

Configurable On-Line Global Energy Optimization in Multi-Core Embedded Systems Using Principles of Analog Computation

Zeynep Toprak Deniz, Yusuf Leblebici, and Eric Vittoz

Ecole Polytechnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland
{zeynep.toprak,yusuf.leblebici,eric.vittoz}@epfl.ch

Abstract. This work presents the design of an on-line energy optimizer unit, which is capable of dynamically adjusting power supply voltages and operating frequencies of multiple processing elements (PE), tailored to the instantaneous workload information and is fully adaptive to variations in process and temperature. The circuit design borrows some of the basic principles of analog computation to continuously optimize the system-wide energy dissipation of multiple cores. The analogy between the energy minimization problem under timing constraints in a general task graph and the power minimization problem under Kirchhoff's current law (KCL) constraints in an equivalent resistive network is exploited. To our best knowledge, this is the first study of its kind to demonstrate an on-line solution to complex, multi-variable energy optimization problem which allows dynamic adjustment of individual operating frequencies and supply voltages of multiple processing elements.

1 Introduction

The continuing exponential growth of complexity in VLSI systems is largely supported by the advances in silicon processing technology, which enable integration of ever more complex functions on a single chip. Future Systems-on-Chip (SoC) are generally envisioned as high-performance embedded systems composed of a heterogeneous network of processing elements (meaning non-identical elements in functionality, size, performance and even the design methodology), providing integrated solutions to challenging design problems in the mobile telecommunication, consumer electronics and multimedia domains. Application demands and the continuous trend towards mobile, distributed systems have also made battery-powered portable electronic systems very popular and virtually ubiquitous. Nowadays such systems are widely used in many applications, such as mobile computing, information appliances as well as various industrial, medical and military applications. Hence, energy dissipation and energy/performance trade-offs have emerged as major factors in determining the weight, the size and the life-time (autonomy) of portable devices. Thus, the ultimate energy management goal in such complex systems is to reduce "system-level" or global

energy consumption, rather than concentrating on local minimization. A number of system-level energy optimization techniques have been presented in the literature recently [1, 2].

The significance of the problem of energy optimization in multi-core systems where the individual energy demands of various processing elements (PE) are governed by instantaneous workload requirements is underlined by the increasing prominence of multi-core systems that must operate under strict energy budget constraints, in mobile applications. A range of solutions have been proposed over the last few years, which are mostly based on static, off-line calculation of a limited set of operating points in the form of optimum voltage and frequency assignments, that are subsequently chosen according to actual demands. These observations lead to the conclusion that implementation of sophisticated energy management techniques will be necessary in SoC/NoC (Network-on-Chip) architectures that consist of multiple functional units, where each unit is experiencing a non-uniform workload during operation time. In such systems, fine grained energy management is generally implemented as Dynamic Voltage Scaling (DVS), which refers to varying the operation speed of a processor by changing the clock frequency along with the supply voltage. DVS is usually implemented as an open-loop technique whereby the single digital core is characterized for throughput at a given clock speed and at a given voltage with ample margin allowed for temperature, power supply and fabrication variations necessitating extensive characterizations, to build a hard-coded table of speed versus supply voltage that insures performance criteria for each wafer [3]-[6].

All components in a SoC/NoC, executing a specific application, are expected to have varying energy requirements in time domain which could be easily derived from their instantaneous workload estimations. Based on this "local" information, the optimum energy allocations for all sub-blocks of a complex SoC/NoC could be computed locally in time domain, for any given task. This approach is widely used in the literature to reduce the energy consumption of individual PEs [7, 8]. However, these "locally optimum" allocations may not always coincide with the global optimum for the overall system, especially taking into account the interaction of the various system components. Thus, a system-wide (global), continuous-time optimization approach would be expected to yield better results from a system point of view (see Fig. 1). However, system-level energy optimization under performance constraints is a challenging problem. The concept of system stability needs to be considered when several components adopt dynamic policies to control energy consumption and performance [9]. Possible oscillations in power/performance space that could be caused by applied energy management policies are undesirable, and should be avoided. Hence, it is preferable to implement very efficient on-line dynamic power management techniques, in a centralized fashion, guaranteeing globally optimum results and system-level stability.

Another important issue that needs to be addressed is the power consumption of the optimizer block, i.e. the optimizer, itself. This issue has not been carefully validated until now, and has been largely neglected in the literature. However,

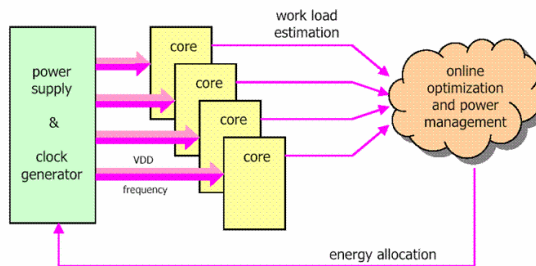


Fig. 1. Conceptual representation of the proposed on-line global power/energy management approach.

the power dissipation of the optimizer unit could be a significant component of the overall system dissipation, especially if an on-line optimization policy is being implemented for multiple components, using a conventional digital processor to solve the optimization problem continuously under varying conditions. To address this issue, we propose using analog circuit principles instead of a digital processor, and thus, saving energy. It is also a known fact that approximate, stable solutions to such multi-variable optimization problems (such as the gradient descent algorithm) can be obtained by using very compact analog circuits [10]-[14]. This approach has the potential advantage of generating sufficiently accurate solutions, while dissipating a small fraction of the power that would be needed by a digital processor to solve the same problem. Furthermore, the energy manager design shall demonstrate the continuous, real-time energy consumption optimization (being independent of the application) to be more response-time efficient than currently proposed energy management policies utilizing discrete power levels [3, 15].

This chapter will explain in detail the basis of the proposed idea of continuously (in time domain) adjusting the control knobs of the overall system in order to minimize the global energy consumption of the embedded system, subject to timing constraints, and it will present the design of the proposed central (global) optimizer unit based on simple analog circuit topologies and design aspects. To do this, the analogy between the problem of minimizing the energy dissipation on a given task graph and the fundamental electrical behavior of resistive networks will be exploited first. It will be also shown that the energy requirement of solving a multi-variable optimization problem in real-time can be minimized based on our approach.

2 Modelling the Multi-Unit Global Energy Optimization Problem

In large scale systems design, it is essential to perform pre-design analysis using extensive modelling tools in order to gain further insight in a complex system, improve understanding of the problem under consideration, find unexpected emer-

gent properties and quantify system parameters before starting physical design. It is also important to abstract the real design problem in order to provide its complete definition and form the basis of its formulation to experiment and to represent any possible solutions. This section explores the hierarchical abstraction and formalization of the energy optimization problem in multi-core systems to consolidate the conceptual variables and constraints of the objective function.

2.1 Cost Function and Non-Linear Constraint Formulation

We start this section with the definitions as well as the key elements of the problem of discourse. It is assumed that the system is composed of real-time dependent tasks with deadlines to be executed on multiple variable-voltage PEs. Task scheduling being known a priori, the functionality of the data-flow of such systems realized as heterogeneous distributed architectures are captured as Directed Acyclic Graphs (DAG) $G_S(t, C)$, an example structure of which is shown in Fig. 2. Furthermore, the formalism describing the relationship between the application software and the time-domain scheduling of various tasks is typically supplied by the concept of task graphs (TG).

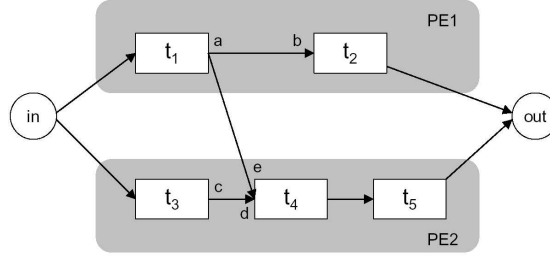


Fig. 2. Task graph of five tasks mapped on two processing element.

