VOSCH: Voltage Scaled Cache Hierarchies

Weng-Fai Wong
Dept. of Computer Science,
National University of Singapore, Singapore
wongwf@comp.nus.edu.sg

Cheng-Kok Koh
School of Electrical and Computer Engineering,
Purdue University, West Lafayette, IN
chengkok@ecn.purdue.edu

Yiran Chen and Hai Li
Seagate Technology
Bloomington, MN
{yiran.chen|helen.li}@seagate.com

Abstract

The cache hierarchy of state-of-the-art—especially multicore—microprocessors consumes a significant amount of area and energy. A significant amount of research has been devoted especially to reducing the latter. One of the most important microarchitectural techniques proposed for the energy management is dynamic voltage scaling *(DVS). In DVS solutions, each cache operates at a number of different voltages. Most of the research in DVS techniques have been around how the voltages can be adjusted and tuned. In this paper, we depart from the use of DVS for energy conservation by examining* static *voltage assignments for caches. We propose the use of* voltage scaled cache hierarchies *(VOSCH) as a means to conserve both static and dynamic energy. In VOSCH, the caches are powered at progressively lower supply voltages as the cache level increases. Compared to DVS solutions, VOSCH is simple, potentially more robust and can conserve more energy. We also experimented with more aggressive designs that included the addition of small cache structures to VOSCH. Even greater energy savings were achieved without having to sacrifice performance.*

1. Introduction

One of the major contributing factors for the dramatic increase in performance of microprocessors is the introduction of on-chip caching. This ensured that aggressive microarchitecture designs are not starved of data to compute on. In 1995, the Intel Pentium Pro had a modest total of 16 Kbyte cache memory. Today, the latest Intel duo core Xeon 7150N processor has 56 Kbyte of L1, 1 Mbyte of L2 per core, and a 16 MByte L3 giving a total of over 18 MByte of cache storage [6]. In that same document, if we are to compare the Xeon 7120N and the Xeon 7130N, we see that power rating goes up from 95W to 150W. Besides a 0.1 GHz frequency difference, the 7130N has a L3 cache that is twice the size of that in the 7120N. This is evidence that caches—especially large caches—can consume a significant amount of energy. Indeed, it has been estimated that caches consume 30%–70% of the processor's energy [13].

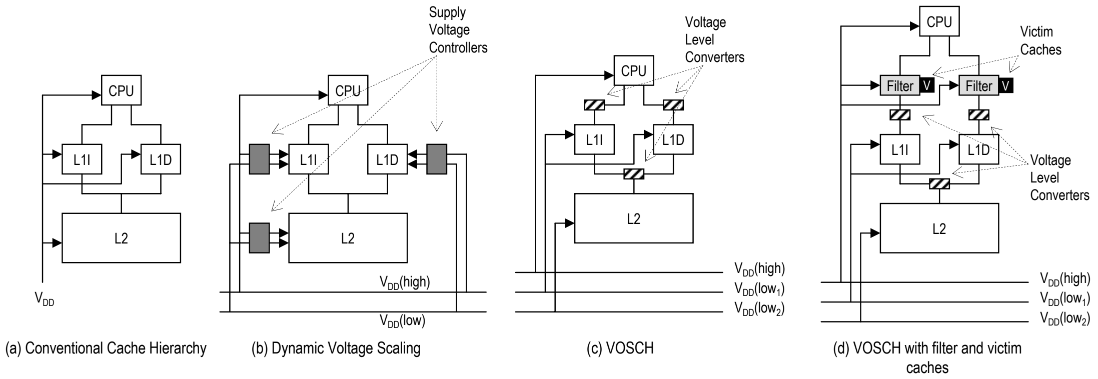
In this paper, we propose *voltage scaled cache hierarchies* (VOSCH). Unlike the conventional cache hierarchy, each level of caches in VOSCH is powered at a different (and lower) V_{DD} as the cache level increases. Level converters are used to bridge the levels (see Figure 1c). We shall show that VOSCH has a better noise margin and also introduces less noise in the power supply network. Our experiments show that VOSCH outperforms DVS solutions in terms of conserving both leakage and dynamic energy—especially for large caches. We shall also study the use of small cache structures, namely the *filter* and *victim* caches, to augment VOSCH. Even greater energy savings can be achieved in these more aggressive designs while maintaining the same performance as our baseline.

