Formal Approaches to Design of Active Cell Balancing Architectures in Battery Management Systems

Invited Paper

Sebastian Steinhorst
Aarhus University, Department of Engineering
Aarhus, Denmark
sebastian.steinhorst@eng.au.dk

Martin Lukasiewycz
TUM CREATE
Singapore
martin.lukasiewycz@tum-create.edu.sg

ABSTRACT

Large battery packs composed of Lithium-Ion cells are continuously gaining in importance due to their applications in Electric Vehicles (EVs) and smart energy grids. To ensure maximum lifetime, safety and performance of the battery pack, complex embedded system architectures consisting of sensors, power electronics and microcontrollers are integrated into the pack as Battery Management System (BMS). In this context, active cell balancing is a promising approach of the BMS to provide equal charge levels across the cells in the battery pack in an efficient manner. The design of such active cell balancing architectures, comprising circuits from the power electronics domain together with complex control schemes, is error-prone and tedious when done in the conventional manual fashion. This paper presents a design flow from a high-level requirements definition to an actual hardware implementation, using design automation approaches such as verification, synthesis and optimization. Here, graph-based models and algorithms from the domain of formal verification are applied to prove the system properties and are extended to enable synthesis of optimized and correct-by-construction active balancing circuit architectures.

1. INTRODUCTION

The progressively increasing introduction of Electric Vehicles (EVs) and smart energy grids with volatile renewable sources necessitates the integration of high-performance Electrical Energy Storages (EESs). Large battery packs formed from series-connected Lithium-Ion (Li-Ion) battery cells are the established choice for storing energy due to their beneficial energy and power density, as well as their cycle lifetime, dominating all other available battery chemistries. However, Li-Ion batteries are highly sensitive with regard to their operating parameters [3]. Using battery cells outside their specified range in terms of voltage or temperature causes irreversible damage, reducing the performance significantly. In severe cases, such abuse can lead to thermal runaway which results in an ignition of the cells and complete destruction of the battery pack, potentially threatening human lives. Consequently, complex embedded system architectures are integrated into battery packs to act as Battery Management Systems (BMSs) in order to monitor and control the operation of the cells and pack at all times [4]. This ensures an efficient, safe and reliable operation of the battery pack.

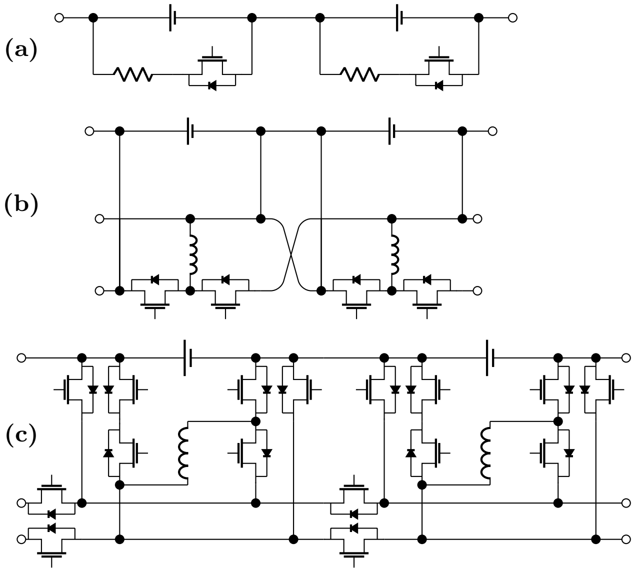

Figure 1: Three balancing circuits: passive using a switched resistor for power dissipation (a), simple neighbor-only active charge transfer (b) from [1] and non-neighbor active charge transfer (c) from [2]. The design of complex architectures such as (c) requires domain-specific automated methods incorporating formal approaches.

