CLUSTERING-BASED SIMULTANEOUS TASK AND VOLTAGE SCHEDULING FOR NOC SYSTEMS

A Thesis

by

YU YANG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2011

Major Subject: Computer Engineering

CLUSTERING-BASED SIMULTANEOUS TASK AND VOLTAGE SCHEDULING FOR NOC SYSTEMS

A Thesis

by

YU YANG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Jiang Hu
Committee Members,	Paul V. Gratz
	Eun Jung Kim
Head of Department,	Costas N. Georghiades

May 2011

Major Subject: Computer Engineering

ABSTRACT

Clustering-Based Simultaneous Task and Voltage Scheduling for NoC Systems.

(May 2011)

Yu Yang, B.S., Zhejiang University;

M.S., Zhejiang University

Chair of Advisory Committee: Dr. Jiang Hu

Network-on-Chip (NoC) is emerging as a promising communication structure, which is scalable with respect to chip complexity. Meanwhile, latest chip designs are increasingly leveraging multiple voltage-frequency domains for energy-efficiency improvement. In this work, we propose a simultaneous task and voltage scheduling algorithm for energy minimization in NoC based designs. The energy-latency tradeoff is handled by Lagrangian relaxation. The core algorithm is a clustering based approach which not only assigns voltage levels and starting time to each task (or Processing Element) but also naturally finds voltage-frequency clusters. Compared to a recent previous work, which performs task scheduling and voltage assignment sequentially, our method leads to an average of 20% energy reduction.

To my parents

ACKNOWLEDGMENTS

This dissertation would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.

First and foremost, my utmost gratitude goes to my advisor, Dr. Jiang Hu. Dr. Hu has supervised, advised and guided me from the very early stage of this research, as well as gave me extraordinary experiences through out the work. I would like to thank other professors as well, who are always willing to discuss with me and give new ideas. Particular thanks to Dr. Gratz and Dr. Kim for their constructive comments on this thesis.

It is a pleasure to pay tribute to my colleagues. Thanks for all your valuable advice. I am particularly much indebted to Yifang Liu for his great help on this thesis.

Last but not the least, I am grateful for my family and friends. Thanks to my parents for their long lasting encouragement and support and thanks to my friends I have made all the way along.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

TABLE Page

LIST OF FIGURES

FIGURE Page

CHAPTER I

INTRODUCTION

A. Network-on-Chip

As the multiprocessors which can achieve a higher performance while exploiting of great cost-effective advantages compared with the traditional single processor emerges, the interconnection is a backbone which supports the above multiprocessor architecture. At first, bus or switch served this purpose and gained popularity especially for small scale multiprocessors. However, the trend towards many-core-based designs entails a large demand for on-chip global communications. Conventional bus structure will be very difficult to keep up with such demand. The new paradigm of Network-on-Chip (NoC) which brings network theories into communication, in contrast, is much more scalable with respect to the complexity and volume of communications, and is gaining substantial popularity. NoC is also amiable to design modularity, which is another means for handling design complexity. Figure 1 illustrates a tile-based multi-core system on a mesh-based NoC architecture. Each tile contains a processing element (PE) and a network router. The PE can be CPU, DSP module, video processor, or embedded memory block. The edges between two neighboring tiles indicate interconnection links. Instead of direct connection like in conventional bus, data are routed through the links and routers toward their destination PEs. More precisely, the PE will generate the data which is transmitted to local router seamlessly via Network Interface (NI). After the local router receives the data, it will determine the next neighbor router that will relay the data according to routing protocols. This process repeats iteratively until the data reaches the final destination. This mechanism utilizes generating, processing and relaying the data through the network infrastructure instead of global wires.

The journal model is *IEEE Transactions on Automatic Control.*

B. Voltage-Frequency Island for Energy Efficiency

In parallel to the communication issue, energy-efficiency is a more and more critical concern. In many modern chip designs, the power density is approaching the limit of chip cooling capacity and becomes the major limiting factor to the performance growth. Energy-efficiency implies that the energy is spent only when it is very necessary. This philosophy is embodied in the recent popular technology of voltage-frequency-island (VFI) which is employed at Intel SCC (Single-chip Cloud Computer) [1]. SCC integrates 48 cores and incorporates NoC as interconnection method. It allows software to dynamically adjust voltage and frequency to achieve low power consumption. In VFI-based designs, one or a set of circuit blocks may have its own voltage and frequency level, which is adjusted based on its performance requirement. For instance, the 3 different grey-scale levels in the tiles of Figure 1 represent different voltage-frequency levels. The energy-efficiency of such systems largely depends on how the voltage and frequency of each block (PE) are assigned. In addition, one must consider that a level shifter is needed when a signal is sent from a low-voltage island to a high-voltage island. The voltage-frequency assignment problem is shown to be NP-hard [2] and it can be integrated in many design stages.

1. VFI in Floorplanning

VFI method has been studied together with the floorplanning problem [3–5].

In [3], a dynamic programming algorithm for supply voltage assignment is employed for System On Chip (SOC). There are several candidate supply voltages for each core. Note that two cores may have different candidate supply voltages. Given a table of voltage-energy table, where each entry represents the energy consumptions for each core under its candidate voltages, the problem is how to choose m supply voltages so that the total energy consumption is minimized. The main idea of dynamic programming algorithm for this

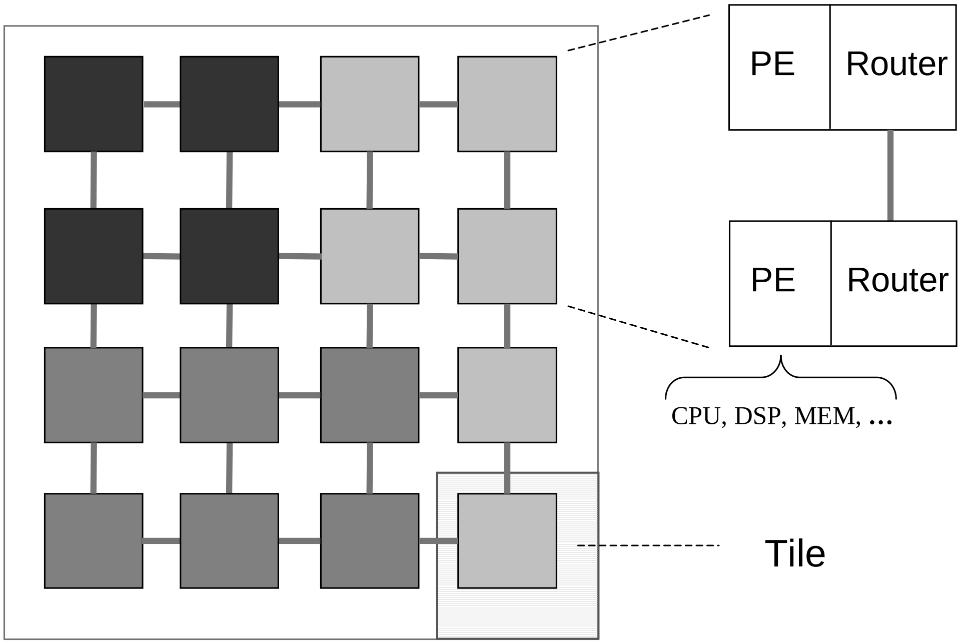

Fig. 1. Tile-based multi-core system on a mesh-based NoC architecture.

problem is as follows: given the optimum supply voltage assignment for i supply voltages, we can get optimum result for the $(i+1)$th supply voltages by comparing the total energy consumptions of replacing each of previous supply voltage with the $i+1$ one and selecting the one which results in minimum energy consumption.

A multiple supply voltage (MSV) problem for SOC has been studied in [4]. The problem that finding optimum supply voltage assignment with minimum energy consumption has been modeled as a convex cost dual network flow problem by transforming the constraints into the objective function as penalty functions and can be solved in polynomial time. This method can be easily integrated into the simulated annealing based floorplanning algorithm due to its fast running time. However, the overlook of communication energy consumption makes it unattractive for more accurate analysis.

