INTEGRATION, the VLSI journal 48 (2015) 21–35

Contents lists available at ScienceDirect

INTEGRATION, the VLSI journal

journal homepage: www.elsevier.com/locate/vlsi

Critical path-aware voltage island partitioning and floorplanning for hard real-time embedded systems

Aminollah Mahabadi [a,d], Ahmad Khonsari [b,d], Behnam Khodabandeloo [b,d,*], Hamid Noori [c], Alireza Majidi [b,d]

[a] Electrical and Engineering Department, Shahed University, Tehran, Iran
[b] School of ECE, College of Engineering, University of Tehran, Tehran, Iran
[c] Computer Engineering Department, Ferdowsi University of Mashhad, Mashhad, Iran
[d] School of Computer Science, IPM, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran

ARTICLE INFO

Article history:
Received 27 July 2013
Received in revised form
28 May 2014
Accepted 29 May 2014
Available online 17 June 2014

Keywords:
Floorplanning
Temperature
Critical path-aware
Mixed integer linear programming (MILP)
Multiple supply voltage (MSV)
Embedded systems

ABSTRACT

Temperature and power are two major issues for multiple supply voltage (MSV)-aware embedded systems that due to their different physical behavior are required to be considered together in the system design especially in applications with hard real-time constraints. In such applications critical path characteristics of a task graph play a key role in finding an MSV-aware floorplan that attempts to optimize temperature and power simultaneously. In this paper, we propose a multi-objective optimization framework to find an MSV-aware floorplan that satisfies these objectives simultaneously in the embedded system design process. This framework is based on integer linear programming (ILP) formulation which is further enhanced with a simulated annealing technique to reduce the complexity of the problem and thus execution time of it. As a trade-off between accuracy and execution time, a heuristic algorithm is also presented for scenarios with rather large design space where finding the optimal solution or Pareto optimal set is a formidable task and time consuming. The experimental results show that the proposed framework suggests floorplans that are more power-efficient compared to the cases that only attempt to optimize the temperature and attains lower temperature compared to the cases that only optimize the power. These results confirm the effectiveness of the proposed approach. Moreover, an interesting and counter-intuitive finding is that by increasing the supply voltage magnitude of the MSV-chip the total power and peak temperature not only do not increase but also decrease in some scenarios. This is due to the impacts of the critical paths of the application graph.

1. Introduction

Aggressive process technology scaling enables designers to integrate more transistors into a single chip to achieve high performance designs. Lower supply voltage is used to mitigate the increased power consumption of the increased number of transistors. However, the decrease of the supply voltage cannot pace with the increased number of transistors, which inclusively results in higher power consumption in these contemporary devices. Higher power density results in higher temperature of the device with many adversarial effects. This high temperature is a challenging issue that affects many parameters of the system design such as leakage power, lifetime of the device and delay [1].

There is a positive feedback loop between temperature and leakage power. The leakage power is proportional to the square of the operating temperature and grows exponentially with temperature [2]. For example when temperature changes from 65 °C to 110 °C, the leakage power increases as much as 38% [2]. In deep submicron technology, leakage power becomes more important and dominates the dynamic power and comprise up to 70% of the total power [3]. With respect to the life time of the device it has been shown that a small change of temperature from 10 °C to 15 °C in the operating temperature may result in a $\sim 2\times$ shorter lifespan of a device [4]. Finally it has been shown that every 20 °C increase in the temperature causes 5–6% increase in interconnects delay for Elmore's delay model [4].

Characteristics of the applications are required to be considered in floorplan design phase of the embedded systems in order to bring lower the temperature and power. Temperature, which is sluggish to variations over space, and power that is prompt to changes from block to block are two major issues for MSV-aware embedded systems that are required to be considered in the system design especially in applications with hard real-time

* Corresponding author.

http://dx.doi.org/10.1016/j.vlsi.2014.05.002
0167-9260/

constraints. E.g. in applications with hard real time constraints, critical path characteristics of a task graph play a key role in the obtained solution of the MSV-aware floorplan that attempts to optimize temperature and power simultaneously. MSV or voltage island (VI) partitioning is an effective method for power reduction of a chip while meeting the chip performance at given voltage levels [5]. The most important problem in MSV design, which controls the power consumption, is assigning suitable voltages to different modules e.g. time critical modules are assigned a higher voltage (faster logic) while nontime-critical modules are assigned a lower voltage (slower logic). However, when voltage islands are fragmented, there are some overheads in power supply network and voltage shifting elements [6].

Beyond the voltage assignment, the MSV-aware floorplanning is a complicated process that involves other major issues including island generation, buffer insertion and floorplanning. There are many researches targeted on developing efficient MSV-aware floorplanning frameworks in recent years [6–10] and are categorized as: before, during or at the post-floorplanning. Most of these studies concentrate on reducing the total power consumption of the chip and to the best of our knowledge do not consider the temperature in their design. Cai et al. [11] consider the temperature in their paper. However, they do not consider applications with real-time constraints and ignore the effects of temperature of the blocks on the delay.

In this paper we propose a new MSV-aware floorplanning framework that considers temperature as an important parameter in the embedded design flow. The proposed MSV-floorplanning framework improves the peak temperature and reduces the hot spots of the design and also attempts to optimize other objectives such as power, wire length, fragmentation cost and area. One feature of particular interest of the paper is an attempt to model the inter-dependency between temperature and leakage power. Finding the optimal solution of a design space that needs to satisfy several objectives and constraints is a challenging and complex problem. The presented method combines the integer linear programming (ILP) formulation and simulated annealing (SA) approach to find optimal points or Pareto optimal set. Although the ILP solution is optimal, due to its long execution time its applicability is limited to solve the small problems. A heuristic solution is presented that trade-offs the accuracy for the execution time in problems with rather a large design space. In summary the contributions of this paper are listed below:

- To automate MSV-aware floorplanning in a hard real-time system a multi-objective optimization formulation is proposed. Due to the importance of temperature and leakage power, we consider these two important factors along with other objectives such as chip power consumption, area, fragmentation cost, wire-delay in the proposed formulation. SA is used to generate chip floorplan and ILP is employed to accomplish VI partitioning.
- Capturing the interdependency of the temperature and the leakage power which are becoming critical parameters in the future nanometer technologies.
- A greedy heuristic voltage assignment algorithm is proposed for VI partitioning that trade-offs accuracy for execution time for the problems with rather a large design space.
- A pertinent test environment has been accumulated that extracts the most important parameters of the 22 nm technology. To the best of our knowledge these parameters have not been reported in the previous studies. In order to evaluate and show the effectiveness of the presented parameters extensive evaluation with a large number of applications including E3S, GSRC and MCNC benchmark suites has been carried out.
- The presented framework generates floorplans that are more power-efficient compared to the cases that only attempt to optimize the temperature and attains lower peak temperature compared to the cases which only optimize the power.
- Due to the impacts of the critical path of the application graph we reached to an interesting and counterintuitive result that when the supply voltage increases the total power and peak temperature reduce in some scenarios.

The remainder of this paper is organized as follows. We discuss the related works in Section 2. Section 3 presents the preliminaries. In Section 4, we present the proposed framework. The mixed integer linear programming (MILP) voltage assignment formulation is described in Section 5. A heuristic solution for MSV voltage assignment problem is explained in Section 6. Section 7 provides the experimental results and evaluates the proposed framework. Finally, Section 8 concludes the paper.

2. Related works

There are many studies that discuss voltage assignment and island generation methods at floorplanning stage of an IC design flow. In these studies, VI partitioning is performed at various design stages such as before floorplanning [6], during floorplanning [12–14] and after floorplanning [15–18]. When the voltage assignment is performed before floorplanning, the results may be too abstract since they do not consider the dynamism of physical environments. As an example of this approach a temperature-dependent optimization leakage-aware floorplanning method is provided in [20] and showed that reducing the peak temperature may not reduce the leakage power. Assigning voltages to the islands after floorplanning lacks efficiency because the floorplan has already been fixed and hence the search space of VI in these works is small and, therefore, the results are far from optimal values.

To capture the physical layer information, voltage assignment needs to be accomplished during floorplanning stage. Thus contemplating physical layout and power optimization simultaneously provide much better results. Hu et al. [12] and Hung et al. [13] consider voltage assignment, voltage island generation and floorplanning problems, simultaneously. However, unlike ours they do not consider temperature and timing constraints. Ma et al. [9] formulate the voltage assignment task as a network flow problem.

Recent studies have targeted on MSV-aware floorplanning to enhance performance or reduce power consumption [21–25]. For example, Logan et al. [25] construct corresponding slicing tree for candidate floorplan and perform optimal island partitioning and voltage assignment using dynamic programming during SA to improve power consumption. However, thermal concerns have been overlooked in these studies. There are studies at the micro-architecture level that consider thermal [21] and leakage power. There are methods for dynamic thermal management of the functional units but do not consider floorplanning issues. Healy et al. [24] present a multiobjective micro-architecture floorplanning algorithm for high-performance processors implemented in two-dimensional and three-dimensional ICs.

Other studies that consider temperature-aware floorplanning and placement [23–25] do not support MSV technique. For example, Marler et al. [26] present a thermal-aware floorplanning to reduce the temperature of chips at the floorplan level by adjusting block utilizations based on the available whitespace in a floorplan. However, they do not consider MSV. To the best of our knowledge there is no study that attempts to minimize the fragmentation cost, temperature, and total power consumption of a chip while satisfying the performance constraints for the automated MSV-aware floorplanning. In this paper we propose an

MSV chip design and consider thermal effects at floorplanning and VI partitioning.

3. Preliminaries

The proposed framework in this paper produces an MSV floorplan for a specific application with hard real-time constraints. The inputs of this framework are firstly the application models with their relations, and secondly the physical characteristics of the application model in a given technology (e.g. 22 nm). Below, we present the models of the inputs to the proposed framework and also express the necessary background for understanding the framework.

3.1. Application model

Each benchmark represents an application for MSV floorplanning that is modeled as a directed acyclic graph (DAG) and is depicted in Fig. 1(a). The nodes represent the tasks and the edges consider control and data dependencies between the nodes. Each task (node) may have a hard real-time deadline and it is illustrated by a value in the DAG which is infinite for tasks with no deadline. The critical path is the longest path in a graph. In the rest of the paper we denote implementation of each task in the DAG in the given technology as a block. It is worth to mention that in the floorplan of the chip some blocks are potentially dead space which means that there are free space with no logics inside them. For example in Fig. 1 we have seven blocks in the DAG (see Fig. 1(a)) and we denote them by b_1 to b_7 (a generic block m is represented by b_m). We assume a fully pipelined implementation of tasks and inputs are streamed data.

We need to define several metrics and definitions for analyzing the experimental results. Firstly, we divide blocks of a task graph into two groups as critical and non-critical blocks. We call a block as critical if it does not satisfy a timing constraint or deadline after changing its supply voltage to a lower one. Also, we present the ratio of the number of critical blocks to noncritical blocks as a metric for defining the complexity degree of a task. Based on the value of this metric and depth of the task graph (length of critical path), the degree of a task graph is classified as simple (e.g. below 30%), normal (between 30% and 60%) and complex (e.g. above 60%).

3.2. Physical characteristics

Below we discuss the physical characteristics of an application model in a given technology.