Besides providing information on the State-of-Charge (SoC) and State-of-Health (SoH) of the battery pack, an important function of the BMS is charge equalization. Individual battery cells in an electrical series-connection behave differently over time regarding their discharge profile due to manufacturing variations and temperature differences. As

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

ICCAD '16, November 07-10, 2016, Austin, TX, USA
© 2016 ACM. ISBN 978-1-4503-4466-1/16/11…$15.00
DOI: http://dx.doi.org/10.1145/2966986.2980088

the charging or discharging of the battery pack has to be stopped once the first cell reaches the upper or lower voltage threshold, charge equalization has to be performed such that all battery cells can provide their maximum energy storage capacity. State-of-the-art implementations utilize passive cell balancing architectures where a switched resistor across each cell enables controlled discharging of cells as illustrated in Figure 1(a). During charging, cells that reach their upper voltage threshold earlier than others can hence be individually discharged such that the SoC of other cells can catch up and all cells reach their maximum SoC to achieve a fully charged battery pack.

Passive balancing, while being easy to control and cheap to implement, is not efficient from an energy perspective as excess charge is dissipated in form of heat during the charging process. By contrast, active cell balancing architectures as illustrated in Figures 1(b) and 1(c) use circuits with temporary energy storage elements to transfer charge between battery cells. Consequently, excess charge is transferred instead of wasted.

While active cell balancing using charge transfer circuits is a promising approach to increase the efficiency of battery packs, most industrial applications continue to apply passive balancing. A reason for this might be the more complex design and control of active cell balancing architectures. Besides additional hardware overhead that needs to be justified by the higher efficiency of the battery pack, circuit networks incorporating transistor switches have to be operated using Pulse Width Modulation (PWM) signals in the kHz frequency range to control the charge transfer via temporary energy storage elements such as inductors, capacitors or transformers. Recent trends in BMS architectures towards modularity [5, 6, 7, 8] will, however, pave the way for a simplified introduction of active balancing from a hardware integration perspective.

Even for simple architectures only providing charge transfer between neighboring cells, as shown in Figure 1(b), their implementation requires thorough verification of both circuits and their control, as switching errors may result in short circuits which may lead to thermal runaway of the cells. For more capable circuit architectures enabling charge transfer between non-neighboring cells, such as shown in Figure 1(c), resulting in faster and more efficient transfers, PWM signals in several phases have to be controlled. This leads to more switching scenarios than what can be efficiently designed and verified manually or using existing tools such as circuit simulation.

As a remedy, approaches introducing design automation for active cell balancing architectures have been proposed recently. Here, domain-specific modeling and formal verification approaches enable a complete analysis of all possible operation scenarios of the active cell balancing architecture, comprising the actual circuit network together with the control scheme [9]. In a further step, this method is extended to synthesis of correct-by-construction active cell balancing architectures which outperform all existing implementations [2]. These approaches to architecture design and verification are complemented by optimal hardware selection and dimensioning approaches [10], as well as using advanced modeling techniques for the design of optimized temporary energy storage elements [11].

Based on the aforementioned recent contributions to design automation of active cell balancing architectures, this paper introduces a complete design methodology from a high-level requirements specification to an optimized hardware implementation.

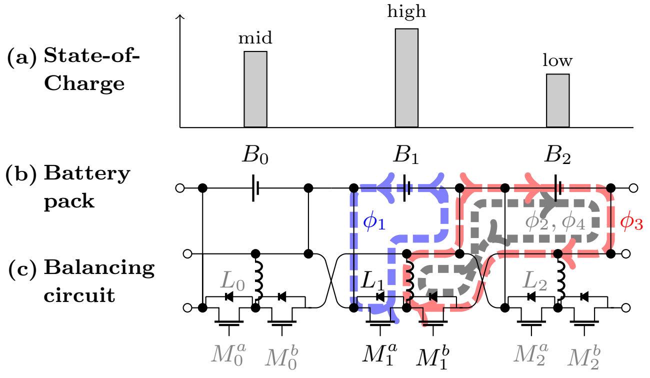

Figure 2: Illustration of the SoC (a) of cells in a battery pack with series-connected cells (b). In order to equalize the SoC of cells, charge is transferred via inductors (as energy storage elements) using the active balancing circuit with MOSFETs that control the current flow (c).

