IJCSES International Journal of Computer Sciences and Engineering Systems, Vol.1, No.2, April 2007
CSES International ©2007 ISSN 0973-4406

Towards Dynamic Voltage Scaling in Real-Time Systems - A Survey

Nasro MIN-ALLAH[1,3], Yong-Ji WANG[2], Jian-Sheng XING[1], Wasif NISAR[1], and Asad-Raza KAZMI[2]

[1]Laboratory for Internet Software Technologies, Institute of Software,
Graduate University, Chinese Academy of Sciences, Beijing 100080, China
E-mail: nasar, wasif, jiansheng@itechs.iscas.ac.cn

[2]State Key Laboratory of Computer Science, Chinese Academy of Sciences, Beijing 100080, China
E-mail: ywang@itechs.iscas.ac.cn, kazmi@ios.ac.cn

[3]COMSATS Institute of Information Technology,
Department of Computer Science, Islamabad 44000, Pakistan
E-mail: nasar@comsats.edu.pk

Abstract

Current trends in hand-held devices pose new challenges to embedded system industry. Among them, the most vitals are that these devices must be lightweight, smart in size and long lasting. The displays, keypads and batteries are usually the most prominent hardware of these devices. Current state of the art technologies such as membrane techniques results in reduced weight and size for keyboards and displays. Unfortunately, advances in battery technology is not in pace with current processing demands and contributes maximum towards device size and weight. Considering the recent advancements in processing technologies, proliferation of ubiquitous computing and wearable devices, more intelligent power consumption of VLSI and VVLSI circuits has become a vital issue.

This paper surveys progress made in the field of battery-powered real time embedded systems based on dynamic voltage scaling (DVS) enabled processors. This paper also enables designers of such systems to develop an understanding of three influential factors namely: i) the capabilities and limitations of the batteries that power these systems, ii) latest power efficient techniques that result in maximizing battery life and, iii) the reasons for incorporating appropriate techniques into their intended power-efficient products.

Key words: *Low power design, Dynamic Voltage Scaling, Energy Efficient Scheduling, Battery Life, Power Reduction techniques.*

1. Introduction

It is estimated that power consumption of future microprocessors will reach 2000 Watts in 2010 [1], i.e. about 20 times more of today, while currently the energy consumption in a server farm is measured in the order of megawatts. With the high increase in number of transistors fabricated on unit area of an integrated circuit, the factor of power dissipation has its own importance. Intel Itanium 2 Processor has nearly 1 billion transistors and if this rate continues, Intel processors would produce more heat per square centimeter than the surface of the Sun [2]. On the other hand, researchers recently demonstrated the first silicon-based chip capable of operating at frequencies above 500 GHz, under extreme cold temperature of 451 degrees below zero Fahrenheit [57]. Eventually with increasing operating frequency the power consumption of the device also upsurges, and consequently with higher power dissipation the real-time systems experience two adverse effects vis-à-vis raised temperature and performance degradation. This factor also causes various negative impacts such as lowering battery life, increasing cooling cost, diminishing reliability, environmental and ambient considerations etc. Owing to potential issues associated with batteries, Dell Corporation recalled for 4 million batteries shipped in last two years 2004-2006 [3].

On the other front, an explosive growth in sales volume of the embedded systems is observed, up to 0.4 million cell phones are sold in a single day in 2006[55], while 60 million personal digital systems (PDA) are estimated to be sold in the year 2008 [56]. Similarly, Sales of portable media players grew from 134.5 million units in 2005 to about 187.7 million units in 2006, according to [58].

Manuscript received January 1, 2007
Manuscript revised March 1, 2007

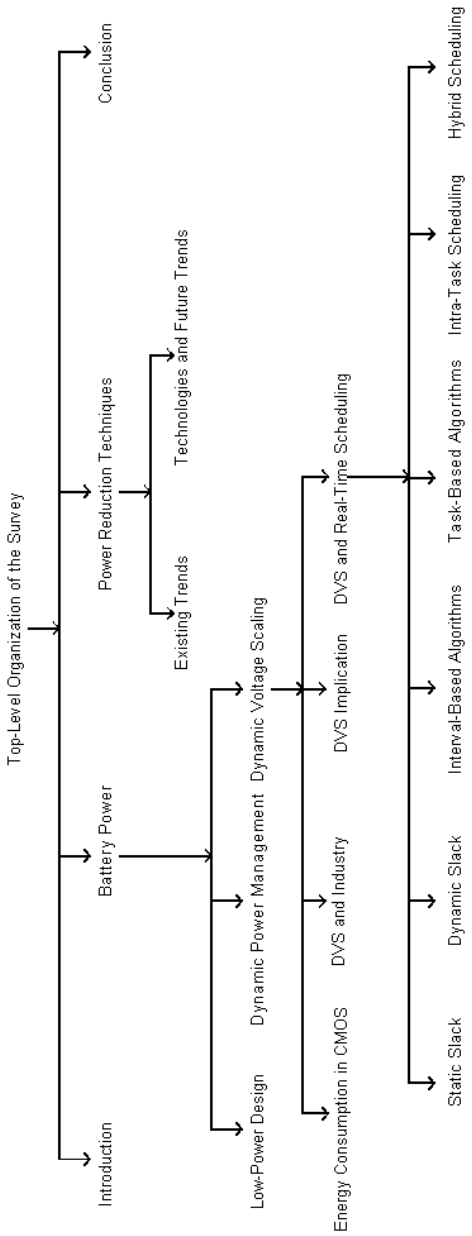

Fig. 1 Outline of the survey

The miniaturization of technology and availability of variety of facilities in one unit are other landmarks, with the rapid growth in information exploration and ease in availability while data is measured in exabytes(10^18). The cellular phones have stopped being just telephones and provide facilities like, 6+ mega pixels camera, a motorized zoom lens, DVD quality video and a high-fidelity music player such as 3G-Enabled phones. 4G-wireless devices are expected to provide more and more facilities at the expense of more power consumption. All these facilities demand high-speed processors, large memories, high-resolution display and more battery life to remain operational for longer time. Due to the efforts of researchers and engineers, the law of Gordon Moore [5] is maintained, which predicts that the number of transistors on an integrated circuit would double every 18 months. However, in practice, along this exponential rate of microprocessor improvement which consumes higher energy, increase in battery capacity has only tripled since 1990[6]. Bridging this gap is a challenge that system designers must face for the foreseeable future.

