

Verification of Balancing Architectures for Modular Batteries

Martin Lukasiewicz, Sebastian Steinhorst, Swaminathan Narayanaswamy
TUM CREATE, Singapore
martin.lukasiewicz@tum-create.edu.sg

ABSTRACT

Large battery packs consisting of a high number of cells are essential in electric vehicles as well as in smart grids as stationary energy buffers. In this context, active cell balancing techniques improve the lifetime and capacity of battery packs significantly by equalizing charge at runtime. Modern balancing circuits rely on switching schemes to transfer charge between cells via energy storage elements such as inductors or capacitors. Verifying correct functionality of complex architectures can become a non-trivial task where circuit and control have to be considered concurrently. For this purpose, we provide a framework for the verification of balancing architectures, using a methodology that takes advantage of graph search algorithms. While this paper focuses on inductor-based architectures, the proposed approach might also be extended to other storage elements such as capacitors or transformers. The experimental results based on several case studies give evidence that a manual verification becomes impractical and our framework is capable of either proving correctness or delivering a counter-example.

Categories and Subject Descriptors

B.7 [Integrated Circuits]: Design Aids—*verification*

General Terms

Design, Verification

Keywords

Battery, Architecture, Cell Balancing

1. INTRODUCTION

Environmental changes and shortage of fossil energy sources are main reasons for an increased interest in electric power solutions. Renewable energy from solar or wind might be directly converted into electric power and fed into the electricity grid. However, due to the volatility of renewables, electric energy buffers become necessary. While there

This work was financially supported by the Singapore National Research Foundation under its Campus for Research Excellence And Technological Enterprise (CREATE) programme.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

ESWEEK'14 October 12 - 17 2014, New Delhi, India
ACM Copyright 2014 978-1-4503-3051-0/14/10 ...\$15.00.
<http://dx.doi.org/10.1145/2656075.2656104>.

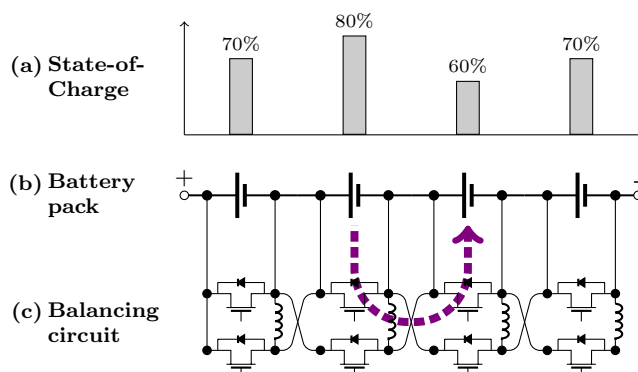


Figure 1: Illustration of a battery pack with series-connected cells. In order to equalize the SoC of cells, charge is transferred via inductors (as energy storage elements) using the balancing circuit with MOSFETs that control the current flow.

are many approaches to buffer electric energy, stationary batteries are considered as important components in decentralized smart grids. To further reduce the dependency on fossil fuels, electric vehicles are a sustainable transportation solution where efficient and cheap battery packs are a crucial component.

Battery packs for smart grids and electric vehicles often consist of a high number of cells. These might range from small cylindrical cells that are commonly used in notebook computers to pouch cells that can weigh more than 1kg. To achieve a high power and energy density, Lithium-Ion (Li-Ion) cells are used in electric vehicles while Nickel-Metal Hydride (NiMH) cells might be used in stationary storage systems. Independent of the underlying chemistry, cells are commonly connected in series to provide a desired power output of battery packs, see Figure 1b. Note that cells in parallel increase the pack capacity and mutually equalize charge by default such that they are considered in the remainder of the paper as one single cell (which is also known as *brick*).

One major challenge in battery pack design with series-connected cells is charge balancing. Among the cells exists a certain variation in terms of capacity that depends on the production process. This variation might even increase over operation time as the cells age due to uneven heat distribution along the pack. In case the cell with the lowest capacity is discharged, the entire battery pack cannot be used anymore even if other cells still have charge. In this case, the capacity of the entire battery pack is reduced significantly. See Figure 1a, where the third cell with the lowest SoC might become a bottleneck.

As a remedy, active cell balancing solutions implement

additional electric circuits, see Figure 1c, that are capable of transferring charge in order to equalize the State-of-Charge (SoC) between cells and increase the usable capacity of a battery pack. While there exist many variants for active cell balancing, this paper focuses on inductor-based solutions which are more efficient than other approaches but also require a more complex control. Using sophisticated architectures, the balancing efficiency can be significantly increased, enabling also, for instance, direct charge transfer between cells that are not adjacent. However, with growing complexity, correctness of the architecture cannot be determined manually in an efficient manner.

Contribution of the paper. A verification of cell balancing architectures becomes necessary to guarantee the correct functionality. This correct functionality ensures also that, for instance, hazardous configurations like short circuits will never occur during operation. In case of growing complexity of balancing circuits and control schemes, manual verification becomes infeasible due to the high amount of scenarios. Note that a manual verification is tedious and might be error-prone, preventing the rapid development of new cell balancing architectures with specific requirements. As a remedy, this paper proposes a framework for the automatic verification of active cell balancing architectures.

The proposed methodology performs the verification in four steps: (1) First, the circuit is transformed into a graph, capturing all necessary characteristics of electric components. (2) The graph is then modified for each phase of the charge transfer. (3) Using the modified graphs, current flows in each phase are determined by a graph search algorithm. (4) Finally, the flows are compared with a set of rules that determine whether the circuit satisfies all requirements or certain violated constraints exist. Such a verification is carried out for a predefined set of scenarios, making it possible to extend the verification results to large battery packs, consisting of a high number of cells.

Using the proposed methodology, a framework was developed that enables the fast design and verification of new cell balancing architectures. Using the framework, experimental results show the verification of existing architectures from literature. For instance, for the architecture in [1], the framework reveals a design flaw. Finally, we propose a development board for inductor-based charge transfer that might be used as platform for the design of novel balancing architectures. Using our framework, we successfully verify a complex non-neighbor transfer control scheme for the development board.

Organization of the paper. The remainder of the paper is organized as follows. Section 2 discusses related work, giving an overview of cell balancing architectures. In Section 3, functionality of balancing architectures is explained and the model for a proper specification of these architectures is introduced. Section 4 proposes the verification methodology. Finally, Section 5 outlines the developed software framework and shows experimental results before the paper is concluded in Section 6.

2. RELATED WORK

In the following, related work in the domain of design automation for battery packs as well as balancing architecture design is presented. Finally, verification and simulation approaches in battery pack design are presented.

Battery Pack Design Automation. Battery packs require complex monitoring and control which is carried out at run time by a Battery Management System (BMS). An overview of BMSs and their tasks such as cell balancing is given in [2]. In [3], a modular cell-array is proposed, including intelligent management and exploiting several battery-related characteristics.

Recently, several design automation techniques deal with

hybrid energy storage systems. These hybrid systems consist of different storage elements such as battery packs or super-capacitors. A major challenge is to design an algorithm that efficiently distributes charge between the elements. This has been applied to photo-voltaic systems [4], electric vehicles [5], and grid applications [6, 7]. A further direction of research is the optimization of hybrid battery packs by choosing an optimal configuration [8]. These works consider each battery pack as a black-box, i.e., cell balancing at battery pack level is not considered.

Balancing Architectures. Cell balancing is an important task performed by BMSs in order to equalize the charge levels of individual cells in a series-connected battery pack. A review of the various cell balancing approaches is done in [9, 10, 11]. Architectures are broadly classified into *passive* and *active* cell balancing.

Passive cell balancing involves dissipation of excess energy of an overcharged cell across a resistor until the charge levels are equalized. The approach might be implemented with a fixed resistor as in [12] or as a controlled shunting where the balancing is performed with a switch or relay [13]. The advantage of this technique is its ease of implementation and low cost in comparison with the active cell balancing techniques. However, the major disadvantage of passive cell balancing is the low efficiency since energy is dissipated as heat across the resistors.

