IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 22, Issue 1, Ser. IV (Jan - Feb 2020), PP 35-40
www.iosrjournals.org

Technique to Minimize the Power Consumption in Microprocessors

Dr.Alhamali Masoud Alfrgani .Ali

Department of Computer Sciences& Information Technology, Technology College of Civil Aviation & Meterology,aspaia, Libya.

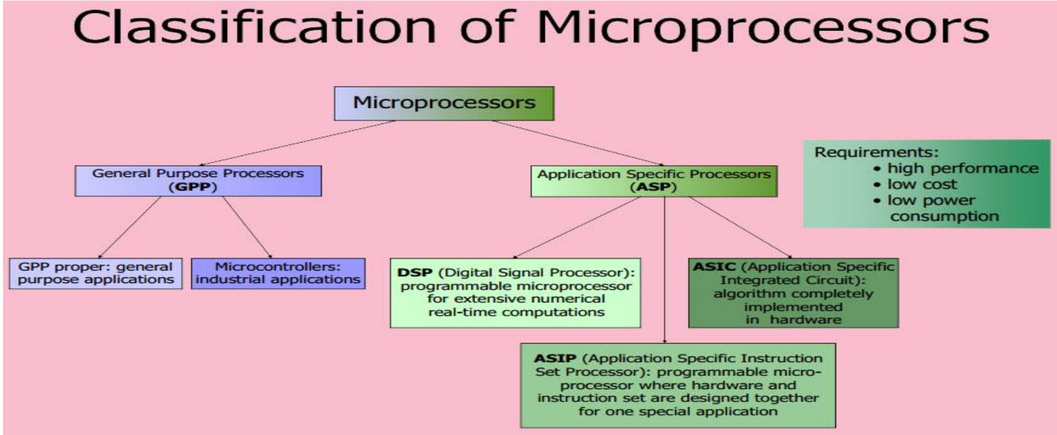
Abstract

Microprocessors are the main part of the computer system. It is very important to reduce the consumption of microprocessors. Power consumption is a major factor that limits the performance of computers. It is based on the survey that reduces the total power consumed by a microprocessor system over time. These techniques are applied at various levels ranging from circuits to architectures, architectures to system software, and system software to applications. They also include high approaches that will become more important over the next decade. We conclude that power management is a multifaceted discipline that is continually expanding with new techniques being developed at every level. These techniques may eventually allow computers to break through the "power wall" and achieve unprecedented levels of performance, versatility, and reliability.

Date of Submission: 12-03-2020 Date of Acceptance: 27-03-2020

I. Introduction

Computer engineers have always tried to improve the performance of computers. But although today's computers are much faster and far more versatile than their predecessors, they also consume a lot of power, so much power, in fact, that their power densities and concomitant heat generation are rapidly approaching levels comparable to nuclear reactors. These high power densities impair chip reliability and life expectancy, increase cooling costs, and, for large data centers, even raise environmental concerns.



Classification of Microprocessors

Microprocessors

General Purpose Processors (GPP)

Application Specific Processors (ASP)

Requirements:
- high performance
- low cost
- low power consumption

GPP proper: general purpose applications

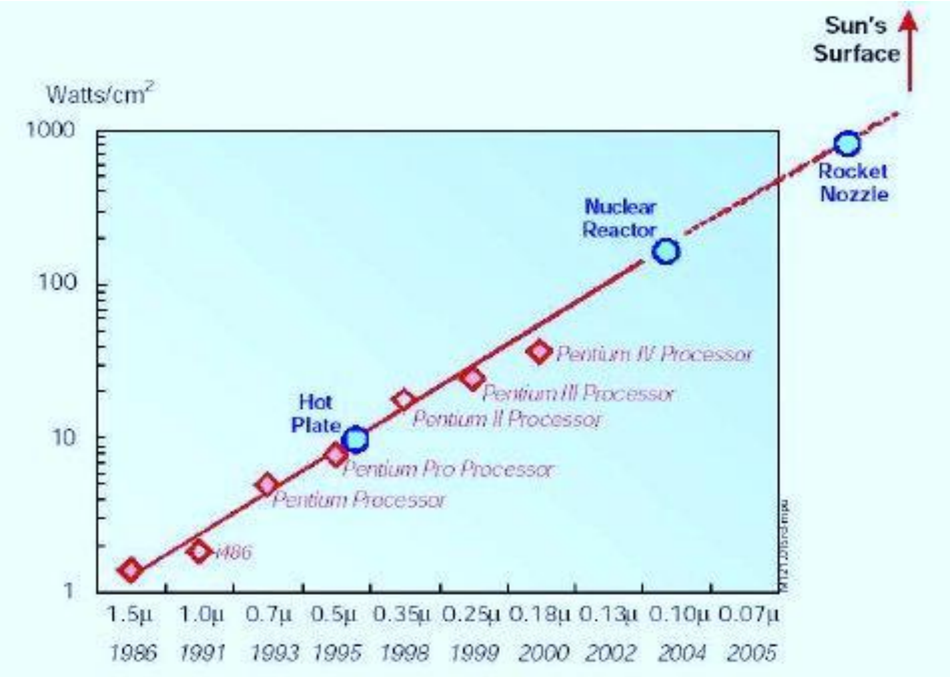
Microcontrollers: industrial applications

