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

Zeynep Toprak Deniz, Yusuf Leblebici, Eric Vittoz
Ecole Polytechnique Féderale de Lausanne
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.

## Keywords

On-line energy optimization, Dynamic Voltage and Frequency Scaling, System-on-Chip, energy management.

## 1. INTRODUCTION

Due to the recent developments in the embedded systems technologies, mobile wireless System-on-Chip (SoC) architectures and other "computing-in-the-small" devices have become very popular and virtually ubiquitous. Such systems are widely used in many applications such as mobile computing, information alliances, as well as various industrial, military and medical applications.

The significance of the energy management problem is underlined by the increasing prominence of multi-core systems that must operate under strict energy budget constraints in mobile applications. In multi-PE systems, 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. This introduces slack times during which the system can reduce its performance to save energy. The key to energy-efficient designs is the ability to tune PE performance to the non-uniform workload.

In cases where performance requirements of a component vary significantly during the active operation regime, dynamic voltage scaling (DVS) is the preferred approach for trading energy dissipation versus performance. DVS is based on reducing the performance level of the component during periods of low utilization so that the task is always completed just-in-time, consuming minimum energy. While the *local* energy dissipation of each PE can be minimized using DVS techniques based on workload predictions, it can be shown that these *local* minima usually do not represent the *global* energy minimum, which can only be reached by considering the relative timing dependencies of all tasks running in the system. This problem of minimizing the overall dissipated energy in a multiple-PE system under timing constraints, and subject to DVS has already been formulated in a rigorous fashion, yet a compact real-time solution methodology has not been offered [1,2,3,4].

Our approach borrows from the basic principles of analog computation to continuously optimize the system-wide energy dissipation of multiple-PEs, converging on the global minima of the constrained optimization problem which are represented as stable operating points of a simple resistive network (RN). The input set of the circuit consists of individual workload estimates for each task and for each PE, while the output consists of assigned supply voltage/frequency values for each PE as well as the allocated time duration for each task as illustrated in Fig. 1.

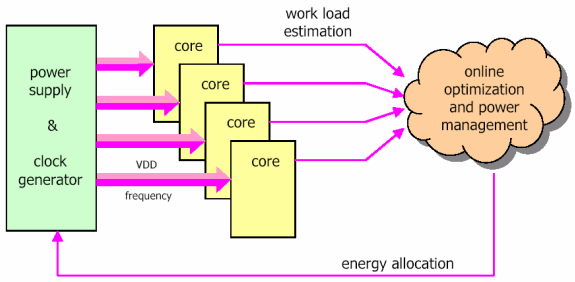



**Figure 1. Block diagram representation of the proposed on-line global energy management unit.**

The remainder of this paper is organized as follows. In Section 2 we concentrate on demonstrating an on-line solution to complex multi-variable energy optimization problem. The closed loop operation principle of the proposed analog optimizer block is described in Section 3. In Sections 4 and 5 the system-level implementation is discussed. Conclusions are provided in Section 6.

## 2. FROM TASK GRAPH TO RESISTIVE NETWORK

In the following, we will assume that the task graph (TG) of the given application 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 and communications between tasks take place. Figure 2(a) shows a simple yet good example for such a case, where each parallel branch represents a PE, each node represents the mapped and scheduled tasks on PEs, and edges represent the

communication information between tasks. Note that, each edge $e(t_u,t_v)$ dictates that the task $t_v$ can only start after the task $t_u$ finishes.

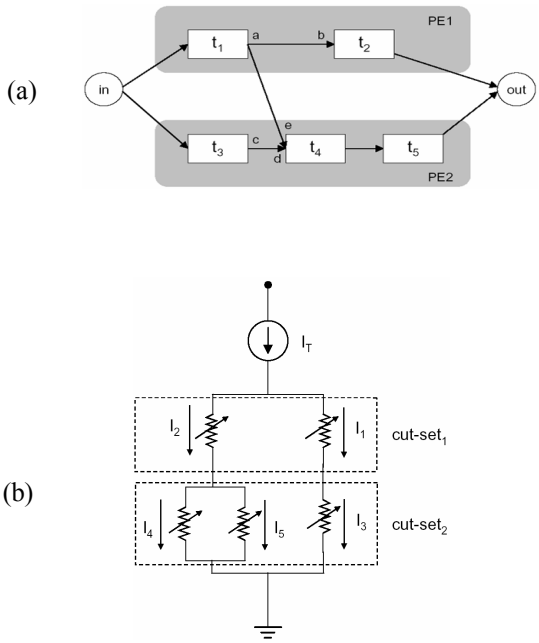



**Figure 2. (a) Task graph of five tasks mapped on two processing elements, and (b) resistive network equivalent of the given TG.**