In [5], a algorithm based on branch and bound is proposed to tackle MSV assignment problem. This algorithm utilizes the work in [4] to get a fast lower bound estimation by transforming the relaxed problem into dual network flow problem. By using this lower bound, many branches will be pruned out thus running time is reduced. This algorithm will guarantee to generate the optimal assignment result.

2. VFI in Post-Placement

Voltage island generation for post-placement is studied in [6]. Instead of generating voltage islands according to logic boundary, the authors propose a dynamic programming algorithm which exploits ”non-natural” (non-logical) boundary such that the the number of voltage islands is minimized under the maximum power budget. The motivation of this work is to avoid the large power delivery overhead caused by fragmented voltage islands.

An improved post-placement VFI generation algorithm is proposed in [7]. The authors eliminate the unnecessary requirement in [6] that each VFI must be a rectangle shape. The algorithm is based on the dynamic programming on the constructed tree which is derived by the placement region.

3. VFI in Network-on-Chip

VFI can be easily integrated with NoC-based systems. Recently, a number of works address the voltage-frequency assignment problem for NoC. In [8], the voltage-frequency assignment and partitioning were performed after the tasks were bound to PEs and mapped to tiles. The whole procedure can be divided into two separate parts: first, it starts with a VFI partition where each PE belongs to a different voltage. Second, based on the result of first stage, an iterative merge process continues. Each time, two different VFIs which will result in the maximum energy reduction if merged, will be merged. This merge process will continue until only one single island remains. The best voltage assignment in all these merge process will be chosen as the output of this algorithm.

In [12], an enumeration-based method was proposed for voltage-frequency assignment. By using this method, the VFI overhead such as mixed clock FIFOs and voltage level converters is reduced by 82%, as well as over 9% energy consumption reduction compared with previous work. Both [8] and [12] assumed that task scheduling has been finished and delay budget has been allocated to each individual task. Therefore, they did not consider task precedence constraints which state that certain tasks must be finished before another task is started. In practice, the precedence constraints often arise from data inter-dependencies among the tasks.

4. Other Techniques for Energy-Efficiency

Besides the multiple supply voltage design, the energy efficiency can also be achieved by communication and task scheduling. In [9], an energy-aware scheduling (EAS) algorithm is proposed under real time constraints which schedules communications and tasks into heterogeneous NoC architecture which consists of different PEs such as DSP or Power PC. Each task can have different energy consumptions and execution times when mapped to different PEs due to heterogeneity. The scheduling problem is to determine the target PE, the time slot when the task can be executed and when all the communication transaction can occur such that the total tasks' energy consumption combined with communications' is minimized under the real time constraints.

In [10], the authors propose an energy-aware mapping for tile-based NoC architecture under performance constraint. Given an Application Characterization Graph (APCG), where each vertex represents one select IP/core, each direct arc represents the communication, and an Architecture Characterization Graph (ARCG), where each vertex represents one tile in the architecture, and each directed arc represents a routing path. The problem is how to find the mapping function which map every vertex from the APCG to one and only one vertex in the ARCG, such that the communication energy is minimized while meeting the bandwidth constraints of all links. A branch and bound algorithm is presented to solve the above problem. Effective upper bound cost and low bound cost methods are used to trim branches of the search tree more quickly. Some other speed-up techniques are also employed like ordering IP according to their communication demand, exploiting symmetry property.

A communication latency aware low power NoC synthesis algorithm is presented at [11]. The NoC topology is a directed graph where each node represents a tile, each edge represents a point-to-point interconnection between two adjacent tiles. The implementation of a NoC topology is a mapping from each edge to a particular wire style and a mapping from each edge to the amount of wiring resources assigned to that edge where different edge has different power consumption, delay and area. The total communication power is minimized while the total total size and total delay are subjected to a constraint. Latency constraints and power minimization objectives are modeled as multi-commodity flow (MCF) problem in a unified manner. A polynomial time approximation schemes (PTAS) is proposed to obtain $(1+\varepsilon)$ optimal solutions in polynomial time, where ε is an input accuracy threshold.

C. Motivation and Contribution

In this work, we propose a clustering-based simultaneous task and voltage scheduling algorithm for post-mapping energy minimization in NoC designs. In the core, it presents a new clustering algorithm guided by Lagrangian relaxation. In fact, clustering technique has been employed in various electronic design automation problems. For example, a novel clustering based technique, which utilizes equi-slack gate clusters to minimize the leakage power of circuits in nanometer technology, was presented in [13]. The equi-slack gate cluster based technique achieves much better results in terms of runtime and leakage power reduction than most of the existing leakage power minimization methods. Another work [14] uses feature extraction based clustering algorithm to enhance the yield by assigning the same body bias to the gates with similar features.

While the simultaneous approach allows more flexibility for energy reduction, it poses significant difficulty to solving problem due to the task precedence constraints in DAGs. We handle this difficulty by proposing a new clustering algorithm and combining it with Lagrangian relaxation. One of this work's main contributions is an innovative way to reformulate the original scheduling problem. From a new perspective, we transform the complex simultaneous task and voltage scheduling problem into a clearly defined clustering problem. This transform is two-fold. First, Lagrangian relaxation integrates the original objective and constraints into one cost function, which is a linear combination of energy consumption and deadline constraint violations. Then, this cost function is mapped to a summation of distances between tasks and voltages in our task clustering space. By doing this, the difficult simultaneous scheduling problem is transformed to a clustering problem. The clustering problem is solved by a customization of the classical k-means algorithm. It makes use of domain-specific analysis to define the centers of the clusters, the distance metric in the clustering space, and the cluster agglomeration procedure.

In the experiment, we compare our method with a sequential approach of task scheduling followed by voltage assignment [12]. The results show that our method can achieve 20% energy reduction on average under the same task deadline constraints.

CHAPTER II

CLUSTERING-BASED SIMULTANEOUS TASK AND VOLTAGE SCHEDULING FOR NOC SYSTEMS

A. Preliminary

An application consists of a set of tasks and data inter-dependencies among them. It can be described by Communication Task Graph (CTG) $G = (P, E)$ which is usually a Directed Acyclic Graph (DAG). Each node $p_i \in P$ represents a task. If a task p_i is assigned with supply voltage v, it has an execution time $d_{ex}^{p_i}(v)$ and corresponding energy consumption $E_{p_i}(v)$. A directed edge (p_i, p_j) implies a precedence constraint between p_i and p_j. This constraint is usually caused by the communication from p_i to p_j with data of volume $\phi(p_i, p_j)$. Hence, edge (p_i, p_j) requires that task p_j cannot start until p_i is finished and data of $\phi(p_i, p_j)$ has been transferred from p_i to p_j. A task without any incoming edges is called *source task* and a task without any outcoming edges is called *sink task*. Let S be the set of all sink tasks in the CTG. For any task p_i in S, there is a deadline D_{p_i} associated with it. For example, Figure 2 shows a CTG with 5 tasks, $p_0, \cdots, p_4$. The communication from p_0 to p_1 has data volume of 3000. The deadline for sink p_4 is 10.