2. BALANCING ARCHITECTURES FOR BATTERY MANAGEMENT

BMSs are the key contribution to battery packs from the embedded systems and power electronics domain, as they perform sensing, computation, control and communication. This complex system architecture continuously monitors basic properties of the battery cells such as their voltage and temperature. Together with current measurement, which can be performed on pack level due to the series-connection of cells and hence same current through all cells, these measurements allow to estimate the SoC of the battery pack, which is determined by the cell with the lowest individual charge. The actual estimation of the SoC is an ongoing field of research and challenging, as noisy observations under load have to be performed and interpreted [3]. Besides the current SoC which needs to be determined as accurately as possible, the SoH takes into account aspects of charge cycling, capacity changes and temperature evolution over time and therefore characterizes both the current performance abilities of the cells as well as their long-term development.

Due to operating temperature differences and manufacturing variations, which are exacerbated over time when the battery pack ages, the individual cells in a pack show different characteristics regarding their charging/discharging behavior. As charging and discharging is only performed on pack-level on the terminals of the series-connected pack, a charge imbalance between the battery cells is hence emerging over time as illustrated in Figure 2(a). Li-Ion cells cannot be charged or discharged beyond their specified operating voltage thresholds and, therefore, require cell balancing which equalizes the charge between cells and ensures that all cells in the pack can be charged to full capacity.

In contrast to the dissipative approach of passive cell balancing, active balancing architectures efficiently transfer charge between battery cells and can therefore also equalize a pack when a weak cell reaches its cutoff threshold. This increases the effective capacity of a battery pack as the complete stored charge in all battery cells can be utilized when performed with appropriate foresight or using sufficiently dimensioned circuits. Figure 2(b) shows an exemplary battery pack consisting of three series-connected cells with a balancing circuit in Figure 2(c) which can transfer charge between neighboring cells. Here, a charge transfer from battery cell B_1 to cell B_2 is performed in four phases ϕ_1 to ϕ_4 as explained in the following. In phase ϕ_1, inductor L_1 is charged from battery cell B_1, controlled by closing the MOSFET switch M_a^1. In phase ϕ_2, all switches are open for a short period of time and freewheeling occurs through the body diode of M_b^1, starting to discharge the inductor into cell B_2. In order to maintain a low path resistance for an efficient charge transfer, M_b^1 is closed soon after M_a^1 was opened and the inductor continues to discharge into the cell in phase ϕ_3. Finally, just before L_1 is completely discharged, M_b^1 is opened for freewheeling phase ϕ_4. These four phases are periodically performed with a frequency in the kHz range, depending on the inductance.

Recently, more complex architectures, such as the one based on the circuit illustrated in Figure 1(c), have been proposed. Such architectures not only enable charge transfer between adjacent battery cells with respect to the series connection, but also allow to transfer charge directly to non-neighboring cells. This further increases the efficiency of the balancing process, due to a higher degree of freedom to assign balancing partners, but also requires a more complex circuit architecture and control. The MOSFET switches in the circuit shown in Figure 1(c) are controlled in five signal phases with three different signals as described in detail in [2]. Here, both circuit and control are the result of an automated synthesis approach, optimizing the number of circuit components required as well as the number of signals. As we will illustrate in the remainder of this paper, it is possible to develop a design flow which uses design automation and formal methods in order to obtain such optimized correct-by-construction active cell balancing architectures from an initial requirements specification.

