Yakovlev A.
[Energy-modulated computing](https://doi.org/10.1109/DATE.2011.5763216).
In: 2011 Design, Automation & Test in Europe (DATE). 2011, Grenoble, France: IEEE.

Copyright:

DOI link to Conference Proceedings:

https://doi.org/10.1109/DATE.2011.5763216

Date deposited:

07/11/2017

Energy-Modulated Computing

Alex Yakovlev

School of EECE

Newcastle University

Newcastle upon Tyne, United Kingdom

alex.yakovlev@ncl.ac.uk

Abstract—*For years people have been designing electronic and computing systems focusing on improving performance but "keeping power and energy consumption in mind". This is a way to design energy-aware or power-efficient systems, where energy is considered as a resource whose utilization must be optimized in the realm of performance constraints. Increasingly, energy and power turn from optimization criteria into constraints, sometimes as critical as, for example, reliability and timing. Furthermore, quanta of energy or specific levels of power can shape the system's action. In other words, the system's behavior, i.e. the way how computation and communication is carried out, can be determined or modulated by the flow of energy into the system. This view becomes dominant when energy is harvested from the environment. In this paper, we attempt to pave the way to a systematic approach to designing computing systems that are energy-modulated. To this end, several design examples are considered where power comes from energy harvesting sources with limited power density and unstable levels of power. Our design examples include voltage sensors based on self-timed logic and speed-independent SRAM operating in the dynamic range of Vdd 0.2-1V. Overall, this work advocates the vision of designing systems in which a certain quality of service is delivered in return for a certain amount of energy.*

Keywords-charge-to-digital converter; energy; energy-frugality; energy-harvesting; power; power-proportionality; self-timed logic; SRAM; voltage sensor

I. INTRODUCTION

Semiconductor technology permits formidable concentration of electronic devices and electrical power on small areas of a silicon die. At the same time, engineering processes, involving software and hardware, can no longer sustain this growth; they require the design and test processes to be much more resource-conscious. In system design we are faced with software and hardware resources such as communication (device drivers, buses), storage (data buffers, SRAM blocks), processing (microcontroller, DSP) – they are in one or another way derivatives of energy, the mother of all resources. Energy has been obedient in serving the needs of system builders for years while the size of silicon ruled the game. Now energy is actively beginning to show its limits.

This is realized by many. Green, energy-frugal, power-proportional are qualities of computer systems that people begin to consider now. Numerous examples bring up the issue of resource and energy-awareness into computing and electronics. From the energy supply perspective, battery life, energy harvesting sources, power control and regulation are

the necessary attributes of modern systems engineering. From the energy consumption viewpoint, the high end of the spectrum is occupied by mammoth data plants (e.g. Google plant in Oregon was estimated to require 103MW of power, enough to supply every home in Newcastle). In the middle, there are many-core chips, such as Intel's 48-core SCC, consuming between 25W-125W. The low end of the spectrum is systems that interface to biological organisms, where power constraints are at the level of microwatts. Over the years system design methodologies developed completely relying on feature scaling and availability of as many resources as needed in order to satisfy their performance appetites. However, architecting systems solely on the principles of hierarchy and object-orientation, without proper account of underlying resources often leads to inefficiency, likewise does the full decentralization of control and distribution of resources on the principles of local optima.

One of the important questions we may want to ask ourselves, is there a fundamental case for evolution of electronic system design that is completely dictated by the energy aspect? While thinking about energy-driven computing, we should consider various issues, involving energy characterization of components and devices of different functionality and nature, interplay between energy, performance and dependability, power constraints and quality of service, optimization methods for resource-driven computing, modeling and meta-modeling techniques to underpin design automation tools. All this is a grand challenge, and parts of this puzzle are being solved by many individual researchers and research teams in some or another, often fragmented, form.

In this paper we try to look at this challenge by being more like a mountaineer who does not start climbing right away but first measures the terrain and thinks about an appropriate gear to prepare for the big assault. The role of this gear is played by a set of design examples of varying complexity and functionality. What unifies them is that they all come from a research project whose main aim is to develop a holistic methodology for designing electronic systems for power sources alternative to batteries, i.e. energy-harvesters (EHs) [1]. The holistic approach to energy generation and utilization within one system, calls for a globally optimized design and brings us to the notion of power-adaptive system design, with which we start our exploration. We then observe important links between power proportionality, energy-modulation, power adaptation and role of self-timing in designing power-adaptive systems.