An option to fill the above gap could be multiple or larger batteries, however, the weight to size ratio is one of the dominating factors in the current development of electronic devices such as cell phones, PDAs, digital players, laptops and wearable systems which are all battery powered. As being a mobile device the battery life has direct impact over the geo-positioning, the rank in topology of a network, and system's utility. The aesthetic beauty through compactness is one of the driving forces in the handheld devices industry. In modern handheld devices, three hardware parts vis-à-vis keypad, display screen and battery have significant contribution in size and weight, while manufacturers have key concern to have competitive edge in wireless industry. New solutions have already been developed for displays and keyboards for example the usage of membrane techniques. It is only battery that impediment in induction new shapes and styles. The increased up time is one of the value added features in the latest devices, while prolonging battery life, through available techniques, means increased size as well as weight which consequently influences system's weight and size.

2. Battery Power

2.1 Existing Trends

Batteries are used in almost all mobile devices, so a large number of batteries are floated in the market today; only in the USA nearly 3 billion dry-cell batteries are sold every year [8]. Both re-chargeable and dry-batteries are used in the modern handheld devices, the former kind always make economic sense for its usage in devices like walkman, digital camera, wireless mouse, cell phones and similar devices, while the later is used to low-draw

devices like smoke detectors, remote controllers, and devices that have long idle time such as emergency flashlights. All batteries have aging factor that is change in amount of charge held when battery grew older and older and thus, result in performance degradation. Also the variation in temperature can causes performance reduction, it is reported that laptop battery ages quickly as compared to other applications when internal temperature rises up to 45 degree Celsius, and life expectancy of a battery reduces to half at high temperature as compared to moderate temperature i.e. 20 degree Celsius, while it reduces to one fourth at 60 degree Celsius [9]. Albeit, improvements in energy densities of rechargeable batteries have taken place, yet they have not satisfied the ever-increasing energy demands of handheld devices. In evaluation of a battery status many considerations like cost, performance, weight, safety and friendliness must be taken into account.

2.2 Existing Technologies and Future Trends

Nickel-Cadmium (Ni-Cd): Ni-Cd is one of the primary forms of rechargeable battery with low cast and high discharge rate, its usage to low cast product owing to low energy density and toxicity.

Nickel-Metal Hydride (NiMH): This kind is generally used for powering laptop computers, and available in standard battery sizes. Although, early versions of NiMH batteries had problem of rapid discharge, it this has been eradicated in the current models. They have approximately twice the energy density as compared to Ni-Cd but are expensive and have shorter life cycle.

Lithium-Ion: On one hand, the stable performance and longer lifetime have made them the most popular battery choice for notebooks, PDAs, and cellular phones, on the other they are expensive and not available in all standard sizes. It can be charged at random, that is full depletion is not required before recharge. The Lithium-Ion enjoys capacity growth of 10% since commercially launched by Sony Corporation in 1991 [6]. The disadvantages in this type are permanent capacity loss and non-uniformity in performance at varying temperatures.

Rechargeable Alkaline: Are different from normal disposable alkaline batteries, and available in 4-packs and 8-packs in standard consumer sizes like AA, AAA, C, D and 9-Volt. Although, in this type initial energy density is higher than Ni-Cd, and it has been found to decrease rapidly with cycle life [7].

Lithium Polymer: This emerging technology enables to build ultra-thin batteries, and suits the needs of lightweight next-generation portable computing and communication devices such as IPAQ with improved energy density and safety. However, these batteries are expensive, and face challenges in internal thermal management.

Fuel Cell: It is used to convert hydrogen and oxygen into water, heat and electricity. In recent years research has been done actively on fuel cells such as Direct Methanol Fuel Cells (DMFC) and Direct Borohydride Fuel Cells (DBFC). In June 2004, Toshiba announced a prototype of the world's smallest DMFC with energy output of 100 milliwatts (mW) that can be integrated into devices as small as digital audio players and wireless headsets for mobile phones. Today, Toshiba is able to use the larger size active fuel cells for laptops on a limited basis now and, hopes to commercialize DMFC for smaller devices in 2007[10].

Batteries are rated by voltage and milliamp hours (mAH) i.e a battery can deliver X milliamps (mA) for one hour (H).Manufacturers usually estimate device battery life in hours under the best-case scenario. For example, the best case scenario for MP3 player is to play only MP3s encoded at 128Kbps, user is wearing bundled ear-buds, volume level is at about 50 to 75 percent, the backlight of your screen turns off automatically within 5 to 10 seconds, equalizer setting is flat or normal, and if applicable, no photos are viewed. The pros and cons of some famous kinds of rechargeable batteries are briefly discussed here.