3.2.1. Thermal model

An accurate thermal model (such as RC model) with specific details (such as leakage power and heatsink overhang modeling) attempts to mimic the real temperature of the chip. Such model which is a faithful representation of the real conditions of the chip can be further used in the analysis and design of the chip. Heat flow circulation starts from the source of the heat (i.e. die) and passes through intermediate layers and finally ends up at the final sink of the heat (i.e. air). The heat flow model follows a coarse grain discrete heat flow as reported in [28]. As depicted in Fig. 2, a chip heat flow model is partitioned into five vertical layers and are listed below. Each layer is then divided into some elements (see Fig. 3):

1. *Die*: This layer is divided into a number of blocks with potentially different sizes that correspond to the micro-architectural and dead space blocks in the chip floorplan. Comparative studies such as [28] do not consider dead space blocks, since it does not affect their results. However, for absolute performance analysis such as ours the impacts of dead space are to be considered. As mentioned earlier we model a dead space block as a micro-architectural block with

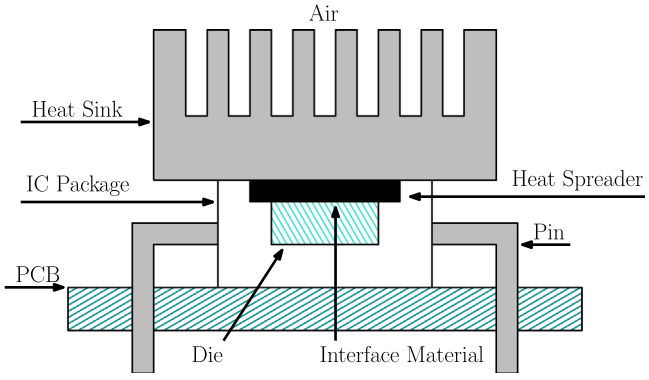


Fig. 2. A typical electronic packaging and its thermal diffusion elements [30].

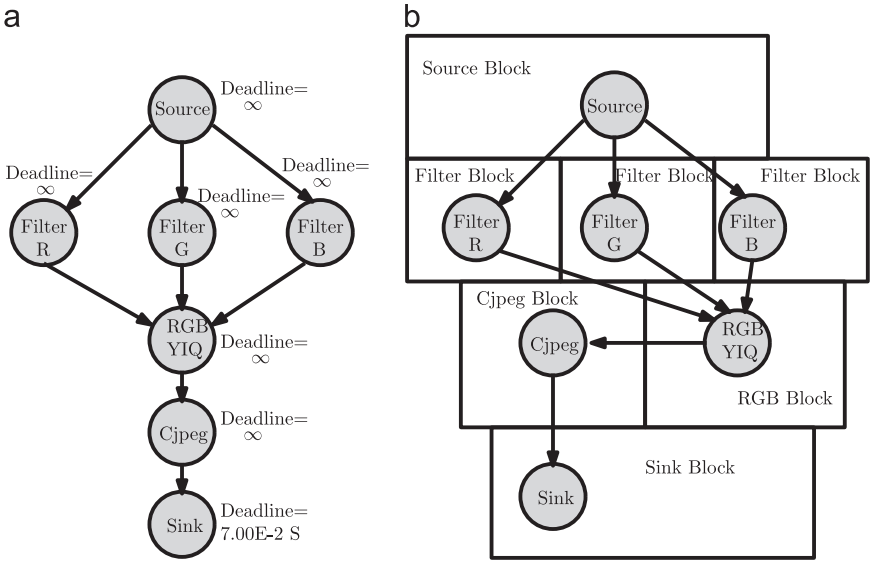


Fig. 1. Consumer DAG of E3S benchmark [27] and allocated blocks. (a) A sample DAG graph. (b) DAG corresponding blocks.

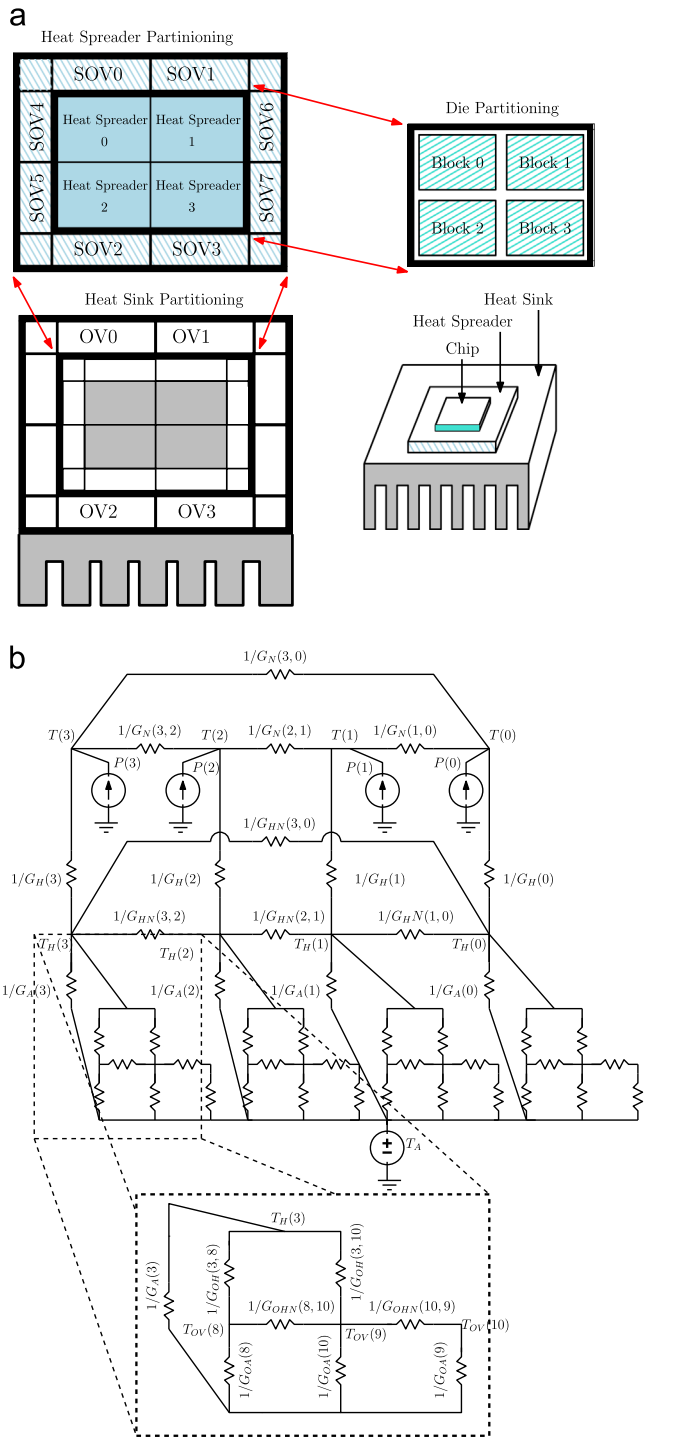


Fig. 3. Heat flow between blocks and heatsink elements. (a) Partitioning of heatsink, heat spreader and die. (b) Equivalent thermal circuit for model. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

zero power consumption. Fig. 3(a) illustrates four blocks in the die layer.

2. *Heat spreader*: This layer is above the die layer. Typically the heat spreader area is larger than the die and it covers the die layer. This layer is partitioned into center part and overhang part. The center part of the heat spreader is above the die layer and has the same shape and size as the die layer. The overhang part is the border part and surrounds the center part. We divide the heat spreader to some elements. To do so we need to divide both center part and overhang part into elements. The elements of the center part correspond to the blocks of the die layer in terms of number, size and shape. As shown in Fig. 3 (a) the elements of the overhang part are obtained by continuing the elements of borderline elements of the center part. The overhang part of the heat spreader is shown as red color in the figure.
3. *HeatSink*: This layer is above the heat spreader layer and transfers the heat to the air. This layer may cover the lower layer and thus it has a center part that matches the layer below and an overhang part as well. Fig. 3(a) shows the elements of the heat sink layer. The heatsink layer is divided into some elements again as what we did in the heat spreader layer.
4. *Thermal interface material*: These materials are placed between any two physical layers. Since the contribution of this layer to the overall heat of the system is insignificant and for the sake of simplicity we do not consider the heat effects of this layer in the model.
5. *The air or outside environment*: In a heat flow model, the air is assumed to be at a fixed ambient temperature.

There are some other negligible elements and layers such as I/O pins and PCB that contribute to the temperature of the system and we do not consider them in the model. In order to achieve high accuracy, we have included most of the heat flow layers discussed above that have major contribution to the temperature of the chip. To simplify the model the thermal effects of the heat spreader layer have been captured in the heatsink level. This simplification trade-offs the accuracy for the execution time and produces an insignificant error in the obtained results.

We use the well-known electrical–thermal duality relationship between heat transfer dynamics and RC electrical circuits in a high level thermal model [30]. Table 1 summarizes the quantities in heat transfer and their equivalence in electrical domain. We refer the interested readers to [31] and reference therein for the details of the related concept. According to the duality relation (as shown in Fig. 3), any heat flow can be expressed as a *current*, heat producer as a *current source* and the passing heat flow is described as an *electrical resistance*. With this duality relationship the temperature difference is equivalent to the electrical potential difference which is known as *voltage*. Electrical resistances describe steady-state thermal behavior. However to describe transient thermal behavior we require the notation of thermal capacitances.

The temperature of block b_m depends on three factors, the ambient temperature, the temperatures of blocks in the neighborhood of block b_m and finally its power consumption. Let us denote the set of neighboring blocks of block b_m by $\mathcal{N}_m = \{b_1, \ldots, b_n\}$, and the thermal conductance of block b_m to a neighbor block b_i by G^N_{mi} and the thermal conductance of block b_m to the heatsink element m (e_m) above it by G^H_m. G^A_k is defined as the conductance from the heatsink element e_k to the ambient, and G^{NH}_{kg} as the conductance between two heatsink elements e_k and e_g. The set of neighbors of heatsink element e_m is denoted by $\mathcal{N}^H_m = \{e_1, \ldots, e_h\}$. The following equations are obtained by thermal analysis described in [28]. Eq. (1) shows temperature balanced block m (b_m) on the die and

Table 1
Analogy of thermal and electrical quantities [30].

Thermal quantity	Unit	Electrical quantity	Unit
Heat flow	W	Current flow	A
Temperature difference	K	Voltage	V
Temperature resistance	K/W	Electrical resistance	Ω
Thermal capacitance	J/K	Electrical capacitance	F

Table 2
The variables used in MILP formulation.

Variable	Type	Definition
v_{ij}	Binary	$v_{ij}=1$ iff (voltage v_j is assigned to block b_i)
v_{ijkl}	Binary	$v_{ijkl}=1$ iff (voltage $v_{ik}=1$ and $v_{il}=1$)
n_{ij}	Binary	$n_{ij}=1$ iff (i and j blocks are neighbors)
t_e^i	Real	The earliest possible starting time of block b_i
t_l^i	Real	The latest possible starting time of block b_i
D_{blk}^i	Real	The gate delay of block b_i
D_e^{ji}	Real	$\forall e_{ij} \in \mathcal{E}$, the delay from block b_i to b_j
D_l^{ji}	Real	$\forall e_{ij} \in \mathcal{E}$, the delay imposed on block b_i by b_j
T_{b_i}	Real	The steady-state temperature of block b_i
T_{e_i}	Real	The temperature of the heatsink element e_i directly above the block b_i
T_{e_a}	Real	The ambient temperature
T_i^{hs}	Real	The temperature of heatsink above block b_i
P_{blk}^i	Real	The power of block b_i
P_{leak}^i	Real	The leakage power of block b_i

Table 3
The parameters of the problem.

