Embedded Operating System Power Management

Roshin Lal R, *Indian Institute of Science, Bangalore*

***Abstract*— With the rapid evolution of submicron technology, manufacturers are integrating increasing numbers of components on one chip. A heterogeneous system on chip (SoC) might include one or more programmable components - general purpose processor cores, digital signal processor cores, or application specific intellectual property cores - as well as an analog front end, on-chip memory, I/O devices, and other application specific logic. A large and increasing number of modern embedded systems are subject to tight power/energy constraints. Historically, low power consumption in embedded processors has been achieved through the use of a number of low power idle and sleep modes. Embedded processors are now performing more sophisticated tasks, which require ever higher performance levels and typically run for considerable periods of time increasing the ratio of run time to idle time significantly. This paper looks at techniques by which power consumed by an embedded system can be analyzed and describes methods by which the power consumption can be reduced.**

Keywords: Embedded power management, operating system, voltage scaling, idle power

I. Introduction

EMBEDDED computer systems are characterized by the presence of a dedicated processor which executes application specific software. The block diagram of a standard heterogeneous system on chip is shown in Figure 1. Recent years have seen a large growth of such systems. There is rapidly growing consumer demand for computing devices that are both compute intensive and battery operated, including personal digital assistants (PDAs), cell phones, wearable computers, handhelds, and laptops [10], [11]. According to a new study released by Strategy Analytics Inc [2], the global cellular market will double in size to 1.7 billion by the end of 2006. This means that one in four of the global population will own a cellphone.

One of the key problems with mobile or wearable computing is energy consumption. In order to make mobile or wearable computing widely applicable, major advances in reducing power consumption are needed. In this paper we took at techniques for reducing system power consumption for an embedded system using policies implemented in the embedded operating system.

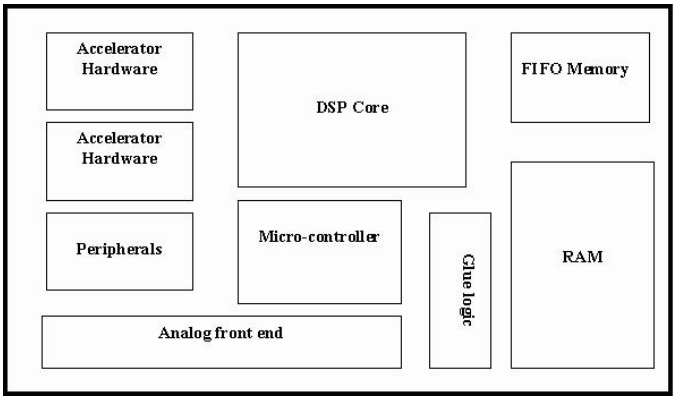

Fig. 1. Heterogeneous System on Chip (Soc)

The paper is organized as follows: Section II explains the factors governing power consumption of an embedded system. Section III gives a set of techniques by which the power consumed by an embedded system can be reduced through the embedded operating system. Section IV gives a short introduction to the TI OMAP 730 processor which will be used as the platform for implementing the techniques mentioned in this paper and finally, section V summarizes the paper and lists topics of interest that needs further work.

II. Power Consumption in CMOS Circuits

Unlike bipolar technologies, where a majority of power dissipation is static, the bulk of power dissipation in properly designed CMOS circuits is the dynamic charging and discharging of capacitances. Thus, a majority of the low power design methodology is dedicated to reducing this predominant factor of power dissipation.