In the DAG in Fig. 2, each node represents computational tasks t_u , while edges indicate the data dependencies between these tasks, indicating that task v can only start after task u finishes. Tasks require a finite number of clock cycles N_u to be executed, depending on the PE on which they are mapped. Further, tasks are annotated with deadlines D_{lu} that have to be met during application run-time and T_{graph} on a task set restricts all tasks' finish time. During voltage scaling N_u remains constant, while CT cycle time and d_u task execution time of u^{th} task change with supply voltage V_{DDu} . CT , d_u and E_{dyn} can be computed as

$$d_u = N_u CT \quad \text{where} \quad CT = \frac{k L_d V_{DDu}}{(V_{DDu} - V_{th})^\alpha} \quad (1)$$

$$E_{dyn} = N_u C_u V_{DDu}^2 \quad (2)$$

where k is a technology dependent constant, V_{th} is the threshold voltage of the devices, α is a technology dependent constant, ranging from 1.2 to 2 for recent technologies, derived in *Alpha-Power Law MOSFET Model* [16, 17] and, C_u is the effective switching capacitance per cycle. Although the dynamic power dissipation is still dominating, the trend to reduce the overall circuit supply voltage and the threshold voltage is increasing concerns about the leakage currents; for advanced technologies (< 90 nm) it is expected that the leakage will account for more than 50% of the total power [18]. The leakage energy is given by Eq. 3, where V_{BSu} is the voltage applied between the body and the source of the transistor (body bias), I_{ju} is the junction leakage current, L_d , N_g are logic depth and average number of gates respectively, k , K_3 , K_4 and K_5 are constant fitting parameters denoting circuit technology dependency [18]-[22]. Another important issue, which often is overlooked in voltage scaling approaches, is the consideration of transition overheads, i.e., each time the PE's supply voltage is altered; this change requires a certain amount of extra energy and time. The energy overhead (E_{OH}), when switching from V_{DDu} to V_{DDv} is given by Eq. 4, where C_r denotes power rail capacitance.

$$E_{leak} = \frac{L_d N_g N_u k V_{DDu} (V_{DDu} K_3 e^{K_4 V_{DDu}} e^{K_5 V_{BSu}} + |V_{BSu}| I_{ju})}{(V_{DDu} - V_{th})^\alpha} \quad (3)$$

$$E_{OH} = C_r |V_{DDu} - V_{DDv}|^2 \quad (4)$$

Hence, the total task energy and the energy of the whole system can be given as Eq. 5 and Eq. 6 respectively. Here, Eq. 6 represents the objective (cost) function to be minimized.

$$E_{task} = E_{dyn} + E_{leak} + E_{OH} \quad (5)$$

$$E_{total} = \sum_{taskcount} E_{task} \quad (6)$$

We have to guarantee that due to voltage scaling technique applied, tasks with deadlines still finish before their deadlines and the last task on each PE finishes no later than T_{graph} . This means that the sum of task durations in each path (from input to output) in the DAG should not be greater than T_{graph} . Note that the number of paths in a DAG can grow exponentially with respect to the number of edges, which implies that path-based optimization methods are not easily applicable and that voltage scaling approaches used on single PE systems cannot be readily extended to solve energy optimization problem on multiple PEs.

The execution time of task u at the highest supply voltage (V_{DDMAX}) is a constant T_u . Since *IN* and *OUT* nodes are conceptual nodes their execution times (T_{OUT} , T_{IN}) are set to 0. Besides the task execution time d_u , each task is also associated with a starting time denoted as D_u . Hence, the timing constraints on the given task graph can be modelled as

$$D_{OUT} - D_{IN} \leq T_{graph} \quad (7)$$

$$D_v - D_u - d_u \geq 0 \forall e(u, v) \in DAG \quad (8)$$

$$D_u + d_u \leq dl_u \forall u \text{ with deadline} \quad (9)$$

$$dl_u \geq T_u, \text{ and } D_u \geq 0, \text{ integer} \quad (10)$$

$$V_{DD_{MIN}} \leq V_{ddu} \leq V_{DD_{MAX}} \quad (11)$$

For a feasible scheduling, if D_{IN} set to be 0, the above constraints guarantee that tasks with deadlines will finish before their deadlines and the finish time of all tasks is not greater than T_{graph} . In order to be compatible with the working range of the PE's the supply voltage of each task is constricted between a minimum and a maximum value, according to Eq. 11. Combining the objective and the constraints given in Equations 7 to 11, we have the IP formulation for the voltage scaling problem in multi-PE systems. Trading-in the increase of delay for energy savings, the relationship between d_u and E_{task} has to be established. If the supply voltage can change continuously, the task execution time d_u can also be assumed to change continuously. In this case E_{task} is a convex function of d_u [22], i.e., we have $E_u = f(d_u)$, where $f(\cdot)$ is a convex function. Substituting E_u with $f(d_u)$, the IP formulation of the problem of minimization of energy dissipation in multi-core systems for continuous voltage scaling approach is completed.

3 Energy Minimization on an Arbitrary Task Graph Using Resistive Network Analogies

In the following analysis, we will consider generic high-performance multi-core systems that are composed of a heterogeneous network of PEs. Due to the diversity of the applications that run within the system and their different degrees of parallelism, the workloads imposed on the system components are non-uniform over time. For many applications, peak performance is required only during some time intervals in such systems. This introduces slack times during which the system can reduce its performance to save energy. The key in energy-efficient designs is the ability to tune PE performance to the non-uniform workload. Here, the first goal is to explore the problem of *energy minimization* on a given TG, and to show the analogy between this problem and the fundamental electrical behavior of a resistive network. Our ultimate goal is to exploit this analogy in the form of a compact solution to the optimization problem.

In this section, an analogy will be introduced which maps the cost function of the energy minimization problem under timing constraints in multi-core systems, into the problem of minimizing power consumption in an equivalent resistive network subject to KCL. In the following, it will be assumed that the TG is mapped and scheduled onto the target architecture (multi-unit PE system), i.e., it is known a priori where and in which order tasks are executed and the communications between tasks take place.

3.1 From Task Graph to Resistive Network

In heterogeneous multi-core systems, the timing relationships (constraints) and the relative ordering between various tasks of an application are usually represented with a task graph which is used to capture the data-flow interdependencies of the entire system, as already established in the previous section. On the other hand, a resistive network is a connected graph (possibly with multiple edges) where each edge e is assigned a positive real number R_e called its resistance (in Ohms). Based on this analogy a given task graph can be mapped onto a resistive network by replacing each task (node) in the task graph with a resistor and edges of task graph as electrical connections in the resistive counter part.

It is a well-known fact that analog processing is usually more efficient than digital processing with respect to power consumption and chip area, when high precision is not required. Resistive networks have been widely used for various applications in analog VLSI [23]. A resistive network (RN) can be described by a system of linear equations based on Kirchhoff's and Ohm's laws. In a parallel resistive network that consists of n resistors, an imposed current i splits into n components proportional to branch conductances (G_n) that act as a current divider. According to Maxwell's heat theorem [24], *any network of linear resistors driven by a constant current, at steady state intrinsically minimizes the power dissipated in the form of heat in the network*. The demonstration of this theorem can be found from the book entitled *A Treatise on Electricity and Magnetism* by James C. Maxwell, first published in 1891 [24].

Tasks in real-world applications usually have control and data dependencies. Processing element sharing can be captured in a TG with multiple PEs with additional edges representing the control relation between dependent tasks. Figure 2 is a simple yet good example for such a case, where each parallel branch represents a PE. In this example, five tasks are mapped and scheduled on two PEs. Tasks mapped on different PEs can run in parallel in time as a basic consequence of parallelism, i.e., the period of each parallel branch in a TG is still equal to T . Notice that the points a through e (labelled on the TG for the sake of easy identification) actually represent the same instant in time. Also recall that t_4 can only start after processing of t_1 and t_3 are finished (DAG). The given TG indicates that t_1 and t_3 have to be finished at the same time for sake of completing work just-in-time (corresponding to minimum energy consumption), which describes a soft deadline for tasks. Similarly, execution time of tasks t_4 and t_5 mapped on the second PE should be equal to that of task t_2 running on the first PE. Hence, from the time point of view, the system shown in Fig. 2 can be presented by a simple sequential task graph as shown in Fig. 3(a). Here, original tasks t_1 and t_3 running in parallel on two different processing elements, are represented as the first combined task $W_1 = f(t_1, t_3)$, and similarly $W_2 = f(t_2, t_4, t_5)$ represents the combined tasks t_2, t_4, t_5 .