The execution time of a task p_i is estimated by the product of clock period and the total number of active cycles, i.e.

$$d_{ex}^{p_i}(v) = R_{p_i} \times \pi(v) \tag{2.1}$$

where R_{p_i} is the total number of active cycles, $\pi(v)$ is the clock period for a supply voltage v. According to [8], $\pi(v)$ can be calculated as follows:

$$\pi(v) = \frac{K_i v}{(v - v_t)^{\alpha}} \tag{2.2}$$

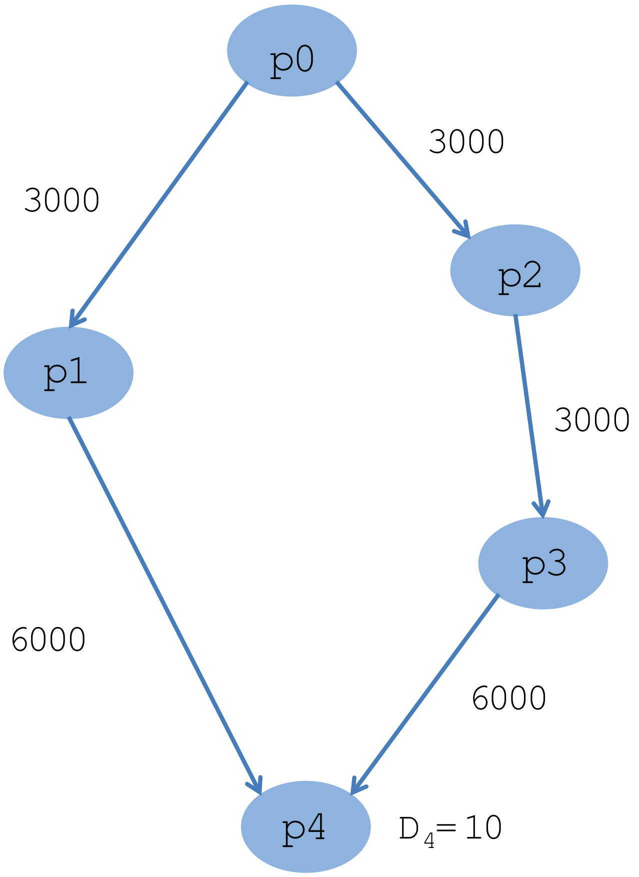

Fig. 2. A communication task graph.

where α is a technology parameter, K_i is a design specific constant [15], and v_t is the threshold voltage.

A task energy consumption which includes the dynamic and static ones is also related to the supply voltage. By using the above notations, the sum of dynamic and static energy consumption associated with each task is defined as follows:

$$E_{p_i}(v) = R_{p_i} C_i v^2 + Q_{p_i} K_i v e^{-\frac{v_t}{S_t}} \tag{2.3}$$

where R_{p_i} and Q_{p_i} are the total number of active and idle cycles for task p_i respectively, C_i is the total switched capacitance per cycle, K_i is a design parameter and S_t is a technology parameter [16].

We assume tile based mesh NoC architecture is used here as depicted in Figure 1. Each tile contains a processing element as well as a router. We denote the set of tiles as $\mathcal{T} = \{T_1, T_2, ..., T_N\}$. Routing in NoC has been shown to have significant impact on energy consumption [17]. In this work, we use a commonly adopted routing algorithm [8], which is similar to wormhole flow control and XY routing algorithm in computer networking. Asynchronous communication across different voltage islands is obtained by mixed-clock/ mixed-voltage FIFOs [18].

In NoC we assume that the tasks have been allocated to each tile. We use function $M : P \to T$ to represent the mapping function which assigns each task to a specific tile in NoC. For example, $M(p_i)$ denotes the tile which task p_i is mapped to. Under the above assumptions, we define Execution Task Graph (ETG) $G' = (P, E')$ which can be derived from CTG by the following procedures: for any pair of tasks p_i and p_j in the same tile, suppose p_i executes earlier than p_j, then if $(p_i, p_j) \in E$ in CTG, $\phi(p_i, p_j) = 0$; otherwise, we add an edge (p_i, p_j) in ETG and let $\phi(p_i, p_j) = 0$. In other words, if two tasks are assigned to the same tile, a precedent constraint is imposed since the PE in this tile can only process one task at a time; if the two tasks have communication requirement, this requirement is eliminated as communication in the same PE can be ignored since the volume data is stored in local memory and can be retrieved in negligible time.

Using the above notation, the communication energy consumption for any edge $(p_k, p_l) \in E'$ is defined as follows:

$$E(p_k, p_l) = \sum_{i \in Q} \phi(p_k, p_l) E_{bit} \frac{v_i^2}{v_{DD}^2} \tag{2.4}$$

where E_{bit} is a bit energy metric [8], which is the total energy consumed when one bit of data is transferred through the link, buffer and switch fabric. Also assume the bit energy metric is measured under v_{DD}. Q means the set of tiles on the path from tile $M(p_i)$ to tile $M(p_j)$ since each link and router belong to a tile in NoC, v_i is the supply voltage for tile i.

Similar to [8], the communication latency for any edge $(p_i, p_j) \in E'$ is represented as follows:

$$t_{co}(p_i, p_j) = \begin{cases} 0, & \text{if } M(p_i) = M(p_j) \\ \sum\limits_{i \in Q} \frac{\mu_s}{f_i} + t_{fifo} \lceil \frac{\phi(p_i, p_j)}{W} \rceil, & \text{otherwise} \end{cases} \tag{2.5}$$

where W is the channel width, f_i is the operation frequency of tile i, μ_s is the number of cycles it takes to traverse a single router and outgoing link, t_{fifo} is the latency of the FIFO buffers. Since we use wormhole flow control mechanism, the first term and the second term of the above equation correspond to the latency of header flits traversing path Q and latency of serialization for the remaining flits, respectively.

The deadline constraint means that for each source-sink path in the ETG, the sum of total execution time and communication delay should not exceed the deadline of the sink at the path end. Let $t_{st}^{p_i}$ and be the starting time, $\forall p_i \in P$. Denote by $I(p_i)$ the set of immediate upstream tasks of task p_i, i.e. $I(p_i) = \{p_j | (p_j, p_i) \in E'\}$. Then the following condition must be satisfied for each task: task cannot start until all of its parents and the corresponding communication transactions have finished, i.e.

$$t_{st}^{p_i} = \begin{cases} 0, & \text{if } I(p_i) = \emptyset \\ \max\limits_{p_j \in I(p_i)} (t_{st}^{p_j} + d_{ex}^{p_j}(v_{p_j}) + t_{co}(p_j, p_i)), & \text{otherwise} \end{cases} \tag{2.6}$$

By Equ. (2.6), we can calculate all tasks' start time in topological order. If for each task in S, its start time plus the execution time is no greater than its corresponding deadline, i.e., $t_{st}^{p_i} + d_{ex}^{p_i}(v_{p_i}) \leq D_{p_i}, \forall p_i \in S$, then the deadline constraint is satisfied.

B. Problem Formulation

The simultaneous task and voltage scheduling ($\mathcal{STVS}$) problem is stated as follows.

Given a NoC architecture with each task has been allocated to a tile, an ETG $G' =$

(P, E') derived from a CTG $G = (P, E)$, tasks' mapping function M as well as a set of supply voltage options V, assign each tile $i \in T$ a voltage v_i and each task p_i a start time $t_{st}^{p_i}$, such that the total application energy consumption is minimized subject to the path deadline constraints, i.e.

$$\begin{aligned}
&\text{Min:} E_{APP} = \sum_{\forall p_i \in P} E_{p_i}(v_{M(p_i)}) + \sum_{\forall (p_i, p_j) \in E'} \phi(p_i, p_j) E(p_i, p_j) \\
&\text{s.t. } t_{st}^{p_i} + d_{ex}^{p_i}(v_{M(p_i)}) + t_{co}(p_i, p_j) \leq t_{st}^{p_j}, \forall (p_i, p_j) \in E' \\
&\quad t_{st}^{p_i} + d_{ex}^{p_i}(v_{M(p_i)}) \leq D_{p_i}, \forall p_i \in S \\
&\quad v_i \in V, \forall i \in T
\end{aligned} \tag{2.7}$$

where S is the set of sink tasks and $v_{M(p_i)}$ is the supply voltage for tile where task p_i locates.

C. Motivation and Main Ideas

Usually, one wants to assign a tile with a voltage level the same as its adjacent tiles if they have similar performance requirement. If they use the same supply voltage, the interface overhead, such as level shifters and FIFOs, can be reduced or avoided. In other words, tiles with similar performance requirement are preferred to be grouped together and assigned to the same supply voltage level. This observation is the main motivation for us to schedule the tasks and voltages by clustering considering performance specifications.

A classic approach for clustering is the k-means algorithm [22]. It starts with k randomly generated clusters. Then, it iteratively assigns every element to the cluster whose center is the nearest to the element according to certain distance metric. The center of a cluster is the geometric mean location of all elements in the cluster in certain coordinate system. After every element is assigned to a cluster, the centers of all clusters are updated.

