Simultaneous Communication and Processor Voltage Scaling for Dynamic and Leakage Energy Reduction in Time-Constrained Systems

Alexandru Andrei, Marcus Schmitz, Petru Eles, Zebo Peng
Dept. of Computer and Information Science
Linköping University
S-58183 Linköping, Sweden
{alean, g-marsc, petel, zebpe}@ida.liu.se

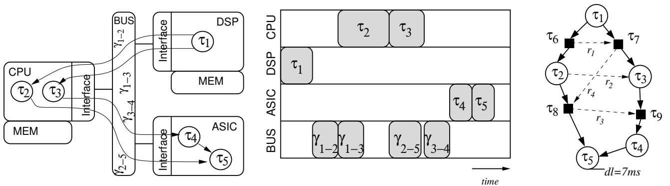
Abstract

Dynamic voltage scaling and adaptive body biasing have been shown to reduce dynamic and leakage power consumption effectively. In this paper, we solve optimally the combined supply voltage and body bias selection problem for multi-processor systems with imposed time constraints, explicitly taking into account the transition overheads implied by changing voltage levels. Both energy and time overheads are considered. The processors are interconnected by buses, considered to be implemented with repeaters or fat wires. The voltage scaling technique achieves energy efficiency by simultaneously scaling the supply and body bias voltages in case of processors and buses with repeaters, while energy efficiency on fat wires is achieved through dynamic voltage swing scaling. We investigate the continuous voltage scaling as well as its discrete counterpart. The continuous voltage scaling problem is formulated and solved optimally using nonlinear programming with polynomial time complexity. For the discrete problem we use mixed integer linear programming for the optimal solution and a time-efficient heuristic. Extensive experiments, conducted on several benchmarks are used to validate the approaches.

1 Introduction

Dynamic voltage scaling (DVS) and adaptive body biasing (ABB) are two system-level approaches that can be used to reduce the energy consumed by processors. Both techniques provide an energy/performance trade-off, which can be exploited during run-time. Conceptually, DVS aims to reduce the *dynamic power* consumption by decreasing the operational frequency and supply voltage [4], while ABB is efficient in limiting the *static leakage power* consumption by reducing the operational frequency and increasing the circuit's threshold voltage via body biasing [7]. Until recently, dynamic power has been the main source of power dissipation. However, in deep-submicron CMOS technology (feature size $< 70nm$), leakage power is predicted to become comparable to the dynamic power [3, 7]. Hence, the combination of DVS and ABB will become essential to manage the total energy dissipation of next generation embedded systems [9]. Approaches for combined DVS and ABB in distributed time-constrained systems have been reported [10].

A negative side-effect of the shrinking feature sizes is the increasing *RC* delay of on-chip wiring [14]. The main reason behind this trend is the ever-increasing line resistance. In order to maintain high performance it becomes necessary to "speed-up" the interconnects. Two implementation styles which can be applied to reduce the propagation delay are: (a) The insertion of *repeaters* and (b) the usage of *fat wires*. In principle, repeaters split long wires into shorter (faster) segments [1, 14] and fat wires reduce the wire resistance [13, 14]. Techniques for the determination of the optimal quantity of repeaters are introduced in [1]. An approach to calculate the optimal voltage swing on fat wires has been proposed in [13]. Similar to processors with supply voltage scaling capability, approaches for link voltage scaling were recently presented in [12, 15]. An approach for communication speed selection was outlined in [8]. Another possibility to reduce communication energy is the usage of bus encoding techniques [2]. In [6], it was demonstrated that shared-bus splitting, which dynamically breaks down long, global buses into smaller, local segments, also helps to improve energy savings. An estimation framework for communication switching activity was introduced in [5].