It is clear from the very simple task graph shown in Fig. 3(a) that the second task (W_2) can start only after the preceding task (W_1) is finished. To optimize the overall energy consumption, these two tasks must share the available time T with respect to their workloads, where the duration of each task is indicated as

d_{W1} and d_{W2} respectively. Clearly, the time constraint: $d_{W1} + d_{W2} = T$ describes the condition that is needed in order to finish the mapped function within its deadline.

The total dissipated energy in the system can be written as the summation of the all task energies. Since the u^{th} task is executed during the time period d_u consuming a power of P_u , the energy dissipation of the u^{th} task can be found as $E_u = P_u d_u$. Hence, the problem of minimizing the overall dissipated energy in a given system (Fig. 3(a)), represented with its task graph under timing constraints, is formulated in Eq. 12.

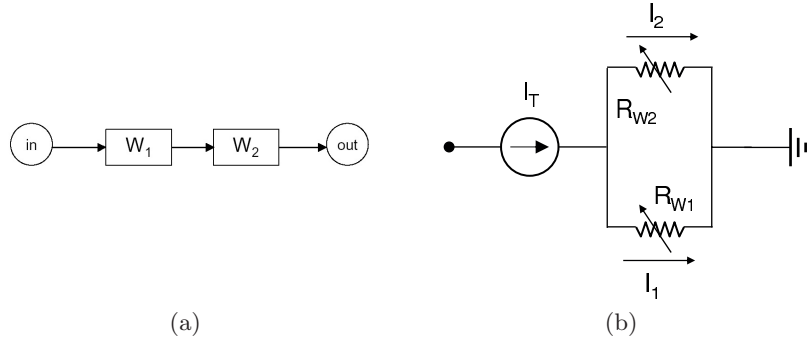


Fig. 3. (a) Simple task graph of two sequential tasks (b) and the resistive-network representation that corresponds to this task graph.

$$\begin{aligned}
 \min E_{total} &= \min \sum_u E_u = \min \sum_u P_u d_u \\
 \text{subject to } \sum_u d_u &= T
 \end{aligned} \tag{12}$$

At this point, we surmise that the equivalent resistive network of this specific task graph consists of two controlled resistors in parallel as shown in Fig. 3(b), where the network is supplied with a constant current I_T . This total current will be split linearly between branches proportional to branch conductances, I_1 and I_2 (Ohm's law). Due to KCL these currents must satisfy the equality $I_1 + I_2 = I_T$.

Now consider the total power consumption in the equivalent RN shown in Fig. 3(b). Intrinsicly, we know that the RN will consume the lowest possible power, P_{total} , at steady-state for a given driving current (I_T) according to Maxwell's Heat theorem. Due to KCL, I_T will be split into parallel branch currents (I_i) that are inversely proportional to branch resistances R_i (proportional to branch conductance G_i). Hence, it can be seen that the parallel resistive net-

work actually realizes the solution of the following minimization problem, under the constraint that the sum of all branch currents is equal to I_T .

$$\begin{aligned} \min P_{total} &= \min \sum_i P_i = \min \sum_i \frac{I_i^2}{G_i} \\ \text{subject to } \sum_i I_i &= I_T \end{aligned} \quad (13)$$

A comparison between Eq. 12 and Eq. 13 reveals the clear analogy between the problem of minimizing energy consumption on a complex system under timing constraints, and the problem of minimizing power dissipation in a resistive network under KCL constraint. Note that the branch currents (I_i) correspond to task durations (d_u) in the former problem. Thus, for the simple case described in Fig. 3, it is shown that the task graph can be represented by the equivalent resistive network.

Still, the simple resistive network equivalent given in Fig. 3(b) is not sufficient to model the actual behavior of the system with respect to individual tasks mapped on two processing elements. Note that, W_1 is a function of the two tasks t_1 and t_3 , that must be executed in parallel on two different processing elements, i.e. these two tasks must have the same duration ($d_{W_1} = d_1 = d_3$). Similarly, in a resistive network branch consisting of two series connected resistors, each resistor must carry the same amount of branch current. Based on this analogy *all parallel tasks can be converted into series-connected branches in the equivalent resistive network*. Furthermore, W_2 is a function of two series tasks executed in parallel to a third task. In this case, execution time of task t_2 must be equal to the sum of execution times of tasks t_4 and t_5 . Consequently, the amount of time necessary for execution of task W_2 will be split among t_4 and t_5 ($d_{W_2} = d_2 = d_4 + d_5$) according to the actual workload of these two tasks. Similarly, in a resistive network branch of parallel connected resistors the main branch current will be shared proportionally between the parallel branches according to KCL. Hence, *all sequential tasks can be represented by parallel-connected branches in the equivalent resistive network*.

Finally, the equivalent RN of the TG given in Fig. 4(a) can be implemented as shown in Fig. 4(b). I_T represents the overall available time, and each device current, (I_i), in the resistive network corresponds to the duration of the related task, (d_u) in the associated TG. Consequently, the problem of minimizing the sum of all task energies in a certain application is mapped onto an equivalent resistive network (Fig. 4(b)) consisting of controlled (pseudo-) conductances, where all parallel tasks are converted into series-connected branches, and all serial tasks are converted into parallel-connected branches. Note that each task sequence (or sub-sequence) in the TG corresponds to a parallel section (or sub-section) in the resistive network where KCL is valid, and is represented by a corresponding cut-set. The equivalence between the two analogous minimization problems is illustrated below.

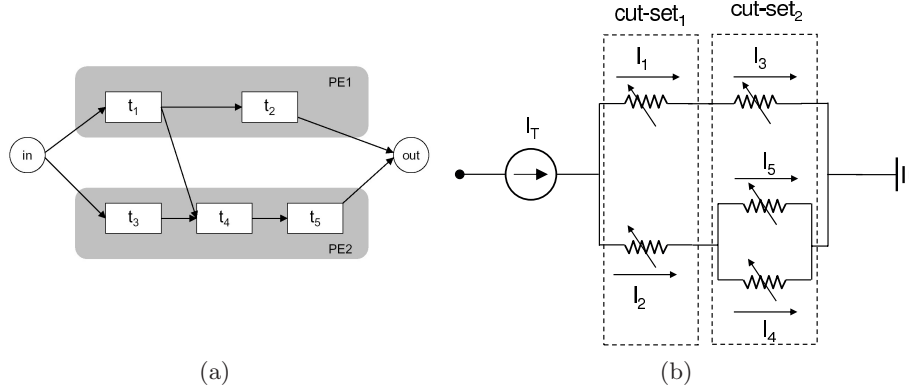


Fig. 4. (a) Task graph of five tasks mapped on two processing element (b) and its parallel-resistive network counter part.

$$\begin{aligned} \min E_{total} &\Leftrightarrow \min P_{total} \\ \min \sum_u E_u = \min \sum_u \frac{P_u}{d_u} d_u^2 &\Leftrightarrow \min \sum_i P_i = \min \sum_i \frac{I_i^2}{G_i} \end{aligned} \quad (14)$$

$$\begin{aligned} \text{subject to } \sum_u d_{u \in j} = T_j &\Leftrightarrow \text{subject to } \sum_i I_{i \in k} = I_{T,k} \end{aligned} \quad (15)$$

$$\forall \text{ task sequence "j"} \quad \forall \text{ cut-sets "k" that correspond to a task sequence "j"}$$

$$\text{where } d_u \Leftrightarrow I_i \quad (16)$$

$$\text{hence } \frac{d_u}{P_u} \Leftrightarrow G_i \quad (17)$$

Note that the summation of branch currents, i.e. elements of the k^{th} cut-set is always equal to the resistive network driving current (I_T). Similarly, summation of the task durations that corresponds to the j^{th} task sequence is equal the task graph period (T) in the given TG. Besides, any subset of the imposed timing constraints, e.g. $d_4 + d_5 = d_2$, are intrinsically modelled by Eq. 15, and hence guaranteed by the resistive network implementation. Thus, the analogy between the energy minimization problem under timing constraints in a general TG and the power minimization problem under KCL constraints in an equivalent RN is demonstrated. Note that, we need to construct a structure to carry out the calculation of P_u (Eq. 17).