In contrast to passive cell balancing, active approaches increase the energy efficiency of battery packs significantly by transferring excess charge from cells with high SoC to cells with low SoC. This is done via temporary energy storage elements such as capacitors, transformers, or inductors.

Capacitor-based cell balancing techniques use capacitors to transfer charge between cells, see [14, 15, 16, 17]. This method is characterized by simple control requirements and low installation space. However, the major disadvantage of this method is the inherent energy loss associated with charging capacitors from cells.

Transformers might be used for transferring energy between cells as proposed in [18, 19, 20, 21]. Fast equalization speed and non-neighbor charge transfers are main advantages of transformer-based active cell balancing techniques. However, transformers can be expensive, heavy, and might require large installation space as well as complex control.

Inductor-based balancing is energy efficient compared to capacitor-based approaches since there is practically no energy loss while charging. Compared to the transformer approach, inductors also require less installation space and are lighter. Thus, inductor-based architectures are good candidates for sustainable active cell balancing. Single-inductor balancing architectures use only one inductor in order to equalize the entire battery pack, see [22]. Cell pairs for balancing are selected by the control algorithm and corresponding switches associated with the cells are controlled in order to exchange charge. This method lacks modularity and does not allow concurrent charge transfers. In contrast, architectures using more than one inductor are a promising approach, see [1, 23]. The main advantage of this approach is its energy efficiency and modular nature of the architecture. Approaches range from simple architectures that can perform charge transfers only between neighboring cells, see [23], to complex architectures that enable non-neighbor transfers and cell bypassing [1]. Due to their modularity, inductor-based approaches are furthermore well-suited for application in emerging smart cell architectures [24].

Verification Techniques. Besides the energy storage elements, a major component in active cell balancing architectures are power MOSFETs that control the transfer. Continuous advances in MOSFET design and development result in improvements in terms of costs, size, and energy-efficiency. As a consequence, complex active cell balancing architectures with specific features such as non-neighbor transfers

become economical. On the other hand, complex balancing circuits in combination with a sophisticated control scheme prohibit a manual verification. Verification techniques for active cell balancing mainly deal with low level electrical properties as presented for instance in [25]. Like many other works, this approach considers SoC models at cell-level in case of balancing and does not consider the balancing architectures. Other approaches from the domain of design automation deal with the optimal design of balancing architectures [26], but do not focus on their verification.

The approach in [27] comes closest to the proposed work in this paper. In [27], the model-checker SAL [28] is used to verify a module-based balancing architecture that is based on transformers. In contrast, we propose a framework that uses graph search and, thus, might result in a better scalability. In fact, early implementations of our framework used a satisfiability (SAT) solver [29], resulting in a too large design space for the verified inductor-based architectures.

To the best of our knowledge, there exists no approach for cell balancing architecture verification at system-level as proposed in this paper. While a Simulation Program with Integrated Circuit Emphasis (SPICE) simulation [30] covers only a specific scenario in a specific configuration, the proposed work is capable of quickly analyzing all possible scenarios and configurations of a balancing circuit. At the same time, the proposed framework performs a high-level verification and further optimization steps of the architectures and their components would have to be carried out at lower levels with tools such as SPICE.

3. CELL BALANCING

In the following, the functionality of cell balancing architectures is introduced, using the example architecture from Figure 1. To formalize cell balancing, a model is presented that describes the circuit and control.

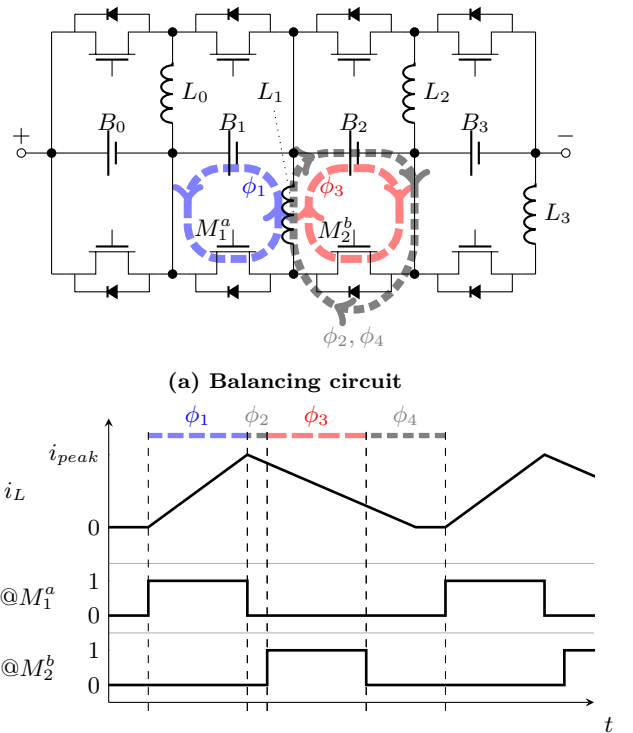
3.1 Balancing Architectures

Active cell balancing for battery packs relies on architectures that are capable of transferring charge between cells. Such an architecture, which is a combination of a balancing circuit and control scheme, is illustrated in Figure 2. The architecture consists of battery cells B_i , a set of MOSFETs M_i^j , and inductors L_i . A MOSFET is controlled like a switch and it always has a diode between the drain and source that blocks current in one direction. Note that if current shall be blocked in both directions, it is necessary to use two MOSFETs with opposing diodes.

By controlling the MOSFETs, charge can be transferred between cells via the inductors as temporary energy storage elements. For this purpose, MOSFETs can be either closed (1), open (0), or controlled by PWM signals. For the example architecture in Figure 2a, charge is transferred if all MOSFETs are open except two that are controlled by non-overlapping PWM signals as illustrated in Figure 2b. The PWM signals result in four phases ϕ_1 to ϕ_4 that are occurring periodically and ensure the charge transfer.

In phase ϕ_1 , the inductor L_1 is charged from B_1 via M_1^a , which is temporarily closed. The duration of this *charging phase* has to be chosen in consideration of the inductance and cell voltage to ensure an optimal utilization of the inductor. Note that the current flow does not pass any diode since this would dissipate energy and unnecessarily reduce the efficiency of the transfer.

In phase ϕ_2 , M_1^a is opened and the charging of the inductor is discontinued. This *freewheeling phase* is necessary to account for the time that MOSFETs require to change between their open and closed state and vice versa. Without phase ϕ_2 , it could happen that both MOSFETs M_1^a and M_1^b would be closed for a short amount of time and a short circuit over B_1 and B_2 via these two switches could occur.



(b) Control scheme and resulting inductor current

Figure 2: Example of a balancing architecture that illustrates a charge transfer between neighboring cells via an inductor. The transfer is carried out in four phases ϕ_1 to ϕ_4 where appropriate PWM signals are applied to the MOSFETs. Note that the balancing circuit is identical to the circuit in Figure 1.

At the same time, it has to be ensured that each charged inductor in the freewheeling phase can discharge via a diode, otherwise the voltage of the inductor would immediately increase and all energy is dissipated via the inductor itself, damaging components.

In phase ϕ_3 , M_2^b is temporarily closed and the inductor discharges its energy into cell B_2 . In this *discharge phase*, the current flow does not go via any diode to ensure an energy-optimal transfer.

In the final phase ϕ_4 , M_2^b is opened and the discharging of the inductor is carried out via the diode. This *blocking phase* ensures that the inductor is fully discharged, but at the same time B_2 cannot start charging the inductor in the opposite direction once the current flow stops.

3.2 Modeling

In the following, a model for balancing architectures is defined. A battery pack consists of a set of $N = \{0, 1, \dots, |N| - 1\}$ series-connected modules where each module $n \in N$ consists of

- the battery cell B_n ,
- a set of MOSFETs M_n^i ,
- a set of diodes D_n^i (which are part of the MOSFETs), and
- an inductor L_n .

The set of elements might be extended by capacitors, transformers, and other components.