Parameter	Explanation
P_{shift}^{kl}	The level shifter power consumption from voltage v_k to v_l
A_{shift}^{kl}	The level shifter area from voltage v_k to v_l
D_{shift}^{kl}	The level shifter delay from voltage v_k to v_l
D_{wire}^{ij}	The wire-delay from b_i to b_j
D_i	The deadline of task i
G_{mi}^N	The thermal conductance of block b_m to a neighbor block b_i
G_m^H	The thermal conductance of block b_m to the heatsink element m (e_m) above it
G_m^A	The conductance from the heatsink element e_m to the ambient
G_{kg}^{NH}	The conductance between two heatsink elements e_k and e_g
N_m	The set of neighboring blocks of block b_m
N_m^H	The set of neighbors of heatsink element e_m
G_i^{Ah}	The thermal conductance of heatsink above b_i to the ambient
d_i^j	The delay of b_i while operating at voltage v_j
P_{dyn}^i	The dynamic power consumption of b_i while operating at voltage v_i
$K_{ij}^{(1)}$	The temperature dependent part coefficient of leakage power of block b_i while operating at voltage v_i
$K_{ij}^{(2)}$	The constant part coefficient of leakage power of block b_i while operating at voltage v_i
$\mathcal{B}$	Set of rectangular floorplan blocks, $\forall b_i \in \mathcal{B}$, $b_1, \ldots, b_m$
$\mathcal{L}$	Set of supply voltage levels, $v_1,\ldots,v_M$ for block b_i
$\mathcal{E}$	The set of inter-block communication between application blocks (e.g. $e_{mn} \in \mathcal{E}$ denotes blocks b_m and b_n communicate)

Eq. (2) shows the temperature balance of the heatsink element e_m above the block b_m. The list of abbreviations used in the above equations is given in Tables 2 and 3:

$$0 = \sum_{n \in \mathcal{N}_m} (T_{b_m} - T_{b_n}) G_{nm}^N + C_m \frac{dT_{b_m}}{dt} + (T_{b_m} - T_{b_n}) G_m^H - P_m \quad (1)$$

$$0 = \sum_{g \in \mathcal{N}_m^H} (T_{e_m} - T_{e_g}) G_{mg}^{NH} + C_m^H \frac{dT_{e_m}}{dt} + (T_{e_m} - T_{b_m}) G_m^H + (T_{e_m} - T_{e_a}) G_m^A \quad (2)$$

where P_m, T_{b_m}, T_{e_m} and T_{e_a} denote the power consumption of block b_m, the temperature of block b_m, the temperature of the heatsink element e_m directly above the block b_m, and the ambient temperature, respectively. Here C_m is the thermal capacitance of block b_m and C_m^H is the thermal capacitance of heatsink element e_m. The thermal conductance of block b_m to the direct heatsink element above b_m (i.e. e_m) can be computed as [28]. In (2), if heatsink element e_k is an overhang element then the term $(T_{e_m} - T_{b_m})\, G_{mm}^H$ is set to zero.

3.2.2. Power model

Power consumption of block b_m in DAG is denoted by P_m and is composed of dynamic (P_m^{dyn}) and static power (P_m^{leak}) and is given by

$$P_m = P_m^{dyn} + P_m^{leak}$$

where the components are calculated through [3]

$$P_m^{dyn} = kC_{ef}(V_{DD_m})^2 f$$
$$P_m^{leak} \simeq V_{DDm} I_m^{linear} T_{b_m}$$

where k, C_{ef}, V_{DD_m} and f represent the switching activity, the switch capacitance, the supply voltage and the operating frequency of block b_m, respectively. Let T_{b_m} denotes the temperature of block b_m and I_m^{linear} represents the leakage current of block b_m that is a function of T_{b_m} and is given by

$$I_m^{linear} = F_m A_m (M_m T_{b_m} + N_m)$$

where A_m presents the area of block b_m, F_m represents the leakage current per unit area, M_m and N_m are leakage coefficients that are obtained by curve fitting in the piecewise linear model [3]. The leakage power consumption of block b_m operating at the voltage level v_j and temperature T_{b_m} is denoted by P_{mj}^{leak} and using the above relations it is computed as

$$P_{mj}^{leak} = K_{mj}^{(1)} T_{b_m} + K_{mj}^{(2)} \quad (3)$$

where $K_{mj}^{(1)}$ and $K_{mj}^{(2)}$ are constants that depend on block b_m and the voltage level v_j. We extract these constants with SPICE simulation for inverter circuit with FO4 as

$$P_{mj}^{leak} \propto (0.0016 T_{b_m} + 6.4426)$$

3.2.3. Block delay model

Temperature has negative effect on the execution time of the tasks as well. Higher temperature increases execution time of the tasks and thus reduces the number of executed critical blocks for a given deadline. Therefore, there can be some cases that if one ignores the temperature of a task it will meet the deadline, while it may miss the deadline because of the temperature effect on the execution time. The delay of a gate depends on its supply voltage V_{DD} and operating temperature T. In order to calculate the block delay model that operates at a voltage level V_{DD} and temperature T, we use the gate delay model reported in [2]

$$D \propto \frac{T^{\mu} V_{DD}}{(V_{DD} - V_{th})^{\alpha}} \quad (4)$$

where μ and $\alpha = 1.2$ are technology dependent empirical constants. Below, we calculate the unknown parameters in the above equation for 22 nm technology based on the calculations reported in [32]. Curve fitting in SPICE simulation for the circuit delay of an inverter with the FO4 load in 22 nm technologies provides $\mu = 1.61$. Since the relation between temperature and delay is non-linear and for MILP formulation a linear delay model is needed. Using a linear curve fitting the nonlinear model can be linearized as

$$D(T) = \lambda T_i + \gamma_i \quad (5)$$

where λ_i and γ_i coefficients depend the supply voltage level $V_{DD_i} \in \{0.8, 0.9, 0.1, 1.1\}$ through the following relationships:

$$\lambda_i \propto \frac{V_{DD_i}}{(V_{DD_i} - V_{th})^{\alpha}}$$
$$\gamma_i \propto \frac{V_{DD_i}}{(V_{DD_i} - V_{th})^{\alpha}}$$

For inverter circuit with FO4 with load at a ($V_{DD_i} = 0.8$ V) supply voltage, the linear model is $D(T) = (91.93T - 12\,918)$. Using SPICE

simulation we found that the absolute error due to the above linearized model is within 6%.

Finally V_{th} is given by $V_{th} = V_{th_0} - V_{DD}e^{-\alpha_{dibl}L_{eff}}$ where V_{th_0}, α_{dibl} and L_{eff} are the threshold voltage for long channel transistors, the DIBL coefficient and the transistor's gate length, respectively [29]. L_{eff} is given by $L_{eff} = L_{gate} - 2L_{int} + X_L$ where L_{int} and X_L represent the channel-length offset parameter and the channel-length offset due to mask effect, respectively. The value of V_{th_0} and α_{dibl} depends on the technology and according to [32] $V_{th_0} = 0.5118$ and $\alpha_{dibl} = 30$ are good candidates for 22 nm devices. It is noteworthy to mention that we do not consider the effects of process variation in the above model and it is a good line of the future work to consider these effects to have a more accurate model.

3.2.4. Wiredelay model

Wire delay is a parameter of paramount value in nowadays technologies and its value cannot be overlooked. We use the wire delay model proposed in [33,34]. It is shown in Fig. 4. Based on the location of the wire in the chip the wire delay is different for the global and intermediate levels. As reported in [34] the delay of a wire with length l being segmented by optimal length l_{opt} defined below for both global and intermediate levels and is given by

$$D(l) = 2l\sqrt{r_w c_w r_0 c_0} + \left(b + \sqrt{ab\left(1+\frac{c_p}{c_0}\right)}\right) \tag{6}$$

where c_p is parasitic output capacitances, r_o is the resistance and c_o is the input capacitance of a minimum sized inverter. The r_w and c_w parameters are resistance and capacitance of the wire per unit length, respectively. The a and b parameters depend on the switching model. For a switching model with 50%-swing selecting $a=0.7$ and $b=0.4$ is a typical choice [33].

c_o and r_o for the global level depend on the characteristics of a wire in a given technology and are related to the optimally segmented interconnect (l_{opt}) and the buffer size (S_{opt}) [33]. Using SPICE simulation of a ring oscillator that consists of an odd number of inverters connected by wires that tries to minimize the ratio of its stage, c_o and r_o are obtained as: $c_o = 0.181$ fF/u and $r_o = 10.860$ mΩ/u.

The same procedure is carried out to calculate these parameters for the intermediate level as: $c_o = 0.144$ fF/u and $r_o = 41.6$ mΩ/u. The values of r_w and c_w parameters of the interconnect copper wire are the same for both global and intermediate levels and are calculated as [35]

$$r_w = \frac{\rho}{WT} \quad c_w = 2(c_p + c_c) \quad c_p = \varepsilon\left(\frac{W}{H} + 2.04\left(\frac{S}{S+0.54H}\right)^{1.77}\left(\frac{T}{T+4.53H}\right)^{0.07}\right)$$

$$c_c = 2.37\left(\frac{W}{W+0.31S}\right)^{0.28}\left(\frac{H}{H+8.96S}\right)^{0.76} e^{-2S/(S+6H)} + \varepsilon\left(1.14\frac{T}{S}e^{-4S/(S+8.01H)}\right)$$

where ρ, c_p, c_c, W, T, H, S and ε are material resistivity, the plate capacitance, fringing capacitance, wire width, wire thickness, inter-layer dielectric height, inter-wire spacing, and dielectric constant, respectively. These parameters are related through

$$\rho = \rho_0\left(\frac{1}{3}\left(\frac{1}{3} - \frac{\alpha}{2} + \alpha^2 - \alpha^3\ln\left(1+\frac{\alpha}{2}\right)\right) + \frac{3C(1-p)}{8}\left(\frac{\left(1+\frac{H}{W}\right)}{\frac{H}{W}}\frac{\lambda}{W}\right)\right)$$

$$\alpha = \frac{\lambda R}{D(1-R)} \quad T = W \times AspectRatio, \quad H = 4T, \quad S = W$$

where ρ_0, λ, D, p, R and C are the bulk resistivity of copper wire, mean free path, average distance between grain boundaries, specularity parameter, reflectivity coefficient at grain boundaries, and a constant for rectangular cross-sections [35]. The values of the above parameters for global and intermediate levels are reported in [1].

3.2.5. Fragmentation cost model

Using the MSV method may lead to a highly fragmented power network that complicates the design of power network in two aspects namely: power supply noise issues, and the increased number of level shifters that may affect the design overhead as well. We adopt a simple fragmentation cost model based on the difference of voltages between two neighboring blocks and formulate the power network planning complexity problem as an ILP see (29) in Section 5.6. We use such a simple model for the fragmentation cost to attain a tractable solution in most scenarios of interest [7].

3.3. Problem definition

Fig. 5 illustrates the multi-voltage floorplanning and island partitioning inputs and outputs presented in this paper.

Inputs:

1. A given block set that is generated by a user and each block in the block set contains the area of the block, the number of voltage levels and power and delay information for each voltage level of the block.
2. A given DAG of application is provided by user.
3. Thermal properties of the chip that are reported in chip specifications manual of manufacturers.
4. Level shifters information provided by the user.

Outputs: A best floorplan containing floorplanning and VI partitioning with thermal distribution information. The output is obtained through execution of the proposed framework that optimizes multiple objectives while satisfying the hard timing constraints of the application.

4. Proposed framework

SA-based floorplanning algorithm has been successfully employed for area and wire optimization in the previous studies [37,22]. However, VI partitioning for voltage islanding purpose has not been implemented in such studies. We integrate an ILP formulation to capture VI partitioning into the classic SA-based slicing tree floorplanning algorithm [19].

Two approaches may be taken to implement this integration, namely Post FloorplanVI (PFVI) partitioning and During Floorplan VI (DFVI) partitioning. Fig. 6(a) and (b) depicts PFVI and DFVI, respectively. As illustrated in Fig. 6(a) PFVI approach is a straightforward extension of SA-based floorplanning algorithm that attempts firstly to obtain a floorplan using the SA approach and then finds the VI partitioning using the MILP approach. SA algorithm iterates the below three phases to converge to the near optimal floorplan:

1. Candidate floorplans generation based on current estimates.
2. Evaluate function at proposed candidate floorplan.
3. Accept new candidate floorplan if it improves solution.

For the sake of clarity we have indicated in Fig. 6 the SA-based floorplanning algorithm section of the framework as 'F'. After accomplishing the SA algorithm we then use MILP to find the solution of the VI partitioning section of the problem. The VI partitioning section of the framework is indicated as 'V' in Fig. 6.