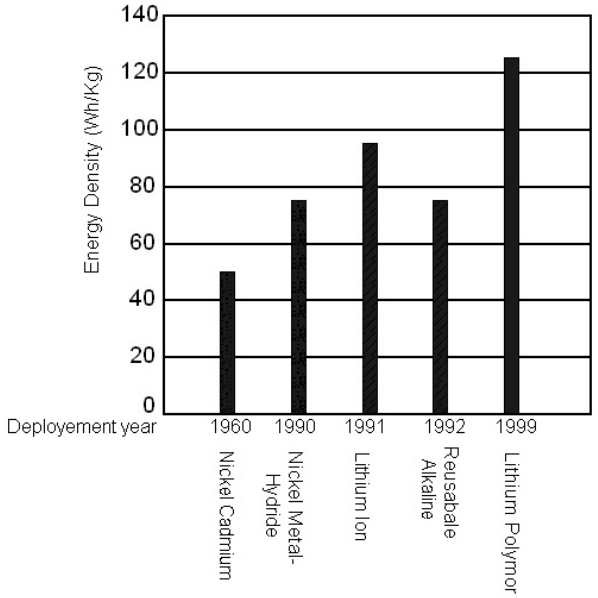

Fig. 2 Energy density of available technologies

3. Power Reduction Techniques

The most effective techniques to manage power consumption in digital systems are low power hardware design, dynamic power management and dynamic voltage scaling. We summarize each in the following:

3.1 Low Power Design

The rapid growth in fabrication of large number of transistors on unit area of integrated circuits induced the techniques of lowering voltage levels. In the decade of 1990's Intel Pentium processors had 3 million transistors, while in the current phase Intel Itanium processor has nearly 1 billion transistors [2]. Thus, employing lower voltage levels results in high-energy reduction that is from 5.0V to 3.3V that reduced energy consumption by 56% [11]. In 1995 Intel, further lowered the operating voltage to 2.9V.

3.2 Dynamic Power Management (DPM)

With this technique the unused system hardware like hard disk or processor are powered down. In DPM the power saving is reported to be 66% [12]. In smaller devices, display eats up a major portion of total energy and efficient solutions -dimming the screen or more efficiently lighting up only the portion of screen that is being used- is in practice today[14]. Many of the latest embedded processors include run-time power modes that can be used to scale down power consumption. The most common of these is idle mode, in which the instruction-executing portion of the processor core shuts down while all peripherals and interrupts remain powered and active. Similarly, intelligent shutdown saves power primarily because it avoids the processor-intensive and time-consuming task of rebooting: by saving the context of lowest-level registers in the system, keep refreshing DRAM and power important internal peripherals, such as the real-time clock. Advanced Configuration and Power Interface (ACPI) is a one of the latest standards for energy management of laptops, desktops and servers that allow several modes of operation, several sleep levels, and the ability to turn off some parts of the device after a preset period of inactivity [13].

3.3 Dynamic Voltage Scaling

3.3.1 Energy Consumption in CMOS Circuitry

The dynamic clock and voltage adjustments represent the cutting edge of power reduction capabilities in CMOS technologies. The relation between frequency and voltage/power provides foundation for dynamic voltage scaling in modern processors [15, 16, 54]. Specially

$$E = Pt \tag{1}$$

with E being the energy consumed and t time taken to run an application with average power P. The average power dissipation p_{av} in a microprocessor has four constituents:

$$P_{av} = P_{cap} + P_{leak} + P_{std-by} + P_{short} \tag{2}$$

where P_{cap}, P_{leak}, P_{std-by}, and P_{short} represent capacitive, leakage, standby and short-circuit power respectively, although P_{leak}, P_{std-by} and P_{short} are important, they are least significant as compared to P_{cap}, and hence, these three term are omitted hereafter for not being within the scope of this paper.(see [54] for details). The capacitive switching power P_{cap} being a dominating term in Equation (2) can be expressed as:

$$P_{cap} = \alpha C V_{dd}^2 f \tag{3}$$

with α_a transition activity dependent parameter, C the switched capacitance, V_{dd} the supply voltage, and f being the clock frequency. Equation (3) indicates the quadratic dependence of V_{dd} and f, furthermore it reflects that lowering the supply voltage is the most effective factor in order to lower the dynamic power consumption. However, lowering V_{dd} increase the circuit delay (t_{dealy}) as:

$$t_{delay} = \frac{kV_{dd}}{(V_{dd} - V)^2} \tag{4}$$

k being a constant depends upon gate size and capacitance, and V_t is threshold voltage, which is the minimum voltage to make it functional. Since f and t_{dealy} are inversely related, therefore, Equation (4) indicates that energy reduction in CMOS devices is possible on the expense of performance delay. The processor clock frequency is:

$$f = \frac{(V_{dd} - V_t)^2}{kV_{dd}} \tag{5}$$

Equation (5) reflects that the clock frequency is linearly related to the supply voltage. Therefore, by assuming $P = P_{cap}$, we can rewrite Equation (3) as:

$$P = \alpha C V^2 f \quad (6)$$

Thus, energy per cycle is proportional to V^2 but this requires running at a slower speed. It is evident from Equation (6) that by varying clock speed and voltage, affect power consumption, linearly and quadratically respectively. An ideal processor would be the one that supplies continuous voltages levels. Although, using continuous variable voltages is not feasible [17] owing to some cost involved in supporting several levels, however, latest processors are capable of supporting a fixed number of discrete-level speeds between minimum and maximum levels [18]. It has been shown in [21] that energy-speed curve is convex in nature and, thus, due to Jenhen's inequality $(\overline{E(r)} \geq E(\bar{r}))$ [60], it is more energy efficient to execute tasks at constant speed instead of changing from one task to the other, as long as deadlines are met. Furthermore, from Equation (6) the energy per transition is given as:

$$P_{av} = \alpha C V^2 \quad (7)$$

with the option of availability of discrete-levels of speed (voltage and frequency) of the processor may be adjusted on the fly for many applications to reduce the processor power consumption. This voltage (speed) adjustment on run time is called, dynamic voltage scaling (DVS), which is an effective means for the reduction of processor power consumption.