A transfer scenario is defined by \mathcal{T} where a single transfer $T = (T_S, T_D) \in \mathcal{T}$ is defined by the source modules $T_S \subseteq N$ and destination modules $T_D \subseteq N$. For a transfer T , it holds that sources and destinations are disjoint, $T_S \cap T_D = \{\}$. Additionally, each transfer defines a domain by

$$d(T) = \{\min(\{n|n \in T_S \cup T_D\}), \dots, \max(\{n|n \in T_S \cup T_D\})\} \quad (1)$$

and it holds that domains are disjoint:

$$\forall T, \tilde{T} \in \mathcal{T} : d(T) \cap d(\tilde{T}) = \{\} \quad (2)$$

To enable charge transfer, certain MOSFETs have to be either statically open or closed, or controlled by predefined PWM signals. The set of all signals is defined by Σ and for each $\sigma \in \Sigma$ the function

$$\delta : \Sigma \rightarrow \mathbb{R} \times \mathbb{R} \quad (3)$$

determines the start and end time when the signal is active, i.e., the respective MOSFETs are temporarily closed.

For a certain transfer scenario \mathcal{T} , the configuration of each MOSFET M is determined by the function

$$c(\mathcal{T}, M) \in \{0, 1\} \cup \sigma. \quad (4)$$

This function determines if the respective MOSFET is open, closed, or controlled by one of the given signals $\sigma \in \Sigma$.

From the signals, phases Φ are determined such that a single phase $\phi \in \Phi$ is defined by a subset of signals $\phi \subseteq \Sigma$. Corresponding to the signals, each phase has a start and end time determined by

$$\delta : \Phi \rightarrow \mathbb{R} \times \mathbb{R}. \quad (5)$$

Two phases $\phi, \tilde{\phi} \in \Phi$ are consecutive if there exists a $t \in \mathbb{R}$ such that $\delta(\phi) = (t', t)$ and $\delta(\tilde{\phi}) = (t, t'')$. The requirement for consecutive phases

$$|(\phi \cup \tilde{\phi}) \setminus (\phi \cap \tilde{\phi})| = 1 \quad (6)$$

ensures that only a single signal changes. When the duration of each phase is longer than the off/on times of the used MOSFETs, this ensures that there will not exist any inconsistent state.

4. VERIFICATION

In the following, our verification methodology is proposed which is outlined in Figure 3. In a first step, the circuit is transformed into a graph, capturing all specific elements such as cells, inductors, MOSFETs, and diodes. In the next step, this graph has to be modified for each charge transfer and phase, depending on the switching rules. Here, vertices are removed that represent MOSFETs which are open during the respective phase. In the following, possible charge flows on the graphs are identified by an introduced graph search algorithm. Finally, the flows are evaluated using a predefined rule set that ensures the correctness of the designed circuit.

4.1 Circuit Transformation

In the first step of the verification methodology, the circuit is transformed into a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ where \mathcal{V} is the set of vertices and \mathcal{E} represents the set of directed and undirected edges, respectively. In order not to lose information about the original circuit, different types of edges and vertices have to be introduced.

The vertices are defined as follows:

- Each MOSFET of the original circuit is transformed into a corresponding vertex. The set of MOSFET vertices is given as $\mathcal{V}_M \subset \mathcal{V}$ where each MOSFET M_n^i corresponds to a vertex in \mathcal{V}_M .

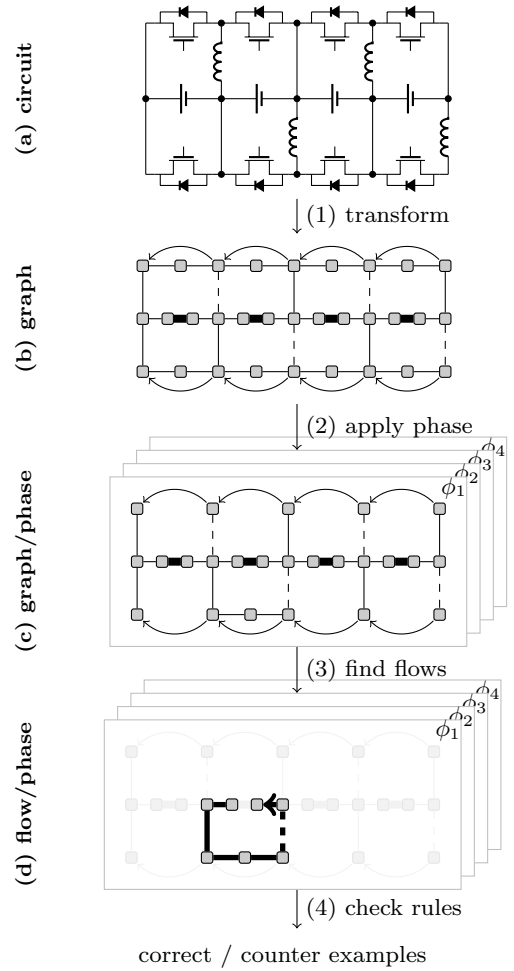


Figure 3: Illustration of the proposed verification methodology. (1) The circuit is transformed to a graph, (2) open MOSFETs are removed for each phase, (3) current flows are determined using graph search algorithms, (4) and, finally, the current flows are checked to determine whether the charge transfer is feasible.

- Each battery cell is transformed into two vertices (and one edge). The vertices are given as $\mathcal{V}_B \subset \mathcal{V}$ with $\mathcal{V}_B = \mathcal{V}_{B^+} \cup \mathcal{V}_{B^-}$ where \mathcal{V}_{B^+} represents the set of positive terminals and \mathcal{V}_{B^-} the set of negative terminals, respectively.

- All remaining vertices $\mathcal{V} \setminus (\mathcal{V}_B \cup \mathcal{V}_M)$ are used for linking elements and correspond to connecting terminals in the circuit.

The edges are defined as follows:

- Each inductor of the original circuit is transformed into a corresponding edge. The set of inductor edges is $\mathcal{E}_L \subset \mathcal{E}$ where each inductor L_n corresponds to an undirected edge in \mathcal{E}_L .
- Each battery cell is transformed into an undirected edge that is connected by the respective positive and negative vertices. The set of battery edges is defined as $\mathcal{E}_B \subset \mathcal{E}$.
- Each diode is transformed into a directed edge. The set of diodes is given as $\mathcal{E}_D \subset \mathcal{E}$.

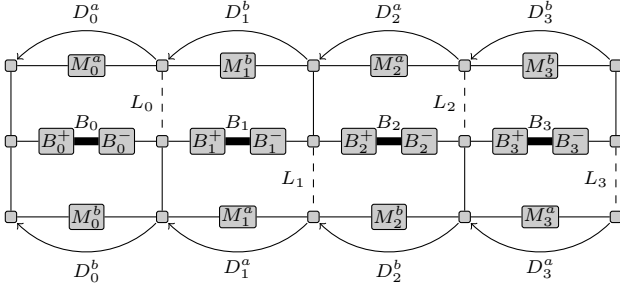


Figure 4: Graph for the circuit in Figure 2a. MOSFETs, cells, inductors, and diodes are transformed into vertices ($v \in \mathcal{V}$) and edges ($e \in \mathcal{E}$), respectively.

- All remaining edges $\mathcal{E} \setminus (\mathcal{E}_B \cup \mathcal{E}_L \cup \mathcal{E}_D)$ represent electrical connections that correspond to the original circuit.

Given a circuit in a suitable netlist format such as SPICE [30], the corresponding graph can be obtained with an appropriate parser. It is also possible to model the circuit manually with the transformation rules given above. Exemplary, the circuit in Figure 2a is transformed to the graph in Figure 4.

4.2 Phase Application

Depending on the phase, the circuit graph is modified by removing MOSFET vertices and their incident edges. Removing a MOSFET vertex equals to the case the respective MOSFET is in the open state (0). The verification is performed at various levels in order to identify counterexamples in different stages. The first stage is invariant of signals to identify flaws that are independent of the PWM signals. In the next stage each transfer is considered separately (local) before the architecture is verified for all possible concurrent transfers (global).