3. ACTIVE CELL BALANCING ARCHITECTURE DESIGN FLOW

In this Section we will introduce a complete computer-aided design flow for active cell balancing architectures. The flow starts with a requirements specification which is translated into design goals for the architecture synthesis and electrical parameters for the component selection and hardware optimization. In the next stage, a choice of circuit and signal templates based on the architecture requirements is used to synthesize a correct-by-construction balancing architecture. Here, the architecture is synthesized, which consists of a balancing circuit together with the control signals required during the different phases of the circuit operation such as charging the inductor from the source cell and discharging it into the destination cell. In the next step, based on the number of components required for the chosen architecture, a component selection is performed, using further optimization approaches in order to reach a desired trade-off between cost, installation space, performance and efficiency. Finally, the synthesized signals are mapped to the actual implementation platform and parameterized regarding, e.g., switching frequency and timing patterns. The resulting implementation can be investigated in a system-level analysis to validate the operation of all design choices made. This design flow is illustrated in Figure 3 and detailed in the following subsections.

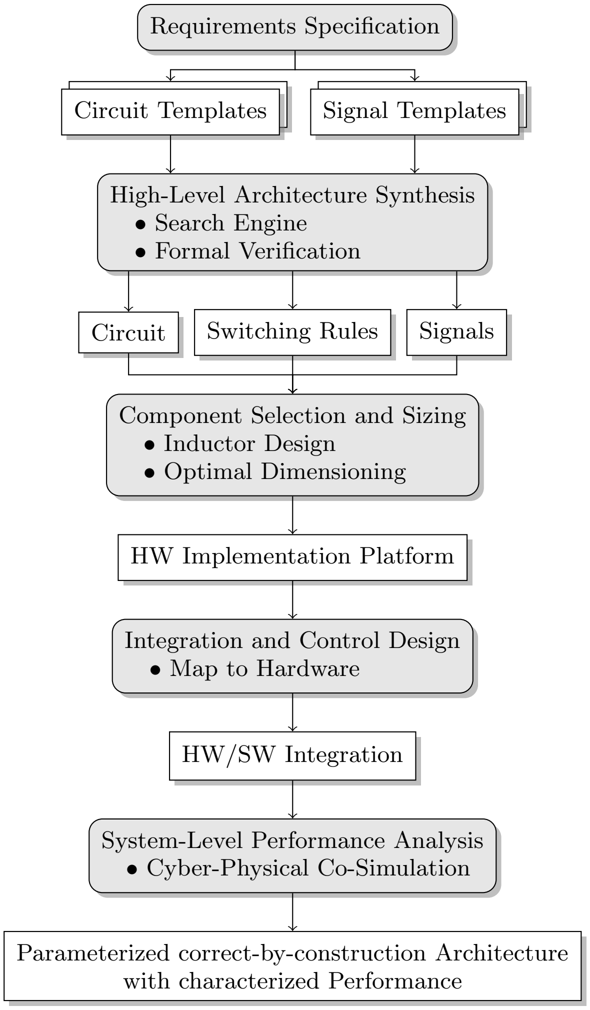
Requirements Specification
Circuit Templates
Signal Templates
High-Level Architecture Synthesis
• Search Engine
• Formal Verification
Circuit
Switching Rules
Signals
Component Selection and Sizing
• Inductor Design
• Optimal Dimensioning
HW Implementation Platform
Integration and Control Design
• Map to Hardware
HW/SW Integration
System-Level Performance Analysis
• Cyber-Physical Co-Simulation
Parameterized correct-by-construction Architecture
with characterized Performance

Figure 3: Design flow for active cell balancing architectures using design automation and formal approaches. Starting with a requirements specification, a parameterized correct-by-construction architecture with characterized performance is obtained.

3.1 Requirements Specification

Before designing an active balancing circuit for a battery pack, some key questions need to be answered. First, it has to be defined whether the circuit shall allow balancing between neighboring cells in the series-connection or also between cells that are not neighbors. In the latter case, the maximum distance for the charge balancing, i.e., how many cells might be bypassed, has to be part of the specification. Further constraints or design objectives, respectively, for the circuit involve the number of maximum MOSFETs as well as the complexity of the switching scheme.