As real time systems are usually not fully utilized up to maximum extent, and thus, a promising target for DVS. Table.1 [22] shows maximum power consumption by a processor when fully loaded. Interestingly, it can be seen that processor, which is the main component of an embedded system, is an ideal candidate for DVS technique. A similar study also reveals that processor core (StrongARM SA-1100) eats major portion of the system energy [23](Table 2). The reduction in energy consumption is obtained through DVS technique by varying voltage and frequency operating points. Fig 4 shows power consumption of StrongARM processor versus clock frequency [20]. In [19] authors show that DVS-enabled ARM7D processor when operating, at lowest and highest mode gives, 185 MIPS/WATT and 579 MIPS/WATT respectively.

Table: 1 Power consumption of major parts of the systems

Screen	CPU Subsystem	Disk	Power in Watts
ON	Idle	Spinning	13.5
ON	Idle	Standby	13.0
OFF	Idle	Standby	7.1
OFF	Maximum Load	Standby	27.3

Table: 2 Power consumption by applications

Benchmark	P_{core} (%)	P_{main} (%)	P_{LCD} (%)	P_{DRAM} (%)
MPEG-I	28.8	16.5	0.5	28.4
DECtalk	33	24.7	0.8	23
WAV	18.7	24.6	1.1	5.5
Idle	49	17.6	3.4	15.3

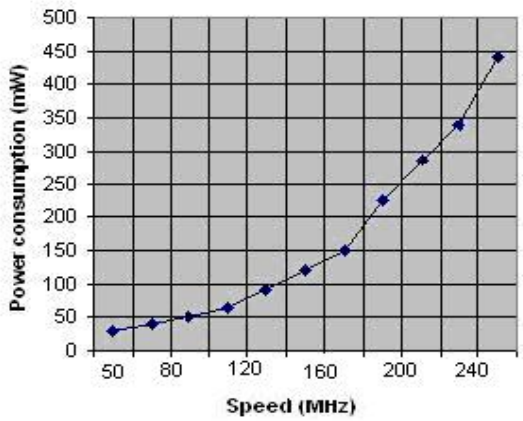

Fig. 3 Power consumption vs. speed for the StrongARM (courtesy [20]).

3.3.2 DVS and Industry

Commercial response is very positive and manufacturers highlight the power efficiency of their products loudly such as Enhanced SpeedStep by Intel and PowerNow by AMD. Similarly, Long Run is a hardware technique developed by Transmeta to dynamically measure system-utilization and vary the CPU speed accordingly [24]. Today Transmeta, Motrolla, AMD and Intel sell processors with dynamic voltage scaling (DVS) capability.

Table: 3 Latest Variable voltage processors

Processor	Speed	Voltage	Manufacturer
PXA250[25]	100 400	0.85 1.3	Intel
StrongARM SA-2[25,26]	150 600	0.75 1.30	Intel
Cursoe(TM5400)[27]	200 700	1.10 1.65	Transmeta
ARM7D[35][25]	20 33	5.0 3.3	Intel
PowerPC860[26]	25 50	2.4 3.3	Motrolla
Mobile K6[28]	192 588	0.9 2.0	AMD
Intel Xscale[25]	150 1000	0.75 1.80	Intel

3.3.3 DVS Implications

A DC-DC converter is used to generate variable supply voltage which takes a finite amount of time and power overhead during voltage transition (from operational level to desired level). In all DVS processors a transition from one power level to another level has a time (25 to 150 micro-secs) and energy (up to 4 μJ) overhead [4].This overhead is because of changing the supply voltage and allowing it to become stable. Specifically, commercial processor have different frequency overheads such as Intel's Strong-Arm takes up to 150 microseconds [35], while XScale takes around 30 micro seconds[29]. A full voltage transition can be performed in less than 300 msec in Transmeta Cursoe 5400 processor [30]. Since a DVS algorithm usually lowers the system speed to reduce energy consumption as frequency is related to processor's power consumption. Such arrangements results in increased execution time of a task and also increases the number of task preemption, which, in turn, increases the number of memory access. Applying DVS techniques to real time systems, system utilization is increased by running tasks at lower speed. However, conflicting issue of performance degradation emerges as a consequence of DVS.

3.3.4 DVS and Real time scheduling

This section is mainly focused on deploying DVS to real time scheduling. Two techniques can impact real time system scheduling namely, DVS and DPM. DPM aims to shut off system hardware that is not currently in use, whereas, DVS run tasks at different speeds in order to fill up the idle periods in the schedule. Its always recommended to exploit DVS first when both DVS and DPM are available, while DPM can be applied at later stages [31] .In the following sections, we explore the integration of DVS techniques to real time system from scheduling point of view.

a. Exploiting Static slack

It has been observed in enormous systems that, during runtime, even if all task instances run for their WCET (worst case execution time), the processor utilization is often far lower than 100% and results in idle intervals. This slack that inherently exists in the system due to low processor utilization is henceforth referred to as static slack. It can be exploited to reduce energy consumption by statically slowing down the processor and operating at a lower voltage. While processor slowdown improves utilization, excessive slowdown may lead to deadline violations. Hence, the extent of slowdown is limited by the schedulability of the task set at the reduced speed, under the scheduling policy used. The following example illustrates the use of static slowdown to reduce energy consumption. To elaborate the effectiveness of DVS in task scheduling, system such as given in Table.4 [59], can be statically scheduled, which leads to a decrease in power consumption from 420mW to 184mW, using the power vs. frequency curve for the StrongARM processor given in Fig 2.

Table: 4 Tasks with Timing Parameters

Tasks	Time Period	WCET	Deadline
Audio decoding	60	10	60
Protocol processing	70	15	70
Video decoding	120	40	120

The following two examples [59] explain how DVS could be exploited in real time systems.