Notice that the points *a* through *e* (labeled on the TG for the sake of easy identification) actually represent the same instant in time. This fact indicates that $t_1$ and $t_3$ have to be finished at the same time for completing the work just-in-time, corresponding to minimum energy consumption. Furthermore, as a basic consequence of parallelism, 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.

$$\begin{aligned} &\min E_{total} = \min \sum_u E_u = \min \sum_u P_u\, d_u \\ &\text{subject to } \sum_u d_u = T \end{aligned} \qquad (1)$$

Let $E_{total}$ be the overall energy dissipated in the given TG, $d_u$ the duration of each task, $P_u$ the power consumption during the task it is associated with and finally *T* the overall available time for all tasks in the TG to be performed (TG period). Recall that, in order to avoid any functionality errors or performance degradation in the system, all tasks in the given TG should be finished within *T*. The total dissipated energy in the system can be written as the summation of the all task energies (see Eq. 1). Hence, the problem of minimizing the overall dissipated energy in a given system, represented with its TG under timing constraints, is formulated as provided in (1). The formal algorithmic solution of (1) is certainly possible in real time, yet the computational overhead that is needed may become prohibitive especially when taking into account realistic timing/delay models and secondary effects such as leakage dissipation.

At this point, we surmise that the equivalent RN of the given TG in Fig. 2(a), consists of five controlled resistors in the configuration as shown in Fig. 2(b), where the network is supplied with a constant current $I_T$. Intrinsically, the resistive-network will consume the *lowest possible power* $P_{total}$, at steady-state for a given driving current according to Maxwell's Heat theorem [5]. Due to KCL, $I_T$ will be split into parallel branch currents ($I_i$) that are proportional to branch conductances ($G_i$). Hence, it can be seen that the simple RN actually realizes the solution to the dissipated power minimization problem; under KCL constraints (see Eq. 2).

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

The comparison between (1) and (2) 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 RN under KCL constraints*. Recall that, the two tasks $t_1$ and $t_3$, must be executed in parallel on two different PEs, i.e. these two tasks must have the same duration ($d_1 = d_3$). Similarly, in a RN branch consisting of two series connected resistors, each resistor must carry the same amount of branch current. Based on this analogy all parallel tasks are converted into series-connected branches in the equivalent RN. However, tasks mapped in series on a single PE can only be executed sequentially in time. Note that, $t_2$ is executed on the first PE in parallel to the two sequential tasks on the second PE. As a consequence of parallelism, the amount of time necessary for the execution of task $t_2$ will be split among $t_4$ and $t_5$ ($d_2 = d_4 + d_5$) according to the actual workload of these two tasks. Similarly, in a RN branch of parallel connected resistors the main branch current will be shared proportionally between the parallel branches according to KCL. Hence, all sequential tasks are represented by parallel-connected branches in the equivalent resistive network. Consequently, the problem of minimizing the sum of all task energies in a certain application is mapped onto an equivalent RN (see Fig. 2) consisting of controlled (pseudo-) resistors. Note that each task sequence (or sub-sequence) in the TG corresponds to a parallel section (or sub-section) in the RN where KCL is valid, and is represented by a corresponding cut-set.

The equivalence between the two analogous minimization problems is illustrated in Table 1. Here, the time durations ($d_u$) allocated to each task are represented by the device currents ($I_i$), and the total task graph period (T) is represented by the constant current source ($I_T$). Individual tasks are modelled with branch conductances ($G_i$), controlled by ($d_u / P_u$). Note that the summation of device currents, i.e. elements of the $k^{th}$ cut-set is always equal to the RN driving current ($I_T$). Similarly, summation of the

task durations that corresponds to the $j^{th}$ task sequence is equal to the TG period (T).