Given the configuration function c , the invariant version is defined by:

$$c_0(\mathcal{T}, M) = \begin{cases} 0 & \text{if } c(\mathcal{T}, M) \in \Sigma \\ c(\mathcal{T}, M) & \text{otherwise} \end{cases} \quad (7)$$

Here, each MOSFET that is controlled by a PWM signal is assumed to be permanently open (0). Such an invariant configuration can be checked for invariant rules such as short circuits and is particularly helpful to determine flaws in the configuration that are independent of the PWM switching.

In case of a local transfer analysis, only a single transfer $T \in \mathcal{T}$ is verified. In this case, the modified configuration function is defined as follows:

$$c_T(\mathcal{T}, M) = \begin{cases} 0 & \text{if } c(\mathcal{T}, M) \in \Sigma \wedge d(M) \notin d(T) \\ c(\mathcal{T}, M) & \text{otherwise} \end{cases} \quad (8)$$

Here, each MOSFETs that is controlled by a PWM signal in another domain is set to be permanently opened (0). Note that $d(M)$ returns the module of a MOSFET. This local transfer configuration is used to check for flaws in a single transfer independent of PWM signals outside its domain.

Finally, a global verification becomes necessary in case of concurrent transfers. Here, the configuration function remains unchanged and the actual behavior of the balancing circuit and its control is modeled. Note that in case there exists no concurrent transfer, the local transfer configuration equals the original configuration function.

Depending on the phase $\phi \in \Phi$, the respective configura-

tion is applied to the graph as follows:

$$c(\mathcal{T}, M, \phi) = \begin{cases} 1 & \text{if } c(\mathcal{T}, M) \in \{1\} \cup \phi \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

If the result of this function is 0, the vertex is removed from the circuit graph. That means that all MOSFET vertices that are either closed (1) or controlled by a signal that is active in the current phase, remain in the graph and can, therefore, be used for a current flow.

4.3 Flow Determination

To determine the current flows in each phase, the obtained and modified circuit graphs are used. Here, each current and voltage source has to be taken into account. A battery cell is a voltage source, i.e., the voltage is constant while the current is determined based on the resistance. On the other hand, charged inductors are current sources where the voltage is adapting to enable the current flow.

Using the *Dijkstra* algorithm on the graphs with appropriate weights that reflect the resistance of certain elements is a feasible method to determine the path of least resistance. However, current flows might be taking various paths and this has to be taken into account. In the following, an algorithm is presented that is capable of determining the flow between two vertices in the graph. The flow between a source vertex s and destination vertex d is defined by a set of paths $\Pi_{(s,d)}$ such that each path π has at least one edge that is not used by other paths:

$$\forall \pi \in \Pi_{(s,d)} : \exists e \in \pi : \forall \tilde{\pi} \in \Pi_{(s,d)} \setminus \{\pi\} : e \notin \tilde{\pi} \quad (10)$$

This ensures that each path in the flow contributes at least one edge, i.e., non-redundant information. In contrast, a k -shortest paths algorithm could result in a very high number of paths and, thus, redundant information that would have to be taken into account in the subsequent rule checking phase.

Algorithm 1 is capable of determining the flow as defined in Eq. (10). It requires a circuit graph (at a certain phase) and a function that defines edge weights (lines 1,2). These edge weights are determined based on the source of the current flow which is explained subsequently. The function $in(v)$ contains the shortest path from the source to the vertex v and correspondingly $w_{in}(v)$ determines the distance of this path. The function $in(v)$ is initialized with an empty set for each vertex except the source while the distances are set to ∞ except for the source that has a distance of 0 (lines 3,4). Correspondingly, the function $out(v)$ contains the shortest path to the destination vertex and $w_{out}(v)$ determines the distance (lines 5,6).

Initially, the set of paths $\Pi_{(s,d)}$ is empty (line 7). In each iteration, the next shortest path π is determined. This is done iteratively by considering all possible sub-paths from \tilde{s} to \tilde{d} that are within a path from the source or to the destination, respectively (line 11-18). In case such a path exists, it is added to the flow and its edges are removed from the graph (line 19-21). In case no further path exists, the algorithm is stopped. Finally, the functions $in(v)$, $out(v)$, $w_{in}(v)$, $w_{out}(v)$ have to be updated with the recent obtained path (line 25-34).

To determine the current flow $\Pi_{(s,d)}$, appropriate edge weights w have to be chosen. This function depends on whether the source is a battery cell or an inductor as explained in the following.

Voltage source edge weights. To determine the current flow of a battery cell, the source is set to be the corresponding positive terminal vertex in \mathcal{V}_{B^+} and the destination is the corresponding negative terminal vertex in \mathcal{V}_{B^-} . At the same time, the respective edge of the battery cell is temporarily removed from the circuit graph. The current flow is determined by consideration of resistances along possible

```

1  $\mathcal{G}(\mathcal{V}, \mathcal{E});$ 
2  $w : \mathcal{E} \rightarrow \mathbb{R};$ 
3  $in(v) = \begin{cases} \{s\} & \text{if } v = s \\ \emptyset & \text{otherwise;} \end{cases}$ 
4  $w_{in}(v) = \begin{cases} 0 & \text{if } v = s \\ \infty & \text{otherwise;} \end{cases}$ 
5  $out(v) = \begin{cases} \{d\} & \text{if } v = d \\ \emptyset & \text{otherwise;} \end{cases}$ 
6  $w_{out}(v) = \begin{cases} 0 & \text{if } v = d \\ \infty & \text{otherwise;} \end{cases}$ 
7  $\Pi_{(s,d)} = \{\};$ 
8 while true do
9    $\pi = \emptyset;$ 
10   $w_\pi = \infty;$ 
11  foreach  $\tilde{s}, \tilde{d} \in \mathcal{V}, w_{in}(\tilde{s}) \neq \infty, w_{out}(\tilde{d}) \neq \infty$  do
12    determine shortest path  $\tilde{\pi}_{(\tilde{s}, \tilde{d})}$  in  $\mathcal{G}(\mathcal{V}, \mathcal{E});$ 
13     $w_{\tilde{\pi}} = w_{in}(\tilde{s}) + w(\tilde{\pi}_{(\tilde{s}, \tilde{d})}) + w_{out}(\tilde{d});$ 
14    if  $w_{\tilde{\pi}} < w_\pi$  then
15       $w_\pi = w_{\tilde{\pi}};$ 
16       $\pi = (in(\tilde{s}), \tilde{\pi}_{(\tilde{s}, \tilde{d})}, out(\tilde{d}));$ 
17    end
18  end
19  if  $w_\pi \neq \infty$  then
20     $\Pi_{(s,d)} = \Pi_{(s,d)} \cup \{\pi\};$ 
21     $\mathcal{E} = \mathcal{E} \setminus \{e | e \in \pi\};$ 
22  else
23    break;
24  end
25  foreach  $e_x = (v, \tilde{v}) \in \pi = (e_1, \dots, e_n)$  do
26    if  $w_{in}(v) = \infty$  then
27       $in(v) = (e_1, \dots, e_{x-1});$ 
28       $w_{in}(v) = w(in(v));$ 
29    end
30    if  $w_{out}(v) = \infty$  then
31       $out(v) = (e_x, \dots, e_n);$ 
32       $w_{out}(v) = w(out(v));$ 
33    end
34  end
35 end

```

Algorithm 1: Graph search algorithm that determines the current flow by determining a set of shortest paths in the circuit graph.

paths. Therefore, the edge weights w resemble the resistance of elements as follows:

$$w(e) = \begin{cases} R_B & \text{if } e \in \mathcal{E}_B \\ R_D & \text{if } e \in \mathcal{E}_D \\ R_L & \text{if } e \in \mathcal{E}_L \\ 10^{-12} & \text{otherwise} \end{cases} \quad (11)$$

Here, R_B is the inner resistance of a battery cell, R_D the resistance of a diode, and R_L the resistance of an inductor. These resistances depend on the used components and peak current. However, in general we can make the assumption that $R_L \gg R_D \gg R_B$. It is therefore not necessary to know the detailed components at design time to determine all current flows as it is possible to assume values that satisfy the stated assumption.

