

LEAKAGE POWER REDUCTION IN MULTICORE CHIPS VIA ONLINE DECAP MODULATION

Leilei Wang* and Pingqiang Zhou

School of Information Science and Technology, ShanghaiTech University, Shanghai, China

*Corresponding Author's Email: wangll@shanghaitech.edu.cn

ABSTRACT

The decap leakage constitutes a significant part of total leakage in multicore processors. Leaking decap is widely used in chips to provide transient power to function blocks, and the amount of decap placed over multicore chip is determined by the worst case at design time. Considering that the decaps do not have to be kept 100% “on” when the chip is running workloads, in this paper we propose an ideal online approach to adapt the percentage of “on” capacitance in the decaps to the dynamic need of the core loads, and therefore to reduce the leakage power consumption in the chip. Our approach is based on the equivalent resistance model of the power sources (power supply pins and local decaps) and the core loads, and a novel fast algorithm is developed to determine the required amount of decap at runtime. Results on a set of multicore test cases show that our approach can achieve up to 48% saving in decap leakage.

1. INTRODUCTION

Driven by Moore's Law [1], and through joint advances of transistor, circuit, architecture and compiler, the computational power of chips is increasing by 2.8x per process generation. However, due to the gap between transistor density improvement and energy efficiency gain, only 1.4x of the benefit can be achieved, because large portion of the chip has to be turned off (also known as *dark silicon*) due to the limitation on total chip power consumption [2]. Therefore, power has become a primary challenge in the system design, even after the shift to multicore era around the year of 2004.

Leakage power constitutes an important part of total on-chip power. Although technologies such as Hi-k metal, SOI and Dual-Gate help to effectively reduce leakage power consumption in sub-45nm technologies, leakage power is still a big concern in sub-22nm technology [3, 4].

Power is delivered from the power source to each transistor on chip by a power delivery network. The parasitics in the power delivery network, together with temporal variations in the current drawn by a circuit, result in a time-varying voltage drop/surge at nodes in the power delivery network. These variations can adversely impact

the performance and the reliability of a circuit, thus on-chip decoupling capacitors (Decaps) are usually applied to increase the reliability of the power supply network, by providing transient power to the functions blocks in a chip [5].

Unfortunately, the deliberately-added decap can occupy more than 20% of the total chip area in high-end processors and its leakage can contribute to more than 20% of the total power consumption [6]. There has been very limited prior work on the reduction of decap leakage. In [7] power gating is used to turn off part of the power grid, together with the associated local decaps, so as to eliminate their leakage. Obviously, this approach has limited usage because it only applies to the case when the circuit blocks under the local power grid is inactive. The authors in [6] proposed active decap to reduce leakage, by introducing operational amplifier to boost the performance of conventional decaps, so that less decap resource is required for a given chip.

Our work is motivated by the power profile of each core in a multicore chip. Figure 1(a) shows a 4-core chip and the current trace of core1 is shown in Figure 1(b) when the chip is running a PARSEC 2.1 benchmark [8]. It is clear that the core is not always busy during the simulation period, implying that its demand on the amount of decap is dynamically changing, and there is big room for us to explore in order to save decap leakage.

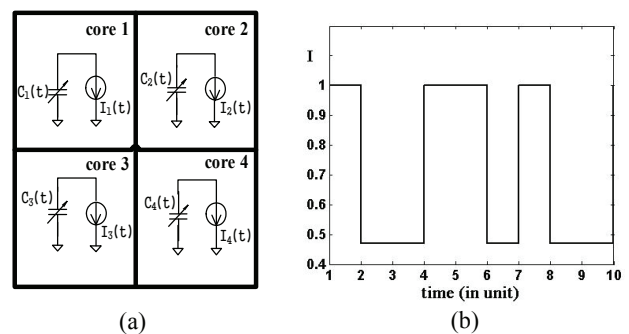


Figure 1: (a) A 4-core chip with one dynamically adjustable decap for each core. The stable supplies such as the voltage regulators are not shown here for simplification. (b) The normalized current trace of core 1. The x-axis shows the simulation time. One unit = one thousand cycles.

This work was supported by the NSFC grant 61401276.

In this work, we propose a novel approach to reduce decap leakage by dynamically adjusting the amount of “on” capacitance of each pre-placed decap in a multicore chip (as shown in Figure 1(a)) while meet the runtime need of the workloads on transient power supply. More specifically, we provide just enough “on” decap for each core in the chip, instead of keeping the whole pre-placed decap leaking at runtime. Therefore, our approach has better flexibility than the work in [7], and can also work in tandem with the adaptive decap idea in [6].

