Energy Efficient Application Mapping to NoC Processing Elements Operating at Multiple Voltage Levels

Pavel Ghosh and Arunabha Sen
Department of Computer Science and Engineering
Arizona State University, Tempe, AZ
Email: {pavel.ghosh, asen}@asu.edu

Alexander Hall
Department of EECS
UC Berkeley, CA, USA
Email: alex.hall@gmail.com

Abstract

An efficient technique for mapping application tasks to heterogeneous processing elements (PEs) on a Network-on-Chip (NoC) platform, operating at multiple voltage levels, is presented in this paper. The goal of the mapping is to minimize energy consumption subject to the performance constraints. Such a mapping involves solving several subproblems. Most of the research effort in this area often address these subproblems in a sequential fashion or a subset of them. We take a unified approach to the problem without compromising the solution time and provide techniques for optimal and heuristic solutions. We prove that the voltage assignment component of the problem itself is NP-hard and is inapproximable within any constant factor. Our optimal solution utilizes a Mixed Integer Linear Program (MILP) formulation of the problem. The heuristic utilizes MILP relaxation and randomized rounding. Experimental results based on E3S benchmark applications and a few real applications show that our heuristic produces near-optimal solution in a fraction of time needed to find the optimal.

1 Introduction

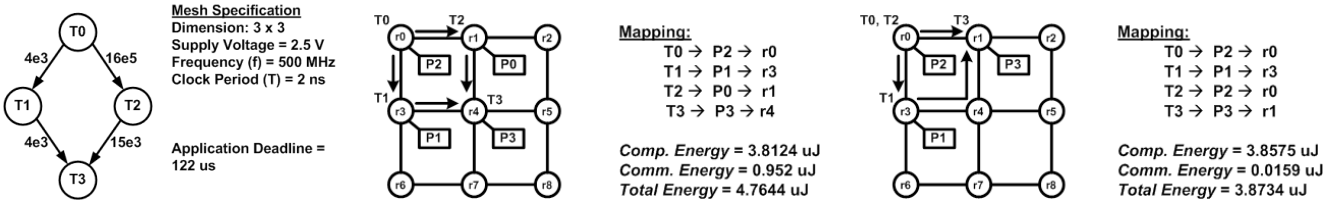
In recent years System-on-Chip (SoC) design has become extremely challenging due to the increasing complexities in processor and semiconductor technologies. Multi-core SoC based embedded systems may contain either all homogeneous generic processing cores, or a varying number of heterogeneous PEs. These heterogeneous PEs may represent programmable general purpose cores, task specific co-processors or hardware accelerators, etc. Network-on-Chip (NoC) architectures provide an alternative to the bus-based communication mechanism that can meet the challenging requirements of performance, scalability and flexibility [1, 2]. As the number of PEs on a SoC and the data traffic between them continues to grow, energy minimization subject to performance constraint has become one of the most important objectives.

The problem of minimizing energy consumption during application execution while satisfying the performance constraints can be divided into four main subproblems: (i) mapping of the application tasks to the PEs, (ii) mapping of the PEs to the routers of the NoC architecture, (iii) assigning operating voltages to the PEs (in case they can operate at multiple voltages) and (iv) routing of data paths, i.e., traffic movement on the NoC architecture. As consideration of all four subproblems simultaneously increases the complexity of the problem, most of the research effort in this domain [4, 6, 8, 9] either solve problems (i), (ii), (iii) and (iv) in a sequential fashion, or solve only a subset of them.

To find an energy efficient application mapping, all four problems (i)-(iv) have to be solved. There are two options available - solve them sequentially or solve them in a unified way. The sequential approach has manifold disadvantages. Firstly, decision taken at an early phase may turn out to be expensive later. Secondly, because of some earlier decisions may lead to violation of constraints at some later phase, and thus resulting in re-execution of all the steps multiple times involving an enormous amount of computation.