**Table 1. Analogy between the two optimization problems.**

$$\min\left(E_{total}=\sum_u E_u\right) \Leftrightarrow \min\left(\sum P_{total}=\sum_i P_i\right)$$
$$\text{subject to}\sum_u d_u = T \Leftrightarrow \text{subject to}\sum_i I_i = I_T$$
$$\forall \text{ task sequence "j"} \quad \forall \text{ cut}-\text{set "k"}$$
$$\text{where } d_u \Leftrightarrow I_i$$
$$\frac{d_u}{P_u} \Leftrightarrow G_i$$

Although the applied mapping scheme resembles to the problem of finding the *dual* of a given TG, it is important to emphasize that the mapping of a given TG to its equivalent RN is based on converting the time domain relation between tasks into equivalent RN currents. Hence, we do not consider this procedure to be equivalent to finding the dual of a given task graph.

## 3. CLOSED-LOOP OPERATION OF THE RESISTIVE NETWORK

Figure. 3 shows the simplified block diagram implementation of the feedback loop for one branch 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 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 branch conductance value also changes according to $I_{Gi}$. This change in the value of branch 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 branch 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. 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 ($P = fCV^2 = IV \Leftrightarrow I = fCV$). 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 ($I_{gi}$).

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 [6]. Similarly, weak-inversion based translinear loops (TLL) are used as single quadrant current multipliers/dividers to compute the controlling variables of the loops where each TLL consists of only 4 transistors. Thus, the entire feedback loop can be implemented with a very small number of devices, which leads to significant savings in silicon area and power dissipation.

Figure 4 shows the simulated 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 (branch 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 (Ni) for all tasks are updated at regular intervals of 5 μs, 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. Any change in the duration will naturally force operation frequency and hence supply voltage changes, providing minimum possible energy dissipation under the new circumstances for the whole system. 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 branch 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.

