

Efficiency-Driven Design Time Optimization of a Hybrid Energy Storage System with Networked Charge Transfer Interconnect

Qing Xie¹, Younghyun Kim², Donkyu Baek³, Yanzhi Wang¹, Massoud Pedram¹, and Naehyuck Chang³

¹University of Southern California, CA, USA, ²Purdue University, IN, USA, ³Korea Advanced Institute of Science and Technology, Korea
Email: ¹{xqing, yanzhiwa, pedram}@usc.edu, ²yhkim1@purdue.edu, ³{donkyu, naehyuck}@cad4x.kaist.ac.kr

Abstract—This paper targets at the state-of-art hybrid energy storage systems (HESSs) with a networked charge transfer interconnect and solves a node placement problem in the HESS, where a node refers to a storage bank, a power source, or a load device, with its distributed power converter. In particular, the node placement problem is formulated as how to place the nodes in a HESS such that the optimal total charge transfer efficiency is achieved, with accurate modelings of all kinds of different components in the HESS. The methodology of FPGA placement problem is adopted to solve the node placement in HESS by properly defining a cost function that strongly relates the charge transfer efficiency to the node placement, properties of HESS components, as well as applications of the HESS. An algorithm that combines a quadratic programming method to generate an initial placement and a simulated annealing method to converge to the optimal placement result is presented in this paper. Experimental results demonstrate the efficacy of the placement algorithm and improvements in the charge transfer efficiency for various problem setups and scales.

Keywords—hybrid energy storage system; networked charge transfer interconnect; placement

I. INTRODUCTION

Energy storage systems (ESSs) store the excess energy generated from power sources for future use of load devices. This is a promising means to enhance the energy efficiency and reliability of the power grid and renewable power sources. Many ESSs have been implemented for commercial purpose. For instance, Golden Valley Electric Authority (GVEA) in Alaska runs a grid-scale NiCd battery-based ESS [1]. It is composed of 13,760 NiCd battery cells and capable of providing 27 MW of power for 15 minutes when generation or transmission related power outage happens. Unfortunately, no energy storage technology can simultaneously satisfy all the desirable features of an ideal ESS (e.g., low total cost, high storage density, high output power level, low self leakage, eco-friendliness, and so on). Designing hybrid energy storages systems (HESSs) that consist of multiple, heterogeneous energy storage elements is an emerging solution that enables a cost-effective and energy-efficient approach by fully exploiting strengths of different energy storage technologies as much as possible [2]. Proper system-level design and management methodologies that are modular, scalable, and highly optimized should be developed for grid-scale HESSs.

The design of a HESS system is based on *nodes*, which are defined as a storage bank, a power source, or a load device with its own distributed input/output power converter. The HESS consists of various types of storage banks and input/output power converters. Power sources and load devices are also connected to the HESS and thereby, we consider them as part of the HESS for general purpose. A charge transfer interconnect (CTI) provides paths to transfer electric energy among the storage banks, power sources and load devices. The architecture of the CTI has a significant impact on the HESS charge transfer efficiency. The state-of-art CTI architecture adopts

a networked CTI architecture [3] that achieves high charge transfer efficiency even there are a large number concurrent transfer tasks in the HESS. The networked CTI provides independent paths for concurrent charge transfer tasks using dynamically configurable CTI routers and links.

Routability in the HESS is defined as how to perform more charge transfer tasks concurrently and efficiently subject to limited amount of resources, where the resource refers to the CTI links and routers. The routability in the HESS heavily depends on the placement of the nodes. A poor node placement may put frequently connected source and destination nodes far away from each other so that the routing has to use a lot of CTI links and routers to connect them, which results in a heavy congestion and, therefore, is undesirable. In this paper, we develop a placement algorithm similar to the conventional integrated circuits (IC) placement algorithms. Unfortunately, existing IC placement algorithms cannot be directly applied to the HESS placement problem because the typical objective of the IC placement is to minimize the total wire length, which is an explicit function of the component locations. The HESS placement problem in the networked CTI, however, has a very different objective - it must result in a solution that, after routing is done, would yield the maximum charge transfer efficiency for a given set of nodes and charge transfer tasks. The key challenge is that the charge transfer efficiency is not an explicit function of the placement of source and destination nodes. In addition, multiple charge transfer tasks in the HESS can share paths at the cost of a charge transfer efficiency degradation due to the CTI voltage issue, which has no counterpart concept in conventional IC placement problem formulation.