Example 1: Consider a simple multimedia terminal, the system receives real-time audio and video streams over a wireless link, and plays them out. Thus, the three main tasks that run on a processor embedded in this system are protocol processing, audio decoding, and video decoding.

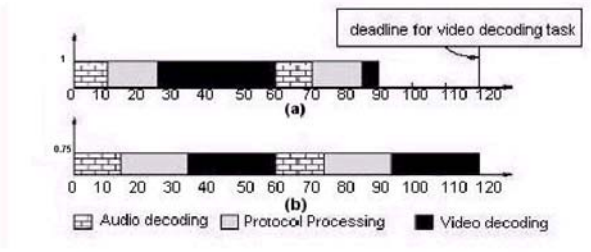

Fig. 4 Task execution schedule: (a) Original, (b) statically optimized (source [59])

The timing parameters for these tasks are listed in Table 4. For simplicity, the initial phase offsets for the tasks are set to zero. The resulting schedule for the time interval [0, 120] when this task set is scheduled on a single processor using the RM priority assignment scheme, is shown in Fig. 4(a). It can be seen from the Fig that the system is idle during time interval [90, 120]. This slack can be utilized to lower the operating frequency and supply voltage, thereby reducing the energy consumption. Fig. 4(b) shows the schedule for the same task set, when the processor is slowed down by a factor of 4/3. As seen from the Fig 3, processor slowdown leads to a reduction in slack. So this speed reduction means lowering voltage level, which leads to a decrease in power consumption from 420mW to 184mW, using the power vs. frequency curve for the StrongARM processor, shown in Fig. 3. The energy consumption over the interval [0, 120], therefore, decreases by 41%, compared to a shutdown based policy.

b. Exploiting slack dynamically

To satisfy timing constraint, real time tasks are scheduled under WCET scenarios. However when executed, WCET are not in accordance with actual execution timings; actually many jobs complete early. In practice, there is significant variation between the upper and lower bound for both worst case and best case execution times (experimental details area available in [32]), variations up to 87% between actual and WCET is reported in [33]. Table 5 shows variations between BCET and WCET for multimedia applications.

Table: 5 Variations in execution times for multimedia benchmarks [59]

Program	Description	BCET	WCET
DES	Data encryption	672,912	672,298
DJPEG	JPEG decompression	12,703,432	122,838,368
FDCT	JPEG forward DCT	5,587	16,693

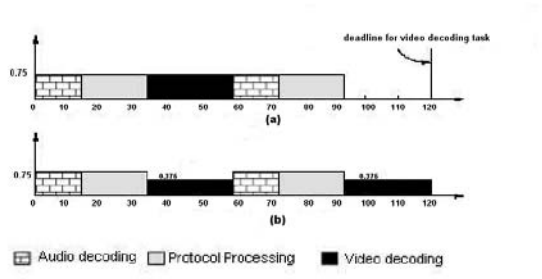

Fig. 5 Task execution schedule for example 2: (a) Statically optimized (b) After dynamic slowdown (source [59]).

Example 2: Consider the statically optimized schedule for the task set of Example 1. Fig. 5(a) shows the resulting schedule when the video decoding task instance requires only 20 time units to complete execution at maximum processor speed. As a result, the video decoding task now completes execution at time t=60 units, and does not get preempted. As seen from the Fig 5(a), this execution time variation creates some dynamic slack. To utilize this slack, the processor frequency and supply voltage are dynamically reduced further during the execution of the video decoding task. If the frequency were to be dropped by a factor of 2 over the already statically optimized value, the slack gets filled up again as shown in Fig. 5(b). Accompanied by appropriate voltage scaling, the energy consumption for the interval [0,120] now reduces by a further 14% compared to the statically optimized schedule of Fig. 5(a).

More promising DVS techniques are needed to respond well to dynamic systems, where workload fluctuates such as given in Table-5. For the task set given in Table.5, when exploited dynamic slack, 14% more energy reduction is reported in [34] as compared statically optimized schedule. In [35], authors exploited the above feature (WCET is pessimistic case) of real time task and use DPM when DVS is not applicable.

c. Interval-based algorithms

In their pioneer work, Weiser et al. [36] exploited DVS to maximize millions-of-instructions per Joule (MIPJ), where time is divided into 50-100 msec intervals, and one year later, the authors in [37] proposed DVS algorithms, mainly for non-real time applications aiming at reducing energy consumption of processor by dividing time into fixed-length intervals. In their work, average throughput is the metric of performance while ignoring real-time constraints. Based on the CPU utilization of previous intervals, their algorithms predict the CPU utilization of the next interval and accordingly adjusting speed[36] i.e. If the processor is more busy than idle in previous segment, speed up, otherwise slow down , while sophisticated heuristics are used to predict how busy the processor would be in the near future. As they assumed regular workload, flaws were pointed out latter on in such interval-based strategies, e.g. the arbitrary chosen interval boundaries do not correspond to real deadlines [38, 39]. Predicting the future workload from the current situations is difficult, and can seriously reduce the gains of voltage-scaling.

d. Task-based algorithms

Unlike interval based strategies, more promising task based (use task information like deadline etc) strategies were proposed. Task-based DVS techniques are applied at the task Level i.e. task dispatch or release time. These algorithms, also known as inter-task, assign different speed to different tasks and remains fixed for entire execution time. To minimize power consumption, Yao et. al. [39] presented a polynomial time static off-line preemptive scheduling algorithm for real time applications. Their solution is optimal when task CPU requirements and deadlines are known, assuming a continuous voltage range. The aim of a task-based scheduler is to execute tasks with speeds just high enough to meet these deadlines with reasonable probability. Although promising, such approaches are unrealistic as systems do not generally have definite knowledge of task CPU requirements [40].