This iteration repeats with assigning elements to the clusters resulted from the previous iteration, followed by clusters' update according to the new elements' assignment. The iteration continues till certain convergence criterion is met. Figure 3 shows an example of one iteration in k-means clustering. In Figure 3(a), the elements in black squares are assigned to the clusters closest to them. The line in the middle separates the two clusters. In Figure 3(b), the centers of the two clusters are moved to the mean point of the elements in them, respectively. After the adjustment of the cluster centers, the next iteration begins with the calculation of distances between the elements and the cluster centers.

Despite their similarity, there is a gap between classical clustering and the voltage assignment in our case. This gap manifests on two related aspects. On one hand, clustering requires a well-defined distance/coordinate metric which is not obviously available in the voltage assignment problem. On the other hand, two tiles are assigned to the same voltage only if they have similar performance requirements (otherwise, one tile may be unnecessarily assigned with a high voltage and energy waste is induced). However, the performance requirement for each tile is not clear since the task scheduling has not been done yet.

We propose to bridge this gap by using Lagrangian relaxation. Lagrangian relaxation converts the complex constraints of an optimization problem into a part of the objective function, which is a set of penalty terms to any violations to the constraints. The converted problem, called Lagrangian subproblem, attempts to minimize a linear combination of the original objective function and the constraint violations. In our case, the objective (cost) function of the Lagrangian subproblem is a linear combination of energy and deadline constraint violations. Then, we define the distance metric based on this cost function. By doing so, both energy cost and performance requirement are handled in a unified manner. Lagrangian relaxation comes with a dual problem which finds the appropriate values for the penalty coefficients (Lagrangian multipliers) for the Lagrangian subproblem. In Section VII, a subgradient approach for solving the dual problem will be introduced.

D. Lagrangian Relaxation

The $\mathcal{STVS}$ problem formulated in Section III is solved under the Lagrangian relaxation framework, which is also adopted by other complicate multi-constrained optimization in electrical design automation area [23]. For each constraint in $\mathcal{STVS}$, we specify a non-negative Lagrangian multiplier λ and obtain the Lagrangian function:

$$\begin{aligned} & L_\lambda(v, t_{st}) = \\ & \quad \sum_{\forall p_i \in P} E_{p_i}(v_{M(p_i)}) + \sum_{\forall (p_i,p_j) \in E'} \phi(p_i,p_j) E(p_i,p_j) + \\ & \quad \sum_{\forall (p_i,p_j) \in E'} \lambda_{ij}(t_{st}^{p_i} + d_{ex}^{p_i}(v_{M(p_i)}) + t_{co}(p_i,p_j) - t_{st}^{p_j}) + \\ & \quad \sum_{\forall p_i \in S} \lambda_i (t_{st}^{p_i} + d_{ex}^{p_i}(v_{M(p_i)}) - D_{p_i}) \end{aligned} \tag{2.8}$$

The Lagrangian subproblem is to minimize the Lagrangian function for a specific set of Lagrangian multipliers and is formulated as:

$$\begin{aligned} \text{Min:} \quad & L_\lambda(v, t_{st}) \\ \text{s.t.} \quad & v_i \in V, \forall i \in T \end{aligned} \tag{2.9}$$

According to KKT conditions [21], the Lagrangian subproblem can be reduced as in [19]. After the reduction, variables t_{st} is eliminated and the subproblem becomes:

$$\begin{aligned} L_\lambda(v) = & \sum_{\forall p_i \in P} E_{p_i}(v_{M(p_i)}) + \sum_{\forall (p_i,p_j) \in E'} \phi(p_i,p_j) E(p_i,p_j) + \\ & \sum_{\forall (p_i,p_j) \in E'} \lambda_{ij}(d_{ex}^{p_i}(v_{M(p_i)}) + t_{co}(p_i,p_j)) + \\ & \sum_{\forall p_i \in S} \lambda_i (d_{ex}^{p_i}(v_{M(p_i)}) - D_{p_i}) \end{aligned} \tag{2.10}$$

In the simplified Lagrangian subproblem, the execution times and communication de-

Algorithm 1 $LR\_clustering\_framework$

```
1: initialize $(v_i, \lambda)$;
2: for all $k \in \{0, 1, 2, 3, ...\}$ do
3:     Perform TILE_clustering (assigning $v_i$'s) on all tiles based on TILE-voltage distance,
       while the center of each cluster is a voltage option in the candidate voltage set;
4:     Update Lagrangian multipliers $\lambda$ with our sub-gradient calculation technique;
5:     If no improvement, stop with the best clustering solution satisfying the timing constraint
       till $k$th iteration;
6: end for
```

lays are independent of each other. Therefore, the subproblem becomes easier to solve. This subproblem will be tackled by a clustering algorithm described in Section VI.

Besides the subproblem, one needs to find proper values for the Lagrangian multipliers such that the original $\mathcal{STVS}$ problem is solved. This is the so-called Lagrangian dual problem and will be discussed in Section VII.

Although variables t_{st} are eliminated in Equ. (2.10), a legitimate task scheduling solution is still largely specified by the results from our method. Since our method tries hard to trade the delay slack for energy reduction, the slack for each task, which is defined by the difference between ALAP and ASAP schedules [20], is minimized. In other words, the starting time for each task in close to fully specified.

The overall framework of the clustering method guided by Lagrangian relaxation is outlined in Algorithm 1. Line 2 and 6 indicate the Lagrangian iteration loop. Line 3 solves the Lagrangian sub-problem by performing tile clustering based on tile-voltage distance measurement. Line 4 in every Lagrangian iteration updates multipliers to solve Lagrangian dual problem.

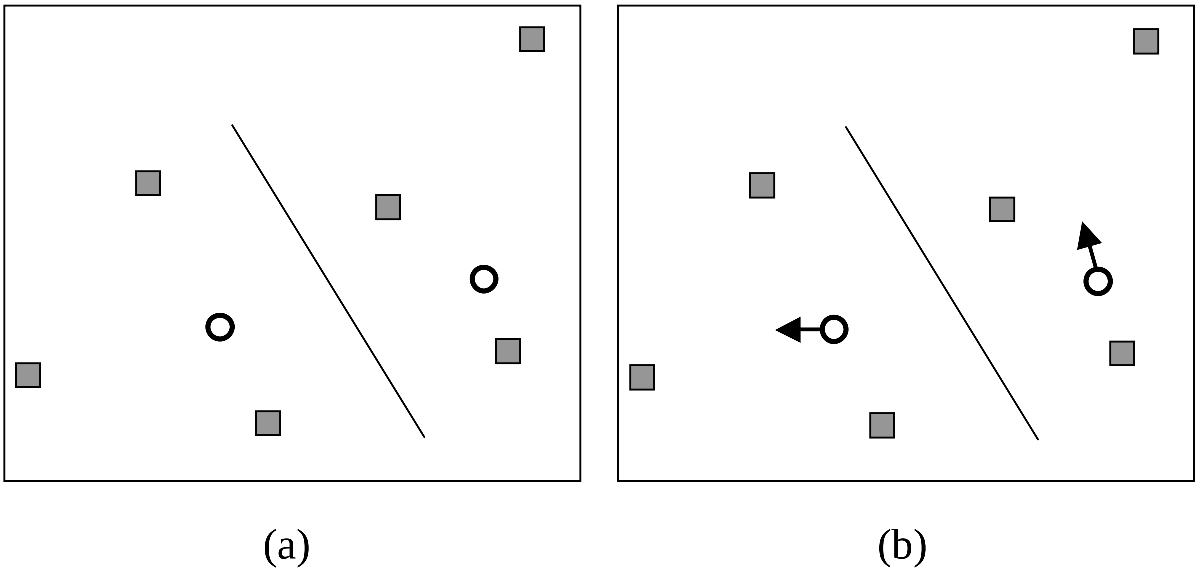

Fig. 3. An iteration in k-means clustering. (a) cluster assignment. (b) center point moves to the mean of all elements in the cluster.