It might be necessary to remove several paths from the flow if these paths contain a very high amount of diodes as the voltage drop might be higher than the cell voltage. However, for realistic scenarios this is not necessary since such paths should never exist.

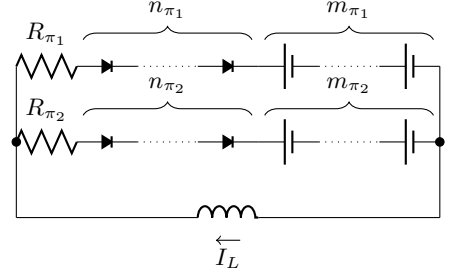


Figure 5: Illustration of the threshold constraint in Eq. (15). A current flow only exists on path π_2 in the presence of path π_1 if the constraint is satisfied.

Current source edge weights. To determine the current flow of an inductor, the source is set to the positive terminal vertex and the destination to the negative terminal vertex of the inductor. Since the inductor edge should be directed (from the negative vertex to the positive vertex) after it is charged, it is not necessary to remove it. The inductor is a current source and, therefore, the voltage is increased until the desired current flow is obtained. Thus, edge weights of the circuit graph are determined by the voltage drop across the components as follows:

$$w(e) = \begin{cases} V_D & \text{if } e \in \mathcal{E}_D \\ V_B & \text{if } e \in \mathcal{E}_B \\ 10^{-12} & \text{otherwise} \end{cases} \quad (12)$$

Note that the current I_D through a diode is determined by

$$I_D = I_s (e^{\frac{V_D}{k \cdot V_T}} - 1) \quad (13)$$

with reverse bias saturation current I_s , the voltage across the diode V_D , the nonideality constant k and the thermal voltage V_T . Thus, the voltage drop is determined as follows:

$$V_D = \ln\left(\frac{I_D}{I_s} + 1\right) \cdot k \cdot V_T \quad (14)$$

For our calculations, we choose typical values for silicon diodes with $I_s = 2.52 \cdot 10^{-12}$ A, $k = 1$ and $V_T = 0.026$ V at room temperature. For a peak current $I_D = I_{peak} = 1$ A, the voltage drop is $V_D = 0.694$ V.

Note that the nominal voltage range of most Lithium-Ion battery cells ranges from $V_{B,max} = 4$ V down to $V_{B,min} = 2$ V. We choose $V_B = 3$ V for our experiments.

Finally, a threshold is necessary to determine if a certain path is in the current flow. This threshold depends on the path with the lowest voltage drop. For this purpose, we determine the minimal and maximal voltage drop across diodes. Using Eq. (14), we use the minimal voltage drop $V_{D,min} = 0.395$ V with $10 \mu\text{A}$ and the maximal voltage drop as $V_{D,max} = 0.694$ V with 1 A. Thus, in case a path π_1 exists, a current flow over π_2 exists if the following requirement is fulfilled (compare Figure 5):

$$n_{\pi_2} \cdot V_{D,min} + m_{\pi_2} \cdot V_{B,min} \leq n_{\pi_1} \cdot V_{D,max} + m_{\pi_1} \cdot V_{B,max} + R_{\pi_1} \cdot I_{peak} \quad (15)$$

Let us assume there exists a path π_1 without a diode and a path π_2 with a diode. This is the case in phase ϕ_3 in our example from Figure 2a where also a path via the diode in module 2 exists which equals to the current flow in phase ϕ_2 and ϕ_4 , respectively. In this case, Eq. (15) becomes $V_{D,min} \leq R_{\pi_1} \cdot I_{peak}$ and with a peak current of 1 A, the resistance of path π_1 would have to satisfy $R_{\pi_1} \geq 0.395 \Omega$. However, we know that the resistance along π_1 is generally

much lower than this (otherwise too much energy would be dissipated) and, therefore, there exists no current flow through π_2 .

4.4 Rule Checking

The obtained flows have to be checked to determine whether charge is transferred correctly. This is done by a set of formal constraints that rely on the universal quantifier \forall and the existential quantifier \exists . These constraints define rules that check the determined current flows. In the following, we use $\mathcal{V}_n \subseteq \mathcal{V}$ and $\mathcal{E}_n \subseteq \mathcal{E}$ as all vertices and edges, respectively, of module $n \in N$.

Invariant. There are constraints that have to be satisfied in all phases. The first constraint ensures that short circuits do not occur at any time:

$$\forall e = (v_+, v_-) \in \mathcal{E}_B : \quad (16a)$$

$$\forall \pi \in \Pi_{(v_+, v_-)} : \quad (16b)$$

$$\exists e \in \mathcal{E}_L : e \in \pi \quad (16c)$$

This constraint holds for each cell and ensures that for each path in the flow there is at least one inductor that is passed.

Note that a closed circuit over a battery without an inductor leads to a short circuit with theoretically infinite current, also in case a forward-biased diode is in the closed circuit. We calculate the current through I_D for $V_C = 2V$ and the previously introduced diode parameters:

$$I_D = 2.52 \cdot 10^{-12} \text{A} (e^{\frac{2V}{0.026V}} - 1) = 6.44 \cdot 10^{21} \text{A} \quad (17)$$

Such a high current flow equals a short circuit.

The second invariant constraint ensures that all cells that are not sources of a transfer are not discharged:

$$\forall n \in N \setminus \bigcup_{T \in \mathcal{T}} T_S, e = (v_+, v_-) \in \mathcal{E}_B \cap \mathcal{E}_n : \Pi_{(v_+, v_-)} = \{ \} \quad (18a)$$

This constraint holds for each battery cell that is not a source and ensures that there is no existing flow.

(1) Charging. Initially, it has to be ensured that the inductors are charged by the source cells of each transfer:

$$\forall T \in \mathcal{T} : \exists \mathcal{E}_{L,T} \subseteq \mathcal{E}_L : \quad (19a)$$

$$\forall n \in T_S, e = (v_+, v_-) \in \mathcal{E}_B \cap \mathcal{E}_n : \quad (19b)$$

$$\forall \pi \in \Pi_{(v_+, v_-)} : \pi \cap \mathcal{E}_L = \mathcal{E}_{L,T} \quad (19c)$$

$$\forall e \in \mathcal{E}_{L,T} : e \in \mathcal{E}_L \cap \bigcup_{n \in d(T)} \mathcal{E}_n \quad (19d)$$

For each transfer, we define $\mathcal{E}_{L,T}$ as set of inductors that are charged from the source cells. That means that each path in a flow from the source cells passes exactly these inductors as stated in (19c). For the sake of modularity, only inductors in the domain of each transfer can be used as defined in (19d).

If an inductor is charged in a specific phase, the inductor edge is transformed to a directed edge in the next phase (with the same direction as the flow). Note that while an inductor is charged from a battery, it cannot be discharged over a reverse-biased freewheeling diode simultaneously as illustrated in Figure 6. Hence, the current through the diode is practically zero.

As the diode is reverse-biased, the diode voltage V_D is the inverse of the cell voltage V_C :

$$V_D = -V_C \quad (20)$$

Therefore, we obtain:

$$I_D = I_s (e^{\frac{-V_C}{k \cdot V_T}} - 1) \quad (21)$$

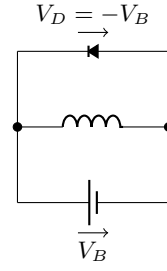


Figure 6: While an inductor is charged from a cell B , it cannot be discharged over a reverse-biased freewheeling diode of an open MOSFET as determined in Eq. (22). This ensures that inductors that are charged in a certain phase cannot discharge at the same time.

