Model-based design of asynchronous controllers for flexible on-chip power buffers

Delong Shang, Fei Xia, Xuefu Zhang, Alex Yakovlev
University of Manchester and Newcastle University

***Abstract* – Conventional Energy Harvesting (EH) methods tend to concentrate on improving the quality of the output voltage – i.e. the Vdd supplied to computation loads, which incur considerable costs. This does not consider load heterogeneity. To match energy heterogeneity with load heterogeneity, the on-chip Capacitor Bank Block (CBB) technology was proposed to deliver variable Vdd from EH directly to Vdd variation tolerant loads. To realize its full potential in optimal energy-load matching, this asynchronous and reactive PDU needs a control subsystem. This work is on the model-based design of asynchronous event-driven controllers for such PDUs.**

I. Introduction

In Energy Harvesting (EH) systems, the energy availability and the capabilities of a harvester can be highly variable in time, resulting in energy heterogeneity [1] [2].

Modern systems such as systems on chip (SoC) [3] may include different components on the same die and must deal with heterogeneity in the computation loads and in their energy use and availability.

Conventional EH system designs concentrate on making the energy supplied to the load higher quality, i.e. providing a constant and stable Vdd to the load [1]. The cost of doing this includes longer waiting time for energy accumulation and higher power overheads in the power conditioning [2].

Delivering harvested energy directly to the load, bypassing any storage in the middle, has also been proposed [1]. But if a stable Vdd is required, it still needs power regulators, such as Low-Dropout Regulators (LDOs) and Switched Capacitor Converters (SCCs) that may lower the power delivery efficiency and limit the voltage scaling capability [4].

The requirement for stable Vdds derives from synchronous computation loads running under global clock signals, which have low Vdd variance tolerance. Asynchronous loads, on the other hand, can tolerate Vdd variances [6].

A power delivery method employing on-chip capacitors capable of delivering harvested power directly to asynchronous loads can be found in [5]. These on-chip capacitors follow a more general asynchronous reactive structure called Capacitor Bank Block (CBB), of which conventional clock-driven SCCs are a special case.

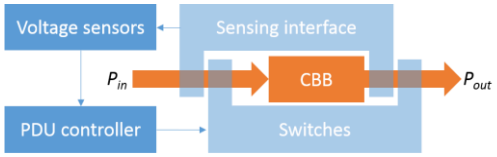

Figure 1. CBB PDU structure.

Model-based design, especially where a graphical language is used, facilitates a holistic design flow including synthesis and verification. In this work, we develop a graphical model-based design method for asynchronous CBB PDU controllers for optimal energy to computation matching without requiring stable Vdds.

A CBB with controller structure is shown in Figure 1. The PDU controller takes, from sensors, input and capacitor voltages, and controls the charging and discharging of capacitor banks (CBs) within the CBB based on these inputs.

II. Controller Design

We run characterization experiments with a CBB PDU connecting a highly heterogeneous EH source and a number of heterogeneous asynchronous computation loads and explore energy to workload efficiency for various charging and discharging combinations in different scenarios. It is found that the optimal energy to workload efficiency may be obtained from connecting capacitors in the CBB of different values to charging and discharging according to operating conditions [9].

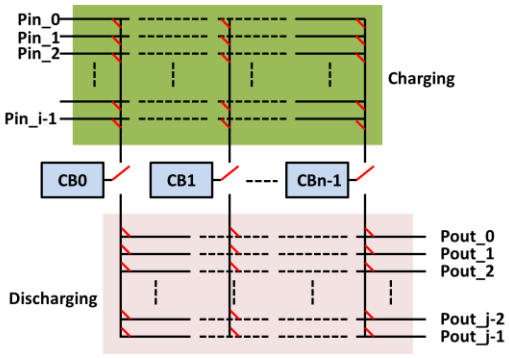

Figure 2. Charging and discharging networks.