In addition to task execution time and deadlines, the work of T. Ishihara and H. Yasuura in [21] requires the switches capacitance per cycle for each task. To study effects of voltage levels on energy consumption, authors in [21] concludes that to reduce energy consumption of a processor 1) with continuously variable voltages , it is shown that a unique voltage should be used for each task and, 2)at most two voltages levels are sufficient to execute any task for processors which offer discrete voltages such as Transmeta's Cursoe processor. A significant work has been done on voltage scaling algorithms for hard real time systems[41,42,43,and 36]. P. Pillai and K. G. Shin's work[41] provides real time guarantees for real time tasks .Their scheduler assumes complete knowledge of the deadlines and worst-case CPU requirements of all tasks in the system. Their static algorithm guarantees in attaining deadlines and, to further reduce energy consumption they proposed cyclic conserving algorithm to exploit WCET scenarios. Their cycle conserving RT-DVS for EDF recomputes utilization at task level (completion, release) and provides required speed for remaining tasks, as jobs might complete earlier than their WCET. As actual execution time is not known in advance, their algorithm uses the earliness of job completed and assuming WCET of remaining jobs to compute utilization and suggest frequency so that remaining task met deadline in worst case. At next invocation of a task, its WCET is used to compute utilization so that actual task sets remains schedulable. Their algorithm switch frequency only at task level i.e. Task completion and release time.

e. Intra-Task Scheduling

Intra-task voltage scheduling, first introduced by authors in [44], has the ability to adjust task speed within the boundaries of task by varying the supply voltage so that not to finish the task earlier than the statically scheduled point. The authors in [44] shows that the intra-task DVS technique is more energy efficient than intertask. More energy reduction is obtained by partitioning each task into several time slots according to the worst-case execution profile. Such algorithms usually require compiler support to insert power management points into the compilation code in order to call OS system services for speed reduction[45]. Shin et al. discussed a static timing analysis in [46] to select the best program locations for partitioning of the given task. They proposed an intertask DVS with a worst case execution path based scheme for aperiodic tasks, Although efficient, their work doest not consider the likelihood of different possible executing paths.

Recently seon at al.'s work in [47] achieves optimal energy savings employing path locality by considering the branch probabilities of CFG (control flow graph). However, obtaining all the branch probabilities for a large program (with large degree of path locality) is impractical [11]. Unlike the previous work which is based on the most probable path, authors in [11] proposed a DVS scheduling algorithm based on the knowledge of frequently executed paths in CFG and eventually showed better results.

Realizing the poor performance of pure DVS techniques for system with fluctuating workloads, recently Zhu yifan and Muller F in [43] extended feedback control techniques for soft real time systems [44,46, 47] to hard real time systems. They proposed a feedback EDF scheduling for hard real time systems exploiting dynamic workload characteristics, where actual execution and WCET exhibits a significant variation. Their greedy scheme splits highest priority job into two subtasks i) the first subtask exploits available slack and executes at lowest speed to reduce energy consumption while ii) enough time is reserved for second subtask to met deadline guarantees running at max speed following a last chance approach[24]. As tasks do not fully utilize WCET , tasks are ideally expected to complete during first subtask. This combination of feedback scheme and task splitting, guarantees in deadline requirement of real time tasks while reducing energy consumption. Only highest priority job is scaled to use available slack while assuming all other task execute at full speed. They proposed and implemented real-time DVS (RT-DVS)algorithms that therefore execute tasks as slowly as possible and only use higher speeds to guarantee on-time completion, when necessary. With Intratask DVS, performance comes at the price of maximum task portioning, which on the other side

increase number of overhead involved changing voltage levels.(more portions , more DVS overhead), so a tradeoff is required.

f. Hybrid Tasks

Although, embedded systems usually consist of periodic jobs where release times of tasks are known a priori, many current real time systems are capable of scheduling aperiodc and sporadic tasks along with hard tasks. Reliability is of primary concern in such systems, applying DVS to mixed tasks demands a compromise between two conflicting terms, performance and energy reduction. Tradeoffs are done while keeping targeted domain in mind. In hybrid systems it is of immense importance to guarantee that all hard deadlines are met and, minimal delay in response time is experienced for aperiodic jobs [48, 49, 50, 52].

Unlike [41], which assign uniform speed to periodic tasks, D.Shin and J. Kim [62] algorithms find different speeds for aperiodic and periodic tasks. Under EDF and RM scheduling policy, they reported up to 32% & 12% reduced energy consumption respectively, over the tasks executed at full speed and power down at idle intervals. This speed adjustment results in reduced energy consumption and average response time up to 11% and 26% respectively over P. Pillai and K. G. Shin's static algorithm [41]. Their offline-static-voltage-scaling provides the opportunity to apply a performance bound on average response times. To reduce average response time they used Total Bandwidth Server (TBS)[55] for handling aperiodic tasks. Yoonmee Doh et al[53] proposed a new technique , where energy budget is allocated for mixed hard and soft real-time tasks. Under the constrained energy budget which is portioned between hard and soft aperiodic tasks so that the response times of aperiodic tasks are minimized and hard deadlines are met. For soft aperiodic task TBS is employed. To successfully execute periodic and aperiodic jobs together , the available energy constraint must be satisfied in priory i.e. $E_c \geq E_p + E_{ap}$, where E_c is given energy budget, E_p and E_{ap} are the energy budget allocated to periodic task and aperiodic tasks respectively. In their work [53], voltage switching cost is neglected. Two voltage levels, low-voltage mode and high-voltage mode with corresponding operating speed are initially explored for mixed tasks and, more adjustments are done latter, depending on the energy budget of the system. They assume a system, having finite mission lifetime or where the battery capacity can be replenished.