Key contributions of this paper are summarized below. First, we discuss the impact of the node placement on the charge transfer efficiency, which provides not only the motivation for the node placement problem, but also clues in deriving the solution method. Second, we present a precise formulation of the inputs, outputs, and objective of the node placement problem for the networked CTI. We keep the problem formulation consistent with the previous research on HESSs and networked CTIs to estimate the charge transfer efficiency accurately. Finally, we present an effective HESS placement algorithm combining a quadratic programming (QP) method to generate a good initial placement and a simulated annealing (SA)-based method to converge to the optimal placement. We demonstrate that the proposed algorithm achieves significant improvements of charge transfer efficiency in experiments on various HESS scales and problem setups.

II. HESS PLACEMENT PROBLEM STATEMENT

A HESS is defined as a set of nodes $N = \{n_i\}$ and a CTI G . A node n_i is either a storage bank, a power source, or a load device. Each node has a power converter to generate controlled voltage or current to interface to the CTI. The CTI is described as a graph $G = (V, E)$, where $V = \{v_i\}$ is a set of vertices and E is a set of bidirectional edges between the vertices. A vertex is a CTI router physically connected to a node, and an edge is a CTI link connecting vertices. We describe

This work is supported in part by a grant from the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NRF-2014-023320) and the National Science Foundation.

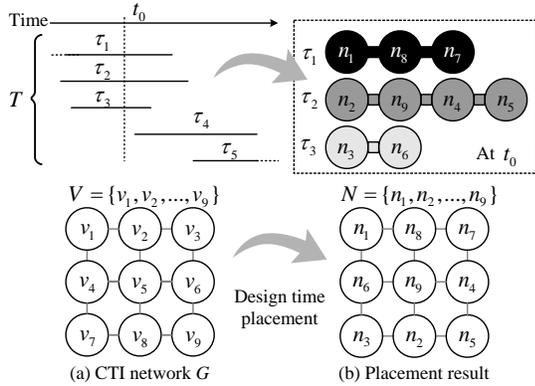


Fig. 1: A placement example of a 3-by-3 networked CTI architecture.

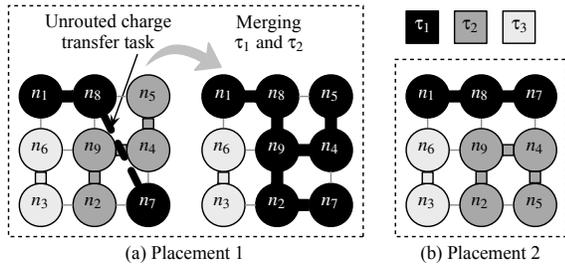


Fig. 2: CTI routing examples of charge transfer tasks in Fig. 1 on two different node placements. The placement shown in (a) requires task merging to complete routing, whereas another placement shown in (b) does not.

the problem setup of the node placement problem in the HESS in this section. We are given a set of charge transfer tasks $\mathbf{T} = \{\tau_i\}$, and the i -th charge transfer task is defined as a five-tuple $\tau_i = (\Sigma_i, \Delta_i, \epsilon_i, R_i, D_i)$, where Σ_i is the source node, Δ_i is the destination node, ϵ_i is the amount of energy to be transferred to Δ_i , R_i is the task arrival time, and D_i is the duration. A task τ_i and the participating nodes Σ_i and Δ_i are *active* during the time period $[R_i, R_i + D_i]$.