Figure 5 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

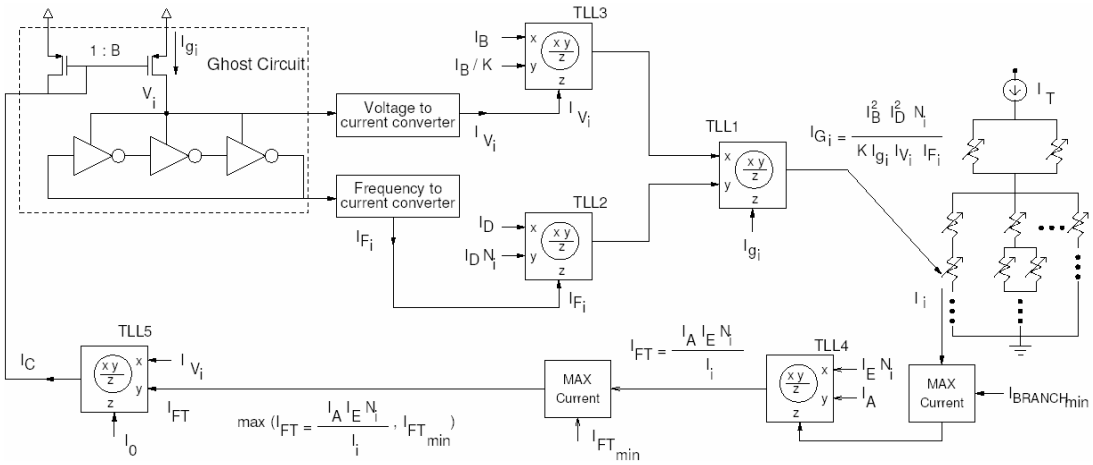



**Figure 3. Block diagram implementation of the optimization feedback loop.**

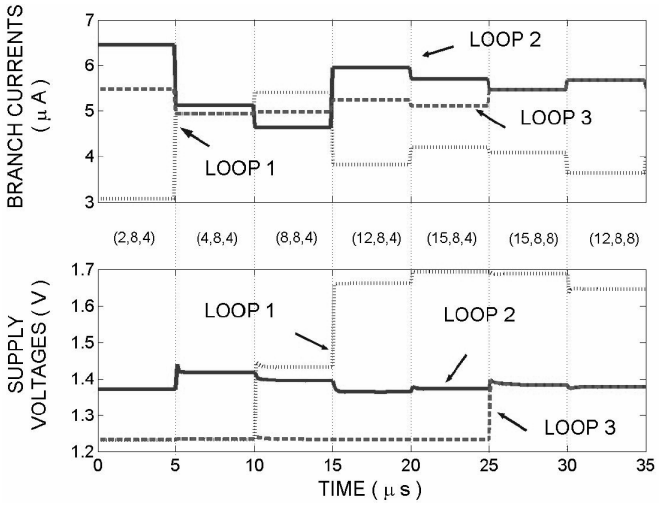



**Figure 4. Simulation results show the branch currents (i.e. task duration) and the corresponding supply voltages which are computed under varying workload combinations as indicated.**

changing workload conditions, calculated from measured voltage/frequency and task duration values. To test the optimality of this solution, the branch 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.

In Table 2 the comparison of the simulated supply voltages (V), operation frequencies (MHz) and task durations (branch 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 (hence, supply voltages and operation frequency) corresponding to a minimization of the total system energy dissipation by optimally using the overall available time (T). The additional energy savings is larger than 11% in the worst case.

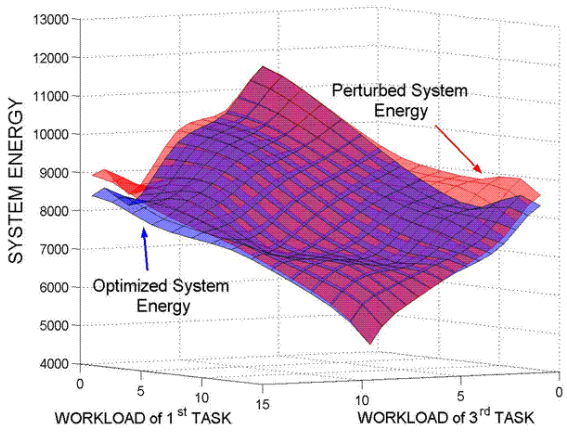



**Figure 5. Energy dissipation of the system composed of three tasks, scheduled in series and mapped on a single PE.**

## 4. CONFIGURABLE PSEUDO-RESISTOR ARRAY

Typically, a large number of diverse applications can be mapped and run on high-performance distributed embedded systems (SoC/NoC). Hence, the proposed circuit architecture should be built with a modular approach to support different software applications as opposed to a hardwired circuit solution. This capability of a modular architecture can be exploited by implementing an array of pseudo-resistors with corresponding control feedback loops (CFL) and various number of necessary type of current mirrors to pick up the branch currents at pseudo-ground nodes of RN as shown in Fig. 6. In addition to these modular building blocks several constant current sources ( $I_{Ti}$ ) can be implemented to model various task graph periods. It should be noted that the switching network is not shown in the figure for the sake of simplicity. Consequently, the implemented array of

**Table 2. The comparison of the simulated global energy optimization approach versus local optimization. The simulated task graph comprises three sequential tasks with varying workload.**