2. Conserving cache energy

The well-known relationship between dynamic power (P_{dyn}), leakage power (P_{leak}) and supply voltage (V_{DD}) can be summarized as follows [11]:

$$P_{dyn} \propto V_{DD}^2, \quad P_{leak} \propto V_{DD}^3 \sim V_{DD}^4.$$

Therefore, both dynamic and leakage power can be drastically reduced by scaling down the supply voltage which is the approach taken by dynamic voltage scaling techniques [5, 9, 15, 11]. In this work, we shall not consider



(a) Conventional Cache Hierarchy (b) Dynamic Voltage Scaling (c) VOSCH (d) VOSCH with filter and victim caches

Figure 1. Conventional caches, dynamic voltage scaled cache hierarchies, VOSCH and VOSCH with filter and victim caches.

orthogonal device-level techniques for leakage power reduction such as the scaling of threshold voltages and optimization of oxide thicknesses.

In [5], a dynamic V_{DD} scaling (DVS)-based cache design, called the *drowsy cache*, was proposed to manage cache power. Periodically, all cache blocks are put in a "drowsy" state with a low V_{DD}. When a drowsy cache block is being accessed, a latency penalty of 1 to 2 cycles is incurred to raise its V_{DD} back to the normal level. The cache block remains active for potential reuse in the near future until it is put into the drowsy mode again at the end of the period. The drowsy cache requires two power routings for the normal and low V_{DD}'s. A single-V_{DD} design to eliminate the additional routing requirement was proposed in [9]. An alternative approach in which the V_{DD} of the pipeline and the cache hierarchy is lowered whenever the pipeline is stalled by a long L2 cache miss was proposed in [11].

In the gated-V_{DD} techniques [16, 1], the power supply to cache lines that are not likely to be reused are selectively turned off with the use of sleep transistors. While these gated-V_{DD} techniques can reduce leakage power, they have negligible effect on the dynamic power of cache.

All these dynamic voltage scaling and gated-V_{DD} techniques share the common feature that the supply voltage of a cache is dynamically adjusted based on the cache access behavior. While substantial savings in cache energy can be expected, all these techniques incur latency or delay penalty when a cache line changes its state from drowsy or gated (low voltage) to active (high voltage). With the exception of [9], they require two or more power routings to realize dynamic voltage scaling or V_{DD}-gating. Moreover, when a cache switches from the drowsy (gated) state to the active state or vice versa, it requires charging or discharging of internal nodes in the cache line. The resultant power supply noise may further reduce the already stringent noise margin that is available.

3. Modeling of Caches

One focus of this work is the evaluation of the efficacy of dynamic V_{DD} scaling (DVS). In particular, we use the drowsy cache scheme presented in [5] as the flag bearer of DVS due to its simple design and its potential for energy reduction; it is a "near perfect" fine-grain voltage scaling solution—switching on when needed and off when it is not. Other solutions, especially those dealing with dynamic power, do not address the issue of static power at all. In contrast, VOSCH maintains the supply voltage to any one of its caches at a constant level, but progressively scales the supply voltage as the cache level increases. Hence, the level(s) of the supply voltage may be different from that of, say, the pipelines. In this section, we will examine several issues regarding these designs.

3.1. Power meshes

Currently the realization of dynamic V_{DD} mainly relies on two techniques: dual-V_{DD} and voltage regulator adjustment. As voltage regulators require a long time for voltage ramping [2], it is not suitable for the dynamic voltage scaling techniques targeting high-performance microprocessors. In dual- or multiple-V_{DD} designs, two or more power supply meshes are implemented with different voltage levels with transistors controlling the switching (Figure 1b).

In VOSCH, dual (or multiple) supply voltage meshes are also required (Figure 1c). However, each cache (logic and SRAM) in VOSCH is powered at a fixed voltage, it is not necessary to supply multiple voltages to the cache circuitry. A VOSCH cache would require only one single power supply mesh over it except at the perimeter, where it has to interface with components operating at a different supply voltage. Level-converters, which are powered by different