(a) Target architecture with mapped task graph (b) Muliple component schedule (c) Extended TG

Figure 1. Architecture and application model

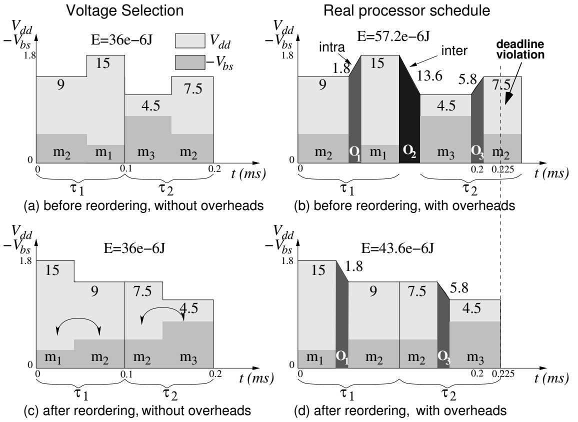
The aim of this work is to introduce a voltage scaling technique, considering scaling of both communication links and processing elements. The approach is based on suitable delay and energy models, in particular for communication links that are implemented via repeaters or fat wires. As opposed to previous system-level approaches, we take also into account the communication leakage power consumption as well as the dynamic adaption of the voltage swing. During the optimization, we consider the transition overheads due to voltage changes, both in terms of energy and delay.

2 Architecture and Application Model

In this paper, we consider embedded systems which are realized as multiprocessor systems-on-a-chip (SoC). Such architectures consist of several different processing elements (PEs), some of which feature DVS and ABB capability. These computational components communicate via an infrastructure of communication links (CLs), like buses and point-to-point connections. Similar to the PEs, the CLs might be equipped with voltage scaling capability. An example architecture is shown in Fig. 1(a).

The functionality of data-flow intensive applications, such as voice processing and multimedia, can be captured by task graphs $G(\mathcal{T},\mathcal{C})$. Nodes $\tau \in \mathcal{T}$ in these directed acyclic graphs represent computational tasks, while edges $\gamma \in \mathcal{C}$ indicate data dependencies between these tasks (i.e. communications). Tasks require a certain number of clock cycles *NC* to be executed, depending on the PE to which they are mapped. Similarly, if two dependent tasks are assigned to different PEs, π_x and π_y with $x \neq y$, then the communication takes place over a CL, involving a certain number of clock cycles for the data transfer. Further, tasks are annotated with deadlines dl that have to be met during run-time.

We assume that the task graph is mapped and scheduled onto the target architecture, i.e., it is known where and in which order tasks and communications take place. Fig. 1(a) shows an example task graph that has been mapped onto an architecture, and Fig. 1(b) depicts a



(a) before reordering, without overheads (b) before reordering, with overheads

(c) after reordering, without overheads (d) after reordering, with overheads

Figure 2. Influence of transition overheads

possible execution order. To tie the execution order into the specification, we perform the following transformation on the original task graph. Firstly, all communications that take place over communication links are captured by communication tasks, as indicated by squares in Fig. 1(c). For instance, communication γ_{1-2} is replaced by task τ_6 and the edges connecting τ_6 to τ_1 and τ_2 are introduced. $\mathcal{K}$ defines the set of all such communication tasks. Secondly, on top of the precedence relations given by data dependencies between processing or communication tasks, we introduce additional precedence relations $r \in \mathcal{R}$, generated as result of scheduling tasks mapped to the same PE and communications mapped on the same CL. In Fig. 1(c) the dependencies $\mathcal{R}$ are represented as dotted edges. After these transformations we obtain an extended task graph $G_E(\mathcal{V}, \mathcal{E})$.

3 Motivational Example

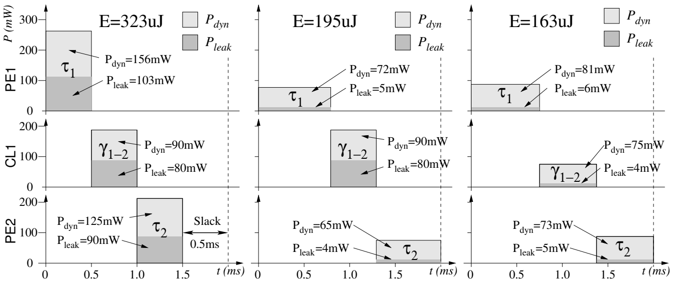
3.1 Voltage Scaling with Transition Overheads