II. POWER-ADAPTIVE SYSTEMS

A. Power-proportional and power-efficient systems

Traditionally, systems are optimized for peak performance as high GHz brings revenue. However, real-life duty cycles, in new applications such as wireless sensor networks, require computing, communication and storage components to operate only at a fraction of maximum load. In idle periods they should not consume much energy. This view brings us to the notion of *energy-proportional computing* [2]. It can be depicted in Fig. 1, where energy-proportionality is characterized as a property in which some useful activity can even be generated at small amounts of energy.
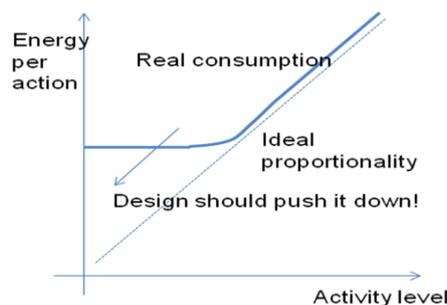
Figure 1.   The idea of energy-proportional computing.

This view is very inspiring, but forms only part of the overall picture in motivating our energy-modulated systems approach, and can be further refined to relate power supply levels (e.g. Vdd levels) with quality of service (QoS) levels, as shown in Fig. 2.   It is often difficult to achieve both power (energy)-proportionality and power-efficiency in the same design, if one follows only one specific discipline in design. We can therefore consider a range of designs, if not system types, some of which are more power-proportional but less power-efficient or vice versa. For instance, Design 1 in Fig. 2 is more power-proportional, in the sense that it starts to deliver the sought QoS at a very low Vdd, where Design 2 cannot deliver at all. However, if the nominal level of power supply is at high Vdd, Design 1 is less power-efficient than Design 2, because it cannot increase QoS at the same rate against power investment.

An example of such a relationship can be found in the area of asynchronous systems [3]. Here, Design 1 is a speed-independent implementation of a circuit which is built of dual-rail components, which uses the property of completion detection. This design is more conservative to delay variations due to low or unstable Vdd, but consumes more power due to its additional logic components (both dynamic and static power).   Design 2 is based on data-bundling delays, thus less timing robust but has much less overhead for a nominal Vdd. If one wants to design a system that is both power-proportional and power-efficient in a broad range of power supply levels, the recommended way would be to produce a hybrid design which combines the strengths of both designs, say, using Design 1 in the depleted power (idle) mode and Design 2 in a full power mode. Thus, an important message from this simple analysis is that truly energy-modulated design has to be power-adaptive. Thus, power adaptation requires good knowledge of the actual power level at run-time, which itself calls for good power meters (cf. voltage sensors in Section III).
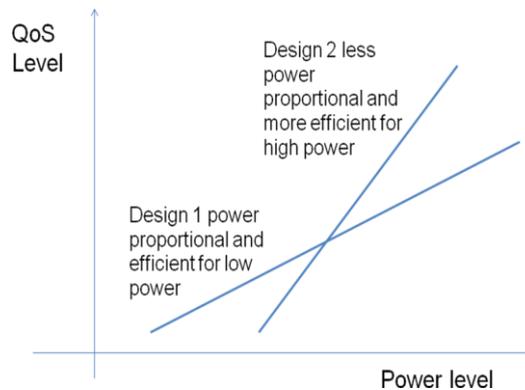

Figure 2.   Power-proportional versus power efficient system design

B.   Designing electronics for energy-harvester power supply

Suppose our main supply of energy comes not from a traditional battery but from a micro-generator which harvests energy from the environment, e.g. mechanical vibration. Or, we can even imagine a more radical scenario, in which we build a system which performs sensing a certain physical magnitude and at the same time scavenges power from the same source that it measures. Let us briefly compare the difference in the way how we may want to design such a system in comparison with the traditional battery-operated design.

Battery can supply finite energy, which depends on the battery capacity, but while it is still operational the available power can be very large. Supply characteristics are stable and known in advance. Consumption depends on the computational load and may vary. System design usually relies on that stability of Vdd level and aims at scheduling the

computational load optimally to meet performance obligations and maximize energy savings. For the latter, techniques such as clock and power gating, dynamic voltage and frequency scaling are used.