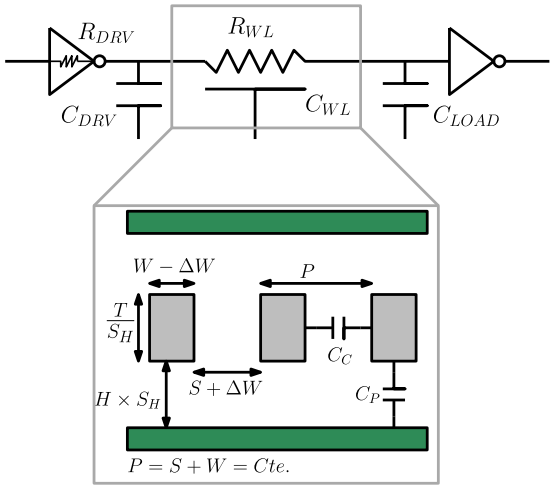
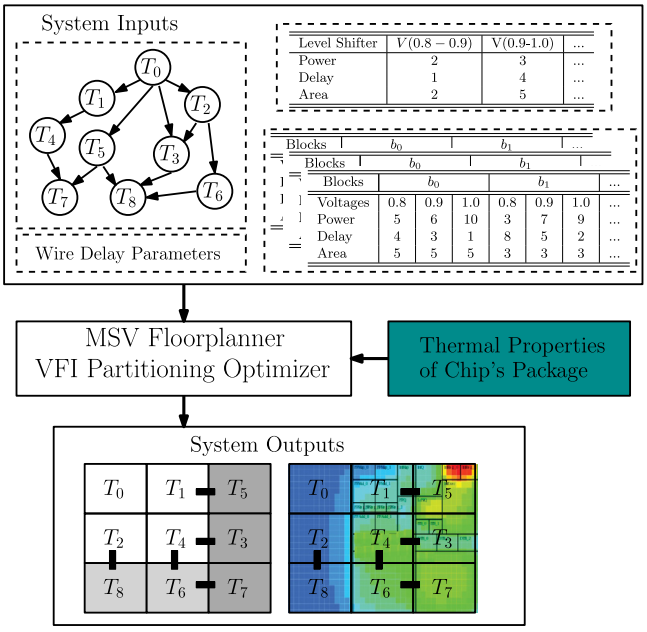


Fig. 4. Typical wire model and interconnect cross section [37].



Level Shifter	$V(0.8-0.9)$	V(0.9-1.0)	...
Power	2	3	...
Delay	1	4	...
Area	2	5	...

Blocks	b_0			b_1			...
Voltages	0.8	0.9	1.0	0.8	0.9	1.0	...
Power	5	6	10	3	7	9	...
Delay	4	3	1	8	5	2	...
Area	5	5	5	3	3	3	...

Fig. 5. Inputs and outputs of the problem.

The above phases are complicated and below we briefly describe the details of each phase. In phase 1, we generate a candidate floorplan using different types of random perturbation moves on the current floorplan solution. To do so firstly, using the method in [20], we generate different candidate floorplans by three types of perturbation moves namely, 'swap two adjacent blocks', 'complement a chain of non-zero length', and 'swap an adjacent block and a cut'. Among the generated candidate floorplans we select the floorplan that returns the minimum value of Eq. (7) and is inputted to phase 2 as candidate floorplan

$$w_A A + w_T T_{\max} + w_W W \tag{7}$$

where w_i, $i \in \{A, T, W\}$ are associated weights of the objectives. Eq. (7) is actually the multiobjective cost function. The objectives in the cost function are the area, the maximum steady-state temperature, and the wire-length cost. The maximum steady state temperature corresponds to the highest voltage level in the MSV system, i.e. with two voltage levels 0.9 and 1.1 V, the temperature is calculated based on the 1.1 V voltage level. The wire-length cost in the multiobjective itself is given by

$$W = \sum_{1 \le i,j \le n} c_{ij} d_{ij}$$

where n is the number of blocks, c_{ij} is the wire density of the interconnection between blocks b_i and b_j, and d_{ij} is the Manhattan distance between their centers. Phase 2 evaluates the candidate floorplan and phase 3 replaces new candidate floorplan if it improves current solution. The second framework (DFVI) attempts to integrate floorplanning and VI partitioning together to search the solution in a larger search space and thus to find a solution that may be overlooked in PFVI. DFVI has inserted VI portioning into standard SA algorithm. So the DFVI has iterates through the following four stages:

1. Candidate floorplans generation based on current estimates.
2. Voltage level assignment and island partitioning for the candidate floorplan
3. Evaluate function at a proposed candidate floorplan.
4. Accept new candidate floorplan if it improves solution.

The details of phase 1 in DFVI are the same as PFVI. However, DFVI differs from PFVI in phase 2. In DFVI the generated candidate floorplan of phase 1 along with application tasks and information for tasks implementation (such as power problems and relations of tasks) are feed to phase 2. Then, we use MILP formulation to compute the voltage level assignment and island partitioning for the candidate floorplan. The objective functions of MILP section depend on the applications and its requirements. Phase 3 evaluates the candidate floorplan and it considers the VI partitioning information, area, and wire-length cost as the objective function and in phase 4 the SA algorithm updates new candidate floorplan if it improves solution. As a result of the above iterative algorithm the best solution is being obtained.

It is noteworthy to mention that to trade-off execution time for the accuracy, in both the above frameworks, one may replace the MILP algorithm with any heuristic algorithm. This replacement has no effects on the proposed frameworks but on the execution time. The presented framework in Fig. 6 is general enough and it may be used for both regular and irregular floorplannings. For the sake of clarity and brevity Fig. 6 shows only the regular floorplan problem.

5. MILP formulation

In this section, the proposed mathematical programming formulations as MILP for voltage assignment and partitioning problem are presented. The definition of variables and input parameters are presented in Tables 2 and 3, respectively.

5.1. Objective function

The weighted multi-objective function of the MILP is

$$\text{Minimize} \quad (w_{temp}\delta_T + w_{power}\delta_P + w_{frag}\delta_F)$$

$$\delta_T = \frac{T_{\max} - T^*_{\max}}{T^*_{\max}} \quad \delta_P = \frac{P_{ower} - P^*_{ower}}{P^*_{ower}} \quad \delta_F = \frac{F_{rag} - F^*_{rag}}{F^*_{rag}}$$

where $T_{\max}$ is the maximum chip temperature, F_{rag} is the fragmentation cost, and P_{ower} is the power consumption and w_i, $i \in \{$temperature, power, fragmentation$\}$ are associated weights of the objectives. $T^*_{\max}$ is the optimal value of the formulation when it has been solved for only temperature as a single objective. The same is applicable for P^*_{ower} and F^*_{rag}.

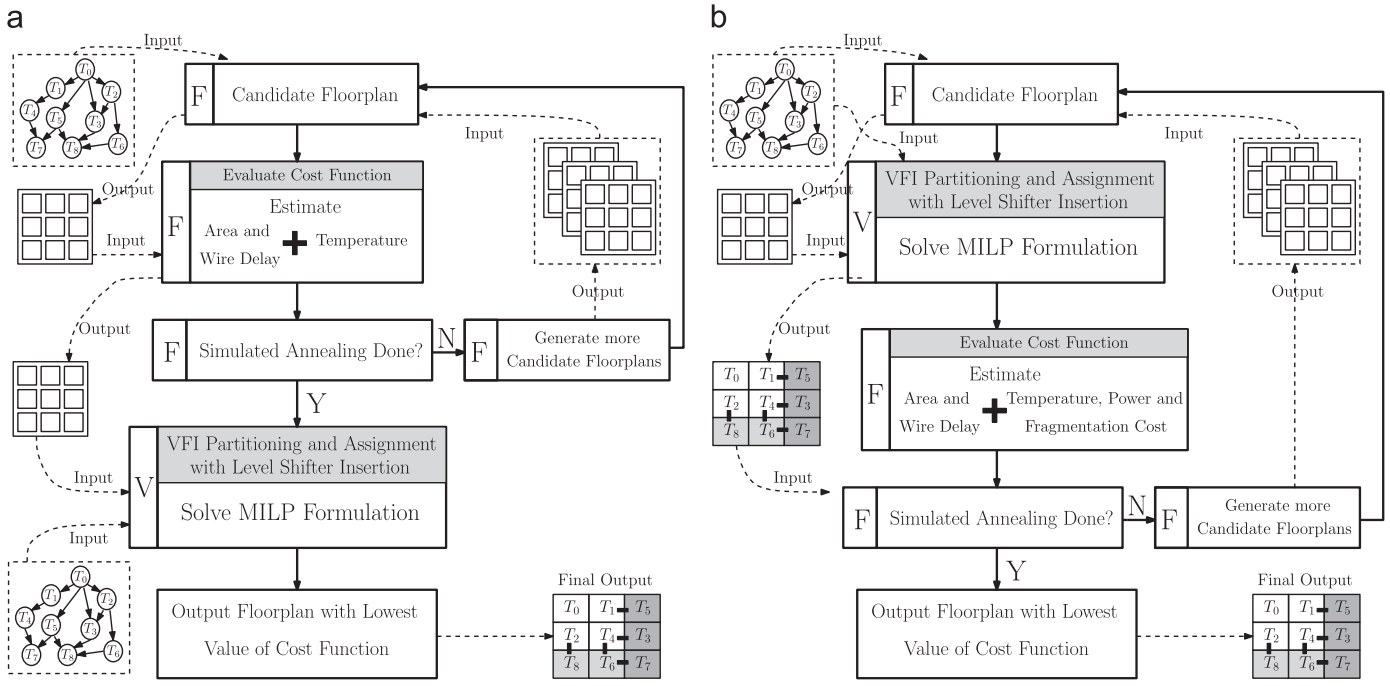


Fig. 6. The proposed VI partitioning and MSV-aware floorplanning framework outline. (a) Post floorplanning VI partitioning (PFVI). (b) During floorplanning VI partitioning (DFVI).

5.2. Supply voltage

The operational voltage of block b_i at voltage v_j is indicated by v_{ij}. Due to the fact that each block can operate at just one voltage level, therefore

$$\forall b_i \in \mathcal{B}, \quad \sum_{j=1}^{M} v_{ij} = 1 \tag{8}$$

where M denotes the number of available voltage levels.

5.3. Level shifter

We denote the necessary level shifter power consumption by $P_{shifters}$. Level shifters are needed for interconnecting two blocks with different voltage levels to facilitate their communication. Thus, we have

$$P_{shifters} = \sum_{e_{ij} \in \mathcal{E}} \sum_{k=1}^{M} \sum_{l=1}^{M} P_{shift}^{kl} v_{ik} v_{jl}. \tag{9}$$

The expression for $P_{shifters}$ is non-linear and is in a quadratic form. It is possible to simplify this expression to a corresponding linear expression by eliminating quadratic terms and adding new constraints. We use the following method to linearize a quadratic expression. Consider a quadratic expression in the form $a \times b$, where a and b are binary variables. We substitute c (a Boolean variable) for the multiplication of a and b using the following equations:

$$\begin{cases} c \geq a+b-1 \\ 2c \leq a+b \end{cases} \tag{10}$$

Substituting (10) instead of $a \times b$ in the quadratic forms will result in linear constraints. We may linearize a quadratic form $a \times b$ for the case that c is an integer using the following constraints. Γ is a large constant:

$$\begin{cases} c \geq 0 \\ c - a\Gamma \leq 0 \\ c \geq b - (1-a)\Gamma \\ c \leq b - (a-1)\Gamma \end{cases} \tag{11}$$

Using the above method we linearize the quadratic form of the constraints (9) by introducing the new binary variable v_{ijkl} to be substituted with quadratic $v_{ik}.v_{jl}$ and the following:

$$\forall b_i, b_j \in \mathcal{B}, \quad \forall k,l \in \mathcal{L} \quad v_{ijkl} \geq v_{ik} + v_{jl} - 1 \tag{12}$$

$$\forall b_i, b_j \in \mathcal{B}, \quad \forall k,l \in \mathcal{L} \quad 2 \times v_{ijkl} \leq v_{ik} + v_{jl} \tag{13}$$

Therefore, the level shifter power consumption ($P_{shifters}$) can be re-written as

$$P_{shifters} = \sum_{e_{ij} \in \mathcal{E}} \sum_{k=1}^{M} \sum_{l=1}^{M} P_{shift}^{kl} v_{ijkl}. \tag{14}$$

Considering, the area overhead of level shifters we reach to the following constraints for the area of the chip:

$$\sum_{e_{m,n} \in \mathcal{E}} \sum_{k=1}^{M} \sum_{l=1}^{M} Area_{shift}^{kl} v_{mnkl} \leq A_{shifters} \tag{15}$$

where $Area_{shift}^{kl}$ represents the occupied area by the level shifters from voltage levels v_k to v_l and $A_{shifters}$ denotes upper bound of the level shifters area and is specified by the user.