Beyond the previously discussed architectural aspects of the balancing circuits, there are certain electrical parameters which need to be specified as they strongly influence the hardware implementation. The major electrical parameter is the targeted average balancing current. The higher this current, the faster the charge is transferred between cells. While faster charge transfer is desirable from a functional perspective, unfortunately, higher currents introduce implementation challenges which usually result in a trade-off between cost, efficiency and performance. Generally, the size of inductors increases with higher inductor currents at same transfer efficiency, bringing issues of installation volume and weight. Reaching higher currents with smaller inductors usually requires increased PWM frequencies and introduces higher internal resistances, which reduces the charge transfer efficiency. Beyond that, there is the general relation of $P = I^2 \cdot R$ which inherently reduces balancing efficiency with higher currents for a given architecture. Achieving an optimal hardware implementation for a targeted average balancing current will be discussed in Subsection 3.3.

3.2 High-Level Architecture Synthesis

The proposed synthesis method is using templates for the circuit and signals which are in the considered cases a result of a manual process. Here, the circuit templates represent general switching networks while the signal templates rely on tried and tested patterns. Given the circuit and signal templates, the optimal architectures are determined within an iterative process. The resulting circuit is based on its template where specific MOSFETs might be removed to reduce their overall number. Correspondingly, the resulting signals are derived from the corresponding template by considering only those signals that are used by the switching rules. The resulting switching rules are the key element as they define which signals and MOSFETs are used, implicitly encoding the optimization objectives such as the number of required MOSFETs.

The proposed iterative synthesis approach for active balancing architectures is illustrated in Figure 4. One iteration consists of two steps: In the first step, a *search engine* determines a potential solution. In the second step, the proposed solution is verified. The synthesis is carried out for exactly one circuit template and signal template. That means, in case of multiple templates, all combinations of circuits and signals can be synthesized and compared.

In the first step, the search engine uses a satisfiability (SAT) solver to determine potential active balancing architectures. For this purpose, a set of binary variables as well as linear constraints is given in [2] to define the search space. Linear constraints are used instead of pure Conjunctive Normal Form (CNF) to allow a higher expressiveness. It should be mentioned that, technically, SAT solvers which are capable of handling these constraints are also known as Pseudo-Boolean (PB) solvers. In practice, SAT solvers can be extended to handle these linear constraints directly or via a translation into CNF [12].

The search space is constrained in a way such that each feasible active balancing architecture equals a variable assignment that satisfies all constraints. On the other hand, not each feasible solution of the constrained problem equals a feasible active balancing architecture. This is due to the fact that prohibitive rules such as the exclusion of short circuits cannot be efficiently defined as SAT problem. Note that prohibitive rules might be formalized using Quantified Boolean Formula (QBF) [13] which are extending SAT with universal and existential quantifiers. However, known solvers for QBF can only address problems about two orders of magnitude smaller than SAT, see [13], making it impossible to solve the discussed problem in a reasonable amount of time.

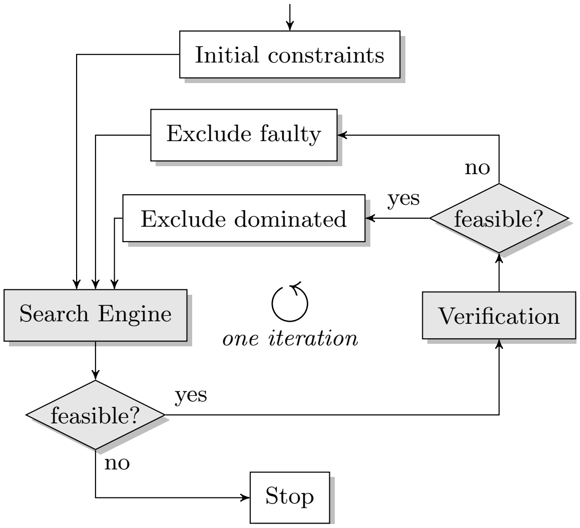

Figure 4: Illustration of the iterative synthesis approach for active cell balancing architectures from [2], consisting of two major steps: the search engine and verification.