E. Tile Clustering for Voltage Assignment

We define TILE-voltage distance $l(i, v_i)$ for any tile i when it is assigned with voltage v_i, i.e.,

$$\begin{aligned} l(i, v_i) = & \sum_{\forall p_j : M(p_j)=i} E_{p_j}(v_i) \\ & + \sum_{\forall Q(p_k,p_j) \text{pass tile } i} \phi(p_k, p_j) \frac{v_i^2}{v_{DD}^2} \\ & + \sum_{\forall p_j : M(p_j)=i} (\sum_{\forall p_k : (p_j, p_k) \in E'} \lambda_{jk}) d_{ex}^{p_j}(v_i) \\ & + \sum_{\forall Q(p_i,p_j) \text{pass tile } i} \lambda_{ij} \frac{\mu_s}{f_i}. \end{aligned} \tag{2.11}$$

where $Q(p_k, p_j)$ is the set of tiles on the path from tile $M(p_k)$ to tile $M(p_j)$, f_i is the operation frequency of tile i.

Based on the distance measurement given in Equ. (2.11), a clustering procedure carried out on all tiles actually performs the optimization to minimize the Lagrangian function

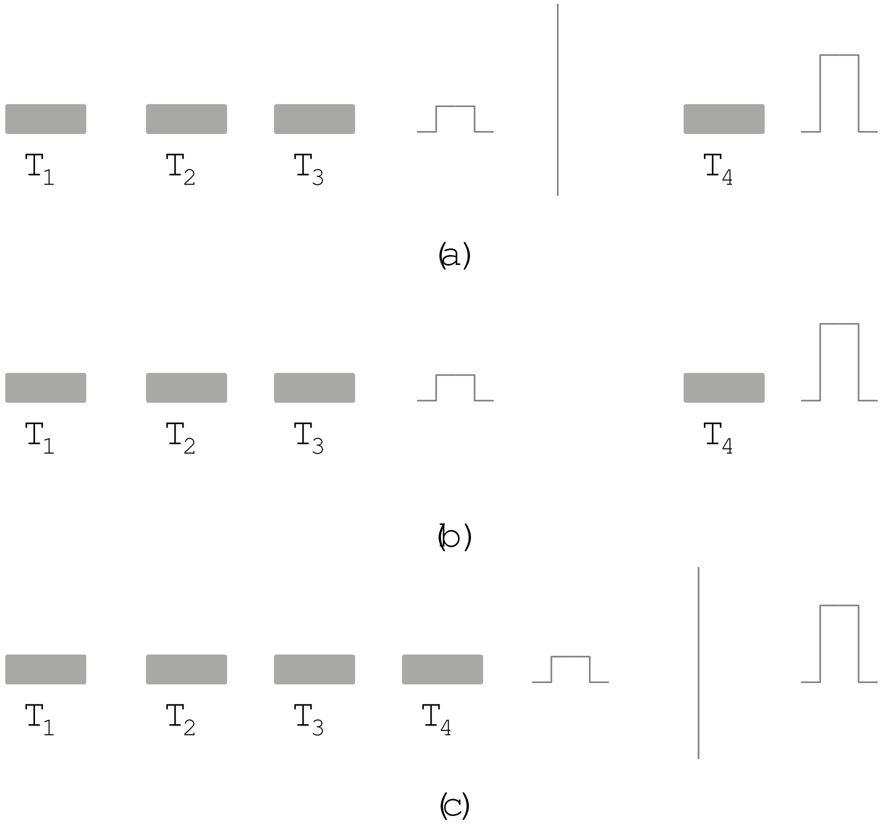

Fig. 4. One iteration in TILE clustering. There are four tiles T_1,T_2,T_3 and T_4. Two supply voltages are available. (a) TILE voltage assignment at the beginning of iteration. (b) T_4's distances to the two voltages are re-evaluated. (c) T_4 is assigned to a new voltage according to the re-evaluated distances.

in Equ. (2.10). In another word, clustering of tiles, using distance measurement in Equ. (2.11), solves the Lagrangian sub-problem in Equ. (2.9).

Our TILE clustering method is performed in a similar way to classic clustering methods, except that in our case the center of each cluster is always one of the voltage options given by the problem. That is, every voltage option corresponds to a cluster and is always the center of the cluster. Therefore, the number of clusters equals the number of voltage options. Also, the distance from each tile to a voltage option depends on the supply voltages of related tiles. When the clustering that results in the minimum overall distance is reached, the iteration of clustering method terminates. When the clustering procedure finishes, there may be a few empty clusters, where there is no tile assigned to them. The number of non-empty clusters is the best number of voltages for the problem, and the TILE assignment of the non-empty clusters is the best voltage assignment solution.

Given a set of multipliers, all clustering iteration works like an iteration in k-means clustering. First, the distances between every tile and all the voltage options are calculated. Then, based on these distances, each tile is assigned to the cluster whose center - a voltage option - is closest to the tile. An iteration is completed without re-evaluation of the center of each cluster, because the center of each cluster is fixed to a voltage option in our case.

In refinement iterations, the tile-voltage distance uses the formula in Equ. (2.11), accommodating several factors: tasks' energy consumption, tasks' communication energy, tasks' execution time, and communication delay. The example in Figure 4 illustrates the change of a TILE's voltage assignment in one iteration. Figure 4(a) shows the clustering at the beginning of the iteration, where T_4 is in the high voltage cluster. Figure 4(b) re-evaluate the distances of T_4 to all voltages, and T_4 gets closer to the lower voltage. Figure 4(c) assigns T_4 to lower voltage cluster.

The clustering algorithm is outlined in Algorithm 2. Line 3 performs distance evaluation between tiles and voltage options; Line 4 assigns tiles to specific clusters according to the updated TILE-voltage distances. Algorithm 2 implements the clustering step in a Lagrangian relaxation iteration in Algorithm 1.

F. Solving Lagrangian Dual Problem

The goal of the outer loop of the Lagrangian relaxation framework in Algorithm 1 is to solve Lagrangian dual problem, which basically tunes the multipliers λ to maximize the minimal value (optimized by adjusting v_i and t_{st} in the sub-problem) of the Lagrangian

Algorithm 2 $TILE\_clustering$

```
initialize $v_i$ to minimize $E_i(v_i) + \lambda_i d_{exec}(v_i), \forall p_i \in P$;
for all $k \in \{0, 1, 2, 3, ...\}$ do
    evaluate all TILE-voltage distances, i.e.,
        $l(i, v_i), \forall i \in T, \forall v_i \in V$;
    make voltage assignment, i.e.,
        $v_i \leftarrow \text{argmin}_{v_i \in V} l(i, v_i)$;
    If no change of assignment made, stop with the current cluster assignment in the $k$th iteration;
end for
```

function, $\min\limits_{v, t_{st}} L_\lambda(v, t_{st})$. In a formal formulation, the dual problem is expressed as:

$$\begin{aligned} \text{Max:} \quad & \min_{v, t_{st}} L_\lambda(v, t_{st}), \\ \text{s.t.} \quad & \lambda \geq 0. \end{aligned} \tag{2.12}$$

The function $L_\lambda(v, t_{st})$ is a concave function of $\lambda \geq 0$. However, it is non-differentiable. Therefore, the subgradient method is employed to solve the dual problem instead [24]. The method works as follows. First, initial λ values are given. Then, every λ for a constraint is updated to a new value in the subgradient direction. In our case, in iteration k, we first solve the Lagrangian subproblem by using the cluster based method; then, we define the subgradient direction to be the left hand side minus the right hand side of the constraints in Equ.(2.7). The values of $t_{st}^{p_i}, d_{ex}^{p_i}, \forall p_i \in P$ and $t_{co}(p_i, p_j), \forall (p_i, p_j) \in E'$ needed in this computation are calculated by a topological traversal of the ETG and ASAP (As Soon As possible) scheduling method, after we get the supply voltage for each tile in the current iteration. We use a step size ρ_k for current iteration k, multiply it with the subgradient

Table I. Energy consumption minimization under task deadline constraint

Benchmark	Previous work [12]		Our method	
	energy	normalized energy	energy	normalized energy
office automation	541	1	423	0.78
telecommunication	232	1	157	0.67
auto-industry	100	1	84	0.84
consumer	646	1	538	0.83
networking	379	1	329	0.86
average	379.6	1	306.2	0.80

direction, and add it to the current λ value, that is:

$$\begin{aligned}
&\lambda_{ij} = \lambda_{ij} + \rho_k(t_{st}^{p_i} + d_{ex}^{p_i}(v_{M(p_i)}) + t_{co}(p_i, p_j) - t_{st}^{p_i}),\\
&\forall(p_i, p_j) \in E';\\
&\lambda_i = \lambda_i + \rho_k(t_{st}^{p_i} + d_{ex}^{p_i}(v_{M(p_i)}) - D_{p_i}), \forall p_i \in S; \qquad (2.13)
\end{aligned}$$

This whole process continues until it converges, which means:

$$\begin{aligned}
&\sum_{p_i \in P} E_{p_i}(v_{M(p_i)}) + \sum_{\forall(p_i,p_j)\in E'} \phi(p_i, p_j)E(p_i, p_j) - L_\lambda(v, t_{st})\\
&\leq \text{error bound} \qquad (2.14)
\end{aligned}$$

It is also known that if the step size ρ_k satisfies when $k \rightarrow \infty, \rho_k \rightarrow 0$, and $\sum_{i=1}^{k} \rho_i \rightarrow \infty$, then the subgradient method will converge to its optimal value.