2. ON-LINE DECAP MODULATION

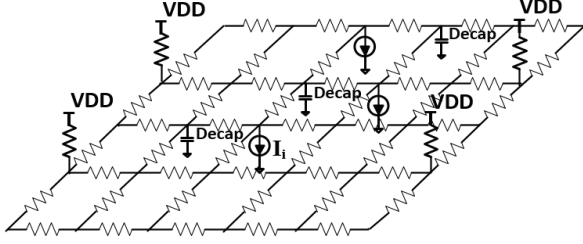


Figure 2: Model of the simplified on-chip power system in a multicore chip. Each core load is modeled as a current source.

Figure 2 shows the power delivery system in a multicore chip. Given

- 1) the power trace of each core in the chip, and
- 2) pre-placed decaps over the chip,

our approach finds the runtime demand on each decap in the following steps:

- Step0: The power required by each core can be provided by either the local decaps surrounding the core, or the power pins (the VDDs in Figure 2) powered by remote off-chip voltage regulators. In our work, we build an equivalent resistance model among the power pins, the pre-placed decaps and the core loads, based on Macromodeling approach [10] (see Section 2.1). This step is done offline, before the execution of workloads on the chip.
- Step1: Given the power traces of the cores, at the i -th transition point (i.e., when the power demand of any core changes) of the traces, we update the amount of required capacitance for each pre-placed decap. At this step, we use the equivalent resistance model built at Step0, and determine the ratio of power supplied to the core loads by the decaps. We developed a fast algorithm for this step (see Section 2.2).
- Step2: Once the amount of required capacitance is known for each decap at each transition point, we can adapt the idea of Gated decap [9] to set the “on” capacitance of each decap, by turning on or off part of the decap at runtime. Alternatively, the amount of decap may be controlled using the digital capacitance modulation approach [11].

- Step3: we repeat Step1 and Step2 for the $i+1$ transition point of core power traces and the whole process iterates.

2.1 Equivalent Resistance Model

To calculate the amount of capacitance needed by the core loads for each decap, we first use the Macromodeling approach [10] to build the equivalent resistance model between the power suppliers (VDD pins and decaps) and the power consumers (core loads) in the power delivery system, as shown in Figure 3. For the details of Macromodeling approach, the readers are referred to [10].

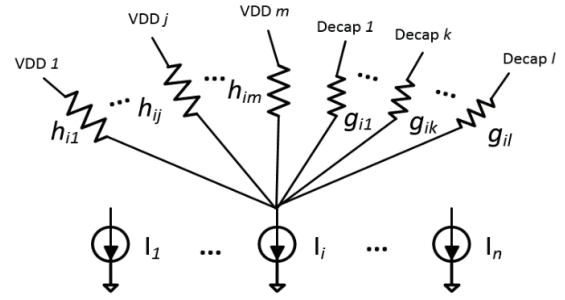


Figure 3: The equivalent admittances between the power suppliers and consumers. Each current load is supplied by the m VDDs and l decaps on the chip.

2.2 Capacitance Estimation at Each Transition Point

Let I_i be the average amount of current required by the core load i in a clock cycle T , then the total amount of charge required by load i in one cycle is

$$q_i = I_i \cdot T \quad (1)$$

This amount of charge q_i is provided by the m VDD pins and l decaps. Assuming that the amount of charge from each source is proportional to the equivalent admittance between the source and the core load [12], we can find the amount of charge from decap k to load i in one cycle is

$$q_{ki} = \frac{g_{ik}}{\sum_{j=1}^m h_{ij} + \sum_{k=1}^l g_{ik}} \cdot q_i \quad (2)$$

For decap k , the total amount of charge it provides to all the core loads on chip is

$$Q_k = \sum_{i=1}^n q_{ki} \quad (3)$$

Furthermore, we assume that the allowed maximum voltage drop of the decap in the discharging phase is 10% of the supply voltage VDD [12], then we can find the amount of required capacitance for decap k is

$$C_k = \frac{Q_k}{0.1VDD} = \frac{\sum_{i=1}^n \left(\frac{g_{ik} \cdot T}{\sum_{j=1}^m h_{ij} + \sum_{k=1}^l g_{ik}} \cdot I_i \right)}{0.1VDD} \quad (4)$$

Given the power traces of the cores, I_i s, it is fast for us to calculate the amount of C_k for each decap k . This step can be improved by the observation that at each transition point, only part of the core loads change their power demands, which implies that we only need to use Equation (1) and (2) to recalculate q_{ki} s for changed I_i s, and then incrementally update corresponding C_k s by Equations (3) and (4).

3. EXPERIMENTAL RESULTS

We build 7 test cases from the IBM Power Grid Benchmarks [13]. Figure 4 shows the layouts of the two benchmarks. The power traces for each core load is obtained from the simulation of PARSEC 2.1 benchmarks [8] by the full system multicore simulator GEM5 [14].