Since the original SAT problem does not exclude each potentially infeasible balancing architecture, a verification step becomes necessary which is detailed in [9]. Infeasible balancing architectures can be identified and the reasons for the infeasibility are subsequently excluded from the search space. This is done by extending the SAT problem with additional constraints that prohibit specific undesired current flows such as short circuits. On the other hand, if the balancing architecture is determined to be feasible by the verification, the objectives are obtained and all dominated solutions are excluded from the search space. Since the considered objectives are linear, this is simply achieved by extending the problem with a set of additional constraints.

With this iterative approach, the synthesis is carried out until the remaining search space is empty, i.e., the search engine does not find any feasible solution anymore, and all optimal balancing architectures are determined.

Note that the approach shares similarities with Satisfiability Modulo Theories (SMT) [14] where a SAT solver is extended with background theories to address first-order logic problems. For the discussed synthesis of balancing architectures, the problem is not fully described by first-order logic and, thus, the approach has to rely on a loose coupling of the proposed search engine and verification. Nevertheless, the proposed synthesis follows the *eager* SMT approach, see [14], where the SAT solver proposes a solution that is either accepted by the background theory or rejected with an additional constraint that is added to the set of constraints.

The verification methodology is outlined in Figure 5. 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

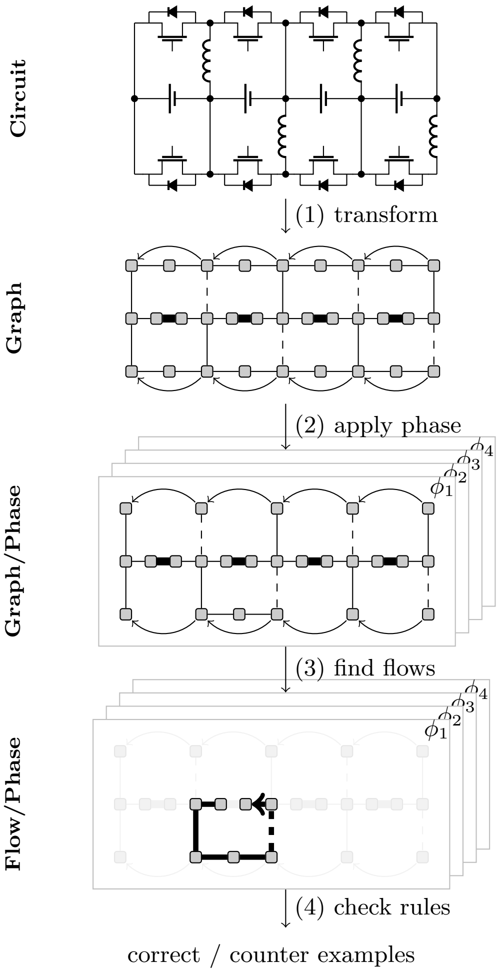

Figure 5: Illustration of the verification methodology from [9]. (1) The circuit is transformed to a graph, (2) open MOSFETs are removed for each phase, (3) current flows are determined using graph search algorithms, (4) and, finally, the current flows are checked to determine whether the charge transfer is feasible.

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. The search algorithm is assigning certain edge weights to the graph depending on the voltage or current source, respectively. Here, 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. As a remedy, in [9] an algorithm is presented that is capable of determining the flow between two vertices in the graph. The flow between a source vertex and destination vertex is defined by a set of paths such that each path has at least one edge that is not used by other paths. This ensures that each path in the flow contributes at least one edge, i.e., non-redundant information. By 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.

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 and the existential quantifier. These constraints define rules that check the determined current flows. Any violation of these rules is explicitly recorded and is either returned to the designer or to the synthesis method.

3.3 Component Selection and Sizing

With the architecture for the active balancing circuit determined using the approach discussed in the previous subsection, the next step is to choose appropriate hardware components for the implementation. Due to the modular characteristics of the balancing circuits, they are generally built on a Printed Circuit Board (PCB) per cell in a discrete fashion. Manufacturing an integrated circuit might be possible for the transistor switches in the architecture, but the inductor cannot be efficiently integrated with existing technologies.