The node placement is a design time problem that seeks to find an injective function (one-to-one function) f_p from N to V , where each n_i is mapped to one v_j . Each v_j has a fixed location (x_j, y_j) , and we call that n_j is placed at (x_i, y_i) if n_j is mapped to v_i . Fig. 1 illustrates an example of placement and routing procedure. Given a task set \mathbf{T} and a CTI G shown in Fig. 1(a), we have a placement result shown in Fig. 1(b) by mapping n_1 to v_1 , n_2 to v_8 , n_3 to v_7 , and so on. Physical meaning of mapping a node to a vertex is that we plug in the power converter of the node to a CTI router that corresponds to the vertex. Without loss of generality, we limit the networked CTI topology to a mesh grid (a regular matrix of vertices with edges that connect neighboring vertex pairs) like the example shown in Fig. 1(a) throughout this paper, but the proposed concept is also applicable to other network topologies.

The CTI routing is a runtime procedure that finds independent routing paths, i.e., assigns the CTI links to each τ_i to make paths, that connects all the nodes in Σ_i and Δ_i during the period of D_i for each $\tau_i \in \mathbf{T}^a$, where $\mathbf{T}^a \subset \mathbf{T}$ is the set of active charge transfer tasks. The CTI routing procedure involves a charge transfer optimization, which finds the optimal CTI voltage V_{cti}^{opt} of the charge transfer task that maximizes the charge transfer efficiency. The networked CTI periodically performs the CTI routing to maintain the best CTI configuration at all times for the given \mathbf{T}^a . However, the number of CTI links is limited, and it may be impossible to assign independent

paths to all tasks when there is a large number of concurrent tasks. Merging task resolves the congestion issue [3][4] and, however, degrades the charge transfer efficiency. Although unnecessary task merging can be avoided by performing routing properly, the solution quality of the CTI routing strongly depends on the node placement in the HESS. As an example, Fig. 2 shows routing results of a task set $\mathbf{T}^a = \{\tau_1, \tau_2, \tau_3\}$ at t_0 in two different placements of the 3-by-3 networked CTI architecture shown in Fig. 1. The placement in Fig. 2(a) requires to merge τ_1 and τ_2 to obtain a feasible routing, which results in a charge transfer efficiency degradation due to the sharing of CTI voltage. In contrast, the placement in Fig. 2(b) does not require task merging and all tasks can be performed at their V_{cti}^{opt} .

III. SOLUTION METHOD

A. General Approach

The placement problem in IC designs, which determines locations of components such as circuit modules or logic cells in ASICs and maps the circuit's subcircuits into programmable logic blocks in field-programmable gate arrays (FPGAs), is one of most difficult issues in electronic design automation domain. Various approaches have been explored to develop effective and efficient IC placement algorithms in past decades [5]. Force-directed placement methods model connectivity of components as pulling and pushing forces and place the components where the force is minimized [6][7]. Partitioning-based methods iteratively divide components into two sets such that the number of nets connecting the two sets is minimized [8][9]. SA-based placement algorithms iteratively move components and evaluate the cost and reduce the movement range as the temperature decreases [10].

In spite of the diverse placement algorithms, the HESS placement problem has some special features aforementioned so that no existing IC placement algorithm can be applied directly. Compared to ASIC placement problems, the HESS placement problem is more similar to an FPGA placement problem because both seek an mapping from a given set of nodes (or subcircuits) to a pre-defined CTI architecture (or an FPGA). The networked CTI architecture in a HESS has a mesh-grid structure that is similar to the array of logic blocks in an FPGA. Therefore, we adopt the methodology of solving FPGA placement problems in this work. Among all approaches, the simulated annealing-based method, such as the well-known *versatile place-and-route* (VPR) tool [11], produces best results in practice. Although the SA-based method is inherently slow, it is tolerable for HESS placement problems because the scale of the HESS is typically smaller than that of an FPGA. In addition, a good initial placement can still significantly reduce the runtime of the SA method. We adopt the quadratic programming method, which is an analytic algorithm, to derive the initial placement.