We show with a motivating example here and later with extensive experimental results that the sequential approach may lead to sub-optimal solution. To the best of our knowledge, our proposed technique is the first that unifies all the four subproblems under a single problem formulation and develops optimal and heuristic solutions for the problem. Although scaling down voltage levels of PEs is favorable for reduction of energy consumption, excessive number of voltage islands may be detrimental from the perspective of physical design as it creates voltage island fragmentation of the chip and increases the complexity of the power delivery network. Therefore, the number of voltage islands on the chip should follow an upper bound. In literature [7, 8], the constraint on the maximum number of voltage either has not been captured properly or involves a huge computation

978-1-4244-4143-3/09/$25.00 ©2009 IEEE



(a) Application Task Graph (b) Mapping using 2-step approach (c) Mapping using 1-step approach

Figure 1. Task Graph and Mappings

time for the solution. We incorporate this constraint by an efficient formulation of the upper bound on the number of voltage islands created.

The motivation behind our unified approach comes from the following example. Fig. 1(a) shows an example application task graph consisting of four tasks $T0$, $T1$, $T2$ and $T3$. The edges represent the task dependencies and the labels on the edges represent the inter-task communication volume in *number of bytes*. Tables 1 and 2 show the execution time and power consumption of these four tasks on four available PEs $P0$, $P1$, $P2$ and $P3$, respectively.

Following a sequential approach, the resultant mapping is as shown in Fig. 1(b), with computation energy consumption $3.8124\mu J$ and communication energy consumption $0.952\mu J$. Thus, the overall energy consumption for the application is $(3.8124 + 0.952) = 4.7644\mu J$. With such a mapping, the execution finish-time of the application is $86.104\mu s$, well within the specified deadline of $122\mu s$. For our proposed unified mapping, as shown in Fig. 1(c), the total energy consumption is $3.8734\mu J = 3.8575\mu J + 0.0159\mu J$, leading to 18.7% save of energy as compared with the sequential approach. With this mapping, the execution finish-time of the application is $84.852\mu s$, still within deadline of $122\mu s$.

Energy consumption can be further reduced if we take advantage of operating the PEs of the NoC at multiple voltage levels. When the PEs are allowed to operate at different voltages, a certain amount of energy will be consumed by the generation of additional clock signals, voltage level converters and mixed clock-mixed voltage FIFOs used by the level shifters connecting adjacent PEs in NoC architecture.

From the example above, it is clear that a unified approach to application mapping to NoC PEs, operating at multiple voltages leads to better energy utilization than a sequential approach. The energy consumption model considered in this paper has three components - (i) energy consumption due to computation, (ii) energy consumption due to communication and (iii) voltage transition energy between adjacent PEs operating at different voltages.

In this paper, we have assumed a regular mesh architecture as the communication infrastructure, where each router has 5 ports. One of the ports is used for connecting it to a PE and the other four are for connection to the neighboring routers. The algorithms proposed in this paper can be used with any other NoC topologies as well. Communication power consumption parameter values used for the evaluation of our techniques are taken from [9].

Table 1. Execution time (in seconds) for task-processor pair

Tasks	P0	P1	P2	P3
T0	7.7e-06	7e-06	6.9e-06	5.8e-06
T1	4.1e-06	5.2e-06	4.9e-06	4e-06
T2	7.2e-06	4.2e-06	5.95e-06	4.4e-06
T3	5.47e-05	4.15e-05	5.95e-05	7.2e-05

Table 2. Task Power (in Watts) for task-processor pair

Tasks	P0	P1	P2	P3
T0	0.16	0.14	0.12	0.15
T1	0.07	0.045	0.065	0.077
T2	0.102	0.25	0.131	0.21
T3	0.048	0.084	0.041	0.028

2 Problem Formulation

In this section, we provide formal definition of the application mapping problem. The input instance to this problem is explained in Table 3. The output of the problem is as shown in Table 4. The objective is to minimize the overall energy consumption, such that:

1. All the application tasks finish their execution before deadline D.
2. Dependencies among the tasks are maintained.
3. Bandwidth constraint on each router link is satisfied.
4. The total number of voltage islands created does not exceed κ.
5. Total energy consumption (computation + communication + voltage transition) is minimized.