| Workload | $Vdd_1$ (V) | $Vdd_2$ (V) | $Vdd_3$ (V) | $f_1$ (MHz) | $f_2$ (MHz) | $f_3$ (MHz) | $d_1$ (s) | $d_2$ (s) | $d_3$ (s) | $P_{TOTAL}$ (uW) | $E_{TOTAL}$ (uJ) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 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 | 1087.67 | 385.36 |
| (4, 8 , 4) | 1.25 | 1.56 | 1.28 | 175.5 | 252.7 | 192.8 | 0.33 | 0.34 | 0.33 | 1220.49 | 411.64 |
| (8, 8, 4) | 1.46 | 1.50 | 1.37 | 227.5 | 243.6 | 189.1 | 0.36 | 0.31 | 0.33 | 1370.14 | 455.36 |
| (12, 8, 4) | 1.66 | 1.35 | 1.31 | 273.1 | 242.2 | 191.1 | 0.25 | 0.39 | 0.35 | 1629.06 | 522.96 |
| (15, 8, 4) | 1.70 | 1.58 | 1.32 | 282.2 | 256.3 | 189.7 | 0.28 | 0.38 | 0.34 | 1755.98 | 577.72 |
| | Local Energy Optimization | | | | | | | | | | |
| (2, 8, 4) | 1.23 | 1.49 | 1.248 | 176 | 255 | 204 | 0.08 | 0.21 | 0.13 | 1160.89 | 486.64 |
| (4, 8 , 4) | 1.24 | 1.49 | 1.25 | 204 | 255 | 204 | 0.13 | 0.21 | 0.13 | 1228.44 | 478.97 |
| (8, 8, 4) | 1.49 | 1.49 | 1.25 | 255 | 255 | 204 | 0.21 | 0.21 | 0.13 | 1463.70 | 516.13 |
| (12, 8, 4) | 1.62 | 1.49 | 1.25 | 288 | 255 | 204 | 0.28 | 0.21 | 0.13 | 1641.88 | 592.35 |
| (15, 8, 4) | 1.71 | 1.49 | 1.25 | 300 | 255 | 204 | 0.33 | 0.21 | 0.13 | 1764.67 | 668.41 |

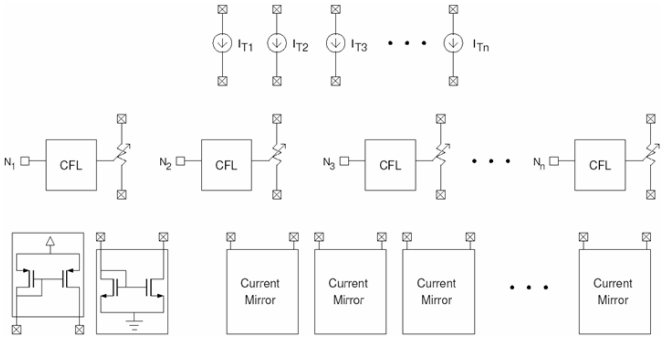
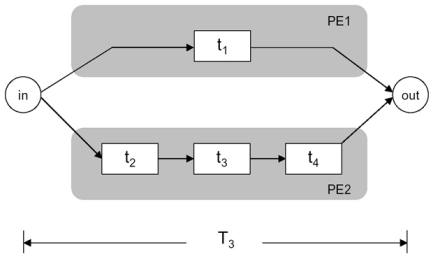



**Figure 6. Modular system approach supporting different task graphs topologies.**

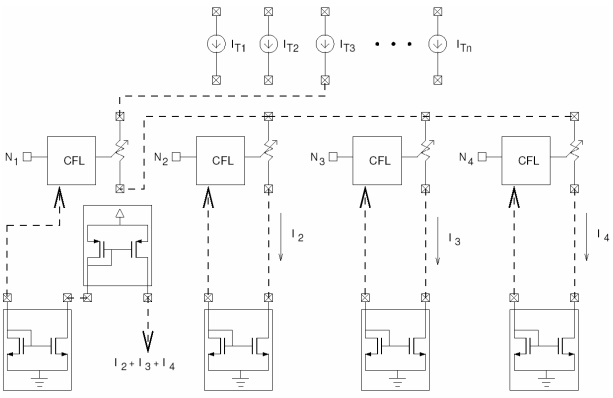
pseudo-resistors can be easily expanded to support any arbitrary TG that can be mapped on the given system of PEs.

Recall that branch currents model the corresponding task durations. Hence, each branch current should be picked up and fed back to the related CFL. Besides, extracting currents flowing to the pseudo-ground is preferred in order not to influence the branch currents. Hence, it is favored to configure the RN in such a way that in any constructed architecture the maximum number of parallel branch currents flow to the pseudo-ground nodes of the RN.

A simple example of such a modular configuration, based on built in pseudo-resistor array, is shown in Fig. 7(b) for the TG of Fig. 7(a). Here, the connections necessary for the given configuration are indicated as dashed lines and for the sake of easy identification only the necessary sub-blocks used for the given configuration are shown. In the given TG, $t_1$ on the first PE is executed in parallel to sequential tasks $t_2$, $t_3$ and $t_4$ on the second PE. Therefore, as a consequence of parallelism, the available time $T_3$ will be split among $t_2$, $t_3$ and $t_4$ tailored to their instantaneous workload, where $t_1$ can be executed during the task graph period ($T_3$).