To demonstrate the influence of the transition overheads in terms of energy and delay, consider the following motivational example. For clarity reasons we restrict ourselves here to a single processor system that offers three voltage modes, $m_1 = (1.8V, -0.3V)$, $m_2 = (1.5V, -0.45V)$, and $m_3 = (1.2V, -0.8V)$, where $m_z = (V_{dd_z}, V_{bs_z})$. The rail and substrate capacitance are given as $C_r = 10\mu F$ and $C_s = 40\mu F$. The processor needs to execute two consecutive tasks (τ_1 and τ_2) with a deadline of $0.225ms$. Fig. 2(a) shows a possible voltage schedule. As we can observe, each of the two tasks is executed in two different modes: task τ_1 executes first in mode m_2 and then in mode m_1, while task τ_2 is initially executed in mode m_3 and then in mode m_2. The total energy consumption of this schedule is the sum of the energy dissipation in each mode $E = 9 + 15 + 4.5 + 7.5 = 36\mu J$. However, if this voltage schedule is applied to a *real* voltage-scalable processor, the resulting schedule will be influenced by transition overheads, as shown in Fig. 2(b). Here the processor requires a finite time to adapt to the new execution mode. During this adaption no computations can be performed, i.e., the task execution is delayed, which, in turn, increases the schedule length such that the imposed deadline is violated. Moreover, transitions do not only require time, they also cause an additional energy dissipation. For instance, in the given schedule, the first transition overhead O_1 from mode m_2 and m_1 requires an energy of $10\mu F \cdot (1.8V - 1.5V)^2 + 40\mu F \cdot (0.3V - 0.45V)^2 = 1.8\mu J$. Similarly, the energy overheads for transitions O_2 and O_3 can be calculated as $13.6\mu J$ and $5.8\mu J$, respectively. The overall energy dissipation of the realistic schedule shown in Fig. 2(b) accumulates to $57.2\mu J$.

Let us consider a second possibility of ordering the modes, as given in Fig. 2(c). Compared to the schedule in Fig. 2(a), the mode activation order in Fig. 2(c) has been swapped for both tasks. As long as the transition overheads are neglected, the energy consumption of the two schedules is identical. However, applying the second activation order to a real processor would result in the schedule shown in Fig. 2(d). We can observe that this schedule exhibits only two mode transitions (O_1 and O_3) within the tasks (intra switches), while the switch between the two tasks O_2 (inter switch) has been eliminated. The overall energy consumption has been reduced to $E = 43.6\mu J$, a reduction by 23.8% compared to the schedule given in Fig. 2(b). Further, the elimination of transition O_2 reduces the overall schedule length, such that the imposed deadline is satisfied.

3.2 Voltage Scaling on Repeater-based Buses

Consider an architecture consisting of two voltage-scalable processing elements (PE1 and PE2) that communicate via a repeater-based, shared bus (CL1), which also allows voltage scaling. PE1 has to execute task τ_1 and PE2 runs task τ_2. Task τ_2 can only start after receiving data from task τ_1, and it has to finish execution before a deadline of $2ms$. Fig. 3(a) shows the initial schedule for this system, considering an execution at the nominal voltage settings (highest supply voltage and body bias voltage), i.e., all components run at their highest performance. The diagram shows the power dissipation (dynamic and leakage) of the individual components over time. For clarity reasons we assume in this example that the processors as well as the repeaters of the bus have the same nominal voltage values ($V_{dd} = 1.8V$ and $V_{bs} = 0V$). Further, we assume that the supply voltages and the body bias voltages of all components can be varied continuously in the ranges $[0.6, 1.8]V$ and $[-1, 0]V$, respectively. Given the power consumptions at the nominal voltages, we can compute a total energy consumption of the tasks and communication in the initial schedule as $(156+103)mW \cdot 0.5ms + (90+80)mW \cdot 0.5ms + (125+90)mW \cdot 0.5ms = 323\mu J$. As can be observed, at the nominal voltages the system over-performs, leading to a slack of $0.5ms$.