4. Conclusions

The mismatch between the slow improvement in batteries capacity demands energy efficient hardware, operating systems and applications while making no compromise on performance/QoS issues. Due to fundamental limitation with battery chemistry, breakthrough is not expected in near future. Focus has been shifted to power aware systems that consume less power so that battery life is prolonged. Today, power aware computing is now possible at different levels such as complier, operating system and protocols. Among power efficient techniques, DVS is the youngest and most prominent one.

This paper surveys the existing battery technologies, discusses their pros and cons briefly. We provide a review of developments made in the field of energy efficient systems. Finally, we explore DVS applications into real time systems from scheduling point of view.

As current pure DVS techniques are mainly targeting processor, more enhanced algorithms are needed to integrate memory, IO and other peripherals to synchronize with processor frequency change to further reduce power consumption.

Acknowledgments

This work is jointly supported by COMSATS Institute of Information Technology under faculty development program, the research collaboration between the Chinese Academy of Sciences and the Royal Society of the United Kingdom (Grant Number: 20030389, 20032006) and the State Education Ministry's Scientific Research Foundation for the Returned Overseas Chinese Scholars (Grant Number: 406).

References

[1] S. Borkar, "Design Challenges of Technology Scaling", **IEEE Micro**, Vol.19, No.4, 1999, pp. 23-29.

[2] http://www.intel.com/technology/magazine/computing/multi-core-0705.pdf, Geoff Koch, "Discovering Multi-Core, Extending the Benefits of Moore's Law", **Technology Intel Magazine**, 2006,

[3] http://dell.com, 2006.

[4] N. AbouGhazaleh, B. Childers, D. Mosse, R. Melhem, and M. Craven, "Energy Management for Real-Time Embedded Applications with Compiler Support", in **ACM SIGPLAN Conference on Languages, Compilers, and Tools for Embedded Systems**, 2003, pp. 284–293.

[5] Moore, Gordon E, "Cramming More Components onto Integrated Circuits", **Electronics,** 1965, Vol.38, pp.114-117.

[6] Linden, D. and Reddy. T, **Secondary Batteries-Introduction. In Handbook of Batteries,** New York: McGraw-Hill, 2002.
[7] D. P. Kanishka Lahiri, Sujit Dey and A. Raghunathan, "Battery-Driven System Design, A New Frontier in Low Power Design", in **ASPDAC/ VLSI Design**, 2002, pp. 261-268.
[8] http://yosemite.epa.gov, US environmental protection agency.
[9] http://www.batteryuniversity.com/parttwo-37.htm, 2006.
[10] http://www.toshiba.com/taec/components/Generic/TAEC_DMFC_2_04_06.pdf, 2006.
[11] J. Pouwelse, K. Langendoen, and H. J. Sips. "Application-Directed Voltage Scaling", in **IEEE Transactions on VLSI Systems**, 2003, Vol. 11, No. 5, pp.812-826.
[12] J. Lorch and A. Smith, "Scheduling Techniques for Reducing Processor Energy Use in MacOS", **Wireless Networks**, 1997.
[13] http://delevoper.intel.com/technology/IAPC/tech, 2003.
[14] http://www.hpl.hp.com/news/2005/jan-ar/smartpower.html, 2006.
[15] A.P. Chandrakasan, S. Sheng and R.W. Brodersen. "Low Power CMOS Digital Design", **IEEE Journal of Solid State Circuits**, 1992, pp. 472-484.
[16] T.D. Burd, T.A. Pering, A.J. Stratakos, and R.W. Brodersen, "A Dynamic Voltage Scaled Microprocessor System", **IEEE Journal of Solid State Circuits**, Vol. 35, No. 11, 2000, pp. 1571-1580.
[17] A. P. Chandrakasan and R. W. Brodersen, **Low Power Design**, Dordrecht: Kluwer Academic Publishers, 1995.
[18] http://www.transmeta.com, 2000
[19] Introduction to Thumb, ARM Documentation, Advanced RISC Machines Ltd.
[20] Vijay Raghunthan, Cristiano L.Pereia, "Energy Aware Wireless Systems with Adaptive Power-Fidelity Tradeoffs", **IEEE Transaction on Very Large Scale Integration (VLSI) Systems,** 2005, VOL. 13, NO. 2.
[21] T. Ishihara and H. Yasuura, "Voltage Scheduling Problem for Dynamically Variable Voltage Processors", in **International Symposium On Low Power Electronics and Design**, 1998, pp. 197-202.
[22] B. Shneiderman, **Designing the User Interface: Strategies for Effective Human-Computer Interaction**, MA: Addison-Wesley Reading, 1998.
[23] Marc A. Viredaz and Deborah A. Wallach, "Power Evaluation of a Handheld Computer"**, IEEE Micro**, Vol. 23, No.1, 2003, pp.66-74.
[24] J. Pouwelse, K. Langendoen, and H. Sips, "Dynamic Voltage Scaling on a Low-Power Microprocessor"**,** in **Seventh Int'l Conference Mobile Computing and Networking**, 2001.
[25] http://www.intel.com/design, 2006.
[26] MPC860 PowerPC Hardware Specification, 1998.
[27] TN5400Processor Specification, 2000
[28] AMD PowerNow Technology, AMD Inc., 2000.
[29] XScale micro-architecture, Intel Inc., http://www.intel.com/design, 2006.
[30] Transmeta Corporation, http://transmeta.com/ 2006.
[31] N. K. Jha, "Low Power System Scheduling and Synthesis", in **IEEE ICCAD**, 2001, pp. 259-263.
[32] J. Wegener and F. Mueller, "A Comparison of Static Analysis and Evolutionary Testing for the Verification of Timing Constraints", **Real-Time Systems,** Vol. 21, No. 3, 2001, pp.241-268.
[33] Yifan Zhu, Frank Mueller, "Feedback EDF Scheduling Exploiting Dynamic Voltage Scaling", in **10th IEEE Real-Time and Embedded Technology and Applications Symposium**, 2004.
[34] R. Min, T. Furrer and A. Chandrakasan, "Dynamic Voltage Scaling Techniques for Distributed Microsensor Networks", in **IEEE Computer Society Annual Workshop on VLSI**, 2000.
[35] Y. Shin and K. Choi, "Power Conscious Fixed Priority Scheduling for Hard Real-Time Systems", in **Design Automation Conference**, 1999, pp. 134-139.
[36] M. Weiser, B. Welch, A. Demers, and S. Shenker, "Scheduling for Reduced CPU Energy", in **1st Symposium on Operating Systems Design and Implementation**, 1994, pp. 13-23.
[37] K. Govil, E. Chan , and H. Wasserman, "Comparing Algorithms for Dynamic Speed-Setting of a Low-Power CPU", in **1st ACM International Conference on Mobile Computing and Networking**, 1995, pp. 13-25.
[38] T. Pering, T. Burd, and R. W. Brodersen, "The Simulation and Evaluation of Dynamic Voltage Scaling Algorithms", in **International Symposium on Low Power Electronics and Design**, 1998, pp. 76–81.
[39] F. Yao, A. J. Demers, and S. Shenker, "A Scheduling Model for Reduced CPU Energy", in **IEEE Symposium on Foundations of Computer Science**, 1995, pp. 374-382.
[40] Lorch, J. and Smith, A. J, "Operating System Modifications for Task-Based Speed and Voltage Scheduling", in **First International Conference on Mobile Systems, Applications, and Services**, 2003, pp. 215–229.
[41] P. Pillai and K. G. Shin, "Real-Time Dynamic Voltage Scaling for Low-Power Embedded Operating Systems", in **Proceedings of 18th ACM Symposium on Operating Systems Principles**, 2001, pp. 89-102.
[42] H. Aydin, R. Melhem, D. Mosse, and P. M. Alvarez, "Dynamic and Aggressive Scheduling Techniques for Power-Aware Real-Time Systems", in **IEEE Real-Time Systems Symposium**, 2001, pp. 95-106.
[43] W. Kim, J. Kim, and S. L. Min, "A Dynamic Voltage Scaling Algorithm for Dynamic-Priority Hard Real-Time Systems Using Slack Time Analysis", in **Design Automation and Test in Europe**, 2002, pp. 788-794.
[44] S. Lee and T. Sakurai, "Run-Time Voltage Hopping for Low- Power Real-Time Systems", in **ACM Design Automation Conference**, 2000, pp.806-809.
[45] Hakan Aydin, Rami G. Melhem, Daniel Mossé, Pedro Mejía-Alvarez, "Power-Aware Scheduling for Periodic Real-Time Tasks", **IEEE Trans on Computers**, Vol.53**,** No.5, 2004, pp. 584-600.
[46] D. Shin, J. Kim, and S. Lee, "Intra-task Voltage Scheduling for Low-energy Hard Real-Time Applications", **IEEE Design and Test of Computers**, Vol. 18, No. 2, 2001, pp.20-30.
[47] J. Seo, T. Kim, K. S. Chung, "Profile-Based Optimal Intra-task Voltage Scheduling for Hard Real-Time Applications", in **ACM Design Automation Conference**, 2004, pp.87-92.