Energy-harvester can in principle supply infinite energy, but the power levels maybe small and variable, thereby requiring significant effort (again costing energy!) to maintain the stable Vdd level for the computational load, but putting significant limits on the load's current. It would therefore be reasonable to design the system supplied by an EH under the assumptions where power is not a cost function but a constraint. Namely, specifications determine the allowed power range but the actual power may at any time be unstable within this range. Whilst it is even possible to assume unpredictable supply, it is often reasonable to design the system for a particular regime of power supply and schedule the computations in the load accordingly, to *modulate* them to the supply.

In designing power supply for EH-based systems, people often use the so-called maximum power-point tracking. This is based on a special controller whose aim is to extract maximum power from the micro-generator (e.g., in the case of vibration, by tuning it to the resonant frequency of the energy source). At the same time, on the consumption side of the chain, the design of the computational load can follow one of the two possible strategies for maximizing the amount of computational activity for a given quantum of scavenged energy. These strategies determine the run-time regime of operation and depend on the capabilities of the circuit to cope with the range of Vdd. One strategy is to switch on/off parts of the circuit under the constant (nominal) voltage (cf. Design 2 from the previous section). The other strategy is to operate under the variable voltage, but this requires much more robust circuits, such as classes of self-timed (asynchronous) logic [3].

Examples of using asynchronous logic in EH-based systems and under AC-power supplies exist in the literature. For example, the one in [4] has a fast power-on-reset (4.1nW), 3T DRAM to keep state supply across supply cycles, 135K transistors in 180nm CMOS for an FIR filter. It operates on the following principle. For every power supply cycle: wake up the circuit, perform computation, and shut down the circuit. To time the logic it uses a critical path replica in the heart of the ring oscillator. While the overall direction of this approach is very promising the design of the timing based on the simple replica of critical path may not scale well with the computational load (see below the problems with SRAM delay matching).

The second, more flexible, strategy would require a tracker of the activity as well as dynamic scheduler whose task is to maximize the amount of computational activity for a given quantum of scavenged energy. This brings us to a holistic view upon the system with an EH shown in Fig. 4.
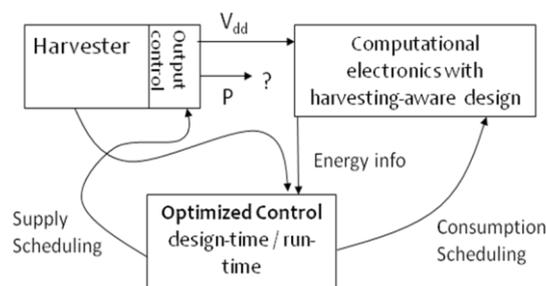

Figure 3.   Power-adaptive computing (holistic view)

Within this holistic approach, useful energy consumption is maximized for a given amount of energy produced. Alternatively, we can minimize the supplied energy for an amount of energy required to carry out the computation.

The adaptation as per Fig.3 can be done at different levels. At the low level, there are power-adaptive cells and components. Here, we can design components robust to Vdd variations, e.g., robust synchronizers [5] and completion detection blocks for self-timed components. It is also possible to use leakage control mechanisms such as body biasing. At the circuit level, adaptation can be achieved via clock/power gating and DVF scaling. Here, again, self-timed logic can prove more energy-efficient [17]. At the system level, it can be done through power sensing, and control of power supply and consumption chains. For instance, a method of optimal control of Vdd for minimum energy per operation has been proposed in [6]. Asynchronous circuit design principles can significantly improve effectiveness and efficiency of both sensing and control in adaptation process.

To illustrate how resilient asynchronous logic is to Vdd fluctuations, consider Fig. 4. It shows the waveforms of the behavior of a 2-bit sequential dual-rail asynchronous counter simulated in Cadence under the AC-supply 200mV+/-100mV and frequency 1 MHz. The self-timed logic of this counter with completion detection is robust to power supply variations.
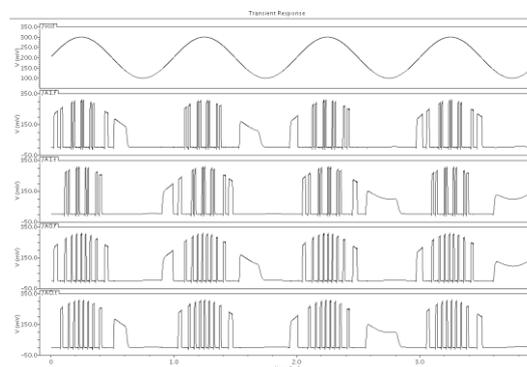

Figure 4.   Operation of 2-bit dual-rail counter

III.   CIRCUITS FOR POWER-ADAPTIVE COMPUTING

Possible directions towards realizing power-adaptive computing can be highlighted by real circuit design examples. The first example is a speed-independent SRAM with completion detection. This example will illustrate a way of designing memory for EH-based systems, which can also be used in ultra-low power systems, when the Vdd levels are drastically lowered down (say, to suit an idle mode).