(a) Schedule without voltage scaling (b) Scaling of PEs (c) Scaling of PEs and CL

Figure 3. Voltage scaling on a repeater-based bus

In order to reduce the energy consumption, we can exploit this slack by scaling the voltages of the processing elements. Using the technique described in [10], it is possible to calculate the optimal voltage settings. The resulting voltages for the execution of tasks τ_1 and τ_2 are $(1.43V, -0.42V)$ and $(1.54V, -0.49V)$, respectively. The corresponding, voltage scaled schedule is shown in Fig. 3(b). The dynamic and leakage power consumptions of the tasks are reduced to $(72mW, 5mW)$ and $(65mW, 4mW)$; however, the execution times have increased to $0.79ms$ and $0.71ms$. Executing the tasks with these settings, the system dissipates an energy of $195\mu J$, a reduction by 39% compared to the energy at nominal voltages.

To demonstrate the importance of combined voltage scaling of the processors and the repeater-based bus, we have produced the voltage scaled schedule shown in Fig. 3(c). Here the supply voltage as well as the body bias voltage of the repeaters have been adjusted with the aim to reduce the dynamic and leakage power dissipation of the bus. The optimal voltage settings for the tasks and the communication can be calculated as $(1.48V, -0.42V)$ for PE1, $(1.77V, -0.61V)$ for

PE2, and $(1.59V, -0.50V)$ for the bus repeaters. Correspondingly the power dissipations are given by $(81mW, 5mW)$, $(75mW, 4mW)$, and $(73mW, 5mW)$, thereby, reducing the overall system energy dissipation to $163\mu J$. Compared to the nominal energy consumption a reduction by 49%, which is 10% better than in the case when only the PEs are voltage scaled. ☐

3.3 Voltage Swing Scaling on Fat Wire-Based Buses

In this example, we demonstrate the influence that a dynamic variation of the voltage swing (the voltage on the wire) has on the energy efficiency of the bus. Fig. 4 shows the total power consumption of a fat wire bus (including drivers and receivers), depending on the voltage swing at which data is sent. These plots have been generated via SPICE simulations using the Berkeley predictive $70nm$ CMOS technology library. The two plots show the total power consumption on the bus for two different voltage settings of the bus drivers and receivers. For example, if the driver connected to PE1 and the receiver at PE2 operate at $1.0V$, the lowest bus power dissipation ($0.55mW$) is achieved by a voltage swing of $0.14V$. Let us assume that the voltages of the driver and receiver are changed during run-time to $1.8V$ due to voltage scaling. The bus power/voltage swing relation for this situation is indicated by the dashed line. As we can observe, by keeping the voltage swing at $0.14V$, the power dissipation on the bus will be $4.5mW$. However, inspecting the plot reveals that it is possible to reduce the bus power dissipation by changing the voltage swing from $0.14V$ to $0.6V$. At this voltage swing, the bus dissipates a power of $2.2mW$, i.e., a 51% reduction can be achieved by changing the voltage swing.

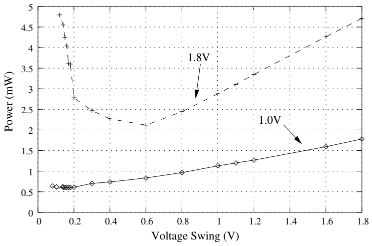


Figure 4. Optimum swing on a fat wire bus

Now assume that the driver and receiver voltages are changed back from $1.8V$ to $1.0V$. Keeping the swing at $0.6V$ results in a power of $0.83mW$, which is, compared to the optimal $0.55mW$ at $0.14V$, 33% higher than necessary. ☐

4 Problem Formulation and Solution

We assume that all tasks and communications of the extended task graph have been mapped and scheduled onto the target architecture. For each task τ_i its deadline dl_i, its number of clock cycles to be executed NC_i, and the switched capacitance C_{eff_i} are given. Each processor can vary its supply voltage V_{dd} *and* body bias voltage V_{bs} within certain continuous ranges (for the continuous voltage scaling problem), or within a set of discrete voltages pairs $m_z = \{(V_{dd_z}, V_{bs_z})\}$ (for the discrete voltage scaling problem). The power dissipations (leakage, dynamic) and the cycle time (processor speed) depend on the selected voltage pair. Such a voltage pair is also referred to as *performance mode*. We consider that a transition between two different performance modes on a processor requires an overhead in terms of energy and time. Tasks are executed cycle by cycle, and each cycle can potentially execute at a different voltage pair, i.e., at a different performance mode.