The goal of the component selection is to achieve an implementation that satisfies the requirements specification while, at the same time, considering the multi-objective optimization problem introduced by the design metrics of energy efficiency, installation volume and balancing current for inductor-based active cell balancing circuits presented in [10]. For this purpose, a set of candidate components, e.g., inductors, needs to be identified and properties such as their inductance, maximum current, DC resistance and volume determined from data sheets. The same applies to the relevant properties of MOSFETs used for the switching network, e.g., their output capacitance, maximum current, on-resistance, volume and delay timings. Once sets of candidate components are identified such that they do not violate the specified requirements for, e.g., balancing current, they can be algorithmically evaluated regarding the design metrics and Pareto-optimal component choices can be determined. For this purpose, all feasible configurations are analyzed regarding their charge transfer efficiency based on a detailed modeling and then further ranked by installation volume and maximum balancing current such that all dominated configurations are excluded. The final configuration choice depends on the weighting used for the trade-off between maximum current, energy efficiency, installation space and cost.

Besides using off-the-shelf inductors, it has been shown that designing a custom inductor for a specific balancing application can be beneficial as it increases charge transfer efficiency by at least 20% compared to the optimal choices made by the selection-based method described above [11]. This increase in efficiency is achieved by using a sophisticated behavioral modeling of the charge transfer process in a geometric programming approach to obtain inductor design parameters minimizing the transfer losses. The trade-off to consider in this context is the initial setup cost for the manufacturing of a custom inductor, which might only be economical for large production numbers.

3.4 Integration and Control Design

Once hardware components have been chosen such that an actual implementation of the active balancing circuit can be made, the integration with the BMS has to be considered. Here, the control signals obtained during architecture synthesis need to be mapped to actual hardware signals, considering the determined optimal inductor switching frequency for maximum balancing current and the transistor delay timings. A common approach for this mapping is to introduce a safety margin such that a single setup for the PWM frequency and timing covers all operation scenarios.

A more sophisticated approach, however, would be to consider that optimal PWM timings to control the charging and discharging of the inductors depend on the voltage relation between the source and destination cells of the charge transfer and hence could be adaptively chosen [15]. This would provide a further optimization of the balancing process, resulting in higher average balancing currents and potentially higher balancing efficiency with shorter freewheeling phases. On the other hand, increased computational effort on the BMS side and higher implementation complexity would be encountered.

3.5 System-Level Performance Analysis

The design flow described in the previous subsections uses symbolic modeling and formal verification methods to ensure the efficiency and correctness of the active balancing architecture. Nevertheless, certain aspects require system-level analysis in order to evaluate the functionality on pack level. The efficiency of a single charge transfer is, for instance, optimized using the above approaches to minimize the number of components and their resistive losses in the current path. However, active balancing is performed with a global goal such that all charge transfers eventually lead to an equalized battery pack. Therefore, beyond an optimal architecture enabling individual highly efficient transfers, the balancing strategy chosen to achieve the equalization is critical. Balancing strategies decide which charge transfers to perform in which sequence, such that losses during balancing are minimized. For this purpose, certain rules have been identified, i.e., the monotonicity of the transfer directions. If charge is transferred from a cell into the direction of a neighbor, the cell giving charge should generally not receive charge from this direction at a later stage as this would render the initial transfer useless.

In order to perform a system-level analysis, considering all cyber-physical aspects of active balancing in battery packs from cell behavior to the actual control algorithms, a detailed modeling of all layers in a co-simulation framework has been proposed in [16].