DSP (Digital Signal Processor): programmable microprocessor for extensive numerical real-time computations

ASIC (Application Specific Integrated Circuit): algorithm completely implemented in hardware

ASIP (Application Specific Instruction Set Processor): programmable micro-processor where hardware and instruction set are designed together for one special application

At the other end of the performance spectrum, power issues also pose problems for smaller mobile devices with limited battery capacities. Although one could give these devices faster processors and larger memories, this would diminish their battery life even further. Without cost effective solutions to the power problem, improvements in micro-processor technology will eventually reach a standstill. Power management is a multidisciplinary field that involves many aspects (i.e., energy, temperature, reliability), each of which is complex enough to merit a survey of its own.



The focus of our survey will be on techniques that reduce the total power consumed by typical microprocessor systems. We will follow the high-level taxonomy illustrated above Figure. First, it will define power and energy and explain the complex parameters that dynamic and static power depends. Next, we will introduce techniques that reduce power and energy. it will discuss some commercial power management systems and provide a glimpse into some more radical technologies that are emerging.

II. Defining Power

Power and energy are commonly defined in terms of the work that a system performs. Energy is the total amount of work a system performs over a period of time, while power is the rate at which the system performs that work. In formal terms,P=W/T

E = P ∗ T, where P is power, E is energy, T is a specific time interval, and W is the total work performed in that interval. Energy is measured in joules, while power is measured in watts. These concepts of work, power, and energy are used differently in different contexts. In the context of computers, work involves activities associated with running programs (e.g., addition, subtraction, memory operations), power is the rate at which the computer consumes electrical energy (or dissipates it in the form of heat) while performing these activities, and energy is the total electrical energy the computer consumes (or dissipates as heat) over time. This distinction between power and energy is important because techniques that reduce power do not necessarily reduce energy.

For example, the power consumed by a computer can be reduced by halving the clock frequency, but if the computer then takes twice as long to run the same programs, the total energy consumed will be similar. Whether one should reduce power or energy depends on the context. In mobile applications, reducing energy is often more important because it increases the battery lifetime. However, for other systems (e.g., servers), temperature is a larger issue. To keep the temperature within acceptable limits, one would need to reduce instantaneous power regardless of the impact on total energy.