The second example is a charge-to-code converter based on a self-timed counter which also acts as an oscillator whose frequency and amplitude are modulated by the Vdd level. While this converter can be used as a core of an ultra-energy efficient ADC or voltage sensor, it also shows an example of a circuit which turns an amount of energy into the amount of computation. It can thus act as conceptual prototype for building computational engines that are directly modulated by energy supplies. Such devices can be constructed to operate in the environments where energy is scavenged very sporadically.

A.   Speed-independent SRAM

SRAM is a fundamental component in designing any computational load for an EH-based system. It also characterizes the problems associated with graceful scaling of power against timing. That's why SRAM is one of our starting points on the way towards building power-adaptive computers.

SRAM is constructed from SRAM cells, address-decoders, pre-charger, write-driver, read-buffer, and controller to synchronize all these components to each other in order to complete the required job. Power and timing are closely

related, from the level of individual cells to that of the SRAM controller. In our experimental design we decided to use the standard simple 6T SRAM cell, although many different structures of SRAM cells are available and they offer various features including greater robustness for process variations. Normally, memory works based on stable Vdd levels and well-kept timing assumptions. The key issue concerning the use of SRAM in the environment with unstable power supply such as EH-based systems is how to design its timing logic. It is therefore necessary to know how timing assumptions are affected under different Vdd levels. Here, we first of all investigated the difference between the latency on the bit line drive and its corresponding typical inverter-chain delay elements used in controllers under different Vdds. The mismatch is shown in Fig. 5.
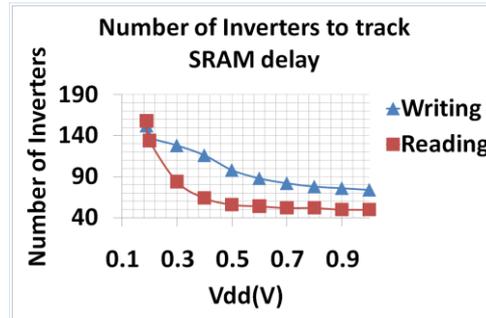

Figure 5. Mismatch between the scaling of SRAM and logic

For example, at 1V Vdd the delay of SRAM reading is equal to 50 inverters whereas at 190mV the delay becomes equal to 158 inverters. This problem has been well known so far. The solutions proposed either use different delay lines in different range of Vdd, or duplicate a column of SRAM to be a delay line to bundle the whole SRAM. These solutions require voltage references and DC-DC converter. We focused on the incorporation of completion detection which avoids using absolute references and relies on the causality inherent in self-timed speed-independent logic.   The details of the design of the controller for speed-independent SRAM can be found in [7]. This controller has a sub-circuit that indicates the completion of the transients in the SRAM during reading or writing. An interesting and original aspect of the design is that the well-known problem of completion detection during writing is solved by performing reading *before* writing. This allows the completion logic to simply wait until the equality between the bit-line state and the new value is established. Building on the genuine completion indication, the control uses handshake protocols to manage precharge, word line and write enable commands (see Fig. 6).
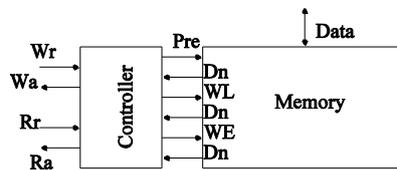

Figure 6. Handshake-based control of self-timed SRAM

The SI SRAM can work smoothly under variable Vdd, as shown in Fig. 7. For example, the first writing works under low Vdd, it takes long time, while the second write, at high Vdd, works much faster.
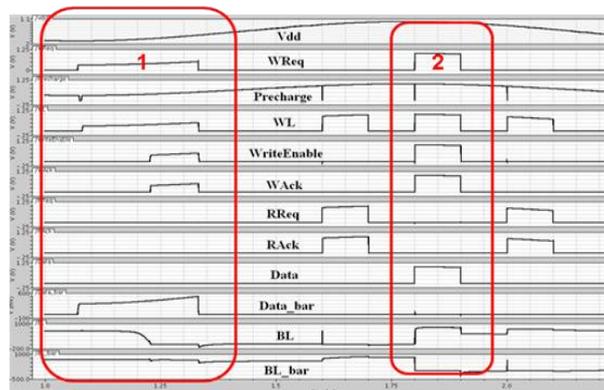
Figure 7.   Operatioon of SRAM under varying Vdd