4. CONCLUSIONS

This paper presents a design flow for active cell balancing architectures which are an important part of efficient battery management. Starting from a requirements specification, we involve a high-level synthesis approach which generates a correct-by-construction architecture from circuit and signal templates using a formal verification back-end. A hardware implementation platform is obtained by using automated tools for optimal component selection. After a final hardware/software integration, a system-level performance analysis using a cyber-physical co-simulation can be performed for the resulting parameterized implementation based on the correct-by-construction architecture. Consequently, design automation and formal approaches significantly contribute to making active cell balancing a viable candidate for implementation in large Li-Ion battery packs.

5. REFERENCES

[1] N.H. Kutkut. A modular nondissipative current diverter for EV battery charge equalization. In *Proc. of the Thirteenth Annual Applied Power Electronics Conference*, 1998.

[2] M. Lukasiewycz, M. Kauer, and S. Steinhorst. Synthesis of active cell balancing architectures for battery packs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 2016.

[3] L. Lu, X. Han, J. Li, J. Hua, and M. Ouyang. A review on the key issues for lithium-ion battery management in electric vehicles. *Journal of Power Sources*, 226(0):272–288, 2013.

[4] 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 the Conference on Design, Automation and Test in Europe (DATE)*, 2012.

[5] F. Baronti, G. Fantechi, R. Roncella, and R. Saletti. Intelligent cell gauge for a hierarchical battery management system. In *IEEE Transportation Electrification Conference and Expo (ITEC)*, 2012.

[6] A. Otto, S. Rzepka, T. Mager, B. Michel, C. Lanciotti, T. Günther, and O. Kanoun. Battery management network for fully electrical vehicles featuring smart systems at cell and pack level. In Gereon Meyer, editor, *Advanced Microsystems for Automotive Applications 2012*. Springer Berlin Heidelberg, 2012.

[7] S. Steinhorst, M. Lukasiewycz, S. Narayanaswamy, M. Kauer, and S. Chakraborty. Smart cells for embedded battery management. In *Proc. of the 2nd International Conference on Cyber-Physical Systems, Networks, and Applications (CPSNA)*, 2014.

[8] S. Steinhorst, Z. Shao, S. Chakraborty, M. Kauer, S. Li, M. Lukasiewycz, S. Narayanaswamy, M. U. Rafique, and Q. Wang. Distributed reconfigurable battery system management architectures. In *Proc. of the 21st Asia and South Pacific Design Automation Conference (ASP-DAC 2016)*, 2016.

[9] M. Lukasiewycz, S. Steinhorst, and S. Narayanaswamy. Verification of balancing architectures for modular batteries. In *Proc. of the 12th International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, 2014.

[10] S. Narayanaswamy, S. Steinhorst, M. Lukasiewycz, M. Kauer, and S. Chakraborty. Optimal dimensioning of active cell balancing architectures. In *Proc. of the Conference on Design, Automation and Test in Europe (DATE)*, 2014.

[11] M. Kauer, S. Narayanaswamy, M. Lukasiewycz, S. Steinhorst, and S. Chakraborty. Inductor optimization for active cell balancing using geometric programming. In *Proc. of the Conference on Design, Automation and Test in Europe (DATE)*, 2015.

[12] N. Eén and N. Sörensson. Translating pseudo-boolean constraints into SAT. *Journal on Satisfiability, Boolean Modeling and Computation*, 2:1–26, 2006.

[13] H. Samulowitz and R. Memisevic. Learning to solve QBF. In *Proc. of the 22nd National Conference on Artificial Intelligence*, volume 7, 2007.

[14] L. De Moura and N. Bjørner. Satisfiability modulo theories: An appetizer. In *Formal Methods: Foundations and Applications*, pages 23–36. Springer, 2009.

[15] M. Caspar and S. Hohmann. Optimal cell balancing with model-based cascade control by duty cycle adaption. *IFAC Proc. Volumes*, 47(3):10311–10318, 2014.

[16] S. Steinhorst, M. Kauer, A. Meeuw, S. Narayanaswamy, M. Lukasiewycz, and S. Chakraborty. Cyber-physical co-simulation framework for smart cells in scalable battery packs. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 21(4):62:1–62:26, 5 2016.