[48] L. Abeni and G. Buttazzo, "Integrating Multimedia Applications in Hard Real-Time Systems", in **IEEE Real-Time Systems Symposium**, 1998, pp. 4-13.

[49] B. Sprunt, L. Sha, and J. P. Lehoczky, "Aperiodic Task Scheduling for Hard Real-Time Systems", **Real-Time Systems**, Vol.1, No.1, 1989, pp. 27-60.

[50] J. K. Strosnider, J. P. Lehoczky, and L. Sha, "The Deferrable Server Algorithm for Enhanced Aperiodic Responsiveness in Hard Real-Time Environments", **IEEE Transactions on Computers**, Vol.44, No.1, 1995, pp.73-91.

[51] D.Shin, J. Kim, "Dynamic Voltage Scaling of Periodic and Aperiodic Tasks in Priority-Driven Systems", in **Asia and South Pacific Design Automation Conference**. 2004, pp. 653-658.

[52] M. Spuri and G. Buttazzo, "Scheduling Aperiodic Tasks in Dynamic Priority Systems", **Real-Time Systems**, Vol.10, No. 2, 1996, pp. 179-210.

[53] Y. Doh, D. Kim, Y.H. Lee, and C. M. Krishna, "Constrained Energy Allocation for Mixed Hard and Soft Real-Time Tasks", in **International Conf. on Real-Time and Embedded Computing Systems and Applications**, 2003, pp. 533-550.

[54] T. Gloker, H. Meyr, **Design of Energy-Efficient Application-Specific Instruction Set Processors**, Dordrecht: Kluwer Academic Publisher, 2004.

[55] http://sanjose.bizjournals.com/sanjose/stories/2001/08/06/daily5.html, 2006

[56] http://www.c-i-a.com/pr1002.htm, Computer Industry Almanac

[57] http://www-03.ibm.com/press/us/en/pressrelease, 2006

[58] http://www.pcworld.com/news/article/0,aid,126198,00.asp

[59] Vijay Raghunthan, Cristiano L.Pereia, "Energy Aware Wireless Systems with Adaptive Power-Fidelity Tradeoffs", **IEEE Transactions on Very Large Scale Integration Systems**, Vol. 13, No. 2, 2005.

[60] J. L. W. V. Jensen, "Sur les fonctions convexes et les inegalites entreles valeurs moyennes", **Acta Math**., Vol. 30, 1906, pp. 175-193.