3 Computational Complexity

In this section we define the *voltage assignment* problem, subproblem (iii) of the application mapping problem and prove it to be NP-complete and inapproximable within any constant factor. If $M_1(t_j) = p_i$, then for each p_i we can find out an allowable set of voltage levels $L_i =$

Table 3. Input of the Problem

Symbols	Explanation
$G(V_T, E_T)$	Set of tasks $V_T = \{t_1, t_2, \ldots, t_m\}$; directed edge $e_{ij} = (t_i, t_j) \in E_T$ representing the dependency of task t_j on task t_i
w_{ij}	For edge $e_{ij} \in E_T$, the data communication volume (in *Mbps*) between tasks t_i and t_j
D	Application deadline, by which all the tasks need to be completed
κ	Maximum number of voltage islands created
P	A set of processing elements $\{p_1, p_2, \ldots, p_k\}$
$P_t \subseteq P$	For each task $t \in V_T$, a subset of PEs, potential to execute the task t
V_p	For each PE $p \in P$, an allowable set of voltage levels $\left\{v_1, v_2, \ldots, v_{n_p}\right\}$ in which this PE can operate, where n_p denotes number of distinct voltage levels it can operate
$\tau(t, p, v_p)$	Execution time of task $t \in V_T$ on PE $p \in P_t$, when p is operating at voltage $v_p \in V_p$
$\epsilon(t, p, v_p)$	Computation energy consumption of task $t \in V_T$ on PE $p \in P_t$, when p is operating at voltage $v_p \in V_p$
$G_R(V_R, E_R)$	An undirected $n \times n$ mesh architecture graph, where each node $r \in V_R$ denotes a router in the NoC architecture, whereas each edge $e_{ij} = (r_i, r_j)$ represents a router link in the NoC architecture between those two router nodes
β	Capacity of each link $e_{ij} \in E_R$
λ	Data communication latency on each link $e_{ij} \in E_R$
L_{ij}	Link length of each link $e_{ij} \in E_R$
ψ_i	Communication energy consumption rate at the router input port = $328nW/Mbps$[9]
ψ_o	Communication energy consumption rate at the router output port = $65.5nW/Mbps$[9]
ψ_l	Communication energy consumption rate at the router links = $79.6nW/Mbps/mm$[9]
$\alpha_{v_k v_l}$	Power consumption by the level shifter when two adjacent PEs are operating at voltages v_k and v_l, respectively

Table 4. Output of the Problem

Output	Explanation
Task to PE Mapping function M_1 :	$M_1(t_i) = p_j$, which means that task t_i has been mapped to PE p_j; $t_i \in V_T$, and $p_j \in P_{t_i} \subseteq P$
PE to Router Mapping Function M_2 :	$M_2(p_i) = r_j$, which means that PE p_i has been mapped to NoC router r_j; $p_i \in P$ and $r_j \in V_R$
PE to Voltage Assignment Function M_3 :	$M_3(p_i) = v_j$, which means that PE p_i has been assigned to voltage v_j; $p_i \in P$ and $v_j \in V_{p_i}$
Task Edge to Path in Mesh Mapping Function M_4 :	If $e_{ix} \in E_T$, $M_1(t_i) = p_j$, $M_2(p_j) = r_k$ and $M_1(t_x) = p_y$, $M_2(p_y) = r_z$, then $M_4(t_i, t_x) = r_k \rightarrow r_{\pi(1)} \rightarrow r_{\pi(2)} \rightarrow \ldots \rightarrow r_{\pi(l)} \rightarrow r_z$, where $\{r_1, r_{\pi(1)}\}, \{r_{\pi(l)}, r_z\}, \{r_{\pi(q)}, r_{\pi(q+1)}\} \in E_R, \forall q \in [1, l-1]$ and $\pi(q) \neq k, z, \forall q \in [1, l-1]$

$\{v_{i1}, v_{i2}, \ldots, v_{in_i}\}$, such that the task t_j can meet its deadline d_j (obtained using deadline D and communication latencies) whenever p_i is operating at the voltage levels in L_i. We consider two components for energy consumption:
(1) Energy consumption by PE p_i when operating at voltage $v \in L_i$. We denote this component by $\eta^v_{p_i}$.
(2) Energy consumption by the level shifter connecting two adjacent PEs p_i and p_j, where p_i is operating at voltage x and p_j is operating at voltage y, denoted by α_{xy}.