Assuming that V_u and I_u are the supply voltage and the current drawn from supply (including dynamic, short circuit and leakage currents) for the u^{th} task, and using d_u as the task duration, the corresponding device conductance G_i can be mapped as follows.

$$\frac{d_u}{P_u} = \frac{d_u}{V_u I_u} \Leftrightarrow G_i \quad (18)$$

From the above explanations, the generalized steps involved for mapping the given task graph into a parallel-resistive network equivalent are as follows:

- Identify and assign the processing elements and the tasks mapped and scheduled on to the given system.
- Insert *IN* and *OUT* nodes into the given TG where edges from *IN* node to the first task of each PE, and edges from the last task on each PE to *OUT* node are added.
- If possible simplify the task graph by replacing the edges representing processing element sharing by equivalent edges capturing the data/control dependencies between PEs.
- Convert all *parallel tasks* in the simplified TG to *series-connected resistor branches* in the resistive network.
- Convert all *series tasks* in the simplified TG to *parallel-connected resistor branches* in the resistive network.
- Replace the *IN* and the *OUT* nodes of TG by a DC current source modelling the TG period and by the *ground* connection providing the necessary current path in the resistive network respectively.

However, not every task graph is in series/parallel configuration. The task graph given in Fig. 5(a) is an example of such non series/parallel configuration. Still, an equivalent resistive network can be mapped from the given TG without violating the corresponding timing constraints as shown in Fig. 5(b) where the cut-sets are highlighted by dashed lines. Here, timing constraints, e.g. $d_1 + d_4 = T$, $d_2 + d_3 + d_4 = T$ and $d_3 + d_4 = d_5$, are intrinsically satisfied due to KCL constraint in the resistive network, i.e. $I_1 + I_4 = I_T$, $I_2 + I_3 + I_4 = I_T$ and $I_3 + I_4 = I_5$ respectively. Hence, the equivalent resistive network of controlled resistors can be mapped for any arbitrary task graph, where each device current represents the available time for the corresponding task. Although the applied mapping scheme has a certain resemblance to creating the dual of a given task graph, it is important to emphasize that the mapping of a given task graph to its equivalent resistive network is based on converting the time domain relation between tasks into equivalent resistive network currents.

4 Implementation of Resistive Network for Global Energy Optimization

4.1 Closed-Loop Operation of the Resistive Network

The ultimate goal of this work is to solve the system-wide energy optimization problem continuously by means of the equivalent power minimization in

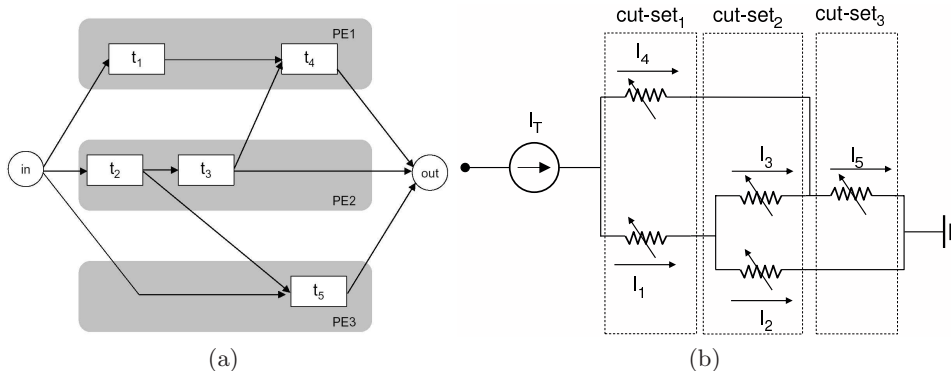


Fig. 5. (a) An example task graph of non series/parallel configuration (b) and the equivalent resistive-network of this task graph.

its resistive network model. In order to fulfil the overall goal, the variations of task-related design parameters given in Eq. 18 must be monitored. The design parameter d_u (task duration), is intrinsically embedded in the resistive-network model of the system, as indicated in Eq. 17.

The premise is that such global optimization (taking into account the complete system) will result in better solutions compared to the case where the energy dissipation of each unit is minimized separately. This concept is illustrated in a simplified manner in Fig. 6. Here, only one loop related to one of the resistive elements is shown, other feedback loops for the rest of the RN are not shown in the figure for sake of simplicity. From now on we will call the shown scheme, modelling individual tasks mapped and scheduled on to the system, as the feedback loop. The working principle of the feedback loop is as follows: due to Kirchhoff's current law (KCL) the branch currents will be divided proportionally with respect to branch conductances. Since the available duration of individual tasks, (d_u) corresponds to the device currents, I_i , the required operation frequency and the necessary minimum possible supply voltage will be calculated in the loop by means of the device currents, I_i . These values will then be used to calculate (estimate) the average supply current as a result of the dictated voltage level, modelling all relevant components such as dynamic, short circuit and leakage currents, and the device conductance will be adjusted according to Eq. 17.

Figure 6 shows the simplified block diagram implementation of the feedback loop for one device conductance, where a current-based approach is used to represent key loop variables. The simple *ghost circuit* (GC) which consists of a ring oscillator replicating the critical path of the PE, is used in each loop to continuously determine the minimum supply voltage and the supply current that correspond to a target operation frequency. The predicted workload information (N_i) is injected into each loop in the form of a 4-bit external control variable. Any

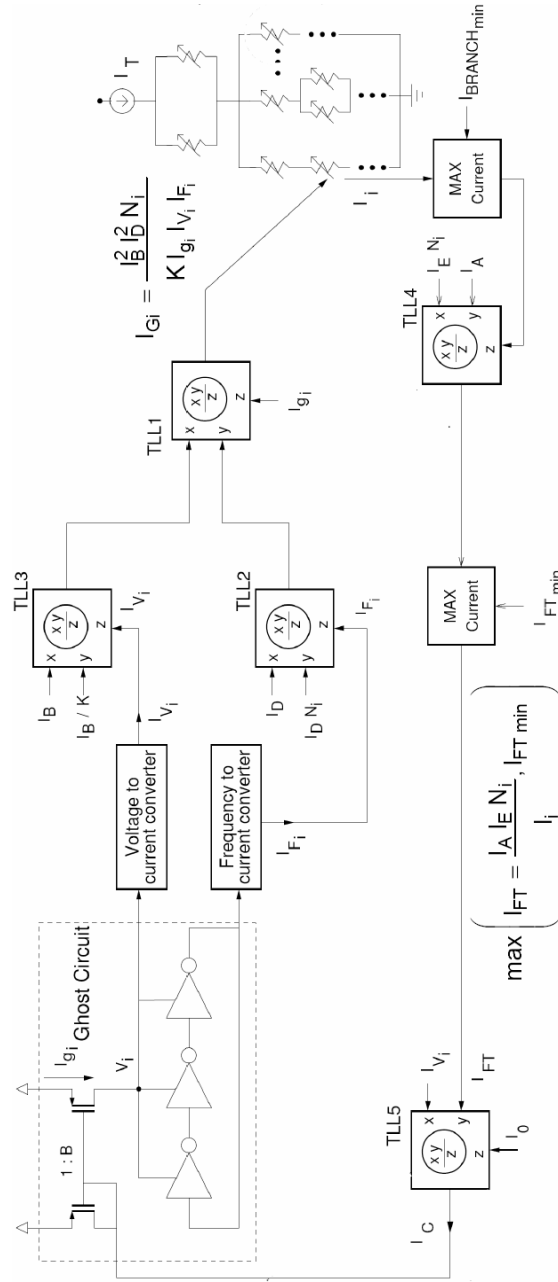


Fig. 6. Detailed representation of the constructed loop including all sub-blocks, e.g. Ghost circuit, translinear loops, voltage and frequency to current converters and the resistive-network mapped from the given task graph.