By inserting values, this yields:

$$I_D = 2.52 \cdot 10^{-12} \text{A} \left(\underbrace{e^{-\frac{2V}{0.026V}}}_{\approx 0} - 1 \right) = -2.52 \cdot 10^{-12} \text{A} \approx 0 \quad (22)$$

This is an important observation as it ensures that inductors that are charging cannot simultaneously be discharged.

(2) Freewheeling. Freewheeling is necessary if an inductor is charged and at the same time not discharging to any cell. In case a charged inductor cannot be discharged over any path, the inductor would increase the voltage until it discharges itself, possibly damaging components. The freewheeling rule is defined as follows:

$$\forall T \in \mathcal{T}, e = (v_+, v_-) \in \mathcal{E}_{L,T} : \quad (23a)$$

$$\exists \pi \in \Pi_{(v_+, v_-)} : \pi \cap \mathcal{E}_L \subseteq \mathcal{E}_{L,T} \quad (23b)$$

All charged inductors in $\mathcal{E}_{L,T}$ have to have at least one path to discharge. This path may only contain inductors that are also charged as uncharged inductors would have an infinitely high resistance in the initial moment of discharge.

(3) Discharging. In the discharging phase, the inductors discharge via the destination cells.

$$\forall T \in \mathcal{T}, e = (v_+, v_-) \in \mathcal{E}_{L,T} : \quad (24a)$$

$$\forall \pi \in \Pi_{(v_+, v_-)} : \quad (24b)$$

$$\pi \cap \mathcal{E}_B = \{e \mid e \in \mathcal{E}_B \cap \mathcal{E}_n, n \in T_D\} \wedge \quad (24c)$$

$$\pi \cap \mathcal{E}_L \subseteq \mathcal{E}_{L,T} \quad (24d)$$

This constraint ensures that each charged inductor in $\mathcal{E}_{L,T}$ is discharged with the following requirements on the current flow. Firstly, the current flow only passes exactly the destination cells as defined in (24c). Secondly, (24d) states that only charged inductors (in the same direction) can be in the current flow to avoid an infinitely high resistance along the discharge path.

(4) Blocking. Finally, the blocking phase has to ensure that inductors are not charged from the destination cells.

$$\forall T \in \mathcal{T}, e = (v_+, v_-) \exists \mathcal{E}_{L,T} : \quad (25a)$$

$$\forall \pi \in \Pi_{(v_+, v_-)} : \pi \cap \mathcal{E}_D \neq \{ \} \quad (25b)$$

The constraint states that for each discharge path, there is at least one diode in the discharge direction.

Checking order. To define the current flow in a certain phase, the presented constraints have to be checked in a specific order. First, the invariant rules have to be checked before it is checked whether source cells are charging inductors. Each inductor that was charged in some previous phase has to be checked whether it is discharged to the destination cells and whether this discharge is blocking. Finally, for

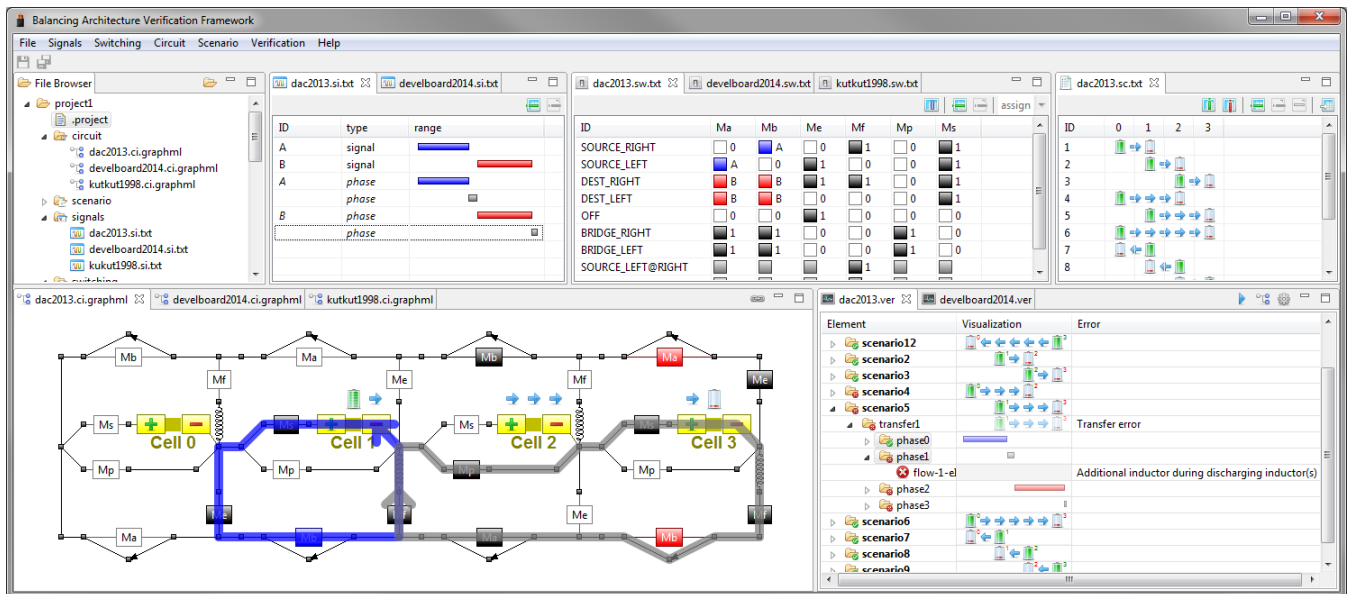


Figure 7: Screenshot of the developed framework for active cell balancing architecture verification. It enables the modification of signals, switching rules, circuit architectures, and scenarios. The verification results show flaws in the design and the respective current flows for the architecture from [1].

each charged inductor, freewheeling has to be ensured if it is not discharged via the destination cells.

5. RESULTS

In the following, experimental results using our verification methodology are presented. The developed software framework is introduced and applied to two architectures from literature. Finally, a development board for cell balancing is proposed and a non-neighbor transfer scheme is exemplary verified. All experiments were carried out on an Intel Core i5 with 2.6 GHz with 8 GB RAM.

5.1 Software Framework

We implemented the proposed verification in a software framework as illustrated in Figure 7. Using a Graphical User Interface (GUI), it is possible to define signals and phases, switching rules of MOSFETs, transfer scenarios, and the circuit architecture. Each of these components may be defined separately. By selecting specific components, the verification checks the correctness by applying the proposed methodology and determining the current flows. In case of conflicts, the verification results show which constraint is violated and it is possible to visualize the respective current flows in each phase.

The framework enables a fast design of balancing architectures with a template-based approach and, thus, a rapid prototyping is made possible. Also various design alternatives can be compared within a short amount of time. At the same time, design flaws can easily be visualized, allowing also a fast re-design of balancing circuits and switching rules.

5.2 Verification Results

We applied the proposed verification methodology to two cell balancing architectures from literature. The architecture in [23] is illustrated throughout this paper, see Figure 2, and is capable of transferring charge between neighboring cells. This architecture also enables concurrent charge transfer, i.e., multiple neighboring cell pairs can transfer charge at the same time. The second architecture is proposed in [1]

which further enables charge transfer between cells that are not neighbors.

Architecture 1. The first architecture is verified to be correct using 13 transfer scenarios for a series of five cells. Note that five cells are necessary to enable a verification of concurrent transfers where at least one module (between the transfer pairs) is acting as buffer with all MOSFETs opened. Each module of the architecture consists of two MOSFETs and one inductor. The considered scenarios include all possible neighbor transfers in both directions in a battery pack with five cells as well as concurrent transfers with a least one module as buffer where all MOSFETs are opened. The verification requires 4.6 seconds and gives evidence that the resulting architecture is correct.

The verification results can be carried over to a battery pack with significantly more than five cells as all possible concurrent transfers are covered by the verification. That means, also architectures with 100 or more cells are correct since the verification results can be applied to any part of the battery pack due to its modular architecture.