Voltage Assignment Problem: Given an $n \times n = N$ grid, where each node represents a router in the NoC architecture with a PE attached to it and a list L_i of allowable operating voltages associated with each PE p_i, the problem is to assign a voltage v_j to PE p_i, where $v_j \in L_i$ for all the PEs, such that the following energy consumption expression is minimized:

$$\sum_{i=1 \text{ to } N, v=M_3(p_i)} \eta^v_{p_i} + \sum_{x=M_3(p_i), y=M_3(p_j), e_{ij} \in E_R} \alpha_{xy}$$

The *voltage assignment* problem has one-to-one correspondence with the *minimum weight grid coloring* problem, defined as follows:

Minimum Weight Grid Coloring Problem: Given a grid graph $G = (V, E)$ of dimension $a \times b = M$. Each node $u \in V$ has an associated set of colors $C_u = \{1, \ldots, K_u\}$ each with a certain *color-cost* $c(k) \in \Re^+, k \in C_u$. Let $u, w \in V$ be neighbors in the grid-graph, i.e., $\{u, w\} \in E$. We are also given *combination-costs* $c(k, l) \in \Re^+$ for each color combination $k \in C_u$ and $l \in C_w$. The goal of the *minimum weight grid coloring* problem is to find a coloring, i.e., for each node $u \in V$ a color $k \in C_u$, which minimizes the following objective function:

$$\sum_{v \in V} c(k_u) + \sum_{\{u,w\} \in E} c(k_u, k_w)$$

We now prove the decision version of this problem, called ***MINCOL***, to be NP-hard by reducing an instance of the known NP-hard problem ***GRAPH 3-COLORABILITY with no vertex degree exceeding 4*** [5]-(page 85) to an instance of ***MINCOL***.

Theorem 1. Min weight grid coloring is NP-hard and even inapproximable within any factor.

Proof. Let $G' = (V', E')$ denote the given planar graph. From this graph we construct an instance of the min weight grid coloring problem as follows.

We embed G' in a grid graph $G = (V, E)$ of dimension $O(n') \times O(n')$, where n' is the number of nodes in G', using a polynomial algorithm for computing an orthogonal representation. Let $W \subseteq V$ denote the subset of nodes in G which corresponds to the embedded node set V' of the given planar graph. For each edge $e' = \{u', w'\} \in E'$ we denote by $p_{e'} = u_1, e_1, u_2, \ldots, e_{j-1}, u_j$ the path corresponding to the embedding of e'. $P = \bigcup_{e' \in E'} p_{e'}$ denotes the set of all such paths. Fig. 2 gives an example of how such an embedding could look like for a simple graph

(a) Example planar graph

(b) Embedding into a grid graph of dimension 6×6

Figure 2. Planar graph and its embedding

with four nodes. With each node $u \in V$ of G we associate 3 colors $C_u = \{1, 2, 3\}$. We set all color-costs $c(k) = 0$ for $k \in C_u$ and $u \in V$. Similarly, all combination-costs of edges $\{u, w\} \in E$ contained in none of the paths, i.e., $\{u, w\} \notin P$, are set to zero as well: $c(k, l) = 0$ for $k \in C_u$ and $l \in C_w$. For each path $p \in P$ we set the combination-costs as follows:

i) For $e_1 = \{u_1, u_2\}$ we set $c(k, l) = 0$ for $k \in C_{u_1}, l \in C_{u_2}$ and $k \neq l$. For remaining combinations with $k = l$ we set $c(k, l) = 0$. In other words, if u_1 and u_2 are assigned *different* colors, we have cost 0, otherwise 1.

ii) For e_i with $i \in \{2, \ldots, j-1\}$ we set $c(k, l) = 1$ for $k \in C_{u_i}, l \in C_{u_{i+1}}$ and $k \neq l$. For the remaining combinations with $k = l$ we set $c(k, l) = 0$. In other words, if u_i and u_{i+1} are assigned *the same* color, we have cost 0, otherwise cost 1.