(a)




(b)

**Figure 7. (a) Task graph of four tasks mapped on two processing elements, (b) resistive network representation of the given task graph.**

Consequently, the pseudo-resistors modeling the tasks mapped on the second PE are connected in parallel and the resulting RN is connected in series to the pseudo-resistor modeling the first task as shown in Fig. 7(b). Note that, the parallel section of the equivalent RN is connected to the pseudo-ground node.

## 5. 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. 8. 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.

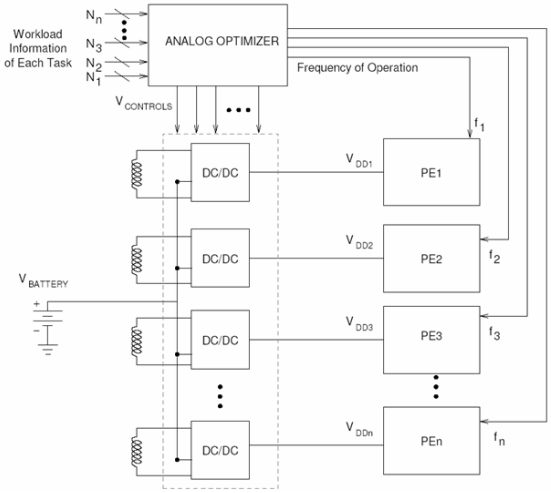



**Figure 8. Block diagram representation of the system architecture in which analog optimizer controls the individual clock frequencies and supply voltages of various PEs.**

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. 9. 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 $I_{Vi}$ 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.

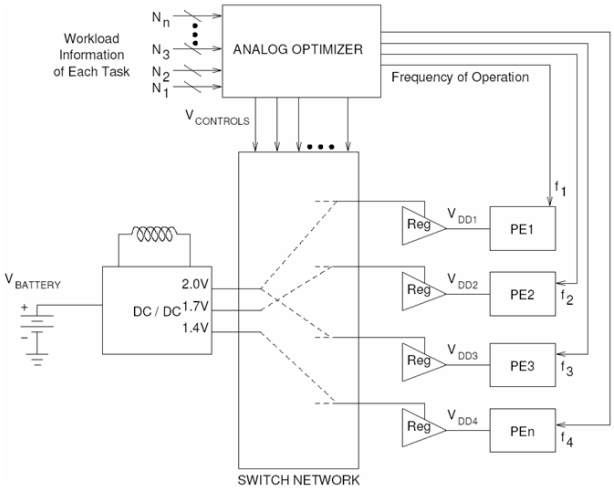



**Figure 9. Block diagram representation of the system architecture in which a single high efficieny, multiple output DC/DC converter is used to generate the supply voltage levels in a certain range.**

## 6. CONCLUSIONS

In this work, the energy optimization problem in SoC/NoC applications is discussed 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.

## 7. REFERENCES


[1] Y. Zhang, X.S. Hu, D.Z. Chen, Energy minimization of real-time tasks on variable voltage processors with transition energy overhead, ASPDAC, pp.65-70, 2003.

[2] A. Andrei, M. Schmitz, P. Eles, Z. Peng, B.M. Al-Hashimi, Overhead-Conscious Voltage Selection for Dynamic and Leakage Energy Reduction of Time-Constrained Systems, DATE 2004, pp.105-118, Paris- France.

[3] M. Schmitz, B. Al-Hashimi, P. Eles, Energy-Efficient Mapping and Scheduling for DVS Enabled Distributed Embedded Systems, DATE 2002, pp.514-521, Paris – France.

[4] Y. Zhang, X. Hu, D. Chen, Task scheduling and voltage selection for energy minimization, DAC 2002, pp.183-188, New Orleans - USA.

[5] J.C. Maxwell, *A Treatise in Electricity and Magnetism*, Vol.1, 3rd ed., 1954, Dover Publications.

[6] E.A. Vittoz, *Analog VLSI for collective computation*, IEEE International Conference on Electronics, Circuits and Systems, pp.3-6, 1998.