Architecture 2. The second architecture is investigated using 103 transfer scenarios, also taking non-neighbor transfers into account. We consider a configuration with seven cells in series to enable also the verification of non-neighbor and concurrent charge transfer. Each module consists of six MOSFETs. Two of the MOSFETs (p and s) have to be Insulated Gate Bipolar Transistors (IGBTs) in case of high currents and voltages. Note that for an actual implementation, the number of power MOSFETs and IGBTs increases to six and four, respectively, since it requires the implementation of two MOSFETs with opposing diodes. Due to the increased complexity and higher number of scenarios, the runtime increases to 40.5 seconds.

The results reveal a flaw in the design for some scenarios in case charge is transferred between non-neighboring cells. The flaw only appears in case the distance between neighboring cells is even, e.g., a transfer from module 1 to module 3 as illustrated in Figure 8 (here we illustrate only four cells for the sake of simplicity). The discharge takes place with an additional inductor in the circuit and, thus, the architecture violates constraint (23b). Such a discharge would increase

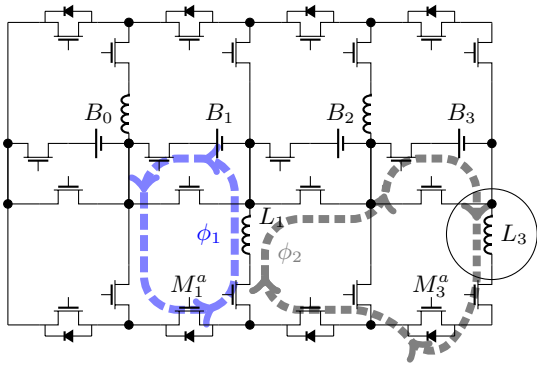


Figure 8: Illustration of the design flow for the architecture from [1]. The charged inductor L_1 is discharged over the inductor L_3 . In this case, the voltage will increase until inductor L_1 discharges via an electrical spark.

the voltage over the inductor rapidly until it discharges itself via an arc. Thus, the freewheeling requirement is violated. At the same time, constraint (19c) is violated which corresponds to the same requirement in the discharging phases.

This flaw can be resolved with the following strategies:

- Transfers between cells that have an even distance can be forbidden. In case charge shall be transferred between these cells, the respective transfer can be split into two transfers.
- An additional MOSFET might be added to the circuit to bypass the inductor at the destination module.

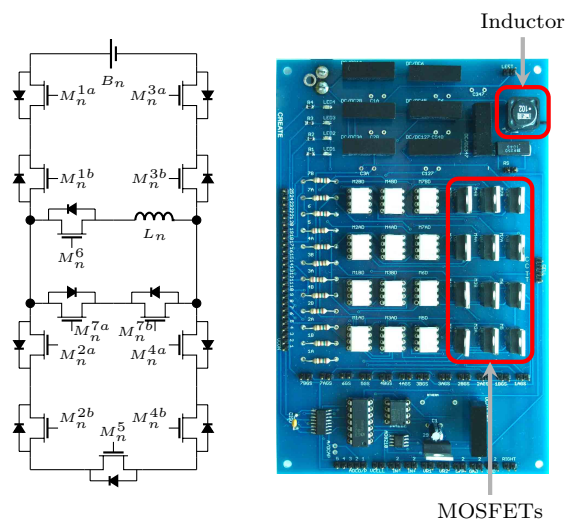
Using our verification framework, we verified both solutions to be correct.

5.3 Development Board

The proposed verification framework enabled the design of a *development board* for active cell balancing. Figure 9 shows the circuit schematic as well as the corresponding Printed Circuit Board (PCB) implementation. The circuit is using an inductor and 12 power MOSFETs and by setting some of these MOSFETs to the closed state permanently, it is possible to model various architectures. Note that the PCB implementation relies on overdimensioned discrete components including individual gate drivers and DC/DC converters, while a final production implementation would be significantly smaller. Furthermore, additional sensor circuitry has been added to evaluate the performance of the board.

We used the development board to implement a non-neighbor charge transfer. In contrast to the architecture from [1], no expensive and large IGBTs are required since none of the MOSFETs is in the series-connected battery string that would require to cope with high currents. The configuration function is given in Table 1, where for each module n one of the six defined states is satisfied and the MOSFETs are controlled correspondingly. Here, a module might be a charge source (state 1 and 2) and depending in which direction charge is transferred, a certain configuration is applied. Corresponding to the charge source rules, the charge destination rules are defined (state 3 and 4). Additionally, a module might be used between a source and destination (state 5) or a module might be in no transfer domain (state 6). An example for a non-neighbor charge transfer with this scheme is given in Figure 10. Here, module 0 is in state 6, module 1 in state 1, module 2 in state 5, and module 3 in state 3.

The verification for seven cells and 103 scenarios requires 80 seconds. The results show that in all considered scenar-



(a) Circuit schematic

(b) PCB

Figure 9: Circuit schematic and Printed Circuit Board (PCB) implementation of the development board for active cell balancing. The battery cell is connected at the pins on the right of the PCB.

ios the charge transfer is correct. Thus, the development board can be used with the proposed switching scheme for non-neighbor charge transfer. Moreover, due to the modularity, the correct behavior can be carried over to the same architecture with an arbitrary number of modules since the verification captures non-neighbor and concurrent transfers at the same time.

6. CONCLUSION

This paper proposes a verification methodology for cell balancing architectures in battery packs. The methodology is based on a transformation of the circuit to a graph representation such that graph search algorithms can be applied to determine current flows. Such a verification becomes inevitable in case of growing complexity of cell balancing architectures. Exemplary, we show the verification of two architectures from literature, revealing a flaw in one of them and proposing potential solutions to fix it. Additionally, we propose a development board for cell balancing and verify a complex scheme for charge transfer between non-neighboring cells.

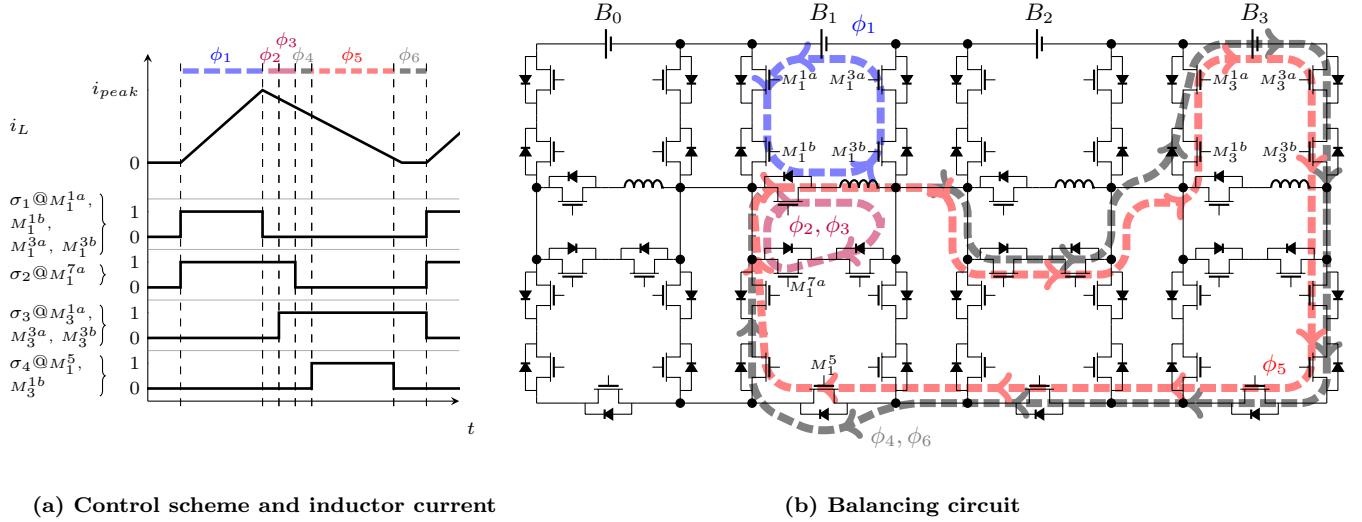
The proposed methodology verifies balancing at system-level. Future work will cover the automatic synthesis of balancing architectures and determination of optimal duration of phases depending on the used electrical components. Additionally, costs and efficiency of circuits shall be evaluated at circuit-level to enable a design space exploration of architectures.