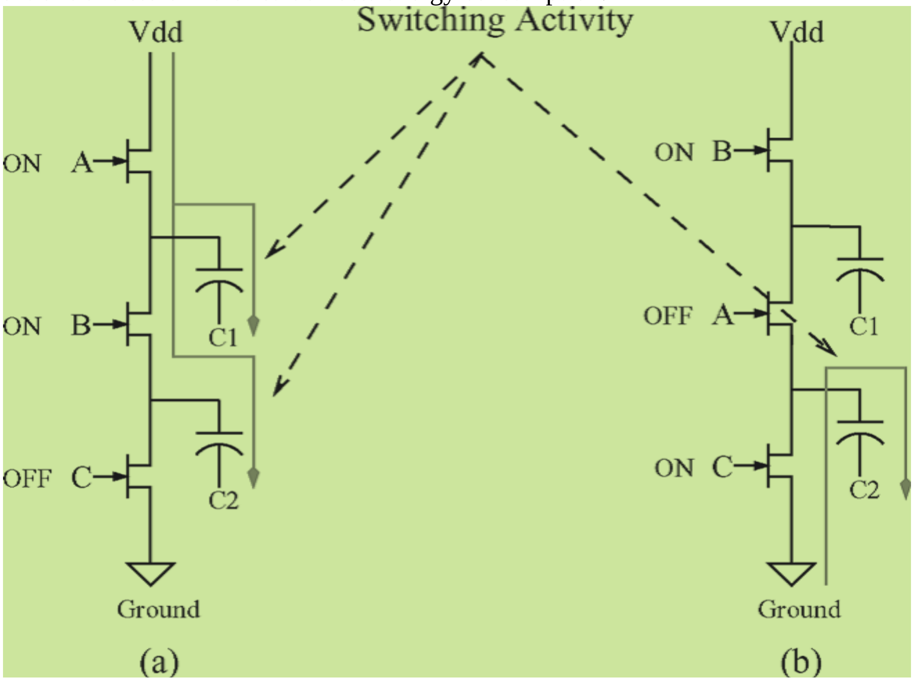
III. Reducing Power

Circuit And Logic Level Techniques

Transistor sizing reduces the width of transistors to reduce their dynamic power consumption, using low-level models that relate the power consumption to the width. According to these models, reducing the width also increases the transistor's delay and thus the transistors that lie away from the critical paths of a circuit are usually the best candidates for this technique. Algorithms for applying this technique usually associate with each transistor a tolerable delay which varies depending on how close that transistor is to the critical path. These algorithms then try to scale each transistor to be as small as possible without violating its tolerable delay.

Transistor Reordering

The arrangement of transistors in a circuit affects energy consumption.



shows two possible implementations of the same circuit that differ only in their placement of the transistors marked A and B. Suppose that the input to transistor A is 1, the input to transistor B is 1, and the input to transistor C is 0. Then transistors A and B will be on, allowing current from Vdd to flow through them and charge the capacitors C1 and C2. Now suppose that the inputs change and that A's input becomes 0, and C's input becomes 1. Then A will be off while B and C will be on. Now the implementations in (a) and (b) will differ in the amounts of switching activity. In (a), current from ground will flow past transistors B and C, discharging both the capacitors C1 and C2. However, in (b), the current from ground will only flow past transistor C; it will not get past transistor A since A is turned off. Thus it will only discharge the capacitor C2, rather than both C1 and C2 as in part (a). Thus the implementation in (b) will consume less power than that in (a). Transistor reordering rearranges transistors to minimize their switching activity. One of its guiding principles is to place transistors closer to the circuit's outputs if they switch frequently in order to prevent a domino effect where the switching activity from one transistor trickles into many other transistors causing widespread power dissipation. This requires profiling techniques to determine how frequently different transistors are likely to switch.

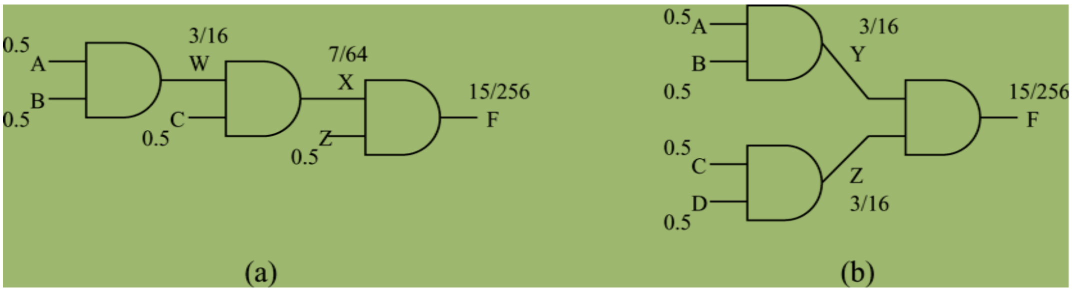
Half Frequency and Half Swing Clocks

Half-frequency and half-swing clocks reduce frequency and voltage, respectively. Traditionally, hardware events such as register file writes occur on a rising clock edge. Half-frequency clocks synchronize events using both edges, and they tick at half the speed of regular clocks, thus cutting clock switching power in half. Reduced-swing clocks also often use a lower voltage signal and thus reduce power.