If we aim for a total cost of 0, the path p will propagate the color chosen for u_j all the way to u_2. For the cost to remain at zero, u_1 and u_2 (and therefore u_j) must be colored with different colors. Therefore, the given planar graph G' is 3-colorable, if and only if there is a solution to the constructed min weight grid coloring problem of total cost zero. This proves the problem to be NP-hard. The inapproximability within any factor follows, since any approximation algorithm with multiplicative approximation ratio ρ and additive factor ρ' could be used to decide whether G' is 3-colorable as well: simply multiply all combination costs by $\rho' + 1$. If G' is 3-colorable, the optimal solution has cost 0 and therefore the approximation algorithm must find a solution with cost $\leq \rho \cdot 0 + \rho' = \rho'$. Otherwise, if G' is not 3-colorable, any solution has cost $\geq \rho' + 1$. Hence, the approximation algorithm distinguish between two cases. □

4 Optimal Solution for Application Mapping

In this section we use the mathematical programming techniques to solve the application mapping problem. We formulate the problem as a Mixed Integer Linear Program (MILP). In Table 5 we define a few of the variables used in the MILP formulation and also declare their types. The parameters used for calculation of energy consumption and execution time are as follows:

τ_{tpv} = execution time of task $t \in V_T$ on PE $p \in P_t$ at voltage level $v \in V_p$ (in *sec*)

ϵ_{tpv} = computation energy consumption for task $t \in V_T$ on PE $p \in P_t$ at voltage level $v \in V_p$ (in *Joules*)

$\alpha_{v_1 v_2}$ = voltage transition energy consumption parameter for two adjacent nodes operating at voltage levels v_1 and v_2, respectively (in *Joules*)

Table 5. Varibales used in MILP Formulation

Variable	Type	Definition
δ_{tp}^{v}	Binary	1, if $t \in V_T, p \in P_t \subseteq P, v \in V_p$, and $M_1(t) = p$ and $M_3(p) = v$; 0, otherwise
ζ_{rv}	Binary	1, if $r \in V_R$ operating at voltage v, i.e., $M_2(p) = r$ and $M_3(p) = v$ for some $p \in P$; 0, otherwise
f_{xy}^{ij}	Binary	1, if $e_{ij} \in E_T, e_{xy} \in E_R$, and $e_{xy} \in M_4(e_{ij})$; 0, otherwise

The objective of the application mapping problem is to minimize the energy consumption of the system subject to the application deadline constraint, mesh interconnection link bandwidth constraint and maximum allowed number of voltage islands constraint, i.e.,

$$\text{Obj: } minimize\ E = E_c + E_r + E_l + E_{vt}$$

where E_c is the computation energy consumption, E_r and E_l are the communication energy consumed at the router ports and links, respectively and E_{vt} is the voltage transition energy consumption, and calculated as:

$$E_c = \sum_{t \in V_T} \sum_{p \in P_t} \sum_{v \in V_p} \left(\delta_{tp}^{v} \epsilon_{tpv}\right)$$

$$E_r = \sum_{\forall e_{xy} \in E_R} \sum_{\forall e_{ij} \in E_T} \left(f_{xy}^{ij} w_{ij}\right) (\psi_i + \psi_o)$$

$$E_l = \sum_{\forall e_{xy} \in E_R} \sum_{\forall e_{ij} \in E_T} \left(\left(f_{xy}^{ij} w_{ij}\right) L_{xy}\right) \psi_l$$

$$E_{vt} = \sum_{v_1 \in V} \sum_{v_2 \in V} \sum_{\forall e_{xy} \in E_R} \left(\zeta_{x v_1} \zeta_{y v_2}\right) \alpha_{v_1 v_2}$$