change in the workload information (N_i) influences the current corresponding to the target operation frequency (I_{FT}) in the feedback loop. Hence, the simple GC determines the supply voltage level to be applied to the PE for achieving the target frequency as well as the resulting current consumption. These values are then converted into current representations in order to calculate the pseudo-resistor controlling currents (I_{Gi}), with several translinear loops used to carry out necessary calculations as current operators, while the device conductance value also changes according to I_{Gi} . This change in the value of device conductance forces all the branch currents in the RN to be adjusted by means of KCL. As the system settles to its new operating point, the new device currents in the pseudo-resistor network are determined by KCL, dictating the optimum task duration with the prescribed supply voltage and operating frequency for each PE to minimize system-wide energy dissipation. Implications with respect to overall stability of the loops will be discussed later in Stability Analysis Section. It can be shown that the dynamic behavior of each branch control loop is governed by a single-dominant-pole transfer function, and that the entire system always converges to a stable operating point for a given set of (N_i) values. Also, note that the GC can effectively capture the actual frequency-voltage-power relationship of the PEs, including the influence of leakage power dissipation, eliminating any analytical approximation of physical behavior that is inherently prone to inaccuracies. These circuits are capable of reflecting actual operating conditions on-chip, inherently taking into account local variations of temperature, as well as process-related fluctuations of device parameters.

In this solution, the GC is driven by its supply current (I_C) rather than the supply voltage since the instantaneous operation of the oscillator is imposed by the calculated power dissipation based on the required frequency of operation ($P = fCV^2 = IV \iff I = fCV$). This is done with the assumption that the dynamic power consumption is dominant. If necessary, a static GC is added to the loop to mimic the static current consumption of the PE, proportional to the total number of gates (that may be different for different PE). Then, this current is added to the dynamic current consumption (I_{gi}).

Current-mode processing in each feedback loop is carried out by single quadrant current multiplier/dividers labelled as TLL_i . Each current operator is implemented by the simple alternating topology translinear loop (TLL) of four transistors operated in weak inversion with their bulks connected to the common substrate resulting in Eq. 19 as shown in Fig. 7(a). Thus, the entire feedback loop can be implemented with a very small number of devices, which leads to significant savings in silicon area. Here, a clockwise element (CW) is the one whose gate-to-source voltage is a voltage drop in the clockwise direction of the loop. So we shall consider a counterclockwise element (CCW) as the one whose gate-to-source voltage is a voltage increase in the clockwise direction of the loop [25].

$$I_{out} = \frac{I_x I_y}{I_z} \quad (19)$$

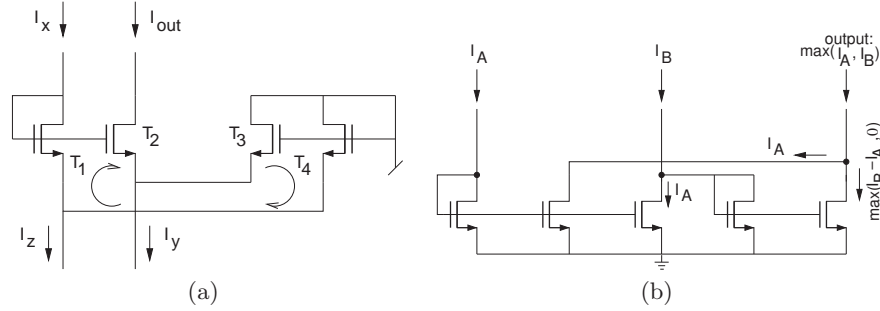


Fig. 7. (a) A subthreshold MOS translinear loop consisting of two CW transistors and two CCW transistors, constructed to operate as an inverse current multiplier, (b) A subthreshold MOS translinear loop consisting of two CW transistors and two CCW transistors, constructed to operate as an inverse current multiplier.

This is a single quadrant current multiplier/divider, thus all currents should be positive. Such multiplier/divider schemes are utilized in the feedback loop used to implement the analog optimizer for converting duration (time) to frequency (where all variables are represented as currents) and to implement the pseudo-resistor controlling current definition by means of ratio of current multiplications. The simulation results of the implemented single quadrant current multiplier/divider can be found in [26, 27].

Minimum current limiter (Maximum current selector) blocks are used to restrict the operation range to a defined value. The upper limit of the operation is intrinsically limited by the technology due to the fact that the maximum allowed core operating voltage is fixed to a constant level. In order to guarantee that the lower limit of operation, which is 1.2V or 150MHz is not violated under any circumstances two minimum current limiter blocks are used in all the feedback loops [28, 27]. For this purpose a combination of NMOS transistors is used to carry out addition/subtraction of the replicas of the two input currents as given in Fig. 7(b).

Each pseudo-resistor is realized as a single MOS transistor operating in weak inversion where the equivalent conductance value of each transistor is controlled independently by a current by means of a control transistor - thus, utilizing only a few transistors. Note that the linear pseudo-Ohm's law (Eq. 20) is valid and the network of controlled resistors remain linear with respect to currents only [23].

$$G^* = \frac{1}{R^*} = \frac{I_S}{V_0} \exp\left(\frac{V_G - V_{T0}}{2U_T}\right) \quad (20)$$

A resistor connected to ground potential in the RN corresponds to a saturated MOS transistor (operated in weak inversion) that provides a pseudo-ground in the equivalent pseudo-resistive network (refer to Fig. 8). Any current flowing to the pseudo-ground can be easily extracted without influencing the branch current ratio, by means of a grounded current mirror made of transistors com-

plementary to those of the network as shown in Fig. 8 [23]. Hence, grounded current-mirrors are used to sense each device current separately, to be further used in the feedback loop.

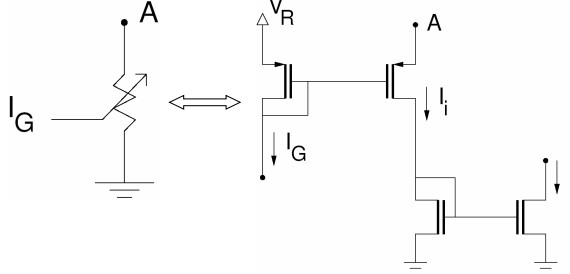


Fig. 8. Implementation of a grounded pseudo-resistors realized using CMOS process.

Figure 9 shows the simulated and the measured operation of a three-loop optimizer network which is used to model the behavior of a task graph comprising three sequential tasks. Here, the task durations (device currents) resulting in the optimum system energy dissipation are shown for various workload combinations as indicated. The workload information of three sequential tasks are shown in parenthesis for each simulation interval as (N_1, N_2, N_3) combination. The normalized workload estimations (N_i) for all tasks are updated at regular intervals of $5 \mu\text{s}$, ranging from $(2,8,4)$ in the first interval to $(12,8,8)$ in the last interval. The available time is shared among the three tasks for all workload conditions; guaranteeing timing constraints and optimizing the dissipated energy in the system by means of optimally utilizing the available time. As it can be seen from the figure, the supply voltage level for the second task ($Loop_2$) varies with respect to other task workloads condition although there has not been any change in its own workload. The corresponding supply voltage and the device current (task duration) values indicate that the proposed analog optimizer is capable of responding to varying operating conditions with fast settling times and a wide dynamic range (supply voltage variation between 1.2 and 1.74 V), dictating the optimum operating voltage and duration of all three tasks mapped on the PE for minimum system energy consumption [28, 27].

The three-loop demonstrator circuit of the proposed analog optimizer architecture has been implemented using a $0.18 \mu\text{m}$ standard digital CMOS process (Fig. 10(a)). The overall circuit area of the optimizer is $(250\mu\text{m} \times 700\mu\text{m})$ excluding decoupling capacitors, while each loop circuit occupies only $(180\mu\text{m} \times 120\mu\text{m})$. The circuit is capable of supporting the desired frequency range of 170MHz-290MHz, as well as the voltage range of 1.2V-1.8V. The average power consumption of the entire three-loop optimizer is 6.5mW [28, 27].