5.4. Performance

Increasing the temperature of the chip results in severe performance degradation in real-time applications. In order to model system performance, let us define three parameters.

1. Absolute earliest start time (AEST), t_e^i, is the earliest possible starting time of the block b_i. That is, the time at which block b_i receives the latest signal.

2. Absolute latest start time (ALST), t_l^i, is the latest possible starting time of the block b_i. That is, the latest time at which block b_i can start execution in order to meet its deadline.
3. Slack time, t_i^{slack}, is the difference between the earliest start time of the block b_i and the latest start time of the block b_i. The block b_i can meet its deadline provided that t_i^{slack} is positive.

The delay imposed by the level shifters in the link e_{ij} can be computed as

$$\forall e_{ij} \in \mathcal{E}, \quad D_{shift}^{ij} = \sum_{k=1}^{M} \sum_{l=1}^{M} D_{shift}^{kl} v_{ijkl} \tag{16}$$

where D_{shift}^{ij} is the level shifter delay from the voltage level v_k to v_l voltages. The gate delay of block b_i is given by

$$\forall b_i \in \mathcal{B}, \quad D_{blk}^{i} = \sum_{j=1}^{M} d_i^j v_{ij} \tag{17}$$

where d_i^j represents the delay of b_i which operates at voltage level v_j. The delay from block b_i to block b_j, D_e^{ji} is comprised of the following delays:

1. Level shifter delay D_{shift}^{ij}.
2. Gate delay of b_i, D_{blk}^{i}.
3. The wire delay between blocks b_i and b_j, D_{wire}^{ij}.

Thus D_e^{ji} is given by

$$\forall e_{ij} \in \mathcal{E}, \quad D_e^{ji} = t_e^i + D_{blk}^i + D_{wire}^{ij} + D_{shift}^{ij} \tag{18}$$

The delay imposed on the block b_i by the block b_j can be modeled as

$$\forall e_{ij} \in \mathcal{E}, \quad D_l^{ij} = t_l^j - D_{blk}^j - D_{wire}^{ij} - D_{shift}^{ij} \tag{19}$$

Now, *Earliest start time* (t_e^i) and *latest start time* (t_l^i) of block b_i can be formulated as

$$t_e^i = \max(0, \max(D_e^{ij})) \tag{20}$$

$$t_l^i = \min(D_i, \min(D_l^{ij})). \tag{21}$$

In order to guarantee that blocks will meet their deadlines, the slack time of each block must be greater than or equal to zero. That is

$$\forall b_i \in \mathcal{B}, \quad t_l^i - t_e^i \geq 0. \tag{22}$$

The max and min operations are not linear. According to (22) in the ILP solver we replace max and min operations by the following linear equations:

$$\forall i,j \in \mathcal{B}, \quad t_e^i \geq D_e^{ij}, \quad t_l^i \leq D_l^{ij}. \tag{23}$$

5.5. Temperature and power

Due to the high computational complexity of dynamic thermal analysis, we only used the steady-state temperature, T_i, of block b_i, in the formulation. One advantage of steady-state temperature analysis is the ability to estimate long-term temperatures in the short simulation intervals. We denote the steady-state temperature of the heatsink above the block b_i by T_{hs}^i. The presented thermal model in (1) and (2) are rewritten in (24) and (25), respectively. Here, due to the steady-state analysis, capacitance values are set to zero:

$$T_{b_m} G_m^H = \sum_{j \in \mathcal{N}_m} (T_{b_j} - T_{b_m}) G_{mj}^N + T_{e_m} G_m^H + P_{blk}^m \tag{24}$$

$$T_{b_m} G_m^H = \sum_{j \in \mathcal{N}_m^H} (T_{e_m} - T_{e_j}) G_{ij}^{NH} + T_i^{hs} G_i^H + (T_i^{hs} - T_A) G_i^{Ah} \tag{25}$$

where P_{blk}^i is the power consumption of block b_i that presented as

$$P_{blk}^i = P_{dyn}^i + P_{leak}^i \tag{26}$$

where P_{dyn}^i and P_{leak}^i represent the dynamic power and the leakage power consumption of block b_i, respectively. Dynamic power consumption of block b_i (P_{dyn}^i) can be computed as

$$P_{dyn}^i = \sum_{j=1}^{M} v_{ij} P_{ij}^{dyn} \tag{27}$$

where P_{ij}^{dyn} represents the dynamic power consumption of block b_i while operating at voltage level v_j. The leakage power consumption of block b_i (P_{leak}^i) is calculated as

$$P_{leak}^i = \sum_{j=1}^{M} v_{ij} P_{ij}^{leak} \tag{28}$$

where P_{ij}^{leak} that is given in Eq. (3) represents the leakage power consumption of block b_i when operating at voltage level v_j. Since (3) is non-linear we convert this to an equivalent linear relation. To do this we modify (28) using (11) with the $a = v_{ij}$, $b = P_{ij}^{leak}$ and $c = P_{leak}^i$.

5.6. Fragmentation cost

Fragmentation cost model, F_{rag}, is formulated to control the VI partitioning as

$$F_{rag} = \sum_{i=1}^{m} \sum_{j=i+1}^{m} \sum_{k=1}^{L_i} \sum_{l=1}^{L_j} n_{ij} v_{ik} v_{jl} \quad (v_{ik} \neq v_{jl}). \tag{29}$$

As explained in Eq. (9) we simplify the above equation to a corresponding linear expression by eliminating the quadratic terms $v_{ik}v_{jl}$ which is defined in Eqs. (12) and (13). Thus the linearized version of (29) is given by

$$F_{rag} = \sum_{i=1}^{m} \sum_{j=i+1}^{m} \sum_{k=1}^{L_i} \sum_{l=1}^{L_j} n_{ij} v_{ijkl}. \tag{30}$$

6. Voltage assignment heuristic

To balance accuracy and execution time, especially for the scenarios with large design space, we present a greedy heuristic algorithm for voltage assignment that can be used within a given thermal modeling tool such as hotspot [38]. It is based on the list scheduling (LS) algorithm that usually attains reasonable results [39] and is shown in Algorithm 1. Employing LS algorithm with static priority assignment may not always lead to an optimal schedule [39]. Because assigning a new voltage to a block may change the critical path of the DAG, due to changes in propagation delays. Thus we need to calculate the critical path after each assignment. This imposes some modifications in the algorithm so as to assign dynamic priority to each block to achieve an effective assignment. In each step of the assignment procedure we recalculate the priority of each block. Algorithm passes through the following three steps iteratively to converge the solution:

1. Recalculating the new priorities of each block.
2. Choosing the highest priority block as the new candidate block.
3. Assigning the voltage level to the candidate block such that it meets the timing constraints and achieves higher utility based on the objectives.

Algorithm 1. Voltage assignment heuristic.

1: **for all** blocks in list **do**
2: Assign HighestVoltageLevel to block

```
3:  for all blocks in list do
4:    Calculate AEST, ALST, and Priority
5:  while not all blocks voltage assigned to fixed do
6:    Sort all blocks based on priority
7:    b_i: Select a block with HighestPriority;
8:    Decrease Voltage Level of block b_i
9:    if Real-Time Constraint valid then
10:     if VoltageLevel of b_i equal to LowestVoltageLevel then
11:       Mark voltage of block b_i as fixed
12:   else
13:     Increase VoltageLevel of block b_i
14:     Mark voltage of block b_i as fixed;
15:   for all blocks in list do
16:     Update AEST and ALST and Priority
```

Step 1 recalculates the priority of all unmarked or *not fixed* blocks. Step 2 performs selecting the candidate block which is the block with the lowest slack time among the unmarked blocks. In step 3 we reduce the voltage level of the candidate block. If it either causes to go beyond the lowest voltage level or miss the deadline constraint, the block is marked and the previous voltage level is *fixed* for that block. The algorithm terminates when all blocks are marked. The difference between the $AEST(m)$ and the $ALST(m)$ of block b_m is known as the slack time. This metric shows the distance of a block b_m to the critical path and the lower the value of the metric, the higher the priority. The priority of the block b_m is defined as

$$priority(m) = \frac{\alpha T(m) + \beta P(m) + \gamma F(m)}{SlackTime(m)} \tag{31}$$

where α, β and γ are weights for the temperature, the power and fragmentation cost of block b_m, respectively. Fragmentation cost of block b_m ($F(m)$) is the difference between the current fragmentation cost and the new fragmentation cost caused by decreasing its voltage to a lower level. In the initialization phase of the algorithm all blocks are unmarked. We assign the highest voltages level to all blocks and thus the lowest execution time is considered for all blocks. The advantage of choosing these values is that we start from a feasible point in our design space. It can be shown that the algorithm with the choice of diminishing for the population size parameter of *not fixed* block in each step convergences to the solution provided that it is in the feasible region. As a result, the time complexity of our dynamic critical path algorithm has been shown to be $O((n^2 v)$. $O(ThermalSimulator)$, where n, v and $O(ThermalSimulator)$ are the number of blocks, legal allocation voltages and the worst-case execution time of thermal simulator tool, respectively.

7. Experimental results

The presented framework and also the proposed heuristic algorithm were validated experimentally using standard benchmarks. E3S [27], GSRC [40] and MCNC [41] benchmark suites. In E3S benchmark, there exist 34 processing elements (PEs) which we randomly select all of them to map the blocks. To adapt GSRC and MCNC for our study we add extra information to them for the experiments. In the version we use, the dimensions of the GSRC benchmarks are in the range of tenths of a micron, and for the MCNC benchmarks the dimensions are in the range of microns and thus it is necessary to scale them down to be applicable for 22 nm technology. We implement our algorithms in the C++ programming language and run it on an Intel Core i7 CPU860 2.80 GHz PC with 4GB memory and Microsoft Windows 7 OS. The MILP formulation is solved using the IBM CPLEX 11.1 solver [42].

7.1. Assumptions and experimental setup

The settings and assumptions of the proposed framework are similar to other studies [7,43] and we use the same method reported in [43]. Using the information provided by the net file of the MCNC and GSRC benchmarks we produce the DAG by assigning directions to the connections of the DAG. Moreover, we need to assign timing and power consumption to each node (i.e. block) in the DAG according to its occupied area. The base value for delay and power consumption of each block is selected by assigning the highest V_{DD} for the block. Then, using these base values, the delay and power consumption of the block for other V_{DD} s are calculated. This procedure applies for both static and dynamic powers of each block. The delay for the highest V_{DD} for each block in the DAG is generated randomly based on the typical delay values of the given technologies. The next step is assigning the timing constraints of each block of the DAG. These are chosen to be $1.1\times$ of the critical path delay of the DAG in the given benchmark. The power density information has not been reported in MCNC and GSRC benchmark data sets. Thus, we use the values reported in [1] for the 22 nm technology that are in the range of 2.92–13.81 w/mm^2 [44].

To E3S benchmarks information for 22 nm technology has not been released yet and we calculated the required parameters by scaling the values reported in [36,45,46]. The area and power consumption of level shifters are required to be considered as well. We use SPICE simulation to obtain power consumption and delay of a level shifter of dimensions 10×10 [17]. In modeling the wire-delay we ignore the process variation for the key parameters of transistors including L_{eff} and V_{th}. Wire-delay model of the net is proportional to the wire length in the millimeter basis and are summarized in Table 4. Here ρ_0 indicates the resistivity of bulk copper; λ is the mean free path length; D is the average distance between grain boundaries; ρ is the specularity parameter; R is the reflectivity coefficient at grain boundaries and C is a constant value for rectangular cross-section [37]. Finally, the ambient temperature is assumed to be 45 °C.