B. Solving HESS Placement Problems

Solving the HESS node placement problem is complicated due to two obstacles. First, the charge transfer task may arrive and finish at any time, and thus \mathbf{T}^a changes over time. Therefore, the optimal placement of nodes is not fixed but continuously changes. For instance, placing n_7 next to n_8 in Fig. 2 is good while τ_1 is active, but it is no longer beneficial once τ_1 is finished. Second, as we are targeting at maximizing the charge transfer efficiency in the HESS, there lacks an efficient way to estimate the charge transfer efficiency from the information of charge transfer tasks and locations of nodes involved. For the first obstacle, we assume that we are provided with knowledge of the charge transfer tasks such as the probability of having a charge transfer task between two particular nodes and statistical information of ϵ_i 's, R_i 's, and D_i 's. This knowledge is equivalent to the knowledge of functionality and inputs/outputs of blocks in IC placement problems that is used to determine connectivity between nodes. Therefore, this assumption

is reasonable. To overcome the second obstacle, we derive a cost function that strongly relates the charge transfer efficiency to the node placement, properties of HESS components and applications.

Maximizing the charge transfer efficiency is equivalent to minimizing the energy loss during the charge transfer process. The energy loss of a charge transfer task mainly consists of two parts: i) the energy loss during charging and discharging the storage banks, i.e., rate capacity effect; and ii) the energy loss during the current/voltage regulations in power converters. We define a *charge transfer rate* $r_{i,j}$ of task between i -th and j -th bank as $r_{i,j} = \varepsilon_{i,j}/d_{i,j}$, where $d_{i,j}$ is the time to finish this task such that $d_{i,j} \leq D_{i,j}$.

In the HESS with the networked CTI architecture, we need to account for two additional energy loss terms. First, previous analysis assumes that the charge transfer task is carried out at its V_{cti}^{opt} , however, this may not be possible in networked CTI because multiple tasks are sharing the same path and they may have different V_{cti}^{opt} 's. Second, the router requires back-to-back MOSFET switches [3] that introduce resistances along the charge transfer path and burn additional power. When a charge transfer task have a high r_i^{min} , it is better to place related nodes close together. Placing two nodes close means that it is more likely to form a short charge transfer path between and not to share the charge transfer path with another task. Acutally, placing nodes with a high r_i^{min} close in the HESS can be analogous to placing highly-connected blocks close in IC placement problems. Hence, we define a *connectivity* \mathbf{C} between two nodes as

$$C_{i,j} = p_{task}(i,j) \frac{\langle \varepsilon_{i,j} \rangle}{\langle D_{i,j} \rangle} \quad (1)$$

where $p_{task}(i,j)$ is the probability that a charge transfer task occurs between n_i and n_j , and $\langle \varepsilon_{i,j} \rangle$ and $\langle D_{i,j} \rangle$ are corresponding expected energy to be transferred and deadline. A high $C_{i,j}$ in implies that it is more beneficial to place n_i and n_j close. We adopt the probabilistic notations because we are not optimizing placement only for a specific task set, but based on our knowledge of how this HESS is used in practice. Note that the probability values can be inferred from the past task set or the prior knowledge of future task set using statistical methods such as Maximum Likelihood (ML) estimations.

The definition of connectivity in (1) is not enough to capture the efficiency degradation due to the task merging. By placing nodes that are associated to tasks with very different V_{cti}^{opt} 's far away from each other, we reduce the chance that two tasks with different V_{cti}^{opt} 's share the one charge transfer path. Thus, we introduce a *hindrance factor* $H_{i,j}$ that indicates potential congestion on the CTI links between n_i and n_j during routing. The congestion occurs if both nodes are active simultaneously, and it causes a negative impact on the charge transfer efficiency becomes if V_{cti}^{opt} difference between the charge transfers is large. The hindrance factor \mathbf{H} is defined as

$$H_{i,j} = \int p_{act}(i,t)p_{act}(j,t) \left| V_i^{opt}(t) - V_j^{opt}(t) \right| dt \quad (2)$$