A 1-kbit (64x16) SI SRAM, presented in [7], was designed using Cadence and UMC 90nm CMOS. It showed minimum energy point per read or write at 0.4V. It consumes 5.8pJ at 1V for a write of a 16-bit word and 1.9pJ at 0.4V. Further analysis of the SRAM (failure analysis and corner performance analysis) and design of its version which has an intelligent bundling replica (only one column has full completion detection) can be found in [8]. A 90nm CMOS chip with the SI SRAM, aimed at demonstrating its physical operation for variable power supply, is now being made ready for a Europractice run in January 2011. Interesting possibilities exist for improving the SRAM along the line of the QoS vs Power relationship (cf. Fig. 2). For example, its low Vdd limit can be pushed further down in sub-threshold (below 0.3V) by sectioning the completion detection in the column into smaller segments, say, of 8 bit each. This would reduce the loading capacity of the bit lines which are currently affecting the shape and speed of the column completion detection. On the other hand leakage power can be reduced by switching to 8T cells (with two NMOS transistors in stack).

B. Voltage sensing using self-timed logic

Our holistic approach to designing the system consisting of the power supply and the load requires timely and accurate metering of resources. An important resource is power supply, and we should find efficient ways of metering power on a chip, preferably avoiding using complex A-to-D converter schemes. A way to do it could be to follow the very principle of power-proportionality, i.e. build a circuit whose digital performance factor, say, its computation rate is proportional to the measured power level. In the published literature, there are techniques for doing that, for example by using a ring oscillator (inverter ring) whose Vdd line is connected to the power supply signal [6]. The frequency of the output of the ring oscillator is proportional to the Vdd (it is not exactly linear but it can be calibrated and stored in a look-up table for example).

Another possible way to build a power meter by voltage sensing in an EH-based environment is to follow good energy-proportionality of self-timed logic. A piece of such logic can be arranged to convert a quantum of energy (say, represented by electric charge) into an amount of computation activity, which can be integrated in a digital code. This takes us to a simple sample and hold approach, illustrated by Fig. 8. Here, the voltage sensor samples Vin from the DC-DC converter into its sampling capacitor, converts it into a code sent to the controller that controls the output from DC-DC converter.
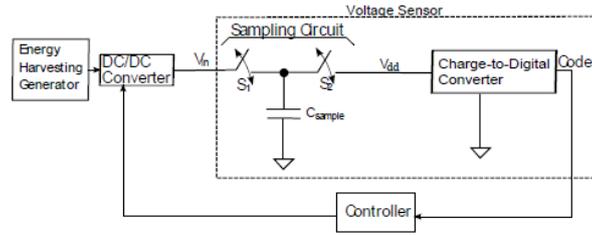
Figure 8.   Voltage sensor in an EH-based power chain

The charge-to-digital converter can be implemented by a self-timed counter, which is connected in a pulse generator (oscillator) mode as shown in Fig. 9. The counter is built of a chain of toggle flip-flops (we can for example use the toggle from [3], shown in Fig. 10).   The main principle on which this converter works is very simple. After we close switch S2 the sampling capacitor starts to supply power (effectively, charge) to the counter. The least significant bit of the counter acts as an oscillator, whose pulsing signal on R0 triggers the toggle T0. The output of the latter is connected in handshake with the input of the next bit's toggle T1, and so on. The switching activity propagates to the more significant bits and the frequency of the pulses on signals Ri is progressively divided by 2. In this circuit each logic gate fires strictly in sequence, without any hazards, and therefore there is a strong proportionality between the amount of charge taken from the capacitor and the number of transitions and, hence, counts performed by the counter.
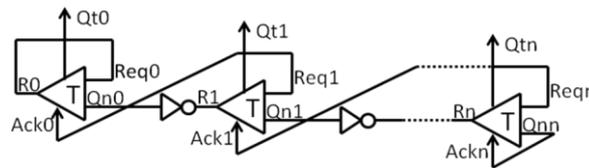