There are three main sources of power dissipation in a CMOS device [4], [5]:

1) Dynamic Switching Power: Capacitive/switching/dynamic power ($P_{dynamic}$) dissipation is the dominant form of power dissipation today. It occurs due to the switching of charge in and out

of the CMOS circuit output load capacitance C_L. When CMOS circuits switch, the output is either charged up to V_{DD}, or discharged down to ground. In static logic design, the output only transitions on an input transition, while in dynamic logic, the output is precharged during half the clock cycle, and transitions can only occur in the second clock phase, depending upon the input values. In both cases, the power dissipated during switching is proportional to the capacitive load, however, they have different transition frequencies. $P_{dynamic}$ is given by

$$P_{dynamic} = C_L N_{SW} V_{DD}^2 f \quad (1)$$

where C_L is the CMOS circuit output load capacitance, N_{SW} is the average number of switching activities per clock cycle, V_{DD} is the supply voltage and f is the clock frequency. Low power techniques reduce one of more of C_L, N_{SW}, V_{DD}, f.

- C_L: fast algorithms, compact layout design
- N_{SW}: encoding
- V_{DD}: voltage scaling, variable voltage processors
- f: low-frequency and/or clock gating

2) Leakage: Ideally, CMOS circuits dissipate no static (DC) power since in the steady state there is no direct path from V_{DD} to ground. However leakage does exist and implies that OFF switches are not really off and becomes significant when system is in idle state. Leakage component ($P_{leakage}$) is becoming more and more dominant as we go to 0.13μm process technology and below. $P_{leakage}$ increases exponentially with reduction in the threshold voltage V_T.

3) Short Circuit Current Power: The short circuit component (P_{sc}) of power dissipation occurs due to the non-zero rise and fall times at the inputs. For the ideal case of step input, the transistors change state immediately, one turning on, one turning off. There is no conductive path from the supply to ground. For real circuits, however, the input signal will have some finite rise/fall time. Therefore, there will be a conductive path open because both devices are on.

The longer the input rise/fall time, the longer the short circuit current will continue to flow, and the average short circuit current increases. To minimize the total average short circuit current power, it is desirable to have equal input and output edge times. The peak magnitude of the short circuit current is dependent on the device size. The average current, however, is roughly independent of device size for a fixed load capacitance. While the peak magnitude of the current increases, the rise/fall time decreases so that the average current is the same. If all devices are sized up so that the load capacitances scales up proportionally, then the rise/fall time remains constant and the average current (and power) scales up linearly with device size.

Short-circuit current power is either linearly or quadratically dependent on the supply voltage, depending on the size of the channel length. While reducing the supply increases the duration of the current linearly due to increased rise/fall times, the peak magnitude of the current is reduced linearly (velocity saturation), such that the average current is approximately constant, and the average power is just a linear function of supply voltage (P = IV). For larger devices that are not velocity saturated, the average current is approximately linear with supply voltage so that the average power is a quadratic function of supply voltage. For most ICs, the short-circuit power dissipated is approximately 5-10% of the total dynamic power [16].

Hardware Block	Current
CPU	34mA
DRAM	29 mA
LCD Display	15 mA
Others	11 mA
Total	89 mA

TABLE I

CURRENT CONSUMPTION AT EACH HARDWARE BLOCK

III. POWER MANAGEMENT USING EMBEDDED OPERATING SYSTEM

A trend gaining in popularity in the world of embedded systems is the use of embedded operating systems [1], [3], [6]. Embedded operating systems are increasingly used in the development and deployment of embedded systems. Their benefits are well known: they provide numerous helpful facilities including cooperative and preemptive multitasking, multithreading, support for both periodic and aperiodic tasks, fixed-priority/dynamic-priority scheduling, semaphores, interprocess communication, shared memory, and memory management; in doing so, they can dramatically reduce the time to design, develop, and test a product [7], [9], [8].