For each communication task τ_{ij}, which captures a communication between processing task τ_i and τ_j, the number of bytes b_{ij} is given. Depending on the employed bus implementation style, either using repeaters or fat wires, we have to distinguish between two subproblems:

Repeater Implementation: The communication speed as well as the communication power on bus architectures implemented through repeaters depend on the supply voltage and body bias voltage. Similar to processing elements, these voltages can be varied within a continuous range, or within a set of discrete voltage pairs $m_z = \{(V_{dd_z}, V_{bs_z})\}$, and transitions between different bus performance modes require an energy and time overhead. Furthermore, an energy overhead is required to adapt the bus voltage to the processor voltage. ☐

Fat Wire Implementation: If communication is performed over fat wires, it is necessary to dynamically adapt the voltage swing at which data is transfered. Furthermore, in order to reduce the power dissipated in the bus drivers and receivers, it is possible to dynamically scale the supply and body bias voltage of these components. While the voltage swing can be scaled without an influence on the bus performance, the operational speed of the bus drivers and receivers is affected through voltage scaling, i.e., the bus performance has to be adjusted in accordance to the driver/receiver speed. In the case of continuous voltage scaling, the value for the voltage swing, the supply voltage, and the body bias voltage can be changed within a continuous range. On the other hand, for the discrete voltage scaling case, the components operate across sets of discrete voltages, referred to as modes. For the voltage swing this set is $n_z = \{V_{sw_z}\}$ and for the bus drivers and receiver the set is $m_z = \{(V_{dd_z}, V_{bs_z})\}$. Of course, changing the voltage swing value as well as the supply and body bias voltages requires an energy and time overhead. ☐

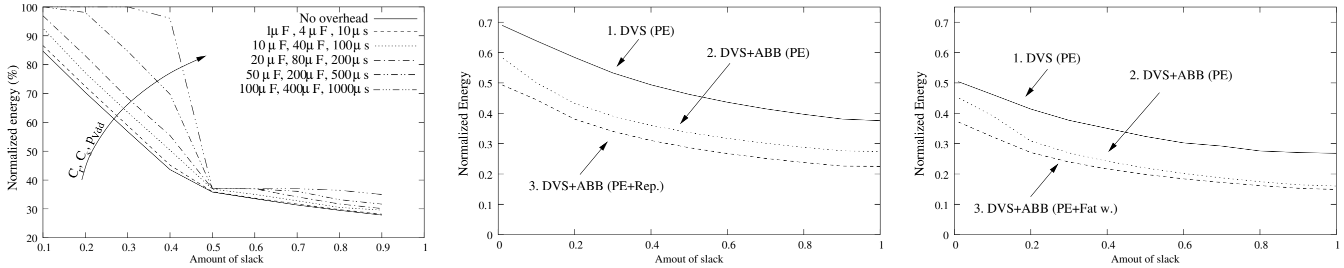
Our overall goal is to find mode assignments for each processing and communication tasks such that the individual task deadlines are satisfied and the total energy consumption, including overheads, is minimal. ☐

We solve optimally in polynomial time the continuous voltage scaling problem using a convex nonlinear formulation. The discrete voltage scaling problem is strongly NP hard. We propose a mixed integer linear programming formulation for the optimal solution and a time-efficient heuristic. More details about the solutions can be found in [10] and [11].