Figure 9.   Self-timed counter used as charge-to-code converter
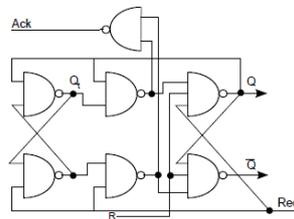

Figure 10. Toggle flip-flop from [3]

With this circuit, we therefore achieve the strong proportionality between the quantity of charge sampled in the capacitor (it is proportional to the voltage Vin if the sampling time is constant) and the binary code accumulated in the counter. This relationship is illustrated in Fig. 11. The full details of circuit operation and the thorough analysis of the the relationship between voltage, energy, transition count and the final code can be found in [9].
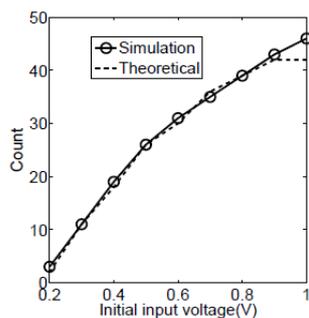
Figure 11. Values of count against the initial value of Vdd in Csample

C. Reference-free voltage sensing

All known ways of voltage sensing or A-to-D conversion require the availability of reference magnitudes, e.g. time, voltage, current or frequency. In designing systems that are powered by varying supply, having a stable reference is a problem (one needs to accumulate enough energy in a supercap or have an additional battery and DC-DC converters to provide voltage references, which usually need to be very accurate as well). One of the interesting by-products of the different scaling between memory and logic under varying Vdd (cf. Fig.5) is a way to build a voltage sensor which is free from time or voltage references. Such a sensor is described in [10]. Basically, the idea is reflected in Fig. 12.
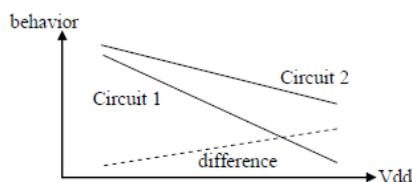

Figure 12. Idea for reference-free voltage measurement

All we need is to have two circuits racing against each other and recording the completion event of one circuit (say Circuit 1) in terms of a 'ruler' provided by the other circuit (Circuit 2). In our case, we used an SRAM cell as Circuit 1 and a chain of inverters as the ruler. In other words for a certain measured value of the voltage which drives both circuits, the completion event in the SRAM will mark a particular length in the inverter chain, which is then reflected in a thermometer code.

The sensor can work under a wide range of Vdd, from 200mV to 1V in 90nm CMOS technology, and provides accurate measurements of voltage over this operating range with an accuracy of 10mV. Unlike existing methods, the voltage information is directly generated as a digital code without any analog circuits.

IV. CONCLUSION

The concept of energy-proportional computing, recently advocated by several researchers from industry and academia (cf. [2]), can be taken further to consider ways of designing systems that operate in such a way that the supply of energy to them directly determines (modulates) their functioning. Energy harvesting systems offer a fruitful application paradigm to explore energy-modulated behavior. For that such systems must be designed in a holistic way rather than considering power electronics separately from the computational component. In this work we addressed the relationship between power-proportionality and power-efficiency, which leads us to designing power-adaptive systems. Such systems must have two-way control and adaptation between the power source and computational load: (i) perform task scheduling

according to the power profile, and (ii) optimize the supply to the load needs. More details about design principles for power-elastic systems and adaptation by means of task concurrency control and "soft arbitration" can be found in [11]. Besides, our new stochastic analysis methods help characterize the behavior of concurrent multi-core and multi-task systems in the context of energy-modulated computing [12].

We have emphasized in this paper the important role that self-timing can play in designing hardware components for power-adaptive systems. In addition to the circuits presented in this paper, self-timed SRAM and voltage sensors, we are currently working on new solutions for on-chip power delivery that exploit the property of self-timed circuits to be more robust to variable Vdd than synchronous [13]. These developments should help in exploring a wider range of design options for power electronics, taking it outside the conventional DC-DC converters, which is crucial for building energy-harvesting systems applicable for microwatt power densities. Within the Holistic project [1] they are complemented by further advancement of power-gating methods for clocked micro-processors.

While we continue work on the system design at the circuit and architectural level, further work will also require the development of mathematical modeling and optimization methods to underpin the energy-modulated computing paradigm. Here, initial steps have been made in creating Petri net based models with energy tokens [15] and game-theoretic power management for dependable systems [16].