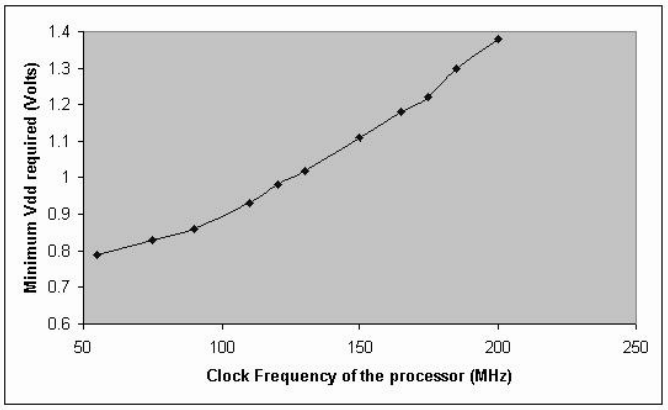

Fig. 2. Minimum operating voltage as a function of the clock frequency

In a portable computer, power is consumed mostly by the screen backlighting, the central processing unit (CPU), the memory, the hard disk, the display, and the keyboard [17]. Table I shows the power consumed by various hardware blocks in a representative system when the CPU load is 10% [24]. Dependence on battery means low power consumption. Since battery technology is fairly mature and the lifetime of a battery is not expected to increase very much over the next decade, low power consumption makes energy conservation a key issue in both hardware and software.

Power consumption is proportional to the capacitance of the wires, square of the voltage swing, and the clock frequency. Therefore, in order to save power:

1) Reduce capacitance by increasing the levels of VLSI integration and multichip module technology.
2) Reduce voltage by redesigning chips operating at lower voltage.
3) Reduce clock frequency by trading off computational speed for power savings.

In software, energy conservation has lead to new classes of energy efficient systems software, data access protocols and algorithms. Since the CPU power consumption is proportional to clock rate, adjustment of clock rate to avoid CPU idle time has been suggested with a new metric of instruction per joule (instruction/joule) and new CPU scheduling policies. SUN Solaris is an example of an operating system with power saving features built in.

The following sections give a list of techniques by which power dissipation of an embedded system can be reduced through the control of the embedded operating system.

A. Dynamic Voltage and Frequency Scaling

The average power consumed by a microprocessor while running a certain program is given by: $P = I \times V_{DD}$, where P is the average power, I is the average current and V_{DD} is the supply voltage. Since power is the rate at which energy is consumed, the energy consumed by a program is given by: $E = P \times T$, where T is the execution time of the program. This in turn is given by: $T = N \times \tau$, where N is the number of clock cycles taken by the program and τ is the clock period.

Voltage control is a promising power and energy reduction technique. For the popular static complimentary CMOS devices, the dominant source of power dissipation, caused by the charging and discharging of parasitic capacitance during switching activities at circuit level, is given in Equation 1. This expression suggests that a reduction in V_{DD} is a highly effective way of reducing power and energy consumption. However, voltage reduction comes at a price of lowered noise immunity, additional level converters, and, most importantly, an increased delay of circuit elements which follows the equation:

$$delay = K \frac{V_{DD}}{(V_{DD} - V_T)^2} \tag{2}$$

where K is a constant and V_T is the threshold voltage of transistors. We therefore have a power-delay tradeoff by controlling the supply voltage: the power consumption varies quadratically with the supply voltage, while the delay increases roughly linearly with the voltage.

The power-delay tradeoff introduced by a voltage reduction has a profound impact on low-power design. As shown in [14], components of a datapath circuit can be supplied with multiple voltages such that the power consumption is minimized while meeting all the latency constraints.

Core Voltage	1.0 V	1.5 V	1.8 V
PLL Frequency	800 MHz	800 MHz	533 MHz
CPU Frequency	33 MHz	200 MHz	266 MHz
PLB Frequency	33 MHz	100 MHz	133 MHz
EBC Frequency	33 MHz	33 MHz	33 MHz

TABLE II

OPERATING POINT DESCRIPTION FOR IBM POWERPC 405LP

For instance, under the Dhrystone 1.1 benchmark programs, an ARM7D processor can run at 33MHz and 5V as well as at 20MHz and 3.3V. The energy-performance measures at these two operation modes are 185 MIPS/watt and 579 MIPS/watt, and the MIPS measures are 30.6 and 19.1 respectively [15]. Thus, if we switch from 33MHz and 5V to 20MHz and 3.3V, there will be around (579 - 185) / 579 = 68% reduction in energy consumption at an expense of (30.6 - 19.1) / 19.1 = 60% increase of processing time. The example prompts a question that whether we can adopt an approach to scale the supply voltage and the operation clock speed such that the reduction of energy consumption can be maximized and performance constraints continue to be met.