Figure 10(b) shows the variation of the overall energy dissipation of the system composed of three tasks, scheduled in series and mapped on a single processor - as a function of changing workload conditions, calculated from measured

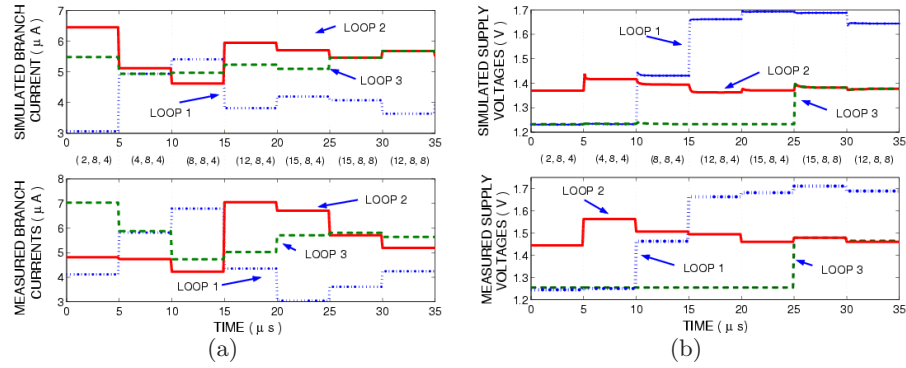


Fig. 9. Simulation and measurement results show (a) the branch currents (i.e. task duration) and (b) the corresponding supply voltages which are computed under varying workload combinations as indicated.

voltage/frequency and task duration values. To test the optimality of this solution, the device current values were slightly perturbed from their actual values (while keeping the sum constant) and the energy surface has been re-calculated. The resulting energy surface is clearly higher than the original solution for all workload combinations and for all branch current perturbations, demonstrating that the original solution indeed is the minimum energy surface.

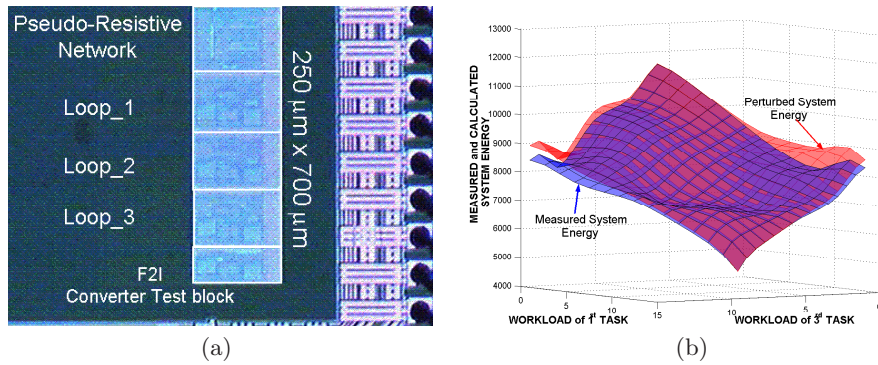


Fig. 10. (a) Chip microphotograph of the three-parallel loop optimizer. The overall circuit occupies only $250\mu\text{m} \times 700\mu\text{m}$ and (b) Energy dissipation of the system composed of three tasks, scheduled in series and mapped on a single PE.

In Table 1 the comparison of the simulated supply voltages (V), operation frequencies (MHz) and task durations (device currents- μA) of the same system are given for the proposed global optimization approach versus local energy optimization applied to each task. Note that only the workload of first task increases throughout the table. Hence, in the local optimization scheme the core supply voltage levels and operation frequency remain constant during the second and the third tasks resulting in a higher power dissipation and energy consumption in the overall system. In contrast, when using the proposed global optimization approach, any change in workload condition of any of the tasks influences all task durations corresponding to a minimization of the total system energy dissipation by optimally using the overall available time (T). The additional energy savings varies between 11% and 20% for different cases.

Table 1. The comparison of the simulated global energy optimization approach versus local optimization.

N_i	V_{dd1} (V)	V_{dd2} (V)	V_{dd3} (V)	f_1 (MHz)	f_2 (MHz)	f_3 (MHz)	d_1 (s)	d_2 (s)	d_3 (s)	P_{total} (μW)	E_{total} (μJ)
Global Energy Optimization (Proposed Approach)											
(2,8,4)	1.24	1.44	1.28	173.8	231.6	191.8	0.21	0.43	0.37	1087.6	385.3
(4,8,4)	1.25	1.56	1.28	175.5	252.7	192.8	0.33	0.34	0.33	1220.4	411.6
(8,8,4)	1.46	1.50	1.37	227.5	243.6	189.1	0.36	0.31	0.33	1370.1	455.3
(12,8,4)	1.66	1.35	1.31	273.1	242.2	191.1	0.25	0.39	0.35	1629.0	522.9
(15,8,4)	1.70	1.58	1.32	282.2	256.3	189.7	0.28	0.38	0.34	1755.9	577.7
Local Energy Optimization											
(2,8,4)	1.23	1.49	1.25	176	255	204	0.08	0.21	0.13	1160.8	486.6
(4,8,4)	1.24	1.49	1.25	204	255	204	0.13	0.21	0.13	1228.4	478.9
(8,8,4)	1.49	1.49	1.25	255	255	204	0.21	0.21	0.13	1463.7	516.1
(12,8,4)	1.62	1.49	1.25	288	255	204	0.28	0.21	0.13	1641.8	592.4
(15,8,4)	1.71	1.49	1.25	300	255	7204	0.33	0.21	0.13	1764.6	668.4

4.2 Overall System Implementation

The proposed analog optimizer determines the supply voltage level and operation frequency of all tasks that are represented in the system task graph, simultaneously. On the other hand, tasks are to be executed in their sequential order on the PEs. This means that the individual operating voltages and frequencies will have to be assigned to the PEs according to their temporal relationships. Hence, the intended system will require an interface between the analog optimizer and the PEs. A possible candidate of such an interface is shown in Fig. 11(a). Here, a separate continuous voltage, high efficiency DC/DC converter is used for each PE individually. The supply voltage levels defined by the optimizer (per task) will be applied to the PEs through these high efficiency voltage converters during the operation of the system, sequentially. The frequency of operation on the

other hand is also defined by the analog optimizer and will be used to drive the clock buffers of the PEs as indicated in the figure.

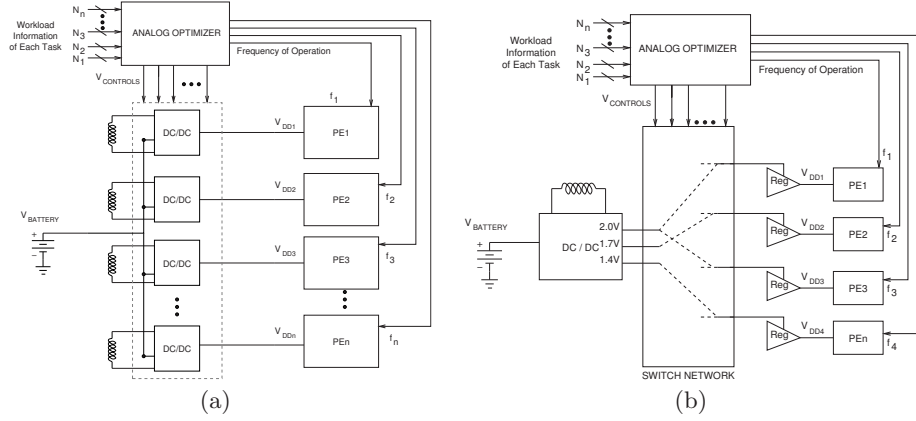


Fig. 11. (a) Block diagram representation of the system architecture in which the analog optimizer controls the individual clock frequencies and supply voltages of various PEs, and (b) Block diagram representation of the system architecture in which a single high efficiency, multiple output DC/DC converter is used to generate the supply voltage levels in a certain range.