REFERENCES

[1] "Next Generation Energy Harvesting Electronics: Holistic Approach", EPSRC-funded Research project, University of Southampton, University of Bristol, Imperial College and Newcastle University, http://www.holistic.ecs.soton.ac.uk/ .

[2] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing", Computer, vol. 40, December 2007, pp. 33-37.

[3] M. A. Kishinevsky, et. al., Self-Timed Control of Concurrent Processes: The Design of Aperiodic Logical Circuits in Computers and Discrete Systems, V. I. Varshavsky, Ed. Norwell, MA, USA: Kluwer Academic Publishers, 1990

[4] J. Wenck, R. Amirtharajah, J. Collier, and J. Siebert, "AC power supply circuits for energy harvesting," Proc. IEEE Symp. on VLSI Circuits, June 2007, pp. 92-93.

[5] J. Zhou, D.J. Kinniment, G. Russell and A. Yakovlev, "A robust synchronizer circuit," Proceedings of the IEEE CS Annual Symposium on VLSI (ISVLSI'06), Karlsruhe, Germany, March 2006, pp. 442-443

[6] Y. K. Ramadass and A. P. Chandrakasan, "Minimum energy tracking loop with embedded dc-to-dc converter enabling ultra-low-voltage operation down to 250 mV in 65 nm CMOS," IEEE JSSC, vol 43, no. 1, pp. 256–265, 2008.

[7] A. Baz, D. Shang, F. Xia, and A. Yakovlev, "Self-timed SRAM for energy harvesting systems," Proc. PATMOS'10, Grenoble, France, 7-10 September 2010, to appear in LNCS, Springer.

[8] A. Baz, D. Shang, F. Xia, R. Ramezani, R. Emery and A. Yakovlev, "Self-timed SRAM with smart latency bundling", NCL-EECE-MSD-TR-2010-161, Microelectronic System Design Group, School of EECE, Newcastle University, December 2010.

[9] R. Ramezani, A. Yakovlev, T. Mak and D. Shang, "Voltage sensing using an asynchronous charge-to-digital converter for energy harvesting circuits," NCL-EECE-MSD-TR-2010-160, Microelectronic System Design Group, School of EECE, Newcastle University, October 2010.

[10] D. Shang, R. Ramezani, F. Xia, and A. Yakovlev, "Wide-range, refererence-free, on-chip voltage sensor for variable Vdd operations," NCL-EECE-MSD-TR-2010-159, Microelectronic System Design Group, School of EECE, Newcastle University, October 2010.

[11] F. Xia, A. Mokhov, Y. Zhou, Y. Chen, I. Mitrani, D. Shang, D. Sokolov and A. Yakovlev, "Towards power elastic systems through concurrency manegement", NCL-EECE-MSD-TR-2010-159, Microelectronic System Design Group, School of EECE, Newcastle University, July 2010.

[12] Y. Chen, I. Mitrani, D. Shang, F. Xia, and A. Yakovlev, "Stochastic Analysis of power, latency and the degree of concurrency," Proc. IEEE International Symposium on Circuits and Systems (ISCAS'10), June 2010, Paris, France. pp.4129-4132.

[13] X. Zhang, D. Shang, F. Xia and A. Yakovlev, "A novel power delivery method for asynchronous loads in energy-harvesting systems," NCL-EECE-MSD-TR-2010-166, Microelectronic System Design Group, School of EECE, Newcastle University, December 2010.

[14] J. N. Mistry, B. M. Al-Hashimi, D. Flynn, and S. Hill, "Sub-clock power-gating technique for minimising leakage power during active mode," Design, Automation & Test in Europe 2011 (DATE 2011), Grenoble, France, March 2011.

[15] D. Sokolov and A. Yakovlev, "Task scheduling based on energy token model," Workshop on Micro Power Management for Macro Systems on Chip (uPM2SoC), DATE Conference, Grenoble, France (March 18, 2011).

[16] J. Docherty and A. Yakovlev, "Game theoretic power management techniques for real-time scheduling," Workshop on Micro Power Management for Macro Systems on Chip (uPM2SoC), DATE Conference, Grenoble, France (March 18, 2011).

[17] C. Ortega, J. Tse, and R. Manohar, "Static power reduction techniques for asynchronous circuits," Proc. of the IEEE Int. Symp. on Asynchronous Circuit and Systems (ASYNC'10), May 2010, pp. 52-61.