Logic Gate Restructuring

There are many ways to build a circuit out of logic gates. One decision that affects power consumption is how to arrange the gates and their input signals.

For example, consider two implementations of a four-input AND gate, a chain implementation (a), and a tree implementation (b). Knowing the signal probabilities (1 or 0) at each of the primary inputs (A,B,C, D), one can easily calculate the transition probabilities (0→1) for each output (W, X, F, Y, Z). If each input has an equal probability of being a 1 or a 0, then the calculation shows that the chain implementation (a) is likely to switch less than the tree implementation (b). This is because each gate in a chain has a lower probability of having a 0→1 transition than its predecessor; its transition probability depends on those of all its predecessors. In the tree implementation, on the other hand, some gates may share a parent (in the tree topology) instead of being directly connected together. These gates could have the same transition probabilities.



Nevertheless, chain implementations do not necessarily save more energy than tree implementations. There are other issues to consider when choosing a topology. One is the issue of glitches or spurious transitions that occur when a gate does not receive all of its inputs at the same time. These glitches are more common in chain implementations where signals can travel along different paths having widely varying delays. One solution to reduce glitches is to change the topology so that the different paths in the circuit have similar delays. This solution, known as path balancing often transforms chain implementations into tree implementations. Another solution, called retiming, involves inserting flip-flops or registers to slow down and thereby synchronize the signals that pass along different paths but re converges to the same gate. Because flip-flops and registers are in sync with the processor clock, they sample their inputs less frequently than logic gates and are thus more immune to glitches.

IV. Low-Power Memories and Memory Hierarchies

One can classify memory structures into two categories, Random Access Memories (RAM) and Read-Only-Memories (ROM). There are two kinds of RAMs, static RAMs (SRAM) and dynamic RAMs (DRAM) which differ in how they store data. SRAMs store data using flip-flops and DRAMs store each bit of data as a charge on a capacitor; thus DRAMs need to refresh their data periodically. SRAMs allow faster accesses than DRAMs but require more area and are more expensive. As a result, normally only register files, caches, and high bandwidth parts of the system are made up of SRAM cells, while main memory is made up of DRAM cells. Although these cells have slower access times than SRAMs, they contain fewer transistors and are less expensive than SRAM cells. The techniques we will introduce are not confined to any specific type of RAM or ROM. Rather they are high-level architectural principles that apply across the spectrum of memories to the extent that the required technology is available. They attempt to reduce the energy dissipation of memory accesses in two ways, either by reducing the energy dissipated in a memory accesses, or by reducing the number of memory accesses

Low-Power Processor Architecture Adaptations

So far we have described the energy saving features of hardware as though they were a fixed foundation upon which programs execute. However, programs exhibit wide variations in behavior. Researchers have been developing hardware structures whose parameters can be adjusted on demand so that one can save energy by activating just the minimum hardware resources needed for the code that is executing.

V. Adaptive Caches

There is a wealth of literature on adaptive caches, caches whose storage elements (lines, blocks, or sets) can be selectively activated based on the application workload. One example of such a cache is the Deep-Submicron Instruction (DRI) cache. This cache permits one to deactivate its individual sets on demand by gating their supply voltages. To decide what sets to activate at any given time, the cache uses a hardware profiler that monitors the application’s cache-miss patterns. Whenever the cache misses exceed a threshold, the DRI cache activates previously deactivated sets. Likewise, whenever the miss rate falls below a threshold, the DRI deactivates some of these sets by inhibiting their supply voltages.

A problem with this approach is that dormant memory cells lose data and need more time to be reactivated for their next use. Thus an alternative to inhibiting their supply voltages is to reduce their voltages as low as possible without losing data. This is the aim of the drowsy cache, a cache whose lines can be placed in a drowsy mode where they dissipate minimal power but retain data and can be reactivated faster.

Compiler-Level Power Management

There are many ways a compiler can help reduce power regardless of whether a processor explicitly supports software-controlled power reduction. Aside from generating code that reconfigures hardware units or activates power reduction mechanisms that we have seen, compilers can apply common performance-oriented optimizations that also save energy by reducing the execution time, optimizations such as Common Sub expression Elimination, Partial Redundancy Elimination, and Strength Reduction. However, some performance optimizations increase code size or parallelism, sometimes increasing resource usage and peak power dissipation. Examples include Loop Unrolling and Software Pipelining. Researchers have developed models for relating performance and power, but these models are relative to specific architectures. There is no fixed relationship between performance and power across all architectures and applications.