In Sections 7.2–7.4 we report the results of the experiments and the effects of the important factors on the results. The results of the experiments reported in Sections 7.2 and 7.3 are based on the MILP method and Section 7.4 that compares MILP and heuristic is based on both MILP and heuristic methods. In the figures the average value and the maximum value of the temperature of each floorplan are being presented. It is noteworthy to mention that the difference between the results obtained in our experiments and those attained by Hotspot has at most 5 °C discrepancy and sounds. Similar results have been reported in [33] and confirm the accuracy and validity of our results. The weights used is the SA algorithm are fixed in the experiments and have been summarized in Table 5.

7.2. The value of the supply voltage and the number of voltage levels results

To use MSV efficiently, we need to consider two key parameters. Calculating the number of voltage levels and then obtaining suitable magnitude of each level. It is worth to mention that as the number of voltage level increases, on the positive side, the magnitude of the objective function decreases and thus the chance of obtaining better solutions increases. On the negative side, however, the complexity of power network plan and execution time of voltage assignment part of the framework increase. To evaluate the effects of the number of voltage level on the framework objective we have considered three scenarios. These numbers have been chosen two, three and four voltage levels. As

Table 4
Parameters for copper resistivity calculation.

Parameter	ρ_0	λ	R	p	C	D
Value	2.04 μΩ cm	37.3 nm	0.32	0.41	1.2	W

Table 5
The weights used in the objectives of SA algorithm.

Benchmark	w_A	w_T	w_W
n10	355 928.28	0.008550	8.6199
n10b	342 439.95	0.008539	9.7009
n10c	319 674.70	0.008760	7.9035
hp	201 057.64	0.00870	9.6726
apte	163 712.58	0.008403	4.966
xerox	160 450.39	0.00843	0.9327
ami33	347 74.12	0.00819	0.576
n30	84 576.64	0.0087	1.646

reported in the previous studies [13], the typical voltages range in contemporary technologies spans 0.8–1.2 V.

The voltage levels for two, three and four levels have been considered as [0.8, 0.9], [0.8, 0.9, 1.0] and [0.8, 0.9, 1.0, 1.1] volts, respectively. The results for these three scenarios have been illustrated in Fig. 7. As the number of voltage levels increases the magnitude of objective function decreases. This is due to the fact that choosing more voltage levels results in a bigger search space and thus implies a potentially bigger solution set. This logically means that the obtained number of non-critical blocks and also the number of candidate floorplan that satisfies the given deadline increase. Thus, with a bigger search space there is a chance to find some solutions that are unexploited in a smaller search space. Bigger search space, however, has a negative impact on the execution time.

Unfortunately, standard E3S benchmarks inconsiderate the benefits of the MSV method and thus makes it inefficient for the purpose of this study. This is due to the fact that the deadlines in the E3S benchmarks are very relaxed and our experiments reveal that all the blocks in the floorplan are running in the lowest voltage level and at the same time almost most of the tasks meet their deadlines. Even, using the E3S benchmarks the proposed MSV-aware framework in average improves the cost function around 7% compared to the single voltage level. Since, SA algorithm finds more candidate floorplans that meet the given deadline in MSV-aware compared to the single voltage level.

To reveal the benefits of the MSV method, we need to use those benchmarks that have flexibility in tuning the deadlines to be tighter. This gives the opportunity that captures the solutions that need to choose higher voltage levels in some blocks. Here, choosing higher voltage levels for some blocks in the critical path results in faster executing of the assigned tasks to critical blocks and thus meeting the deadline. However, the task misses the deadline when the blocks in the critical path operate in the lowest voltage level. Thus we used benchmarks such as GSRC and MCNC that have flexibility in tuning the deadline parameter. Knowing the benefits of multiple voltage levels, in the sequel and as a boundary case we concentrate on two voltage levels which translate to the worst case analysis of the multiple voltage levels. We denote V_{DDH} as the high supply voltage level and V_{DDL} as the low supply voltage level. Now, we need to determine the voltage magnitude of each level. In 22 nm technology the voltage level is bounded below at 0.8 V ($V_{DDL} = 0.8$). V_{DDH} takes on any value above the V_{DDL} and is typically less than 1.2 V. To see the effects of the magnitude of V_{DDH}, we sweep this range in the steps of 0.1 V and we select V_{DDH} as 0.9, 1.0, 1.1 and 1.2 V. Thus we end up four scenarios [0.8,0.9], [0.8, 1.0], [0.8, 1.1], and [0.8,1.2] for V_{DDL} and V_{DDH}. Fig. 8

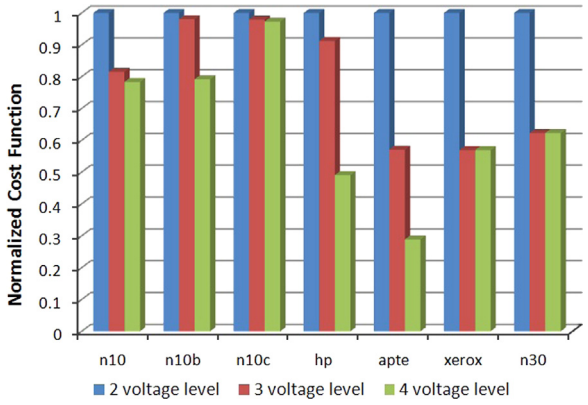


Fig. 7. Evaluation of the proposed framework results under different voltage levels numbers.

(a) presents the behavior of the normalized cost function with respect to different values of V_{DDH} for some benchmarks. The normalized cost function depends on the type of the benchmarks and governs by the following three factors:

1. *Power density*: Choosing higher V_{DDH} results in higher power consumption and higher power density and as a consequence higher temperature of the chip.
2. *The number of critical blocks*: Increasing the magnitude of V_{DDH} allows some critical blocks to be executed faster and as a consequence the number of critical blocks is reduced. For example in *n30*, changing V_{DDH} from 0.9 to 1.0 V decreases five items from the critical block set. Increasing the number of noncritical blocks may lead to a larger search space and thus finding more solutions. On the other hand, larger search space leads to higher execution time of finding the solutions of the voltage assignment part of the problem.
3. *The amount of slack time*: Higher V_{DDH} has two contrasting effects on slack time: (1) on the positive side high V_{DDH} results in a faster execution of the blocks and consequently a higher slack time of the task in a period; (2) on the negative side high V_{DDH} results in higher power density and temperature. High temperature induces larger delays and thus deteriorates the slack time. Thus the attained slack time is the outcome of the above two incongruous effects. It is noteworthy that the impact of V_{DDH} on the execution time and the temperature is complex. This contradicting and odd behavior implies that it is of paramount value to consider both power density and temperature simultaneously in the objective function of the problem and this is the main motivation of attempting to consider a multi-objective problem formulation in this study. Fig. 8 (b) plots this odd behavior and demonstrates the number of blocks that are actually operating at V_{DDH} for different values of V_{DDH}. This figure exhibits that in some benchmarks such as *n10c* slack time is decreased. In such benchmarks there are few numbers of blocks that are operating at V_{DDH} and most of the blocks are operating at a voltage lower than V_{DDH}. On the other hand in benchmarks such as *n10*, *n30* and *ami33* most of the blocks are operating at V_{DDH} and thus the slack time is increased.

The objective function presented in Section 5.1 is affected by temperature and power directly. As argued above factors 2 and 3 i.e. the number of critical blocks and the amount of slack time may decrease max temperature and total power consumption of the chip. On the other, side factor 1 i.e. power density may increase both max temperature and total power consumption. The behavior

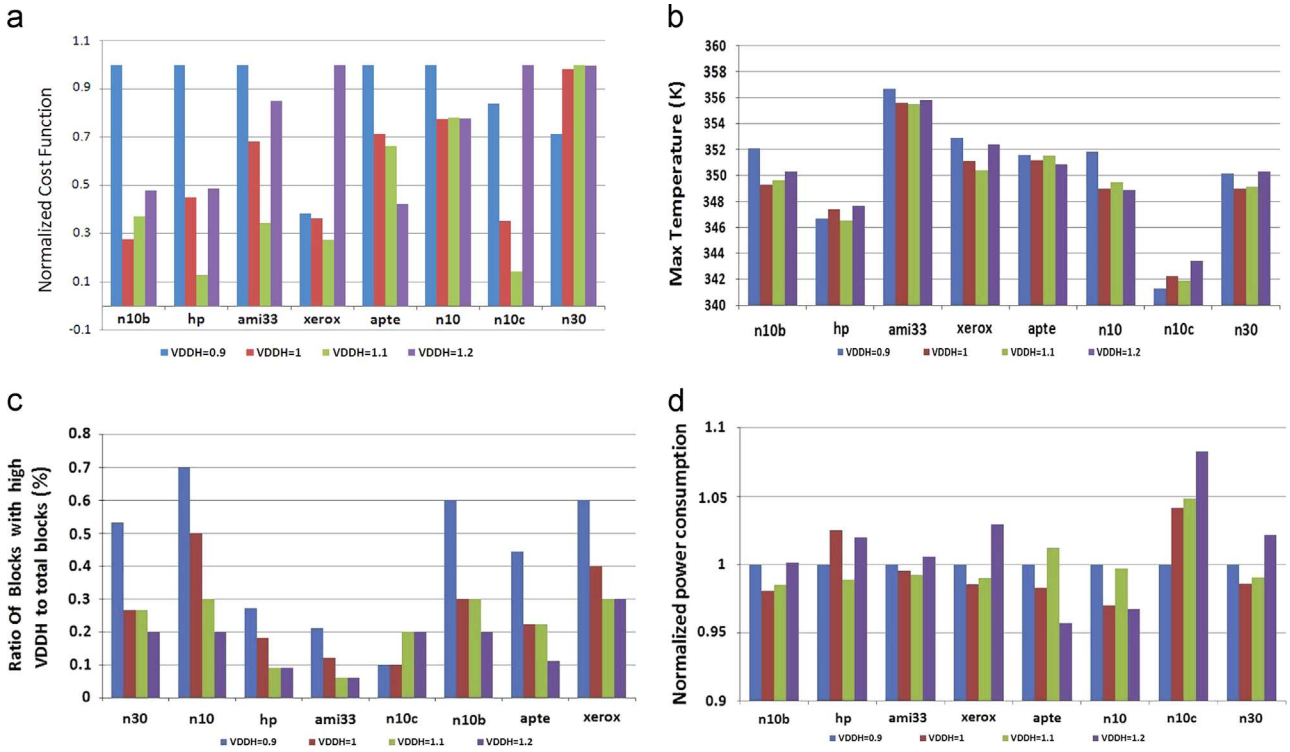


Fig. 8. Evaluation of the proposed frameworks results under different values of VDDH in two level MSV system (VDDL and VDDH). (a) Normalized cost function vs. V_{DDH} value. (b) Max temperature behavior vs. V_{DDH} value. The ratio of the number of blocks that work on high V_{DDH} to total blocks vs. V_{DDH} value. (c) The ratio of the number of blocks that work on high V_{DDH} to total blocks vs. V_{DDH} value. (d) Normalized total power consumption behavior vs. V_{DDH} value.

of the objective function for various values of V_{DDH} is a trade-off between these factors. Fig. 8(c) and (d) illustrates the behavior of the power consumption and temperature with respect to different values of V_{DDH} for different benchmarks.

7.3. Impacts of temperature in MSV design

Recently, it has been demonstrated that the relation of max-temperature and power consumption is not linear and even it does not show monotonic behavior especially when technology scales down [28]. There exist many studies that investigate max-temperature and power consumption separately or together but independently on multi-supply voltage systems. However, the odd dependence of power consumption to temperature has been overlooked in the previous studies in such systems. Some previous studies investigate the effects of the number of islands in power saving and do not consider its effects on temperature [12]. However, considering the effects of both temperature and power produces more accurate results for power consumption of the voltage islands compared to the case that considers only power effects.