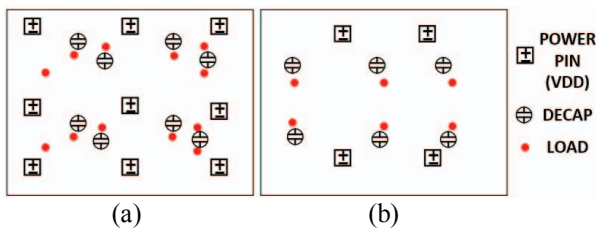


Figure 4: two test cases. (a) the case with 9 VDD pins, 8 decaps and 12 core loads. (b) the case with 4 VDD pins, 6 decaps and 6 core loads.

Figure 5 shows the runtime demand on capacitance of one decap in benchmark 3 of Table 1. We can clearly see that the demand is not constant, and the variation is more than 2X. Therefore, dynamically turning on or off part of the decap while meeting the minimum capacitance demand can potentially reduce the leakage power significantly.

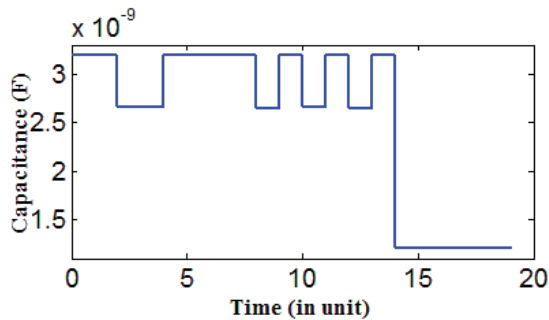


Figure 5: The runtime of capacitance demand for one decap in benchmark 3. One unit equals to 1000 cycles.

Table 1 summarizes the results for all the 7 benchmarks. For each benchmark we present the following results:

- “Cap for case 1”, the total “on” capacitance of the decaps when each decap is 100% on during the simulation period.

- “Cap for case 2”, the sum of total *average* capacitance of the decaps when the decaps are dynamically modulated adapting to the runtime needs of the core loads.
- “Leakage energy saving”, the total saving of leakage energy for all the decaps in a given chip.

We can see that on average, our approach can reduce about 35% of the decap leakage. For several benchmarks such as Benchmark5 and Benchmark7, our approach can achieve close to 50% saving.

Table 1. Results for all the test benchmarks.

Benchmark	Cap for case 1 (nF)	Cap for case 2 (nF)	Leakage energy saving
Benchmark1	15.27	10.41	31.83%
Benchmark2	15.27	11.20	26.65%
Benchmark3	15.27	11.11	27.24%
Benchmark4	15.27	10.06	34.12%
Benchmark5	15.27	8.02	47.48%
Benchmark6	16.39	11.94	27.15%
Benchmark7	16.39	8.48	48.26%
Average			34.67%

REFERENCES

- [1] G. E. Moore, “Cramming more components onto integrated circuits,” *Electronics*, vol. 38, no. 8, April 1965.
- [2] H. Esmailzadeh et al., “Power challenges may end the multicore era,” *Communications of the ACM*, vol. 56, no. 2, Feb. 2013.
- [3] S. Li et al., “The McPAT Framework for multicore and manycore architectures: simultaneously modeling power, area and timing,” *ACM TACO*, vol. 10, no. 1, pp. 5:1-5:29, April 2013.
- [4] H. Xiao et al., “Leakage power characterization and minimization in 3D stacked multi-core chips with microfluidic cooling,” in *SEMI-THERM*, pp. 207 – 212, 2014.
- [5] P. Zhou et al., “Congestion-aware power grid optimization for 3D circuits using MIM and CMOS decoupling capacitors,” in *ASPAC*, pp. 179-184, 2009.
- [6] J. Gu et al., “Design and implementation of active decoupling capacitor circuits for power supply regulation in digital ICs,” *TVLSI*, vol. 17, no. 2, pp. 292-301, 2009.
- [7] T. Xu et al., “Decoupling for power gating: sources of power noise and design strategies,” in *DAC*, pp. 1002 -1007, 2011.
- [8] C. Bienia, “Benchmarking modern multiprocessors,” Ph.D. dissertation, Princeton University, January 2011.
- [9] Y. Chen et al., “Gated decap: gate leakage control of on-chip decoupling capacitors in scaled technologies,” *CICC*, pp. 775-778, 2005.
- [10] M. Zhao et al., “Hierarchical analysis of power distribution networks,” *TCAD*, vol. 21, no. 2, pp. 159-168, 2002.
- [11] Y. K. Ramadass et al., “A fully-integrated switched-capacitor step-down DC-DC converter with digital capacitance modulation in 45nm CMOS,” *JSSC*, vol. 45, no. 12, pp. 2557-2565, 2010.
- [12] S. Zhao et al., “Decoupling capacitance allocation for power supply noise suppression.” *ISPD*, pp. 66-71, 2001.
- [13] IBM Power Grid Benchmarks. Available online at <http://dropzone.tamu.edu/~pli/PGBench/>.
- [14] N. Binkert et al., “The gem5 simulator,” in *ACM SIGARCH Computer Architecture News*, May 2011, pp. 1–7.