7. REFERENCES

- [1] M. Kauer, S. Naranayaswami, S. Steinhorst, M. Lukasiewicz, S. Chakraborty, and L. Hedrich, "Modular system-level architecture for concurrent cell balancing," in *Proc. of DAC*, 2013.
- [2] M. Brandl, H. Gall, M. Wenger, V. Lorentz, M. Giegerich, F. Baronti, G. Fantechi, L. Fanucci, R. Roncella, R. Saletti, S. Saponara, A. Thaler, M. Cifrain, and W. Prochazka, "Batteries and battery management systems for electric vehicles," in *Proc. of DATE*, 2012.
- [3] S. Mandal, P. Bhojwani, S. Mohanty, and R. Mahapatra, "IntellBatt: Towards smarter battery design," in *Proc. of DAC*, 2008.

State	M_n^{1a}	M_n^{1b}	M_n^{2a}	M_n^{2b}	M_n^{3a}	M_n^{3b}	M_n^{4a}	M_n^{4b}	M_n^5	M_n^6	M_n^{7a}	M_n^{7b}
1 $\exists T \in \mathcal{T} : n \in T_s \wedge \tilde{n} \in T_d \wedge n < \tilde{n}$	σ_1	σ_1	1	1	σ_1	σ_1	0	0	σ_4	1	σ_3	0
2 $\exists T \in \mathcal{T} : n \in T_s \wedge \tilde{n} \in T_d \wedge n > \tilde{n}$	σ_1	σ_1	0	0	σ_1	σ_1	1	1	σ_4	1	σ_3	0
3 $\exists T \in \mathcal{T} : n \in T_d \wedge \tilde{n} \in T_s \wedge n > \tilde{n}$	σ_2	σ_4	0	0	σ_2	σ_2	1	1	1	0	0	0
4 $\exists T \in \mathcal{T} : n \in T_d \wedge \tilde{n} \in T_s \wedge n < \tilde{n}$	σ_2	σ_4	1	1	σ_2	σ_2	0	0	1	0	0	0
5 $\exists T \in \mathcal{T} : n \in d(T) \wedge n \notin T_s \wedge n \notin T_d$	0	0	0	0	0	0	0	0	1	0	1	1
6 $\forall T \in \mathcal{T} : n \notin d(T)$	0	0	0	0	0	0	0	0	0	0	0	0

Table 1: Module-based definition of the configuration function $c(\mathcal{T}, M)$ for the development board.



(a) Control scheme and inductor current

(b) Balancing circuit

Figure 10: Example of a non-neighbor charge transfer for the architecture using our proposed development board. The transfer is carried out in six phases.

- [4] Q. Xie, Y. Wang, Y. Kim, N. Chang, and M. Pedram, "Charge allocation for hybrid electrical energy storage systems," in *Proc. of CODES+ISSS*, 2011.
- [5] S. Park, Y. Kim, and N. Chang, "Hybrid energy storage systems and battery management for electric vehicles," in *Proc. of DAC*, 2013.
- [6] D. Zhu, S. Yue, Y. Wang, Y. Kim, N. Chang, and M. Pedram, "Designing a residential hybrid electrical energy storage system based on the energy buffering strategy," in *Proc. of CODES+ISSS*, 2013.
- [7] Q. Xie, X. Lin, Y. Wang, M. Pedram, D. Shin, and N. Chang, "State of Health Aware Charge Management in Hybrid Electrical Energy Storage Systems," in *Proc. of DATE*, 2012.
- [8] W. Chang, M. Lukasiewicz, S. Steinhorst, and S. Chakraborty, "Dimensioning and configuration of EES systems for electric vehicles with boundary-conditioned adaptive scalarization," in *Proc. of CODES+ISSS*, 2013.
- [9] J. Cao, N. Schofield, and A. Emadi, "Battery balancing methods: A comprehensive review," in *Proc. of VPPC*, Sept 2008.
- [10] S. W. Moore and P. J. Schneider, "A review of cell equalization methods for lithium ion and lithium polymer battery systems," *SAE Publication 2001-01-0959*, 2001.
- [11] M. Daowd, N. Omar, P. Van den Bossche, and J. Van Mierlo, "Passive and active battery balancing comparison based on MATLAB simulation," in *Proc. of VPPC*, 2011.
- [12] N. Kutkut and D. Divan, "Dynamic equalization techniques for series battery stacks," in *Proc. of INTELEC*, 1996.
- [13] T. Stuart and W. Zhu, "Fast equalization for large lithium ion batteries," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 24, no. 7, July 2009.
- [14] C. Pascual and P. Krein, "Switched capacitor system for automatic series battery equalization," in *Proc. of APEC*, 1997.
- [15] R. Fukui and H. Koizumi, "Double-tiered switched capacitor battery charge equalizer with chain structure," in *Proc. of IECON*, 2013.
- [16] A. Baughman and M. Ferdowsi, "Double-tiered switched-capacitor battery charge equalization technique," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 6, June 2008.
- [17] C. Speltino, A. Stefanopoulou, and G. Fiengo, "Cell equalization in battery stacks through state of charge estimation polling," in *Proc. of ACC*, 2010.
- [18] J.-W. Shin, G.-S. Seo, C.-Y. Chun, and B.-H. Cho, "Selective flyback balancing circuit with improved balancing speed for series connected lithium-ion batteries," in *Proc. of IPEC*, 2010.
- [19] A. Imtiaz and F. Khan, "Time shared flyback converter based regenerative cell balancing technique for series connected li-ion battery strings," *IEEE Transactions on Power Electronics*, vol. 28, no. 12, Dec 2013.
- [20] M. Einhorn, W. Roessler, and J. Fleig, "Improved performance of serially connected li-ion batteries with active cell balancing in electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 6, July 2011.
- [21] D. V. Cadar, D. M. Petreus, and T. M. Patarau, "An energy converter method for battery cell balancing," in *Proc. of ISSE*, 2010.
- [22] S.-H. Park, T.-S. Kim, J.-S. Park, G.-W. Moon, and M.-J. Yoon, "A new buck-boost type battery equalizer," in *Proc. of APEC 2009*, 2009.
- [23] N. Kutkut, "A modular nondissipative current diverter for ev battery charge equalization," in *Proc. of APEC*, 1998.
- [24] S. Steinhorst, M. Lukasiewicz, S. Narayanaswamy, M. Kauer, and S. Chakraborty, "Smart cells for embedded battery management," in *Proc. of CPSNA*, 2014.
- [25] J. Guerin and W. Liu, "Cell balancing algorithm verification through a simulation model for lithium ion energy storage systems," *SAE Publication 2010-01-1079*, 2010.
- [26] S. Narayanaswamy, S. Steinhorst, M. Lukasiewicz, M. Kauer, and S. Chakraborty, "Optimal dimensioning of active cell balancing architectures," in *Proc. of DATE*, 2014.
- [27] F. Baronti, C. Bernardeschi, L. Cassano, A. Domenici, R. Roncella, and R. Saletti, "Design and safety verification of a distributed charge equalizer for modular li-ion batteries," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, 2014.
- [28] S. Bensalem, V. Ganesh, Y. Lakhnech, C. Munoz, S. Owre, H. Rueß, J. Rushby, V. Rusu, H. Sardi, N. Shankar et al., "An overview of SAL," in *Proc. of LFM*, 2000.
- [29] N. Eén and N. Sörensson, "An extensible SAT-solver," in *Theory and applications of satisfiability testing*, 2004.
- [30] L. W. Nagel and D. O. Pederson, *SPICE: Simulation program with integrated circuit emphasis*. Electronics Research Laboratory, College of Engineering, University of California, 1973.