To control the number of voltage islands in the proposed framework, we use the value of Frag-Cost coefficient. The result of Fig. 9(a) shows the effects of the coefficient on power saving. The figure reveals that maximum power saving is achieved when the Frag-Cost coefficient is set at its minimum. Also the max-temperature of chip depends on the number of islands. Fig. 9 (b) depicts the dependency of max-temperature of the chip on the number of islands and reveals that the voltage islanding improves temperature behavior of the chip.

Table 6 summarizes the comparison of the proposed MSV and the traditional SA approach in terms of max-temperature and total power consumption of a chip. In the proposed method to capture the MSV potentials we used two voltage levels i.e. V_{DDL} and V_{DDH} and in the traditional SA we used a single voltage level i.e. V_{DDH}. The comparison examines total power, max temperature, area, and wire length for different benchmarks. In our comparison we used the proposed PFVI instead of DFVI, although PFVI has a lower performance than DFVI as we see later. In PFVI we did not consider the effects of area and wiring on temperature and power consumption by making zero their coefficients in the objective function. This renders the PFVI to be configured the same as traditional SA and thus a fair comparison. Even, with PFVI, the proposed method saves power consumption spanning from 14% to 28% and on average 21.9%, and reduces max temperature spanning from 5 K to 16 K and on average 9.4 K. It is worth to mention that in some DAGs many blocks operate on lower voltage levels such as ami33, the proposed method achieves better performance in terms of power and temperature.

Now we attempt to compare the results attained by the proposed method in the previous studies. Unluckily, no previous study considers max-temperature in voltage island partitioning. However, in [16] the authors consider power consumption and thus we may relax the temperature effects in our formulation and hence reach to the same configuration as [16]. To do this comparison we consider two scenarios namely low power density and high power density. Tables 7 and 8 demonstrate these two scenarios. Table 7 reveals that in all benchmarks the peak temperature in the proposed method is less than or equal to the results attained in [16]. Discussing about power consumption however, in some benchmarks such as ami33 the proposed method provides better or equal results compared to [16] and in the remaining benchmarks such as n10b [16] outperforms the proposed method. Table 8 shows the corresponding results for high power densities. Thus we conclude that for the applications with low power density, the temperature is in the range 0–65 °C and the gain is not too much. However, for the higher temperature such as 115–145 °C we noticed the gain is higher.

Finally, Table 9 reports the comparison of post floorplanning VI portioning (PFVI method) versus during floorplanning VI

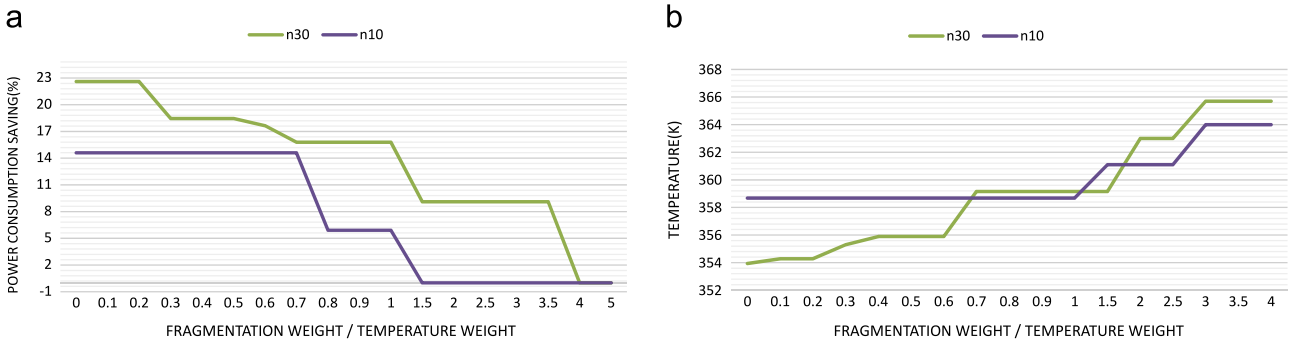


Fig. 9. Temperature and power variation as a function of fragmentation cost coefficient (n10 and n30). (a) Power vs. fragmentation cost coefficient/power coefficient. (b) Temperature vs. fragmentation cost/temperature coefficient.

Table 6
Comparison between the proposed MSV-aware framework and single V_{DD} conventional SA results (P: power (W), T: temperature (K), A: area (mm^2), WL: wire length (mm), execution time (s)).

Benchmark	Area and wire length		SA method (no MSV)			Our MSV platform-PFVI			Our method improvement		
	A (E−06)	WL	Total P	Max T	Execution time	Total P	Max T	Execution time	Total P (%)	Max T	Execution time (%)
n10	2.99	0.129237	19.7874	363.4	220.243	16.8588	358.50	221.402	14.8	4.9	−0.53
n10b	3.07	0.109693	18.9344	358.6	504.32	14.6552	349.5	505.337	22.6	9.5	−0.20
n10c	3.58	0.122913	15.8897	347.9	226.706	11.8060	340.3	227.746	25.7	8.0	−0.46
hp	5.00	0.088825	26.3039	356.6	367.966	20.2803	348.7	368.876	22.9	8.2	−0.25
apte	5.76	0.21404	35.1527	366.8	154.19	27.2785	359.4	154.996	22.4	11.8	−0.52
xerox	6.04	1.03838	37.0397	366.3	219.245	31.0393	358.7	220.23	16.2	5.4	−0.45
ami33	28.4	1.97981	94.461	371.4	3188.35	67.5396	357.2	3213.037	28.5	15.7	−0.77
n30	7.47	0.59270	42.7434	364.6	3589.66	33.2971	354.1	3617.227	22.1	10.6	−0.76

Table 7
Comparison between the proposed and previous power-aware MSV-framework results for low power density benchmarks (P: power (W), T: temperature (K), A: area (mm^2), WL: wire length (mm), execution time (s)).

Benchmark	Area and wire length		Power aware-PFVI [16]			Our MSV platform-PFVI			Our method improvement		
	A (E−06)	WL	Total P	Max T	Execution time	Total P	Max T	Execution time	Total P (%)	Max T	Execution time (%)
n10	2.99	0.129237	16.8588	358.50	220.938	16.8588	358.50	221.402	0.0	0.0	−0.21
n10b	3.07	0.109693	14.1552	352.4	504.998	14.6552	349.5	505.337	−3.5	2.9	−0.07
n10c	3.58	0.122913	12.7160	342.8	227.330	11.8060	340.3	227.746	7.2	2.5	−0.18
hp	5.00	0.088825	22.3803	351.9	368.512	20.2803	348.7	368.876	9.4	3.2	−0.10
apte	5.76	0.21404	27.2785	359.4	154.593	27.2785	359.4	154.996	0.0	0.0	−0.26
xerox	6.04	1.03838	31.0393	358.7	219.902	31.0393	358.7	220.23	0.0	0.0	−0.15
ami33	28.4	1.97981	73.5396	360.8	3203.779	67.5396	357.2	3213.037	8.2	3.6	−0.29
n30	7.47	0.59270	31.7771	358.3	3606.200	33.2971	354.1	3617.227	−4.8	4.2	−0.30

Table 8
Comparison between the proposed and previous power-aware MSV-framework results for high power density benchmarks (P: power (W), T: temperature (K), A: area (mm^2), WL: wire length (mm), execution time (s)).

Benchmark	Area and wire length		Power aware-PFVI [16]			Our MSV platform-PFVI			Our method improvement		
	A (E−06)	WL	Total P	Max T	Execution time	Total P	Max T	Execution time	Total P (%)	Max T	Execution time (%)
n10	3.32	0.136991	41.6412	398.7	251.995	41.6412	398.7	252.398	0	0	−0.16
n10b	2.95	0.099821	33.9725	392.4	464.446	37.8104	380.6	464.91	−11.3	11.8	−0.1
n10c	3.79	0.133975	30.2641	376	252.268	24.7926	365.3	252.798	18.1	10.7	−0.21
hp	5.56	0.097706	55.2793	396.3	383.095	47.4559	381.9	383.631	14.2	14.4	−0.14
apte	6.22	0.246146	69.2874	389.6	170.121	69.2874	389.6	170.496	0	0	−0.22
xerox	7.19	0.986461	74.8047	407.7	235.223	85.3581	392.1	235.646	−14.1	15.6	−0.18
ami33	26.7	2.039204	176.495	416.5	3231.917	148.5871	397.8	3245.167	15.8	18.7	−0.41
n30	7.92	0.669751	79.7605	411.4	3571.056	92.233	389.9	3581.054	−15.6	21.5	−0.28

Table 9
Comparison of the DFVI method results with the PFVI method results in different benchmarks (P: power (W), T: temperature (K), A: area (E−06) (mm^2), WL: wire length (mm), CF: cost function, ET: execution time (s)).

Benchmark	Total P	A	WL	Max T	CF	ET
(a) During floorplanning VI partitioning						
n10	16.42917	3.12109	0.118903	356.97	0.408	324.553
n10b	14.64662	3.05000	0.106910	349.57	0.226	595.85
n10c	11.80544	3.13889	0.125153	342.36	0.187	320.306
hp	20.28797	4.99565	0.088825	348.34	0.080	449.866
apte	27.87874	5.38411	0.201577	359.46	0.253	226.73
xerox	29.33450	6.01690	1.118811	356.75	0.407	307.895
ami33	69.64556	20.4983	1.943318	364.07	0.517	5410.18
n30	32.12703	8.73910	0.590700	352.07	0.180	6070.69
(b) Post floorplanning VI partitioning						
n10	16.8640	2.990	0.129237	357.59	0.486	221.402
n10b	14.6455	3.070	0.109693	349.16	0.256	505.337
n10c	11.8023	3.580	0.122913	339.97	0.300	227.746
hp	20.2880	5.000	0.088825	348.34	0.080	368.876
apte	27.2633	5.760	0.214040	354.97	0.323	154.996
xerox	31.0255	5.650	1.125012	360.89	0.439	220.23
ami33	67.5066	28.40	1.97981	355.75	0.860	3213.037
n30	33.2904	8.471	0.59270	354.02	0.1918	3617.227
Benchmark	Total P (%)	A (%)	WL (%)	Max T (%)	CF (%)	ET (%)
(c) During floorplanning VI partitioning vs. post floorplanning VI partitioning						
n10	−2.575	4.381	−7.9961	−0.60	16.2	−31.78
n10b	0.007	−0.657	−2.5370	0.40	11.8	−15.19
n10c	0.026	−12.36	1.82242	2.38	37.6	−28.90
hp	0.000	−0.080	0.0000	0.00	11.4	−18.00
apte	2.257	−6.520	−5.8227	4.49	21.9	−31.64
xerox	−5.457	6.476	−0.5512	−4.10	73.5	−28.47
ami33	3.168	−27.87	−1.8432	8.30	39.9	−40.61
n30	−3.494	3.171	−0.3374	−1.90	6.05	−37.58

Table 10
Comparison between the proposed heuristic and MILP results (P: power (W), T: temperature (K), A: area (mm^2), WL: wire length (mm), execution time (s)).

Benchmark	Area and wire length		Heuristic			MILP			Difference		
	Area	Wire length	T_{max}	P_{total}	Execution time	T_{max}	P_{total}	Execution time	T_{max}	P_{total} (%)	Execution time (%)
n10	2.99	0.129237	357.73	16.68	0.193	357.65	16.43	1.159	0.08	1.52	600
n10b	3.07	0.109693	353.88	15.34	0.112	350.76	14.55	1.017	3.12	5.43	908
n10c	3.58	0.122913	343.40	11.81	0.201	343.40	11.81	1.040	0.00	0.00	517
n30	7.47	0.59270	350.85	32.08	0.854	350.40	31.20	27.567	0.44	2.82	3228
ami33	28.4	1.97981	357.90	66.72	0.581	357.86	66.57	24.687	0.04	0.22	4249
ami49	36.7	2.33482	351.72	45.33	0.920	351.70	45.08	31.531	0.02	0.56	3427
apte	5.76	0.21404	357.64	28.11	0.105	356.94	27.77	0.806	0.70	1.22	768
hp	5.00	0.088825	347.79	20.76	0.122	347.79	20.76	0.910	0.00	0.00	746
xerox	6.04	1.03838	355.83	29.33	0.141	355.83	29.33	0.985	0.00	0.00	699

portioning (DFVI method) for low power applications. In DFVI, we find better cost functions (on average 21%). PFVI results are less accurate than DFVI results. Thus, trade-offs accuracy for the execution time. Hence, we may reach to the interesting result that difference between the obtained results of PFVI and DFVI is not significant and the lower time complex PFVI may be used for low power applications.