In order to eliminate the quadratic term in the expression of E_{vt}, we define the following decision variable:

$\theta_{xy}^{v_1 v_2} = 1$, if $\zeta_{x v_1} = 1$ and $\zeta_{y v_2} = 1$, otherwise 0

$$\text{Hence, } E_{vt} = \sum_{v_1 \in V} \sum_{v_2 \in V} \sum_{\forall e_{xy} \in E_R} \theta_{xy}^{v_1 v_2} \alpha_{v_1 v_2}$$

Constraints:

Due to the limitations in space, here we omit the mathematical details of the formulations of the constraints.

(1) Each task executes on exactly one PE, at exactly one voltage, connected to exactly one router node.

(2) Each router operates at exactly one voltage level and is attached with at most one processor.

(3) Each task needs to finish within the specified application deadline, and need to maintain task dependencies.

(4) Bandwidth constraint on each router link: the total flow passing through a link does not exceed its bandwidth.

(5) The number of voltage islands is less than or equal to the specified maximum allowed value κ.

5 Heuristic for Application Mapping

In this section we describe our proposed MILP relaxation and randomized rounding based heuristic. The algorithm takes as input the parameters specified in the Section 2 and the corresponding MILP formulation. Output of the algorithm consists of the mappings M_1, M_2, M_3 and M_4, defined in Table 4. The heuristic is explained below.

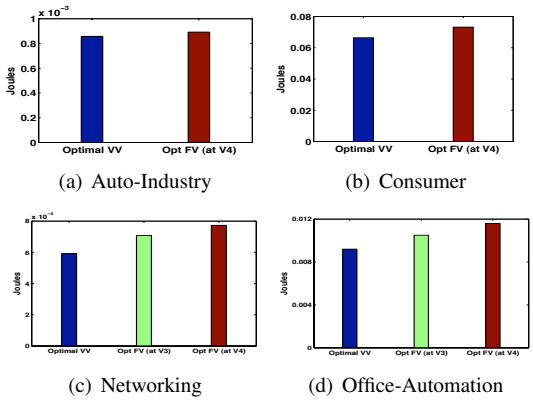
Algorithm 1 Randomized Rounding based Heuristic

```
1:  Relax all the integer constraints in the MILP (/* relaxation */)
2:  Solve the relaxed LP using CPLEX
3:  while (Solution NOT Integral) do
4:      Set the integral variables in LP solution as constants and leave them unaltered during further iterations (/* variable fixing */)
5:      for (each task t ∈ V_T)  (/* rounding */) do
6:          repeat
7:              Round δ^v_tp to 1 with probability as its value
8:          until (δ^v_tp = 1 for some p and v)
9:          repeat
10:             Round γ_tr to 1 with probability as its value
11:         until (γ_tr = 1 for some r)
12:         Round ζ_rv variables following co-relation constraints
13:         Round σ_rp variables following co-relation constraints
14:     end for
15:     Solve modified LP
16:     if (NO Constraint Violation) then
17:         if (Integer Solution) then
18:             Solution Found - break out of outer while loop
19:         end if
20:     else
21:         Eliminate constraint violations by modifying the rounding of δ^v_tp and
            γ_tr variables for the corresponding t ∈ T
22:     end if
23: end while
24: return Solution in terms of functions M_1, M_2, M_3 and M_4
```

6 Experimental Results

In this section we present the experimental results to evaluate our proposed approach. All our experiments can be classified into the following three categories considering the evaluation goal of the experiment:
i) The optimal solution for variable voltage setup is compared with the optimal solution for fixed voltage setup.
ii) The optimal solution of our proposed unified approach is compared with that of the sequential approach.
iii) The quality of the heuristic solution is evaluated by comparing it with the optimal solution.