G. Experiments

In our experiment, the test cases are from Embedded System Synthesis Benchmark Suite (E3S) [25]. E3S contains some example applications from various areas, such as office automation, networking, auto industry, and telecommunication. The number of tasks in the benchmark applications ranges from 5 to 30. These applications are scheduled to 3×3 mesh networks respectively. The supply voltage candidates are 0.8v,1.0v,1.2v,1.4v and 1.6v.

Our method is compared with the voltage assignment method of [12]. In [12], the delay deadline for each individual task is first obtained according to the energy aware scheduling [9]. Then, the voltage assignment for each tile is found by enumeration. Since we only compare the voltage assignment method, we assume that the mapping results are the same for both cases. We implemented both the method of [12] and our method in C++. The experiment was performed on a Windows-based desktop machine with 2.0 GHz Intel core 2 duo CPU and 2 GB memory.

Figure 5 provides details on the supply voltage assignment results for each tile for consumer benchmark from both our method and [12]. The voltage distributions by the two methods differ a lot from each other. This is because our method can handle the energy-performance tradeoff from a global point of view, while the optimization of [12] tends to be restricted to local tradeoff.

Our algorithm's runtime ranges from 0.58s to 2.11s. Figure 6 shows the runtime results for all benchmarks.

The final supply voltage for each tile in five benchmarks is shown in Fig.7,8, 9,10 and Fig. 11 respectively. In these figures, the bottom squares correspond to all tiles and the z values of these tiles represent the assigned supply voltages. From the figures, we can see different tiles are assigned to different supply voltages. Table 1 lists the energy consumption for all benchmarks both for our method and method of [12]. On average

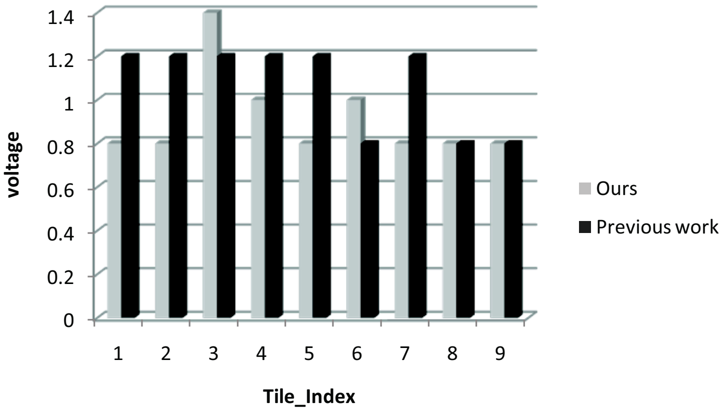

Fig. 5. Comparison on voltage assignment from [12] and our method over 9 tiles on an E3S benchmark - consumer.

over the five benchmarks in E3S, our method achieves 20% energy reduction compared to [12]. The largest energy reduction is 33%. The main reason for this difference is that [12] separates the voltage assignment from task scheduling. Without the information on voltage assignment, the task scheduling may make wrong decisions and incur inappropriate deadline constraints to the subsequent voltage assignment problem.

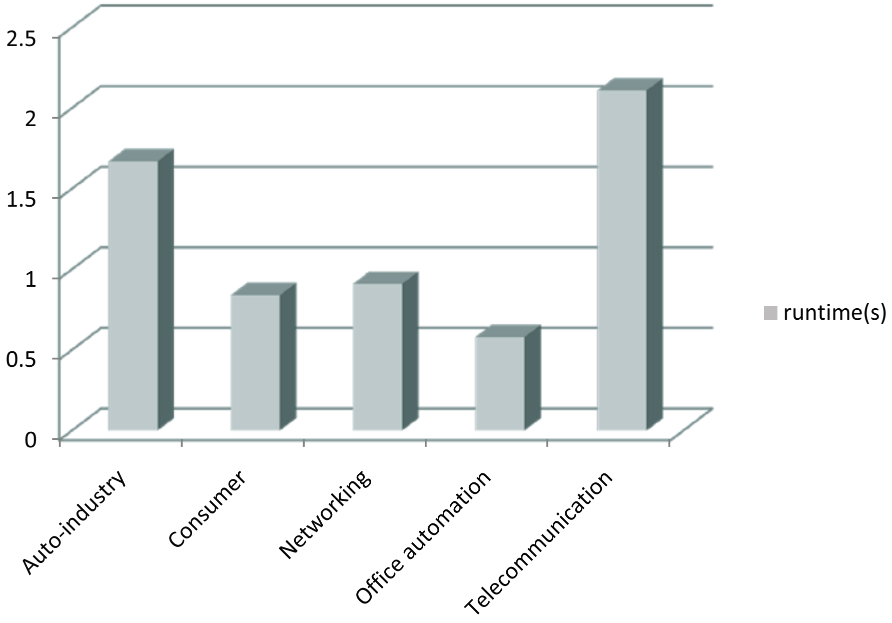

Fig. 6. Runtime for each benchmark.

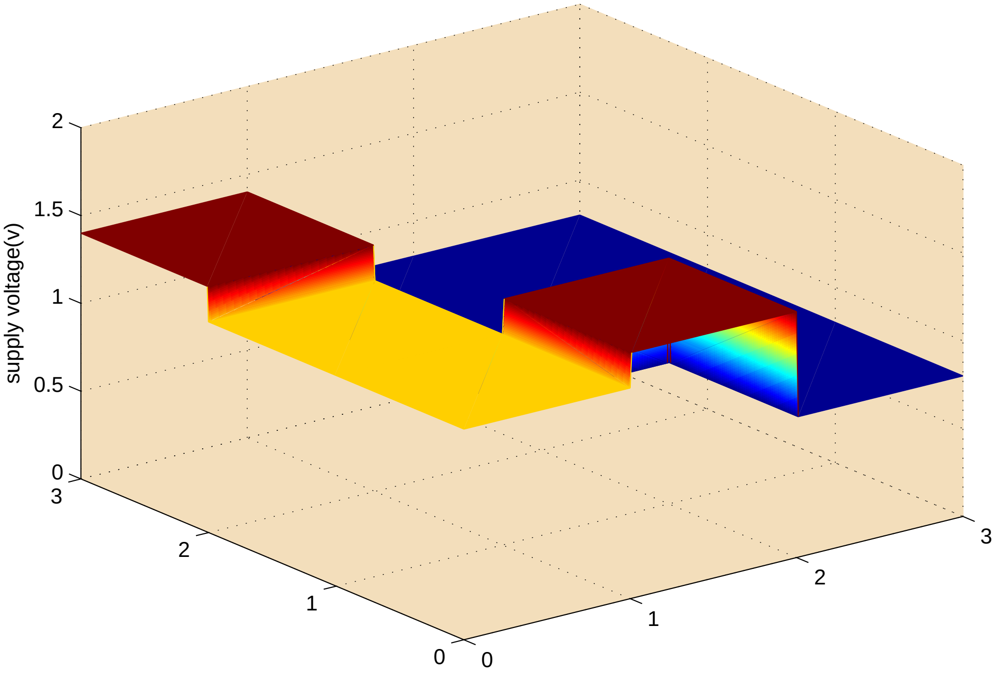

Fig. 7. Supply voltage for each tile in auto-industry.