5 Experimental Results

We have carried out several experiments using numerous generated benchmarks. The automatically generated benchmarks consist of 120 task graphs containing between 50 and 300 tasks, which are mapped and scheduled onto architectures composed of 2 to 5 processors, interconnected via 1 to 4 buses either implemented repeater-based or fat wire-based. In the continuous voltage scaling case the processor voltage pairs (V_{dd},V_{bs}) are varied between $(0.6, -1)V$ and $(1.8, 0)V$, while the discrete voltage levels are $m_z = \{(1.8, 0), (1.4, -0.2), (0.8, -0.6), (0.6, -1)\}$. The voltage ranges for repeater-based systems are identical to the possible processor voltage settings. For the fat wire-based buses the continuous voltage swing values can be set between 0.2 and $1V$, and for the discrete case it can be adjusted to $m_z = 0.2, 0.3, 0.4, 0.6, 1V$. The technology dependent parameters of these processors and buses were considered to correspond to a CMOS fabrication process in $50nm$, for which the leakage power represents approximately 50% of the total power consumed. For experimental purpose the amount of deadline slack in each benchmark was varied over a range 0 to 100%, using a 10% increment. Furthermore, the amount of communication within the generated benchmarks was varied between 10 to 50% of the total execution time, with an increment of 10%.

In order to investigate the influence of transition overheads, we have carried out an additional set of experiments in which the amount of processors' overheads in terms of energy and delay were varied by adjusting the values for C_r, C_s, p_{Vdd}, and p_{Vbs}. In accordance, we use the discrete voltage selection with consideration of overheads. The results are given in Fig 5(a). As expected, the energy dissipation increases for higher values of the overhead determining parameters. For



(a) Influence of scaling the overheads

(b) Energy dissipation of a reapter−based system

(b) Energy dissipation of a fat wire−based system

Figure 5. Optimization Results

instance, while a "hypothetical" processor which requires no transition overheads can reduce the energy consumption by 58% if 40% of slack is available, a realistic processor with $C_r = 20\mu F$, $C_s = 80\mu F$, $p_{Vdd} = 200\mu s/V$, and $p_{Vbs} = 200\mu s/V$ achieves only 42%. This highlights the importance to carefully consider the influence of transition overheads. Moreover, if we consider systems with between 10-40% available slack, the consideration of transision overheads during the optimization results in solutions with an improved energy consumption of 12%.

We conducted another set of experiments with the aim to investigate the energy savings that are achievable when dynamically scaling the supply voltage as well as body bias voltage of bus repeaters. The 32bit-wide bus architecture under consideration consisted of 27 repeaters per bit-line of which each has a total length of $27.4mm$. The capacitance of single wire including the repeaters was estimated as $7.2pF$, using the power optimized data from [1]. During the experiments from Fig. 5(b) we assumed the communication amount to be of 30%, compared to the total execution time. Inspecting the graphs reveals that the highest energy savings are achieved by considering the combined V_{dd} and V_{bs} continuous voltage scaling scheme on the buses as well as on the processors (plot 3). We can also observe that the energy efficiency is increased by approx. 12% if combined voltage scaling is applied on the bus (difference between plot 2 and 3). Generally, the combined V_{dd} and V_{bs} scaling yields higher energy saving (around 30%) than the V_{dd}-only scaling (difference between plot 1 and 2) [1]. In the experiments from Fig. 5(c), we investigated the achievable energy savings on a fat wire-based bus system, assuming the same bus-width as in the repeater based approach. However, fat wires are considered to be suitable only for short distance connections. In the following we consider a length of $4mm$ with a single line capacitance $609fF$. As expected, the fully voltage scalable systems, achieve the best energy savings, with reductions between 4% to 18% compared to systems with fixed bus voltages. Experimental experience has shown that with an increasing amount of communication data, the bus voltage scaling approach achieves increasingly higher energy reductions. If, for example, the time required for communications is around 15% of the total execution time, the energy savings due to bus voltage scaling are around 10%. With communication time around 30%, the energy savings become around 16%. Since all plots in Fig. 5(b) and (c) represent the results for continuous voltage scaling, it is interesting to note that the proposed heuristic for discrete voltage scaling achieves results that are within 4% of the values obtained at continuous voltage levels. It is important to note that the efficiency difference of about 12% on average, between implementations with and without bus voltage scaling is preserved also when discrete voltage levels are used.