Application-Level Power Management

Researchers have been exploring how to give applications a larger role in power management decisions. Most of the recent work has two goals. The first is to develop techniques that enable applications to adapt to their runtime environment. The second is to develop interfaces allowing applications to provide hints to lower layers of the stack (i.e., operating systems, hardware) and likewise exposing useful information from the lower layers to applications. Although these are two separate goals, the techniques for achieving them can overlap.

Application Transformations and Adaptations

As a starting point, some researchers have adopted an “architecture-centric” view of applications that allows for some high-level transformations. These researchers claim that an application’s architecture consists of fundamental elements that comprise all applications, namely processes, event handlers, and communication mechanisms. Power is consumed when these various elements interact during activities such as context switches and inter process communication. They propose a low-power application design methodology. Initially, an application is represented as a software architecture graph (SAG) Power Reduction Techniques for Microprocessor Systems 225 which captures how it has been built out of processes and events. It is then run through a simulator to measure its base energy consumption which is then compared to its energy consumption when transformations are applied to the SAG, transformations that include merging processes to reduce inter process communication, replacing expensive IPC mechanisms by cheaper mechanisms, and migrating computations between processes. To determine the optimal set of transformations, the researchers use a greedy approach. They keep applying transformations that reduce energy until no more transformations remain to be applied.

This results in an optimized version of the same application that has a more energy efficient mix of processes, events, and communication mechanisms. It explored a different kind of adaptation that involves trading the accuracy of computations for reduced energy consumption. They propose a video encoder that allows one to vary its compression efficiency by selectively skipping the motion search and discrete cosine transform phases of the encoding algorithm. The extent to which it skips these phases depends on parameters chosen by the adaptation algorithm. The target architecture is a processor with support for DVS and architectural adaptations (e.g., configurable caches). Two algorithms work side by side. One algorithm tunes the hardware parameters at start of each frame, while another tunes the parameters of the video encoder while a frame is being processed.

Commercial Systems

To understand what power management techniques industry is adopting, we examine the low-power techniques used in four widely used processors, the Pentium 4, Pentium M, the PXA27x, and Transmeta Crusoe.We then discuss three power management strategies by IBM, ARM, and National Semiconductor that are rapidly gaining in importance.

The Pentium 4 Processor.

Though its goal is high performance, the Pentium 4 processor also contains features to manage its power consumption. One of Intel’s goals in including these features was to prevent the Pentium’s internal temperatures from becoming dangerously high due to the increased power dissipation. To meet this goal, the designers included a thermal detection and response mechanism which inhibits the processor clock whenever the observed temperature exceeds a safe threshold. To monitor temperature variations, the processor features a diode based thermal sensor. The sensor sits at the hottest area of the chip and measures temperature via voltage drops across the diode. Whenever the temperature increases into a danger zone, the sensor issues a STOPCLOCK request, causing the main clock signal to be inhibited from reaching most of the processor until the temperature is no longer in the danger zone. This is to guarantee response time while the chip is cooling. The temperature sensor also provides temperature information to higher levels of software through output signals (some of which are in registers), allowing the compiler or operating system, for instance, to activate other techniques in response to the high temperatures. The Pentium 4 also supports the low power operating states defined by the Advanced Configuration and Power Interface (ACPI) specification, allowing software to control the processor power modes. In particular, it features a model-specific register allowing software to influence the processor clock. In addition to stopping the clock, the Pentium 4 features the Intel Basic Speed step Technology which allows two settings for the processor clock frequency and voltage, a high setting for performance and a low setting for power. The high setting is normally used when the computer is connected to a wall outlet, and the low setting is normally used when the computer is running on batteries as in the case of a laptop.