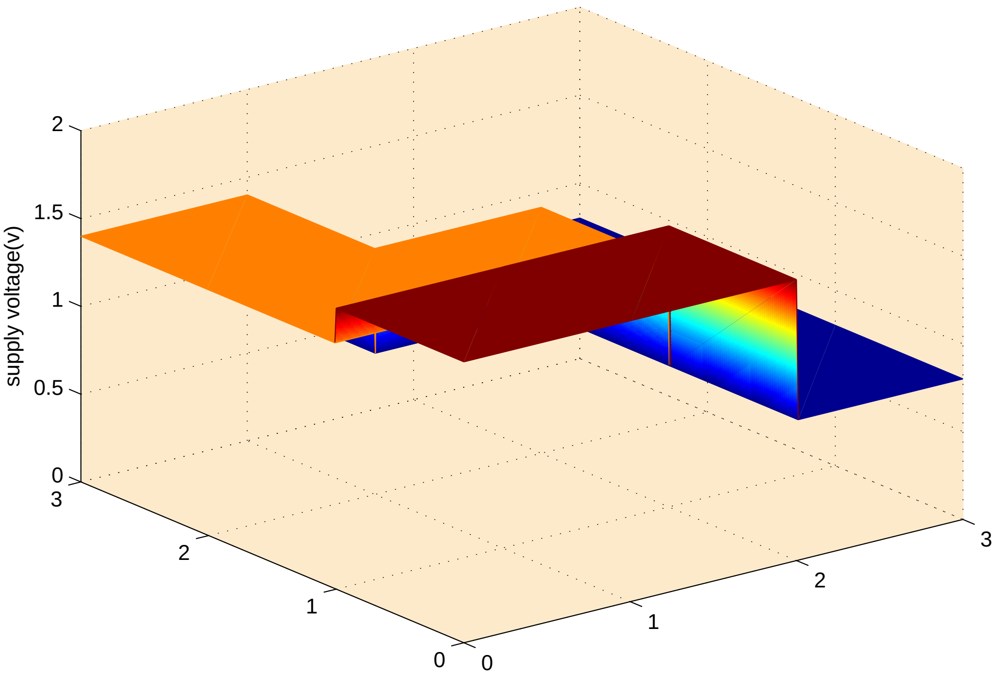

Fig. 8. Supply voltage for each tile in networking.

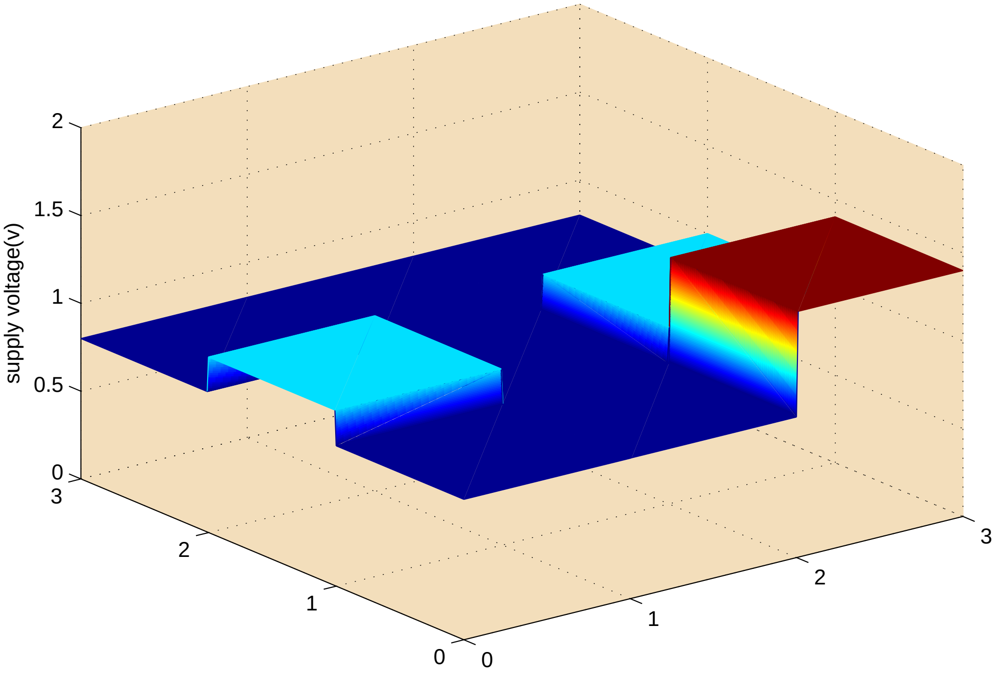

Fig. 9. Supply voltage for each tile in consumer.

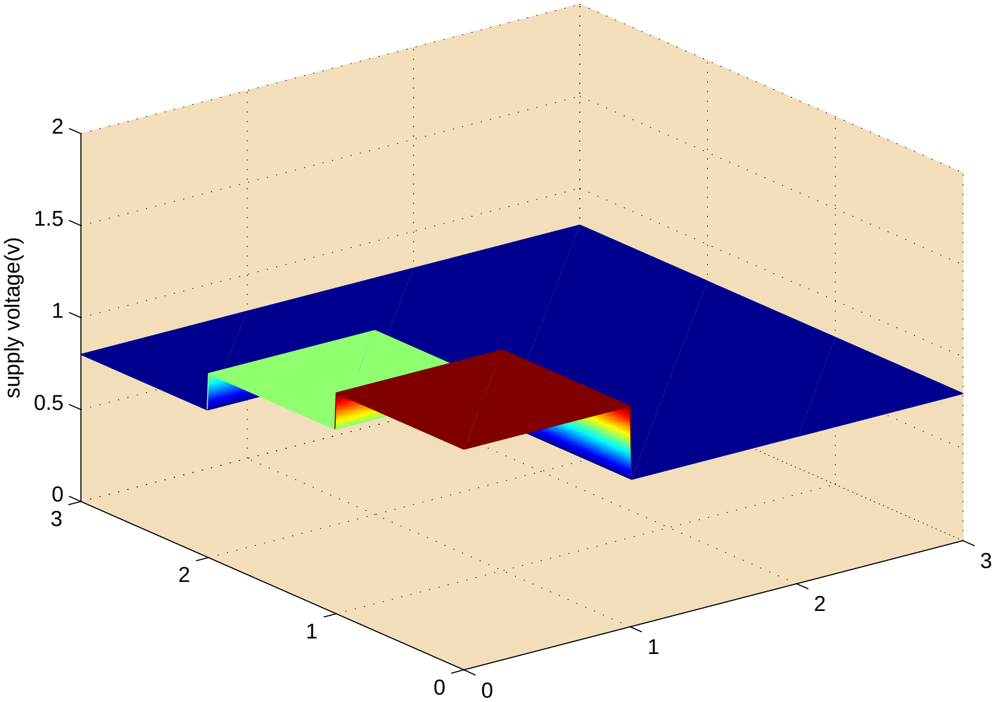

Fig. 10. Supply voltage for each tile in office automation.