where $p_{act}(i,t)$ is the probability that n_i is active at time t , and $V_i^{opt}(t)$ is average $V_{cti}^{opt}(t)$ of charge transfers that n_i is involved at time t . Similar definitions are applied to $p_{act}(j,t)$ and $V_j^{opt}(t)$. Note that participating in the same charge transfer task does not affect $H_{i,j}$ because $V_i^{opt}(t) = V_j^{opt}(t)$. A high $H_{i,j}$ implies that it is more beneficial to place n_i and n_j apart. In general, \mathbf{H} places nodes apart from each other when V_{cti}^{opt} is distinctly different and they are frequently active simultaneously. To reflect the impact of congestion on charge efficiency, we define an *adjusted connectivity weight* as

$$\mathbf{W} = \mathbf{C} - \beta \mathbf{H} \quad (3)$$

where β in (3) is a relative weighting factor.

1) *Generating the initial placement*: A good initial placement can significant reduce the runtime of the following iterative optimization-based method. We use the quadratic programming method to generate the initial placement. Let N be the number of movable nodes in the HESS and (x_i, y_i) the coordinates of the center of i -th node. We define

the cost of a placement as

$$\begin{aligned} \Phi(x,y) = & \frac{1}{2} \sum_{i \in N} \sum_{j \in N} C_{i,j} [(x_i - x_j)^2 + (y_i - y_j)^2] \\ & + \frac{1}{2} \sum_{i \in N} \sum_{f \in F} C_{i,f} [(x_i - x_f)^2 + (y_i - y_f)^2] \end{aligned} \quad (4)$$

where F is a set of fixed nodes and (x_f, y_f) is the coordinate of the f -th fixed node. Without loss of generality, we consider that all nodes connected to power sources and load devices have fixed locations. The objective function $\Phi(x,y)$ can be written in matrix notation as

$$\Phi(x,y) = \frac{1}{2} x^T \mathbf{Q} x + d_x^T x + \frac{1}{2} y^T \mathbf{Q} y + d_y^T y + \text{constant} \quad (5)$$

where \mathbf{Q} is a $N \times N$ symmetric matrix and d_x, d_y are N -dimensional vectors. Note that (5) is the standard form of a QP problem and can be separated into $\Phi(x,y) = \Phi(x) + \Phi(y)$. We use \mathbf{C} , not the \mathbf{W} , in (4), because that all elements in \mathbf{C} are non-negative. This guarantees that \mathbf{Q} is a positive definite matrix, and hence, (5) can be solved in polynomial time by using the ellipsoid method.

2) *Converging to the optimal placement*: The simulated annealing is a generic probabilistic metaheuristic used in the global optimization problem of finding a good approximation to the global optimum of a given function. The HESS placement problem, like IC placement problems, is a hard problem that cannot be optimally solved in polynomial time. Thus, we adopt a SA-based method to further improve the solution quality based on the initial placement. The objective function that our SA-based method minimizes is

$$\begin{aligned} \Omega(x,y) = & \frac{1}{2} \sum_{i \in N} \sum_{j \in N} W_{i,j} [(x_i - x_j)^2 + (y_i - y_j)^2] \\ & + \frac{1}{2} \sum_{i \in N} \sum_{f \in F} W_{i,f} [(x_i - x_f)^2 + (y_i - y_f)^2] \end{aligned} \quad (6)$$

We terminate SA-based method when the temperature reduces below a certain threshold or no better solution is found after a certain number of trials. A near-optimal placement is returned by the presented SA-based method.

IV. EXPERIMENTS RESULTS

A. Experimental Setup

We demonstrate that the proposed node placement algorithm with network CTI examples and evaluate the charge transfer efficiency improvement. The size of CTI network ranges from 5×5 ($|V| = 9$) to 30×30 ($|V| = 900$), and we have the same number of nodes ($|N| = |V|$) with the size of CTI network. A node can be one of four types: power source, load device, and two types of storage banks (battery bank and supercapacitor bank). We consider that charge transfers only take place among different types of nodes only, which is a practical usage scenario. We adopt the power converter model from [12] and the routing algorithm from [3] for the runtime CTI routing and charge transfer optimization.