The experiments are performed using applications (*auto-industry*, *consumer*, *networking* and *office-automation*) from the E3S benchmark suite [3] and three real applications MPEG4, MWD (Multi-Window Display) and OPD (Object Plane Decoder). The following six voltage setups were used, the first one as variable voltage (VV) setup and the other five as fixed voltage (FV) setups: **a)** Case I : Voltage level varies in the range from V_0 to V_4, **b)** Case II : Voltage level is fixed at $V_0 = 1.9V$, **c)** Case III : Voltage level is fixed at $V_1 = 2.3V$, **d)** Case IV : Voltage level is fixed at $V_2 = 2.5V$, **e)** Case V : Voltage level is fixed at $V_3 = 3.3V$, **f)** Case VI : Voltage level is fixed at $V_4 = 3.6V$.



(a) Auto-Industry (b) Consumer

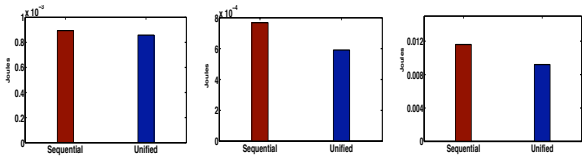
(c) Networking (d) Office-Automation

Figure 3. Optimal with variable voltage levels vs. Optimal with fixed voltage Energy Consumption Comparison for E3S Benchmark

For low voltage setups, no feasible solution was found. On the other hand, for high voltage setups feasible solutions were found with higher energy consumption values. The heuristic is implemented in C++. The MILP for achieving the optimal solution and the corresponding relaxed LP as part of the heuristic were executed on the same machine using ILOG CPLEX 10.0 Concert technology.

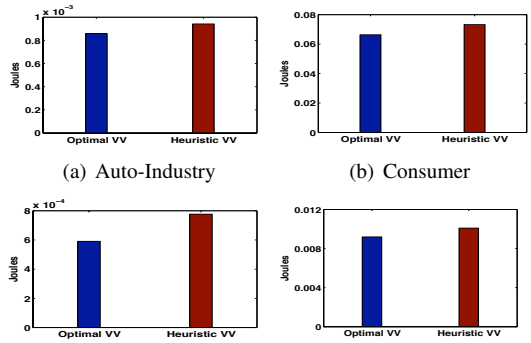
Experiments with E3S Benchmark Applications: Mesh dimension 4×4 was used for the auto-industry application and 3×3 for all the other three applications. The value for the maximum number of voltage islands κ was set to 4 and 3, respectively, for these two different sizes of mesh topologies. Fig. 3 compares the optimal solution while using the variable voltage (VV) levels with the optimal solution while setting a fixed voltage (FV) level, i.e., optimal at Case I with the optimals for all other cases. On average for these four applications, the variable voltage setup can save 18% energy consumption over the fixed voltage setup.We compare the solution quality of our proposed unified approach with the sequential approach in Fig. 4. For these four applications, on average, we are able to save 16% energy consumption. Fig. 5 shows the quality of our proposed heuristic as compared with the optimal solution. i.e., the optimal and heuristic solution for Case I. In all the cases, the heuristic is able to find near-optimal solutions. Fig. 6 shows that the time taken to obtain the heuristic solution is negligible compared to that required to solve the MILP for the optimal solution.

Experiments with Real Applications: Mesh dimension 3×3 for MPEG4 and MWD applications and 4×4 was used for the OPD application. The value of maximum number of voltage islands κ was set to 3 and 4, respectively, for these two different sizes of mesh. Fig. 7 compares the optimal solution while using the variable voltage (VV) levels with the optimal solution while setting a fixed voltage (FV) level, i.e., optimal at Case I with the optimals for all other cases.