The approach of voltage-clock scaling has been investigated in workstations, notebook computers, and PDA devices [12], [13]. The basic idea is to make a scaling decision for the next execution interval based on the processor utilization of previous intervals or anticipated future execution demands. In [13], the response time of interactive users and application deadlines have also been taken into account in the voltage-clock scheduling for individual programs.

Reducing V_{DD} to the minimum required voltage for operation at that frequency is the key to this technique because it leverages the maximum savings in the energy consumption. The minimum V_{DD} values for the various frequencies at which a sample processor can run has been shown in Figure 2. The voltage of operation scales almost linearly with the frequency. Table II shows three abbreviated operating points for a IBM PowerPC 405LP reference design [23].

B. Idle/Sleep States

Power management has spawned a new breed of energy efficient CPUs. By ensuring that the chip spends the maximum time in the low power states, overall system power can be reduced. For example, AT&T's Hobbit chip has two modes of operation: in active mode it consumes about 250 milliwatts, and in inactive mode, it consumes only 50 microwatts [6]. Motorola's Dragon I 68349 CPU draws 300 milliwatts at 16 MHz; when idle it draws 1 milliwatt [6].

Table III shows the power management modes of the ARM11 processor [22]. Standby mode of the ARM11 can be used by the embedded operating system as an "idle task".

Mode	Core		Memory		Power
	Clock	Power	Clock	Power	
Run	On	On	On	On	App Dependent
Standby	Off	On	Off	On	Static Leakage
Dormant	Off	Off	Off	On	Mem Leakage
Shutdown	Off	Off	Off	Off	approx. zero

TABLE III
POWER MANAGEMENT MODES OF ARM11

C. Disabling Clock Domains

Matzke has estimated that as technology scales down to 0.1μm feature size, only 16% of the die will be reachable within a single 1.2GHz clock cycle [18]. Assuming a single chip multiprocessor with two processors per die, each processor would need to have minimum of three equal size clock domains.

The multiple clock domain (MCD) processor provides the capability of independently configuring each domain to execute at frequency/voltage settings at or below the maximum values. This allows domains that are not executing instructions critical to performance to operate at a lower frequency, and consequently, energy to be saved. A multiple clock domain processor potentially offers better energy savings than the global voltage/frequency scaling of commercial processors such as Transmeta [19] and XScale [20] because of the finer grain approach [21]. Figure 3 illustrates the usage of multiple clock domains within a processor.