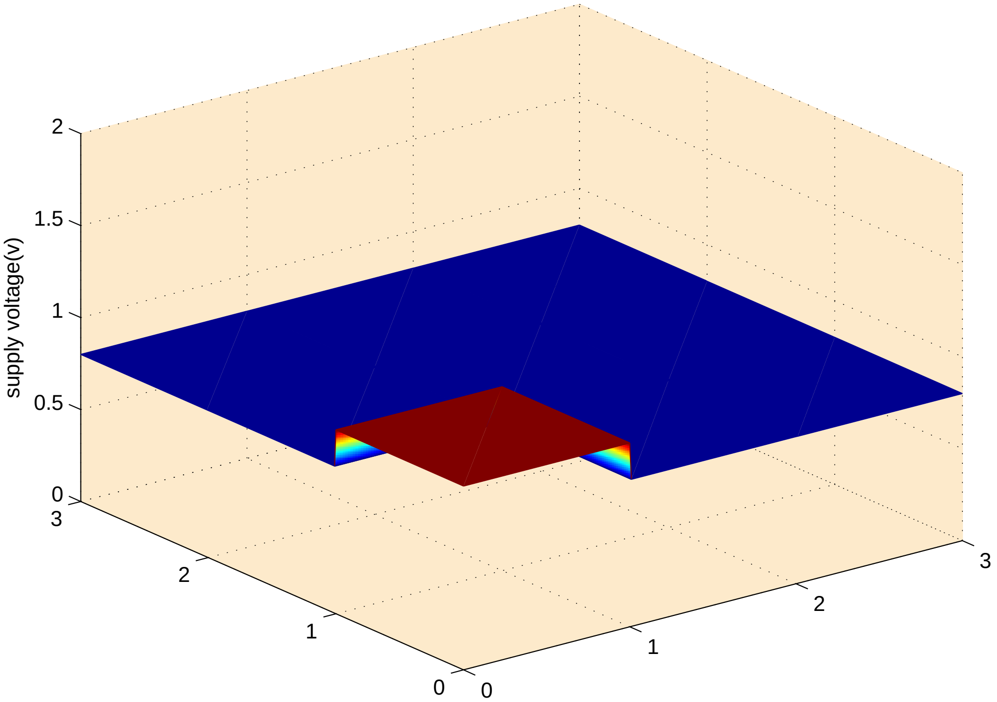

Fig. 11. Supply voltage for each tile in telecommunication.

CHAPTER III

CONCLUSION

In this work, we propose a new clustering approach for voltage-frequency optimization in NoC-based systems. It minimizes a linear combination of energy and latency penalty enabled by Lagrangian relaxation. We use clustering method to solve the Lagrange subproblem and solve the dual problem by subgradient method. Experiments show our method has significant advantage in solution quality over a previous work on the problem of energy minimization under task deadline constraint.

REFERENCES

[1] J. Howard, S. Dighe, Y. Hoskote, S. Vangal, D. Finan et al, "A 48-Core IA-32 message-passing processor with DVFS in 45nm CMOS," In *Proc. of the 58th Annual International Solid-State Circuits Conference*. San Francisco, California. 2010, pp. 108-109.

[2] J. Chang and M. Pedram, "Energy minimization using multiple supply voltages," In *Proc. of the 1st Annual International Symposium on Low Power Electronics and Design*. Monterey, California. 1996, pp. 157-162.

[3] D. Sengupta and R. Saleh, "Application-driven floorplan-aware voltage island design," In *Proc. of the 45th Annual Design Automation Conference*. Anaheim, California. 2008, pp. 155-160.

[4] Q. Ma and E.F.Y. Young, "Network flow-based power optimization under timing constraints in MSV-driven floorplanning," In *Proc. of the 21st Annual International Conference on Computer-Aided Design*. San Jose, California. 2008, pp 1-8.

[5] Z. Qian and E.F.Y. Young, "Multi-voltage floorplan design with optimal voltage assignment," In *Proc. of the 13rd Annual International Symposium on Physical Design*. San Jose, California. 2009, pp. 13-18.

[6] H. Wu, I. Liu, M. Wong, and Y. Wang, "Post-placement voltage island generation under performance requirement," In *Proc. of the 18th Annual International Conference on Computer-Aided Design*. San Jose, California. 2005, pp. 309-316.

[7] R.L.S. Ching, E.F.Y. Young, K.C.K. Leung, and C. Chu, "Post-placement voltage island generation," In *Proc. of the 19th Annual International Conference on Computer-Aided Design*. San Jose, California. 2006, pp. 641-646.

[8] U.Y. Ogras, R. Marculescu, P. Choudhary, and D. Marculescu, "Voltage frequency island partitioning for GALS-based networks-on-chips," In *Proc. of the 44rd Annual Design Automation Conference*. San Francisco, California. 2007, pp. 110-115.

[9] J. Hu and R. Marculescu, "Energy-aware communication and task scheduling for Network-on-Chip architectures under real-time constraints," In *Proc. of the 7th Annual Design, Automation and Test in Europe Conference and Exhibition*. Paris, France. 2004, pp. 234-255.

[10] J. Hu and R. Marculescu, "Energy-aware mapping for tile-based NoC architecture under performance constraints," In *Proc. of the 8th Annual Asia and South Pacific Design Automation Conference*. Kitakyushu, Japan. 2003, pp. 233-239.

[11] Y. Hu, Y. Zhu, H. Chen, R. Graham, and C. Cheng, "Communication latency aware low power NoC synthesis," In *Proc. of the 43rd Annual Design Automation Conference*. San Francisco, California. 2006, pp. 574-579.

[12] W. Jang, D. Ding, and D.Z. Pan, "A voltage-frequency island aware energy optimization framework for Networks-on-Chip," In *Proc. of the 21st Annual International Conference on Computer-Aided Design*. San Jose, California. 2008, pp. 264-269.

[13] X. Ye, Y. Zhan, and P. Li, "Statistical leakage power minimization using fast equi-slack shell based optimization," In *Proc. of the 44th Annual Design Automation Conference*. San Francisco, California. 2007, pp. 853–858.

[14] C. Zhuo, Y-H Chang, D. Sylvester, and D. Blaauw, "Design time body bias selection for parametric yield improvement," In *Proc. of the 15th Annual Asia and South Pacific Design Automation Conference*. Taipei, Taiwan. 2010, pp. 681–688.

[15] T. Sakurai and A.R. Newton, "Alpha-power law MOSFET model and its applications

to CMOS inverter delay and other formulas," *IEEE Journal of Solid-State Circuits*, vol. 25, no. 2, pp. 584-594, April 1990.

[16] J.A. Butts and G.S. Sohi, "A static power model for architects," In *Proc. of the 33rd Annual International Symposium of Microarchitecture*. Monterey, California. 2000, pp. 191-201.

[17] J. Cong, C. Liu, and G Reinman, "ACES: Application-specific cycle elimination and splitting for deadlock-free routing on irregular network-on-chip," In *Proc. of the 47th Annual Design Automation Conference*. Anaheim, California. 2010, pp. 443-448.

[18] T. Chelcea and S.M. Nowick, "A low latency fifo for mixed-clock system," In *Proc. of the 3rd IEEE Computer Society Workshop on VLSI*. Orlando, Florida. 2000, pp. 119.

[19] C. Chen, C.C.N. Chu, and D.F. Wong, "Fast and exact simultaneous gate and wire sizing by lagrangian relaxation," *IEEE Transactions on Computer Aided Design*, vol. 18, no. 7, pp. 1014-1025, July 1999.

[20] G.D. Micheli, *Synthesis and Optimization of Digital Circuits*. NY: McGraw-Hill, Inc., 1994.

[21] M.S. Bazaraa, H.D. Sherali, and C.M. Shetty, *Nonlinear Programming: Theory and Algorithms*. 3rd ed. NJ: WILEY, 2006.

[22] J.A. Hartigan, *Clustering Algorithms*. NJ: WILEY, 1975.

[23] M. Cho, Y. Kun, Yongchan Ban, and D.Z Pan, "ELIAD: Efficient lithography aware detailed router with compact post-OPC printability prediction," In *Proc. of the 45th Annual Design Automation Conference*. Anaheim, California. 2008, pp. 504-509.

[24] J. Hiriart-Urruty and C. Lemarecha, *Fundamentals of Convex Analysis*. NY: Springer, 2001.

[25] R. Dick, “Embedded system synthesis benchmarks suites(E3S),” Univ. of Michigan, DearBorn, MI [online]. Available: http://ziyang.eecs.umich.edu/ dickrp/e3s/

VITA

Yu Yang received both the B.S. degree and the M.S. degree in electrical engineering from Zhejiang University at China in June 2006 and in June 2008 respectively. He received the M.S. degree in computer engineering from Texas A&M University in May 2011. His research interests are mainly on VLSI Computer Aided Design including floorplanning, voltage island partition, multiple supply voltage scheduling. His mailing address is Department of Electrical and Computer Engineering, Texas A&M University, 214 Zachry Engineering Center, College Station, TX 77843-3128.

The typist for this thesis was Yu Yang.