B. Algorithm Verification

We first show that the presented node placement algorithm successfully reduces the network congestion, which will eventually reduce charge transfer task merging during routing. We present an example of a small 5×5 networked CTI for demonstration purpose. We randomly generate a task set \mathbf{T} of 59 tasks. Fig. 3(a) and (b) show results of a randomly generated placement and the optimal placement by the proposed algorithm, respectively. The left-hand side graphs of Fig. 3 show the placement of 25 nodes. The connecting lines denotes $C_{i,j} > 0$ between each pair of nodes, and the color intensity of a line is associated to the magnitude of $C_{i,j}$. Note that we do not show \mathbf{H} to make Fig. 3 more readable because almost every two nodes has a non-zero \mathbf{H} -element. CTI links are more likely to be occupied to form a routing path when it is near a line.

We calculate the congestion of a CTI link by summing up connectivity of tasks close to it. The right-hand side graphs in Fig. 3

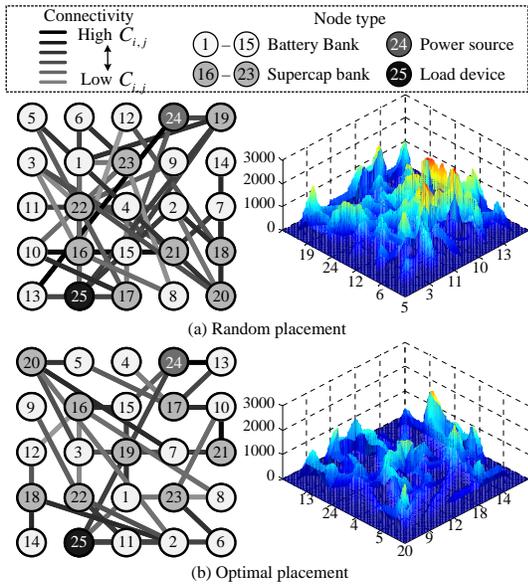


Fig. 3: Connectivity weight $C_{i,j}$ between nodes (left) and routing congestion graphs (right) of (a) a random placement and (b) the optimal placement by the proposed placement algorithm.

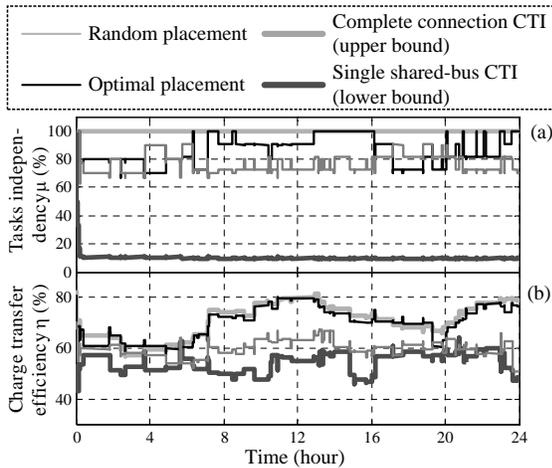


Fig. 4: (a) Task independency μ and (b) charge transfer efficiency η over time for 5×5 CTI in Fig. 3.

are three-dimensional presentations of the routing congestion in the networked CTI. Charge transfer tasks that require congested CTI links are more likely to be merged, which results in a charge transfer efficiency degradation. By comparing Fig. 3(a) and (b), one can see that lines between nodes with high $C_{i,j}$'s are shortened significantly, and the congestion density decreases and evenly distributed across the whole CTI network in the optimal placement result.

C. Charge Transfer Efficiency Enhancement

We next demonstrate that the optimized node placement leads to higher charge transfer efficiency. We define task independency μ as the ratio between numbers of tasks before and after merging (100% on the complete connection CTI). Figs. 4(a) and (b) show μ and η of two placements in Fig. 3, respectively. We compare with two baselines: i) a complete connection CTI for the upper bound of η (no task merging is required and connection resistances are estimated based on the optimal placement); and ii) a single shared-bus CTI for the lower bound of η (all concurrent tasks are merged and connection resistances are estimated based on a random

TABLE I: Charge transfer efficiency (η) for different CTIs.