(a) Auto-Industry (b) Networking (c) Office-Automation

Figure 4. Comparison of the Energy Consumption values using Our Proposed Unified Approach and the Sequential Approach



(a) Auto-Industry (b) Consumer

(c) Networking (d) Office-Automation

Figure 5. Variable Voltage Optimal vs. Heuristic Energy Consumption Comparison for E3S Benchmark Applications

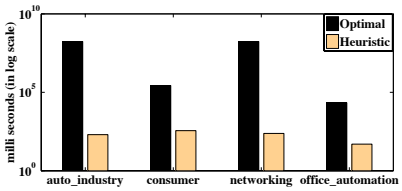
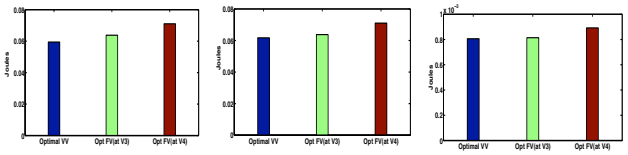


Figure 6. Exec. Time Comparison (log scale)

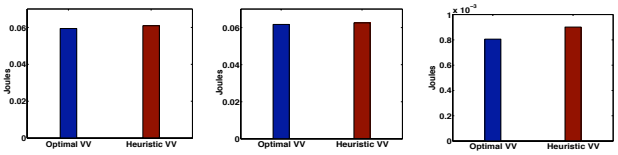


(a) MPEG4 (b) MWD (c) OPD

Figure 7. Optimal with variable voltage levels vs. Optimal with fixed voltage Energy Consumption Comparison for Real Applications

On average for these three applications, the variable voltage setup can save 11% energy consumption over the fixed voltage setup. Fig. 8 shows the quality of our proposed heuristic as compared with the optimal solution. i.e., the optimal and heuristic solution for Case I. In all the cases, the heuristic is able to find near-optimal solutions within seconds.

Thus, the experimental results support our claim of achieving near-optimal solutions from the heuristic. It also



(a) MPEG4 (b) MWD (c) OPD

Figure 8. Variable Voltage Optimal vs. Heuristic Energy Comparison for Real Applications

shows that the optimal solution with flexible voltage levels is more energy-efficient than the optimal at some fixed voltage level. Moreover, it shows that optimal solutions can be achieved using the unified approach as opposed to the sub-optimal solution obtained by the sequential approach.

7 Conclusion

In this paper, we have proposed a unified approach to solve the application mapping problem on a heterogeneous NoC platform for energy minimization. The voltage assignment problem is proven to be NP-hard. Our solution techniques are evaluated using benchmark suite E3S [3] and three real applications. Experimental results demonstrate effectiveness of our heuristic, superiority of the unified approach over sequential approach and advantage of operating the PEs at multiple voltage levels for energy minimization.

References

[1] W. Dally and B. Towles. Route Packets, Not Wires: On-Chip Interconnection Networks. In *Proc. Design Automation Conf.*, pages 684–689, Las Vegas, Nevada, USA, June 2001.

[2] G. De-Micheli and L. Benini. *Networks On Chips*. Morgan Kaufmann, 2006.

[3] R. Dick. Embedded System Synthesis Benchmarks Suite(E3S).

[4] M. A. A. Faruque, R. Krist, and J. Henkel. ADAM: Run-time Agent-based Distributed Application Mapping for on-chip Communication. In *Proceedings of DAC*, Anaheim, California, USA, June 8-13 2008.

[5] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.

[6] A. Hansson, K. Goossens, and A. Rădulescu. A Unified Approach to Mapping and Routing on Network-on-Chip for Both Best-Effort and Guaranteed Service Traffic. *Networks*, 2007, 2007.

[7] W. K. Mak and J. W. Chen. Voltage Island Generation under Performance Requirement for SoC Designs. In *ASPDAC*, pages 798–803, 2007.

[8] U. Y. Ogras, R. Marculescu, P. Choudhary, and D. Marculescu. Voltage-Frequency Island Partitioning for GALS-based Networks-on-Chip. In *Proc. Design Automation Conf.*, San Diego, California, USA, June 2007.

[9] K. Srinivasan, K. S. Chatha, and G. Konjevod. Linear-Programming-Based Techniques for Synthesis of Network-on-Chip Architectures. *IEEE Trans. on VLSI Systems*, 14(4):407–420, April 2006.