Figure 2 is a refinement of the CBB part of Figure 1. The fully connected charging and discharging network of switches lead to potentially arbitrary connections between each CB and any power source for charging and between each CB and any load for discharging, without any need for synchronization.

For such a CBB PDU, the basic PDU controller actions are 1) charging a particular capacitor block (CB) up to a certain voltage from the P_{in} path and 2) discharging a particular CB down to a certain voltage into the P_{out} path.

The CBB PDU architecture supports maximal charging and discharging flexibility. The number of CBs in the CBB and the number of available power wires in the P_{in} path limit the maximal number of simultaneously charging CBs:

$$\max_{CBB} N_{ch} = \min_{CBB}\{N_{CB}, N_{P_{in}}\}, \quad (1)$$

where N_{ch} is the number of CBs in simultaneous charging, N_{CB} is the number of CBs in the CBB, and $N_{P_{in}}$ is the number of

power wires in the P_{in} path.

Similarly, for discharging:

$$\max_{CBB} N_{dsch} = \min_{CBB}\{N_{CB}, N_{P_{out}}\} \quad (2)$$

where N_{dsch} is the number of simultaneously discharging CBs and $N_{P_{out}}$ is the number of wires in the P_{out} path.

The total number of CBs being simultaneously charged and/or discharged together is then governed by the following:

$$\max_{CBB}\{N_{ch} + N_{dsch}\} = \min_{CBB}\{N_{CB}, (N_{P_{in}} + N_{P_{out}})\} \quad (3)$$

With enough power lines in the P_{in} and P_{out} paths, the CBB PDU supports having up to all of its CBs in simultaneous charging or discharging.

It is possible to use a multiple-CB PDU to connect a single EH unit to a single load to enhance energy stability, with $N_{P_{in}} = N_{P_{out}} = \max_{CBB} N_{ch} = \max_{CBB} N_{dsch} = 1$. This can have full random access, similar to RAM memory, or with a first-in first-out policy, similar to FIFO memory buffers.

In addition to connectivity flexibility, the asynchronous CBB also has temporal flexibility, allowing the charging and discharging to be event-driven, which is the most efficient mode of operation. As such, the controller should also be implemented using asynchronous logic.

Signal Transition Graphs (STGs) is a language for the design, synthesis and verification of asynchronous systems [7]. The STG language is supported by an extensive set of tools including Petrify and Workcraft [8]. We develop an STG-based design method for CBB PDU controllers. It is illustrated here with a FIFO controller design.

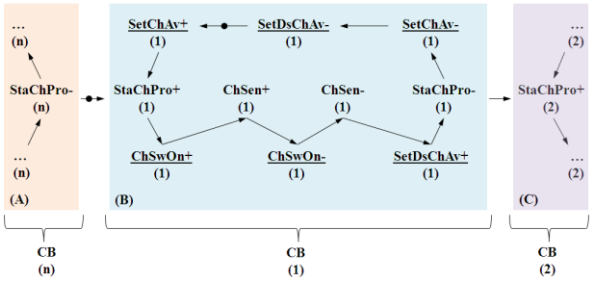

Figure 3. FIFO PDU charge control, (underlined signals are inputs).

The charging control is described by the STG in Figure 3. For a FIFO arrangement, charging is fully sequential and cycles through the CBs one by one. For example, only when the charging of the present CB(1) is completed (StaChPro-(1)), is the next CB(2) available for charging (StaChPro+(2)).

The FIFO discharge control is presented in Figure 4 in STG format. The transitions shown in part (E) guarantee continuous output power, preventing any non-powered gaps so long as the CBB is not entirely depleted. For instance, the discharge switch of CB(1) is disabled (DsChSwOn-(1)) only when the next bank CB(2) has started discharging (StaDsChPro+(2)). Only after that CB(1) will be designated as available for charging (SetChAv+(1)) after SetDsChAv-(1).