6 Conclusions

In this paper we have investigated the combined voltage scaling of processors and communication links, for reduced dynamic and leakage power consumption. Energy savings are achieved by scaling the supply and body bias voltage of processors as well as repeater-based buses. Furthermore, in the case of fat wire-based buses, energy efficiency is obtained by additionally scaling the voltage swing. For this purpose, we have used a set of accurate delay and energy models. Both energy and time overheads are considered during the optimization. Experiments have shown promising results, indicating the necessity to consider combined supply and body bias voltage scaling for processors and communication links as well as the influence of the transition overheads.

References

[1] K. Banerjee and A. Mehrotra. A Power-Optimal Repeater Insertion Methodology for Global Interconnects in Nanometer Designs. *IEEE Trans. on Electron Devices*, 49(11):2001–2006, Nov 2002.

[2] L. Benini, G. De Micheli, E. Macii, D. Sciuto, and C. Silvano. Address bus encoding techniques for system-level power optimization. In *Proc. DATE'98*, pages 861–867, 1998.

[3] Shekhar Borkar. Design Challenges of Technology Scaling. *IEEE Mirco*, pages 23–29, July 1999.

[4] Anantha P. Chandrakasan and Robert W. Brodersen. *Low Power Digital CMOS Design*. Kluwer Academic Publisher, 1995.

[5] W. Fornaciari, D. Sciuto, and C. Silvano. Power Estimation for Architectural Exploration of HW/SW Communication on System-Level Buses. In *CODES'99*, pages 152–156, May 1999.

[6] Cheng-Ta Hsieh and Massoud Pedram. Architectural Energy Optimization by Bus Splitting. *IEEE Trans. on CAD*, 21(4):408–414, April 2002.

[7] C. Kim and K. Roy. Dynamic Vth Scaling Scheme for Active Leakage Power Reduction. In *Proc. DATE02*, pages 163–167, March 2002.

[8] J. Liu, P. Chou, and N. Bagherzdeh. Communication Speed Selection for Embedded Systems with Networked Voltage-Scalable Proceossors. In *Proc. CODES'02*, 2002.

[9] S. Martin, K. Flautner, T. Mudge, and D. Blaauw. Combined Dynamic Voltage Scaling and Adaptive Body Biasing for Lower Power Microprocessors under Dynamic Workloads. In *Proc. ICCAD-02*, 2002.

[10] A. Andrei, M. Schmitz, P. Eles, Z. Peng, B. Al Hashimi. Overhead-Concious Voltage Selection for Dynamic and Leakage Energy Reduction in Time-Constrained Systems In *Proc. DATE-04*, 2004.

[11] A. Andrei, M. Schmitz, P. Eles, Z. Peng, B. Al Hashimi. Simultaneous Communication and Processor Voltage Scaling for Dynamic and Leakage Energy Reduction in Time-Constrained Systems In *Proc. ICCAD-04*, 2004.

[12] L. Shang, L. Peh, and N. Jha. Power-efficient Interconnection Networks: Dynamic Voltage Scaling with Links. *Comp. Arch. Letters*, 1(2):1–4, May 2002.

[13] C. Svensson. Optimum Voltage Swing on On-Chip and Off-Chip Interconnects. *IEEE J. Solid-State Circuits*, 36(7):1108–1112, July 2001.

[14] D. Sylvester and K. Keutzer. Impact of Small Process Geometries on Microarchitectures in Systems on a Chip. *Proceedings of the IEEE*, 89(4):467–489, April 2001.

[15] G. Wei, J. Kim, D. Liu, S. Sidiropoulos, and M. Horowitz. A Variable-Frequency Parallel I/O Interface with Adaptive Power-Supply Regulation. *IEEE J. Solid-State Circuits*, 35(11):1600–1610, Nov 2000.

---

[1] Please note that energy savings are achieved even at zero deadline slack. This means that according to the given schedule, the task set finishes on deadline when executed at the highest voltage. However, even in this case, due to the fact that the application executes on a multiprocessor system, initial slack (idle processor time) is available in the system and can be exploited by voltage scaling.