Cycleaccurate simulation of energy consumption in embedded systems,” Proc. Design Automation Conf., pages 867-872, June 1999.
7. L. Benini and G. De Micheli, “System-level power optimization: Techniques and tools,” Proc. Int. Symp. Low Power Electronics and Design, Aug. 1999.
8. V. Tiwari, S. Malik, and A. Wolfe, “Compilation techniques for low energy: An overview,” Proc. Int. Symp. Low Power Electronics, pages 38-39, Oct. 1994.
9. T. Simunic, G. De Micheli, and L. Benini, “Energyefficient design of battery-powered embedded systems,” Proc. Int. Symp. Low Power Electronics and Design, Aug. 1999.
10. F. Catthoor, F. Franssen, S. Wuytack, L. Nachtergaele, and H. De Man, “Global communication and memory optimizing transformations for low power signal processing systems,” Proc. Int. Wkshp. on Low Power Design, pages 203-208, Apr. 1994.
11. J. L. da Silva, F. Catthoor, D. Verkest, and H. De Man, “Power exploration for dynamic data types through virtual memory management refinement,” Proc. Int. Symp. Low Power Electronics and Design, Aug. 1999.
12. C. L. Su, C-Y. Tsui and A. M. Despain, “Low power architecture design and compilation techniques for high performance processors,” Proc.IEEE Compcon, pages 489-498, 1994.
13. V. Tiwari, S. Malik and A. Wolfe, “Instruction level power analysis and optimization of software,” Journal of VLSI Signal Processing, pages 1-18, 1996.
14. M. Srivastava, A. Chandrakasan, and R. Brodersen, “Predictive system shutdown and other architectural techniques for energy effcient programmable computation," IEEE Trans. on VLSI Systems, Vol. 4, No. 1 (1996), pages 42-55.
15. C.-H. Hwang and A. Wu, “A predictive system shutdown method for energy saving of event-driven computation,” Proc. Int.. Conf. on Computer Aided Design, pages 28-32, November 1997.
16. L. Benini, A. Bogliolo, G.A. Paleologo and G. De Micheli, “Policy optimization for dynamic power management,” IEEE Trans. on Computer-Aided Design, Vol. 18, No. 6 (1999), pages 813-833.
17. E. Chung, L. Benini and G. De Micheli, “Dynamic power management for non stationary service requests", Proc. Design and Test in Europe Conference, pages 77-81, March 1999.
18. B. P. Dave, G. Lakshminarayana, and N. K. Jha, “COSYN: Hardware-software co-synthesis of distributed embedded systems,” Proc. Design Automation Conf., pages 703-708, June 1997.