Non-FIFO charging/discharging protocols can also be implemented with asynchronous controllers designed using the STG method. Partial concurrency between the discharging of two CBs already exists in Figure 4, and extending concurrency to both charging and discharging to various degrees can be incorporated into STG control designs, as STGs represent concurrency and synchronization in a straightforward manner. A number of other PDU controllers with different charging and discharging policies, including various degrees of concurrency, are also synthesized from STG designs [9].

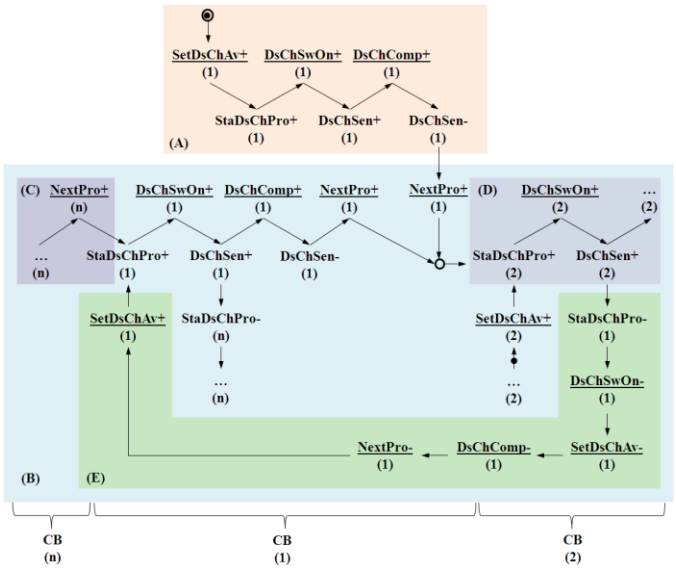

Figure 4. FIFO PDU discharge control. (A) starting from CB(1) after initialization; (B) steady-state; (C) CB(n); (D) CB(2); (E) power continuity provision.

The entire FIFO example CBB PDU including sensors and controller synthsized from the STG models presented in this paper is implemented in Cadence in UMC 95nm CMOS technology. The correctness and efficiency of this design is then additionally validated through simulations.

III. Summary

We present a method of designing asynchronous controllers for the event-driven charging and discharging of CBB PDUs. The use of the STG language facilitates both the design and verification processes.

This work is supported by the EPSRC through projects SAVVIE (EP/K012908/1) and A4A (EP/L025507/1).

References

[1] Christmann, J.-F. et al. "Bringing Robustness and Power Efficiency to Autonomous Energy-Harvesting Microsystems," Design & Test of Computers, IEEE, vol.28, no.5, pp.84-94, Sept.-Oct. 2011.

[2] Weddell, A.S. et al. "Accurate Supercapacitor Modeling for Energy Harvesting Wireless Sensor Nodes," IEEE TCS2, 58(12), Dec. 2011.

[3] Ganguly, A. et al. "Scalable Hybrid Wireless Network-on-Chip Architectures for Multicore Systems," IEEE TC, 60(10), Oct. 2011.

[4] Wibben, J.; Harjani, R., "A High Efficiency DC-DC Converter Using 2nH On-Chip Inductors," VLSI Circuits, 2007 IEEE Symposium on, pp.22,23, 14-16 June 2007.

[5] Zhang, X. et al. "A Novel Power Delivery Method for Asynchronous Loads in Energy Harvesting Systems," ACM JETC, 7(4), December 2011.

[6] Baz, A. et al. "Self-Timed SRAM for Energy Harvesting Systems", Journal of Low Power Electronics, 7(2), 2011.

[7] Cortadella, J. et al. Synthesis of asynchronous controllers and interfaces, Springer, 2002.

[8] Workcraft: https://workcraft.org/.
Petrify: http://www.lsi.upc.edu/~jordicf/petrify/.

[9] Zhang X. Power Delivery Mechanisms for Asynchronous Loads in Energy Harvesting Systems, PhD Thesis, Newcastle University, 2017.