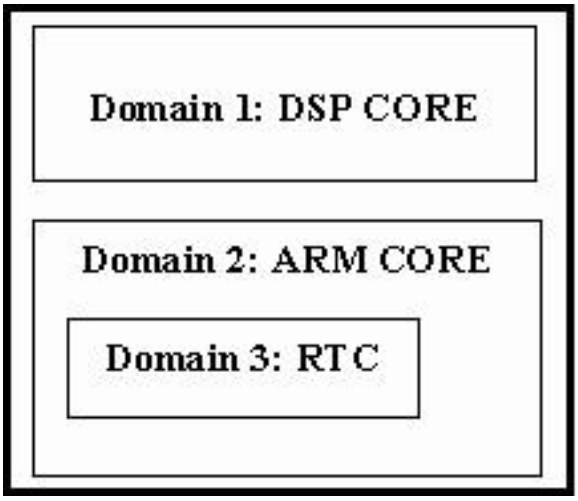

Fig. 3. Multiple Clock Domain Illustration

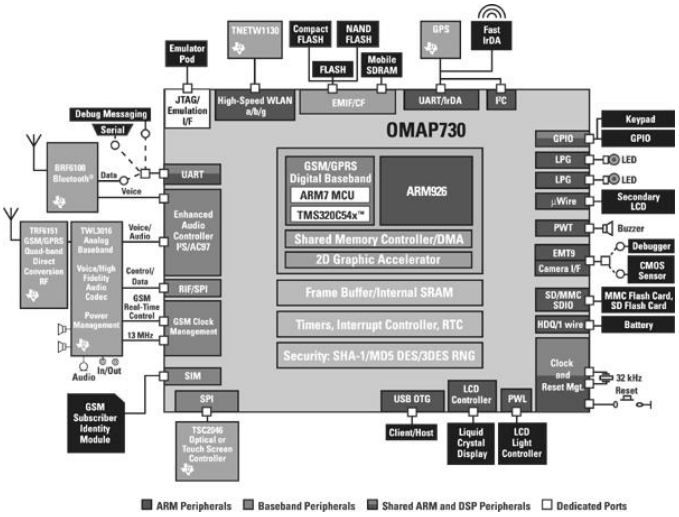

Fig. 4. TI OMAP 730 GSM/GPRS Applications Processor

IV. TI OMAP 730 Processor

The TI OMAP 730 (Figure 4) is a single chip that integrates a ARM926EJ-S core for application processing and a GSM/GPRS digital baseband modem. The GSM/GPRS modem subsystem consists of a ARM7 processor and a TMS320C54x DSP. The ARM9 processor runs the embedded operating system and applications like video and audio players, imaging etc. The GSM subsystem consisting of ARM7 + DSP is used to implement the GSM/GPRS protocol stack.

The OMAP 730 is manufactured on a low-voltage 130 nm CMOS technology and works at a core voltage of 1.1V - 1.5V and an IO voltage of 1.8V - 2.75V. The standby mode power consumption of the OMAP 730 is less than 10 μA. The chip provides split power supplies for application processing, digital baseband and real-time clock thereby enabling precise control over power consumption. Two clocks running at 13 MHz and 32 kHz supply the clocking for the device.

The OMAP 730 integrates 40 peripherals on chip so as to reduce board space and overall system cost. The peripherals include DMA with 4 physical and 17 logical channels, dedicated graphics engine, WLAN interface, enhanced audio controller (EAC), interfaces to SDRAM, flash and SD/MMC, LCD controller, μWire, Bluetooth data/audio interface, USB On-the-Go, 32-bit timers, high speed UARTs and parallel camera port.

V. Summary and Future Work

In this paper, we have described the different techniques available for facilitating low power operation of embedded processor using a high level embedded operating system. The research work aims at applying all the described techniques on an embedded processor platform and estimating the power savings obtained through such optimization techniques. The main focus of work is to reduce the power consumption of the embedded system to the minimum possible while delivering acceptable levels of performance.

The following methods will be adopted for reducing power consumption of the embedded system:

1) Develop heuristics to perform voltage and frequency scaling depending on performance requirements and constraints
2) Maximize the time the system spends in the idle/sleep state
3) Turn off as many unused clock domains and peripherals as possible without affecting functionality and usability of the device

References

[1] Y. Li, M. Potkonjak, and W. Wolf, "Real-Time Operating Systems for Embedded Computing", *Proc. Int'l Conf. Computer Design*, Oct 1997.

[2] Strategy Analytics, *http://www.mobileinfo.com/Reports/StrategyA_cellphone.ht*

[3] C. Ellis, "The Case for Higher-Level Power Management", *Proc. Workshop Hot Topics in Operating Systems*, 1999.