7.4. Heuristic results

In this subsection, we compare the results of the proposed heuristic and its MILP counterpart for different types of benchmarks. In order to do fair comparison, we use the same floorplan in the above methods and consequently they have the same area and wire length. In our setup we consider peak temperature and total power consumption in the objective function of the VI partition part of the problem.

Table 10 demonstrates this comparison. As shown in Table 10, there is a slight discrepancy between the obtained results of the heuristic and MILP approach which is bounded above at 3 K and on average 0.5 K for temperature, and at 5.43% and on average 1.31% for total power consumption. It demonstrates that the proposed heuristic may be employed in scenarios that the computational power is scarce and the execution time is critical. This is achieved with the cost of sacrificing accuracy for the execution time of the algorithm.

8. Conclusion

Emerging embedded systems and applications span a diverse spectrum and finding a general methodology to design such systems that satisfy the requirements of a wide range of applications is a challenge of growing importance. The critical path characteristics of a task graph have impacts on temperature and

power for applications with hard real-time constraints. In this paper we captured these two major issues by presenting a multi-objective optimization formulation that optimizes multiple conflicting objectives subject to certain constraints for embedded systems with substantially different design constraints. In the proposed framework we combined multi-objective ILP formulation with a simulated annealing algorithm to achieve practicality and attain lower execution time compared to only total ILP modeling. Furthermore, a heuristic algorithm is presented for scenarios that their design space is rather large and finding the optimal solution or Pareto optimal set of them is a formidable and time consuming task.

The experimental results reveal that significant improvements are obtained in power saving, peak temperature reduction and island fragmentation for almost all the benchmarks, which confirms the usefulness of the proposed approach. The experimental results show that, due to considering power and temperature simultaneously, the proposed framework suggests floorplans that are more power-efficient compared to the cases that only attempt to optimize the temperature and more temperature-efficient compared to the cases which only optimize the power. Moreover, we reached to an interesting result that demonstrated that by increasing the supply voltage level of the MSV-aware chip the total power and peak temperature may reduce. A future line of this research is studying the critical path of the application graph impacts on temperature and power in the presence of process variation in MSV-aware embedded systems.

References

[1] International Technology Roadmap for Semiconductors, Online at ⟨http://www.itrs.net/Links/2010ITRS/Home2010.htm⟩.
[2] W. Liao, L. He, K. Lepak, Temperature and supply voltage aware performance and power modeling at microarchitecture level, IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 24 (7) (2005) 1042–1053.
[3] Y. Liu, R.P. Dick, L. Shang, H. Yang, Accurate temperature dependent integrated circuit leakage power estimation is easy, in: Proceedings of the Design, Automation and Test in Europe Conference, March 2007, pp. 1526–1531.
[4] A.H. Ajami, K. Banerjee, M. Pedram, Modeling and analysis of nonuniform substrate temperature effects on global VLSI interconnects, IEEE Trans. CAD 24 (June (6)) (2005) 849–861.
[5] W. Jang, D. Ding, D.Z. Pan, A voltage–frequency island aware energy optimization framework for networks-on-chip, in: Proceedings of the International Conference on Computer-Aided Design, 2008.
[6] W.P. Lee, Y.W. Chang, Voltage island aware floorplanning for power and timing optimization, in: Proceedings of the International Conference on Computer-Aided Design, 2006, pp. 389–394.
[7] W.P. Lee, H.Y. Liu, Y.W. Chang, An ILP algorithm for post-floorplanning voltage-island generation considering power network planning, in: Proceedings of the International Conference on Computer-Aided Design, 2007, pp. 650–655.
[8] Q. Ma, E.F.Y. Young, Voltage island-driven floorplanning, in: Proceedings of the International Conference on Computer-Aided Design, 2007, pp. 644–648.
[9] Q. Ma, F.Y. Young, Network flow-based power optimization under timing constraints in MSV-driven floorplanning, in: Proceedings of the International Conference on Computer-Aided Design, 2008.
[10] W. Lee, H. Liu, Y. Chang, Voltage-island partitioning and floorplanning under timing constraints, IEEE Trans. Comput. Aided Des. Integr. Circuit Syst. 28 (May (5)) (2009) 690–702.
[11] Y. Cai, B. Liu, Q. Zhou, X. Hong, A thermal aware floorplanning algorithm supporting voltage islands for low power SOC design, in: Proceedings of the Power and Timing Modeling, Optimization and Simulation, 2005.
[12] J. Hu, Y. Shin, N. Dhanwada, R. Marculescu, Architecting voltage islands in core-based system-on-a-chip designs, in: Proceedings of the International Symposium on Low Power Electronics and Design, 2004, pp. 180–185.
[13] W.L. Hung, G.M. Link, Y. Xie, N. Vijaykrishnan, N. Dhanwada, J. Conner, Temperature-aware voltage islands architecting in system-on-chip design, in: International Conference on Computer Design (ICCD), 2005, pp. 689–696.
[14] B. Liu, Y. Cai, Q. Zhou, X. Hong, Power driven placement with layout aware supply voltage assignment for voltage island generation in dual-vdd designs, in: Proceedings of the Asia and South Pacific Design Automation Conference, 2006, pp. 582–587.
[15] R.L.S. Ching, E.F.Y. Young, Post-placement voltage island generation, in: Proceedings of the International Conference on Computer-Aided Design, 2006, pp. 641–646.
[16] W.K. Mak, J.W. Chen, Voltage island generation under performance requirement for SoC designs, in: Proceedings of the Asia and South Pacific Design Automation Conference 2007, pp. 798–803.
[17] H. Wu, I. M. Liu, M. Wong, Y. Wang, Post-placement voltage island generation under performance requirement, in: Proceedings of the International Conference on Computer-Aided Design, 2005, pp. 309–316.
[18] H. Wu, M. Wong, I. M. Liu, Timing-constrained and voltage-island-aware voltage assignment, in: Proceedings of the Design Automation Conference, 2006, pp. 429–432.
[19] D.F. Wong, D.L. Liu, A new algorithm for floorplan design, in: Proceedings of the Design Automation Conference, 1986, pp. 101–107.
[20] A. Gupta, N. Dutt, F. Kurdahi, K. Khouri, M. Abadir, LEAF: a system level leakage-aware floorplanner for SOCs, in: Proceedings of the 2007 Asia and South Pacific Design Automation Conference (ASP-DAC), 2007, pp. 274–279.
[21] K. Sankaranarayan, S. Velusamy, M. Stan, K. Skadron, A case for thermal-aware floorplanning at the microarchitectural level, J. Instr. Level Parallelism 8 (2005) 1–16.
[22] M.B. Healy, M. Vittes, M. Ekapanyapong, C.S. Ballapuram, S.K. Lim, H.H.S. Lee, G.H. Loh, Multiobjective microarchitectural floorplanning for 2-D and 3-D ICs, IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 26 (1) (2007) 38–52.
[23] M.C. Wu, Y.W. Chang, Placement with alignment and performance constraints using the B*-tree representation, in: Proceedings of the International Conference on Computer Design, 2004, pp. 568–571.
[24] M. Healy, M. Vittes, M. Ekpanyapong, C. Ballapuram, S.K. Lim, H.H.S. Lee, G.H. Loh, Microarchitectual floorplanning under performance and temperature trade off, in: Proceedings of the Design, Automation and Test in Europe, 2006, pp. 1288–1293.
[25] S. Logan, M.R. Guthaus, Fast thermal-aware floorplanning using white-space optimization, in: Seventeenth IFIP International Conference on Very Large Scale Integration (VLSI-SoC), vol. 12, 2009, pp. 65–70.
[26] R.T. Marler, J.S. Arora, Survey of multi-objective optimization methods for engineering, Struct. Multidiscip. Optim. 26 (2004) 369–395.
[27] R. Dick, Embedded Systems Synthesis Benchmarks Suite (e3s). Online at ⟨http://www.ece.northwestern.edu/dickrp/e3s/⟩.
[28] T. Chantem, X.S. Hu, R.P. Dick, Temperature-aware scheduling and assignment for hard real-time applications on MPSoCs, IEEE Trans. VLSI Syst. 19 (10) (2011) 1884–1897.
[29] Y. Cao, L.T. Clark, Mapping statistical process variations toward circuit performance variability: an analytical modeling approach, in: Proceedings of the 42nd Design Automation Conference, June 2005.
[30] K. Skadron, T. Abdelzaher, M.R. Stan, Control-theoretic techniques and thermal-RC modeling for accurate and localized dynamic thermal management, High Perform. Comput. Archit. (2002) 17–28.
[31] Al Krum, Thermal management, in: Frank Kreith (Ed.), The CRC Handbook of Thermal Engineering, CRC Press, Boca Raton, FL, 2000, pp. 2.1–2.92.
[32] Berkeley Predictive Technology Model. ⟨http://www-device.eecs.berkeley.edu/⟩.
[33] K. Banerjee, A. Mehrotra, Global (interconnect) warming, IEEE Circuits Devices Mag. 17 (September) (2001) 16–32.
[34] R.H.J.M. Otten, R.K. Brayton, Planning for performance, in: DAC '98: Proceedings of the 35th Annual Conference on Design Automation, 1998, pp. 122–127.
[35] F. Chen, A. Joshi, V. Stojanovic, A. Chandrakasan, Scaling and evaluation of carbon nanotube interconnects for VLSI applications, in: Proceedings of the Nano-Networks, 2007.
[36] B. Khodabandeloo, A. Khonsari, F. Gholamian, M.H. Hajiesmaili, A. Mahabadi, H. Noori, Scenario-based quasi-static task mapping and scheduling for temperature-efficient MPSoC design under process variation, Microprocess, Microsy. 38 (5) (2014) 399–414.
[37] J. Kong, S.W. Chung, K. Skadron, Recent thermal management techniques for microprocessors, ACM Comput. Surv. (CSUR) 44 (13) (2012), Online at ⟨http://www.cs.virginia.edu/~skadron/⟩.
[38] HotSpot, ⟨http://lava.cs.virginia.edu/HotSpot/⟩, 2011.
[39] G. De Micheli, Synthesis and Optimization of Digital Circuits, McGraw-Hill Book Company, New York, 1994.
[40] GSRC Floorplanning Benchmark Suite. Online at ⟨http://www.cse.ucsc.edu/research/surf/GSRC/progress.html⟩.
[41] MCNC Floorplanning Benchmark Suite. Online at ⟨http://www.cse.ucsc.edu/research/surf/GSRC/MCNCbench.html⟩.
[42] CPLEX 11.1ILOG: ⟨http://www.ilog.com/product/cplex/⟩, 2011.
[43] N.B. Rizvandi, J. Taheri, A.Y. Zomaya, Some observations on optimal frequency selection in DVFS-based energy consumption minimization, J. Parallel Distrib. Comput. 71 (2011) 1154–1164.
[44] W. Huang, K. Rajamani, M.R. Stan, K. Skadron, Scaling with design constraints: predicting the future of big chips, Micro IEEE 31 (4) (2011) 16–29.
[45] G. Link, Vijaykrishnan, Thermal trends in emerging technologies, in: Proceedings of the Seventh International Symposium on Quality Electronic Design (ISQED), 2006, pp. 625–632.
[46] P.H. Jones, J.W. Lockwood, Y.H. Cho, A thermal management and profiling method for reconfigurable hardware applications, in: Sixteenth International Conference on Field Programmable Logic and Applications (FPL), Madrid, Spain, August 2006.