Nevertheless, this solution could become costly due to the number of I/O pins needed for external inductors that are required to ensure the high efficiency of DC/DC converters, and silicon area (dedicated DC/DC per PE) for SoC applications employing numerous PEs. An alternative scenario for the interface between the analog optimizer and the PEs is presented in Fig. 11(b). Here, supply voltage levels defined by the analog optimizer will be applied to the PEs through voltage regulators (current efficient voltage followers) during operation. While the number of external inductors is reduced to one, it is assumed that only one DC/DC converter is utilized with three output levels (1.4V, 1.7V and 2.0V). Each output of the DC/DC converter can be used to generate the supply voltage levels in a certain range, with the help of voltage regulators, e.g. the 2.0V converter output is used to generate 1.8V - 1.51V supply voltage range. It should be noted that in this case, the energy savings obtained by utilizing the analog optimizer will be degraded due to the energy losses in the voltage regulators, by up to 33% (at the "edge" of the regulator output range). However, this drawback can be overcome by taking into account the voltage regulator supply levels in the optimization algorithm, where constant current levels can be used for I_{V_i} to represent task supply voltage levels. Hence, the final solution will still be the optimum energy dissipation for the whole system, including the regulator losses.

5 Stability Analysis

As already mentioned, the concept of system stability needs to be considered when several components adopt dynamic policies to control energy consumption and performance. Possible oscillations in energy/performance space that could be caused by applied energy management policies are undesirable, and should be avoided. In this section it will be shown that the dynamic behavior of each device control loop is governed by a single-dominant-pole transfer function, and that the entire system (the centralized optimizer unit) always converges to a stable operating point for a given set of workload (N_i) values.

In order to derive the characteristic equation of the feedback control loop, the loop is opened on the resistive network. Hence, the device current I_i is treated as the input current (variable) and the pseudo-resistor controlling current I_G is treated as the output current. Note that I_A, I_B, I_D, I_E, I_0 are constant biasing currents used in the feedback loop. From the loop dynamics, the output current I_G can be written as in Eq. 21. Note that one can show the small variations in the value of a variable as $(X + x)$, where lower case represents the variations in the value of the variable. Using this definition, the output current can be written as given in Eq. 22.

$$I_G = \frac{I_B^2 I_D^2 N}{K I_g I_V I_F} \text{ or } I_G = \frac{\text{Constant}}{I_g I_V I_F} \quad (21)$$

$$\begin{aligned} I_G \left(1 + \frac{i_g}{I_g} \right) &= \frac{\text{Constant}}{I_g I_V I_F \left(1 + \frac{i_g}{I_g} \right) \left(1 + \frac{i_V}{I_V} \right) \left(1 + \frac{i_F}{I_F} \right)} \\ &= \frac{\text{Constant}}{I_g I_V I_F} \left(1 - \frac{i_g}{I_g} - \frac{i_V}{I_V} - \frac{i_F}{I_F} \right) \end{aligned} \quad (22)$$

Note that we can express the ratios of the current representations of the ghost circuit supply voltage and the operating frequency as well as the current consumption of the ghost circuit and their variations in terms of the ratio of the input current and its variation as given in Eq. 23.

$$\frac{i_g}{I_g} = -2 \frac{i_i}{I_i} \quad \text{and} \quad \frac{i_V}{I_V} = -\frac{i_i}{I_i} \quad \text{and} \quad \frac{i_F}{I_F} = -\frac{i_i}{I_i (1 + s\tau)} \quad (23)$$

Finally, the device conductance controlling current being the output current and the related device current being the input current the small signal behavior of the feedback loop can be written as given in Eq. 24, since device conductances are linearly proportional to their controlling current I_G . Hence, it is shown that the dynamic behavior of each branch feedback loop is governed by a single-dominant-pole transfer function.

$$\frac{g_i}{G_i} = \frac{i_G}{I_G} = \frac{i_i}{I_i} \left(3 + \frac{1}{1 + s\tau} \right) \quad (24)$$

Now, consider a RN consisting of three parallel branches to illustrate the stability properties of the system. If we write the first branch current in the RN comprising three parallel branches in terms of the RN biasing current and the other branch currents, we get Eq. 25, where G_i and g_i represents the device conductance and the variations in the conductance value respectively. If we replace each $(G_i/\sum G_i)$ quantity in Eq. 25 by (I_i/I_T) , and substituting Eq. 24 where ever suitable, we can finally write Eq. 26. After performing the necessary mathematical operations on the resulting equation set, the characteristic equation of the system can be expressed as Eq. 27.

$$I_1 \left(1 + \frac{i_1}{I_1}\right) = I_T \left(1 + \frac{i_T}{I_T}\right) \frac{G_1 \left(1 + \frac{g_1}{G_1}\right)}{(G_1 + G_2 + G_3) \left(1 + \frac{g_1 + g_2 + g_3}{G_1 + G_2 + G_3}\right)} \quad (25)$$

$$\frac{i_1}{I_1} = \frac{i_T}{I_T} + \frac{g_1}{G_1} \left(1 - \frac{I_1}{I_T}\right) - \frac{I_2}{I_T} \frac{g_2}{G_2} - \frac{I_3}{I_T} \frac{g_3}{G_3} \quad (26)$$

$$4\tau_1\tau_2\tau_3s^3 + [6(\tau_1\tau_2 + \tau_2\tau_3 + \tau_1\tau_3) - 2(R_1\tau_2\tau_3 + R_2\tau_1\tau_3 + R_3\tau_1\tau_2)]s^2 + [6(\tau_1 + \tau_2 + \tau_3) + 3(R_1\tau_1 + R_2\tau_2 + R_3\tau_3)]s + 9 = 0 \quad (27)$$

$$a_3s^3 + a_2s^2 + a_1s + a_0 = 0 \quad (28)$$

$$\text{where } R_i = \frac{I_i}{I_T} \quad \text{thus } \forall \sum_i R_i = 1$$

Now, if we rewrite the characteristic equation of the system as in Eq. 28, we can check the stability of the system by applying the Routh criterion. The principal stability criterion for linear systems states that a system is stable if all poles of its transfer function lie in the left-half of the complex s-plane. Equivalently, a system is stable if the real parts of all roots of its characteristic equation are negative. Note that a root of the characteristic equation is synonymous with a system pole. To apply Routh's criterion, the Routh's Table should be created. The Routh criterion is applied by examining the sign of the coefficient in the column headed by a_3 . The number of sign changes in the elements of this column, taken in order, is equal to the number of roots of the characteristic equation that have positive real parts. Hence, in order to show that the system is stable we should verify that the sign of the expression $a_1a_2 - a_0a_3$ is positive, since all the other components of the first column of the routh table are positive quantities. Note that all R_i and τ_i quantities are positive real values. Thus a_3 , a_1 and a_0 are intrinsically positive for all R_i or τ_i values. Note that a_2 is also always positive for all R_i or τ_i values, since definition of R_i ($I_i/I_T = R_i$) guarantees that the multiplicative factor $(6 - 2R_i)$ in a_2 is always positive. After performing

all the necessary multiplications it is proved that the sign of the mathematical operation $a_1a_2 - a_0a_3$ is always positive, guaranteeing that the proposed system is stable.

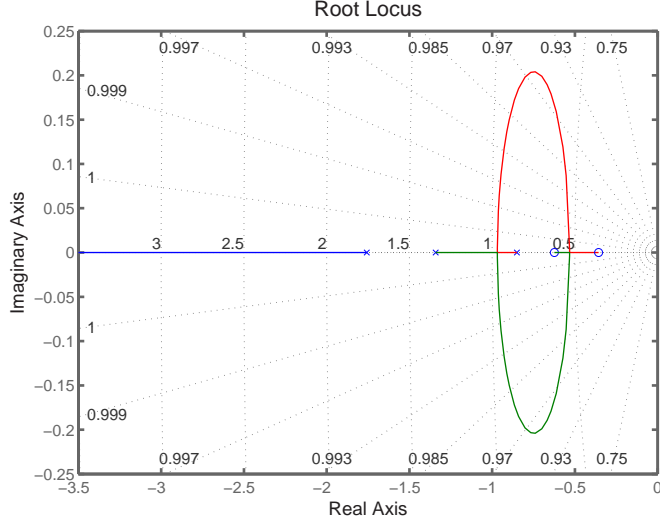


Fig. 12. The root locus plot of the system with three parallel branches for typical R_i and τ_i values.