[4] Naresh R. Shanbhag, "Power Dissipation and Throughput in CMOS ICs", *http://courses.ece.uiuc.edu/ece497esd/lec/shanbhag2.pdf*

[5] Berkeley.edu, "Power Dissipation in CMOS ICs", *http://bwrc.eecs.berkeley.edu/Publications/theses/low.power.CMOS.library.MS/*

[6] A. Vahdat, A. Lebeck, and C. Ellis, "Every Joule is Precious: The Case for Revisiting Operating System Design for Energy Efficiency", *Proc. SIGOPS European Workshop*, Sept. 2000.

[7] J. G. Ganssle, "An OS in a Can", *Embedded Systems Programming*, Jan. 1994.

[8] S. R. Ball, *Embedded Microprocessor Systems: Real World Design*. Boston:Newnes, Butterworth-Heinemann, 1996.

[9] J. G. Ganssle, "The Challenges of Real-Time Programming", *Embedded Systems Programming*, vol. 11, no. 7, pp. 20-26, July 1997.

[10] L. Benini and G. D. Micheli, "System-Level Power Optimization: Techniques and Tools", *Proc. Int'l Symp. Low Power Electronics and Design*, pp. 288-293, Aug. 1998.

[11] J. Scott, L. Lee, A. Chin, J. Arends, and B. Moyer, "Designing the M.CORE M3 CPU Architecture", *Proc. Int'l Conf. Computer Design*, Oct. 1999.

[12] T. Pering, T. Burd, and R. Brodersen, "The simulation and evaluation of dynamic voltage scaling algorithms", *International Symposium of Low Power Electronics and Design*, pp. 76-81, Aug. 1998.

[13] M. Weiser, B. Wilch, A. Demers, and S. Shenker, ”Scheduling for reduced CPU energy”, *Proc. of 1st USENIX Symp. on Operating Systems Design and Implementation*, pp. 13-23, Nov. 1994.

[14] J. M. Chang and M. Pedram, ”Energy minimization using multiple supply voltages”, *IEEE Trans. VLSI Systems*, Vol. 5, No. 4, pp. 436-443, Dec. 1997.

[15] ”Introduction to Thumb”, *ARM Documentation*, Advanced RISC Machines Ltd.

[16] H. Veendrick, ”Short-Circuit Dissipation of Static CMOS Circuitry and Its Impact on the Design of Buffer Circuits”, *IEEE Journal of Solid State Circuits*, Vol. 19, No. 4, August 1984.

[17] Vietanh Nguyen, ”Mobile Computing & Disconnected Operation: A Survey of Recent Advances”, *http://www.cse.ohio-state.edu/ jain/cis788-95/ftp/mobile_comp*

[18] D. Matzke, ”Will Physical Scalability Sabotage Performance Gains?”, *IEEE Computer*, 30(9):37-39, Sept. 1997.

[19] M. Fleischmann, ”Longrun power management”, Technical Report, Transmeta Corp. Jan. 2001.

[20] L. T. Clarke, ”Circuit Design of XScale Microprocessors”, *2001 Symp. on VLSI Circuits, Short Course on Physical Design for Low-Power and High-Performance Microprocessor Circuits*. IEEE Solid-State Circuits Society, June 2001.

[21] Greg Semeraro, David H. Albonesi, Steven G. Dropsho, Grigorious Magklis, Sandhya Dwarkadas and Michael L. Scott, ”Dynamic frequency and voltage control for a multiple clock domain micro-architecture”, *Proceedings of the 35th annual ACM/IEEE international symposium on Microarchitecture*, p356-367, 2002.

[22] ”ARM11 Product Introduction”, *ARM Documentation*, Advanced RISC Machines Ltd.

[23] Bishop Brock and Karthick Rajamani, ”Dynamic Power Management for Embedded Systems”, *Proc. of the IEEE Intl. SOC Conference*, September 17-20, Portland, Oregon, 2003.

[24] Hoseok Chang, Wonchul Lee and Wonyong Sung, ”Optimization of Power Consumption for an ARM7-based Multimedia Handheld Device”, *IEEE International Symposium on Circuits and Systems*, pp.0-0, Jan. 2003.