VI. The Pentium M Processor

The Pentium M is the fruit of Intel's efforts to bring the Pentium 4 to the mobile Domain. It carefully balances performance enhancing features with several power saving features that increase the battery lifetime. It uses three main strategies to reduce dynamic power consumption: reducing the total instructions and micro-operations executed, reducing the switching activity in the circuit, and reducing the energy dissipated per transistor switch. To save energy, the Pentium M integrates several techniques that reduce the total switching activity. These include hardware for predicting idle units and inhibiting their clock signals, buses whose components are activated only when data needs to be transferred, and a technique called execution stacking which clusters units that perform similar functions into similar regions so that the processor can selectively activate the parts of the circuit that will be needed by an instruction. To reduce the static power dissipation, the Pentium M incorporates low leakage transistors in the caches. To further reduce both dynamic and leakage energy throughout the processor, the Pentium M supports an enhanced version of the Intel Speed Step technology, which unlike its predecessor, allows the processor to transition between 6 different frequency and voltage settings.

VIII. Conclusion

Power and energy management has grown into a multifaceted effort that brings together researchers from such diverse areas as physics, mechanical engineering, electrical engineering, design automation, logic and high-level synthesis, computer architecture, operating systems, compiler design, and application development. It have examined how the power problem arises and how the problem has been addressed along multiple levels ranging from transistors to applications. It have also surveyed major commercial power management technologies and provided a glimpse into some emerging technologies. It conclude by noting that the field is still active, and that researchers are continually developing new algorithms and heuristics along each level as well as exploring how to integrate algorithms from multiple levels. Given the wide variety of micro architectural and software techniques available today and the astoundingly large number of techniques that will be available in the future, it is highly likely that we will overcome the limits imposed by high power consumption and continue to build processors offering greater levels of performance and versatility. However, only time will tell which approaches will ultimately succeed in solving the power problem

References

[1]. MARTIN, T. AND SIEWIOREK, D. 2001. Nonideal battery and main memory effects on cpu speedsetting for low power. IEEE Tran. (VLSI) Syst. 9, 1, 29–34.

[2]. MENG. Y., SHERWOOD, T., AND KASTNER, R. 2005. Exploring the limits of leakage power reduction in caches. ACMTrans. Architecture Code Optimiz., 1, 221–246.

[3]. HINTON, G., SAGER, D.,UPTON, M., BOGGS, D.,CARMEAN, D., KYKER, A., AND ROUSSEL, P. 2004. The microarchitecture of the Intel Pentium 4 processor on 90nm technology. Intel Tech. J. 8, 1, 1–17.

[4]. ANAND, M., NIGHTINGALE, E., AND FLINN, J. 2004. Ghosts in the machine: Interfaces for better power management. In Proceedings of the International Conference on Mobile Systems, Applications, and Services. 23–35. DALTON, A. B. AND ELLIS, C. S. 2003. Sensing user intention and context for energy management. In Proceedings of the 9thWorkshop on Hot Topics in Operating Systems. 151–156.

[5]. DE, V. AND BORKAR, S. 1999. Technology and design challenges for low power and high performance. In Proceedings of the International Symposium on Low Power Electronics and Design ISLPED'99 . ACM Press, 163–168.

[6]. HEATH, T., PINHEIRO, E., HOM, J., KREMER, U., AND BIANCHINI, R. 2004. Code transformations for energy-efficient device management. IEEE Trans. Comput. 53, 8, 974–987.

[7]. HO, Y.-T. AND HWANG, T.-T. 2004. Low power design using dual threshold voltage. In Proceedings of the Conference on Asia South Pacific Design Automation IEEE Press, (Piscataway, NJ,) 205–208.

[8]. KANDEMIR, M., RAMANUJAM, J., AND CHOUDHARY, A. 2002. Exploiting shared scratch pad memory space in embedded multiprocessor systems. In Proceedings of the 39th Conference on Design Automation. ACM Press, 219–224.

[9]. KAXIRAS, S.,HU, Z., ANDMARTONOSI,M. 2001. Cache decay: exploiting generational behavior to reduce cache leakage power. In Proceedings of the 28th Annual International Symposium on Computer Architecture. ACM Press, 240–251.

Dr.Alhamali Masoud Alfrgani .Ali ph.D in the computer Engineering From Sam Higginbotom University of Agriculture, Technology & Sciences(2018).M.Tech Computer Engineering From Sam Higginbotom University of Agriculture, Technology & Sciences(2013) , B.tech degree in computer Engineering from Technology College of Civil Aviation & Meterology (1992), Tripoli–Libya. faculty member at Technology College of Civil Aviation from(01-10-2018)