Figure 12 shows the root locus plot for the system with the characteristic equation given in Eq. 27, displaying the closed-loop pole trajectories as a function of the feedback gain. Root loci are used to study the effects of varying feedback gains on closed-loop pole locations. In turn, these locations provide indirect information on the time and frequency responses. As can be seen from the figure all three roots of the systems lies on the left (negative) half plane, demonstrating that the system is stable. Note that the system has two negative zeros in addition to its three poles as indicated within the Fig. 12. The stability of the system is analyzed thoroughly by using MATLAB software tool, where R_i (satisfying the $\sum R_i = 1$ constraint) and τ_i values are randomly selected with 10% mismatch introduced. The graphic is obtained by sweeping the loop gain in the possible range.

6 Conclusion

In this chapter, the energy optimization problem in multi-core applications is addressed with a unique analog implementation approach. The analogy that exists between the energy minimization problem under timing constraints in a general TG and the power minimization problem under Kirchhoff's current law

constraints in an equivalent RN is exploited. The principles of mapping an arbitrary task graph to an equivalent resistive network are presented. A fully analog, current-based solution to implement on-line energy minimization in complex multi-core systems under varying workload conditions is demonstrated, which achieves significant overall energy savings compared to the local energy minimization approach.

References

- [1] M. Sirvastava, "Power-aware design energy consumers & sources, reduction & management." UCLA, EE202A Fall 2002 Lecture notes, 7 and 8, 2002, University of California, LA.
- [2] A. Chandrakasan, R. Min, and M. Bhardwaj, "Power aware wireless microsensor systems," in *Proceedings of the 28th European Solid-State Circuits Conference*, p. In Keynote Paper, 2002.
- [3] D. M. Monticelli, "Taking a system approach to energy management," in *Proceedings of the 29th European Solid-State Circuits Conference*, vol. 1, (Estoril, Portugal), pp. 15–19, Sep. 2003.
- [4] A. Chandrakasan, V. Gutnik, and T. Xanthopoulos, "Data driven signal processing: an approach for energy efficient computing," in *Proceedings of the International Symposium on Low Power Electronics and Design*, (Monterey, California, United States), pp. 347–352, IEEE Press, 1996.
- [5] K. Suzuki, S. Mita, T. Fujita, F. Yamane, F. Sano, A. Chiba, T. Maeda, Y. Watanabe, K. Matsuda, T. Kuroda, and T. Sakurai, "A 300 MIPS/W RISC core processor with variable supply-voltage scheme in variable threshold-voltage CMOS," in *Proceedings of IEEE Custom Integrated Circuits Conference*, pp. 587–590, May 1997.
- [6] K. Usami, M. Igarashi, T. Ishikawa, M. Kanazawa, M. Takahashi, M. Hamada, H. Arakida, T. Terazawa, and T. Kuroda, "Design methodology of ultra low-power MPEG4 codec core exploiting voltage scaling techniques," in *Proceedings of the 35th Design Automation Conference*, (San Francisco, CA), pp. 483–488, June 1998.
- [7] K. Flautner, D. Flynn, and M. Rives, "A combined hardware-software approach for low-power SoCs: Applying adaptive voltage scaling and intelligent energy management software," in *Proceedings of System-on-Chip and ASIC Design Conference, DesignCon*, Jan. 2003.
- [8] C. Poirier, R. McGowen, C. Bostak, and S. Naffziger, "Power and temperature control on a 90nm *Itanium*[®]-Family Processor," in *Proceedings of IEEE International Solid-State Circuits Conference, Digest of Technical Paper*, vol. 2, pp. 304–305, Feb. 2005.
- [9] X. Fan, C. S. Ellis, and A. R. Lebeck, "The synergy between power-aware memory systems and processor voltage scaling," in *Power-Aware Computer Systems, 3rd International Workshop*, pp. 164–179, Dec. 2003.
- [10] D. B. Kirk, *Accurate and Precise Computation using Analog VLSI, with Applications to Computer Graphics and Neural Networks*. PhD thesis, California Institute of Technology, Pasadena, California, March 1993.
- [11] G. Cauwenberghs, "A fast stochastic error-decent algorithm for supervised learning and optimization," in *Advances in Neural Information Processing Systems*, vol. 5, pp. 244–251, 1993.

- [12] L. O. Chua and L. Gui-Nian, "Nonlinear programming without computation," in *IEEE Transactions on Circuits and Systems II*, vol. CAS-31, pp. 182–188, Feb. 1984.
- [13] A. Dembo and T. Kailath, "Model free distributed learning," in *IEEE Transactions on Neural Networks*, vol. 1, pp. 58–70, 1990.
- [14] R. Gregorian and G. C. Temes, *Analog MOS Integrated Circuits for Signal Processing*. New York: John Wiley and Sons, 1986.
- [15] Technical Report, Transmeta Corporation, <http://www.transmeta.com>.
- [16] S. Lee and T. Sakurai, "Run-time power control scheme using software feedback loop for low-power real-time application," in *Proceedings of the Asia and South Pacific Design Automation Conference*, (Yokohama, Japan), pp. 381–386, Jan. 2000.
- [17] S. Lee and T. Sakurai, "Run-time voltage hopping for low-power real-time systems," in *Proceedings of the 37th Design Automation Conference*, (Los Angeles, California, United States), pp. 806–809, June 2000.
- [18] A. Andrei, M. Schmitz, P. Eles, Z. Peng, and B. M. Al-Hashimi, "Overhead-conscious voltage selection for dynamic and leakage energy reduction of time-constrained systems," in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, (Paris, France), pp. 105–118, Feb. 2004.
- [19] B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner, "Theoretical and practical limits of dynamic voltage scaling," in *Proceedings of the 41st Design Automation Conference*, (San Diego, CA, USA), pp. 868–873, 2004.
- [20] S. M. Martin, K. Flautner, T. Mudge, and D. Blaauw, "Combined dynamic voltage scaling and adaptive body biasing for lower power microprocessors under dynamic workloads," in *Proceedings of the International Conference on Computer Aided Design*, (San Jose, California), pp. 721–725, Nov. 2002.
- [21] J. Liu, P. H. Chou, and N. Bagherzadeh, "Communication speed selection for embedded systems with networked voltage-scalable processors," in *Proceedings of the 10th International Workshop on Hardware/Software Codesign*, (Estes Park, Colorado), pp. 169–174, May 2002.
- [22] Y. Zhang, X. Hu, and D. Chen, "Task scheduling and voltage selection for energy minimization," in *Proceedings of the 39th Design Automation Conference*, (New Orleans, LA, USA), pp. 183–188, June 2002.
- [23] E. A. Vittoz, "Pseudo-resistive networks and their applications to analog collective computation," in *Proceedings of the 7th International Conference on Artificial Neural Networks*, (Lausanne, Switzerland), pp. 1133–1150, Oct. 1997.
- [24] J. C. Maxwell, *A Treatise in Electricity and Magnetism*, vol. 1. New York, USA: Dover Publications, Inc., third ed., 1954.
- [25] E. Vittoz, "Analog VLSI for collective computation," in *Proceedings of the 5th IEEE International Conference on Electronics, Circuits and Systems*, vol. 2, pp. 3–6, Sep. 1998.
- [26] Z. T. Deniz, Y. Leblebici, and E. Vittoz, "Configurable on-line global energy optimization in multi-core embedded systems using principles of analog computation," in *Proceedings of 2006 IFIP International Conference on Very Large Scale Integration (VLSI-SoC)*, pp. 379–384, Oct. 2006.
- [27] Z. T. Deniz, *Multi-Unit Global Energy Management and Optimization for Network-on-Chip Applications*. PhD thesis, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, February 2006.
- [28] Z. T. Deniz, Y. Leblebici, and E. Vittoz, "On-line global energy optimization in multi-core systems using principles of analog computation," *IEEE Journal of Solid-State Circuits*, accepted for publication in July 2007 issue.