CTI size	HESS nodes	Complete connection	Single shared-bus	Random placement	Optimal placement
3×3	9	79.5%	68.7%	76.4%	79.5%
5×5	25	71.0%	54.0%	59.5%	69.9%
10×10	100	75.3%	59.9%	64.6%	75.1%
20×20	400	81.9%	71.3%	73.6%	81.9%
30×30	900	79.4%	62.2%	68.4%	79.1%

placement.) One can see that the time-averaged μ is 77.1% on the random placement and 86.0% in the optimal placement. Hence, the optimal placement derived by the proposed placement algorithm requires fewer task merging. Less number of task merging leads to a higher charge transfer efficiency as shown in Fig. 4(b). Tab. I shows charge transfer efficiency results on various networked CTI structures. Results shown in Tab. I demonstrate up to 16.9% improvement in charge transfer efficiency when compared with the single shared-bus CTI, and up to 10.7% when compared with the non-optimal randomly placed networked CTI. The charge transfer efficiency on the optimal placement is very close to the maximum charge transfer efficiency.

V. CONCLUSIONS

The networked CTI requires judicious placement of nodes in order to maximize the routability of the HESS and the benefit of its high scalability. This paper introduced the node placement problem for the first time. We presented a mixed HESS placement algorithm that combines a quadratic programming method to generate a good initial placement and a simulated annealing-based method to converge the optimal placement results. The proposed placement algorithm significantly improves the charge transfer efficiency by placing nodes that are strongly connected close to each other to reduce routing congestion. The experimental results demonstrate a significant charge transfer efficiency improvement up to 17% when compared with the single shared-bus CTI, and up to 9% when compared with the non-optimal randomly placed networked CTI.

REFERENCES

- [1] Golden Valley Electric Authority, "http://www.gvea.com/energy/bess."
- [2] M. Pedram, N. Chang, Y. Kim, and Y. Wang, "Hybrid electrical energy storage systems," in *ISLPED*, 2010, pp. 363–368.
- [3] Y. Kim, S. Park, N. Chang, Q. Xie, Y. Wang, and M. Pedram, "Networked architecture for hybrid electrical energy storage systems," in *DAC*, 2012, pp. 522–528.
- [4] Q. Xie, D. Zhu, Y. Wang, Y. Kim, N. Chang, and M. Pedram, "Efficient scheduling algorithm for multiple charge migration tasks in hybrid electrical energy storage systems," in *ASP-DAC*, 2013, pp. 749–754.
- [5] I. Markov, J. Hu, and M.-C. Kim, "Progress and challenges in VLSI placement research," in *ICCAD*, 2012, pp. 275–282.
- [6] P. Spindler, U. Schlichtmann, and F. Johannes, "Kraftwerk2 —a fast force-directed quadratic placement approach using an accurate net model," *IEEE TCAD*, vol. 27, no. 8, pp. 1398–1411, 2008.
- [7] M.-C. Kim, D.-J. Lee, and I. Markov, "SimPL: An effective placement algorithm," *IEEE TCAD*, vol. 31, no. 1, pp. 50–60, 2012.
- [8] A. E. Caldwell, A. B. Kahng, and I. L. Markov, "Can recursive bisection alone produce routable placements?" in *DAC*, 2000, pp. 477–482.
- [9] M. Can Yildiz and P. H. Madden, "Improved cut sequences for partitioning based placement," in *DAC*, 2001, pp. 776–779.
- [10] K. Vorwerk, A. Kennings, and J. Greene, "Improving simulated annealing-based FPGA placement with directed moves," *IEEE TCAD*, pp. 179–192, 2009.
- [11] V. Betz and J. Rose, "Vpr: A new packing, placement and routing tool for fpga research, field-programmable logic and applications," in *FPL*, 1997.
- [12] Y. Choi, N. Chang, and T. Kim, "DC-DC converter-aware power management for low-power embedded systems," *IEEE TCAD*, vol. 26, no. 8, pp. 1367–1381, 2007.