Cache	**32K L1**					**4M L2**				
V_{DD}	**1.0V**	**0.6V**	**0.55V**	**0.5V**	**drowsy**	**1.0V**	**0.6V**	**0.55V**	**0.5V**	**drowsy**
Core access energy (nJ)	1.0543	0.2482	0.1795	0.1236	1.0543	6.6681	1.6145	1.1673	0.8086	6.6681
Ctrl ckt access energy (nJ)	-	-	-	-	0.1024	-	-	-	-	0.2048
FF transition energy (nJ)	0.0007	0.0011	0.0012	0.0013	0.0007	0.0054	0.0091	0.0096	0.0107	0.0054
Total access energy (nJ)	1.0550	0.2493	0.1807	0.1250	1.1574	6.6735	1.6236	1.1769	0.8193	6.8783
RAM leakage power (W)	0.1234	0.0329	0.0244	0.0205	0.0099	15.6166	4.1656	3.0835	2.5925	1.2502
Logic leakage power (W)	0.0790	0.0146	0.0122	0.0101	0.0790	1.3224	0.2158	0.1960	0.1723	1.3225
Ctrl ckt leakage power (W)	-	-	-	-	0.0097	-	-	-	-	1.2252
FF leakage power (W)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	0.0001	0.0001	0.0001	0.0001
Total leakage power (W)	0.2024	0.0475	0.0366	0.0306	0.0986	16.9391	4.3815	3.2796	2.7649	3.7980
Core access time (ns)	0.5769	0.9574	1.1156	1.3350	0.8654	2.0228	3.5588	4.2696	5.0881	2.3112

Table 1. Power and access delay comparison of different caches designs.

power voltages, are required here. Therefore, compared to DVS, the area overhead due to power meshes and level conversion circuitry is minimal.

3.2. Power and delay modeling

The power and delay parameters used in this study are obtained from SPICE simulations. We simulated a standard sub-banked cache design. To estimate the cache access latency and power consumption under scaled supply voltages, we performed SPICE simulations of various circuit components of the cache model, including the signal driver, cache line, sense amplifier, and decoder, with the PTM 70nm Technology [3]. The netlists were obtained by scaling the post-layout SPICE netlist of a fabricated and tested chip implemented with 0.25μm technology.

The sizes and organization of the level 1 (L1) and level 2 (L2) caches considered in our circuit-level simulations are similar to the Intel Core 2 architecture. In particular, we assumed a 32 Kbyte, 8-way, 64 byte block L1 instruction cache and an L1 data cache of the same configuration. The L2 unified cache is 4 Mbyte, 16-way with a block size of 64 byte. Based on the circuit delay and power parameters obtained from SPICE simulations we modified CACTI [17] to calculate the access time, the levels of leakage power and dynamic power consumption of caches of various sizes and configuration under scaled V_{DD}'s.

Table 1 lists the delay and power/energy of different circuits of 32 Kbyte L1 and 4 Mbyte L2 cache configurations at 1.0V, 0.6V, 0.55V, and 0.5V, as well as in drowsy mode. The delay and energy/power due to level conversion in a VOSCH cache are also accounted for in Table 1 under flip-flop (FF) transition energy and leakage power. To obtain the delay and energy/power due to level conversion, we perform simulation of the level converter design used in [11]. In our simulations for drowsy cache, we use a 0.3V V_{DD} for drowsy mode. The voltage is chosen based on the results in [5] that a supply voltage that is 1.5 times the threshold voltage of 0.2V [7], is sufficient to retain the state of a cache cell. One might be surprised to see the high leakage energy consumed by a drowsy cache. It is important to realize that a drowsy cache scales only the supply voltage of SRAM cells. The peripheral circuitry are all powered at the regular power supply voltage (as shown in Figure 2 of ref. [5]). Hence, it is not possible to reduce the leakage energy of the logic components. Moreover, the leakage between high V_{DD} power supply and low V_{DD} power supply is significant.

V_{DD} (V)	1.0	0.6	0.55	0.5	0.3
SNM (mV)	309	213	198	177	91
SNM/V_{DD}	0.309	0.355	0.360	0.354	0.303

Table 2. Static Noise Margin.

3.3. Noise characterization

A SRAM cell with a lower V_{DD} is more sensitive to noise because the charge required to flip a storage node is proportional to V_{DD}. At first glance, a low voltage cache seems to be less tolerant to noise compared to a high V_{DD} cache and the drowsy cache which operates at high V_{DD} during data access.

We simulated the SRAM static transfer characteristics in the standby mode at different V_{DD}'s with the PTM 70nm technology node. The static noise margin (SNM) is defined to be the noise voltage necessary at each of the cell storage node to shift the static characteristics of the two-cell inverter vertically or horizontally along the side of the maximum nested square so that they intersect at only one point [12, 11]. Table 2 summarizes the SNM of a SRAM cell at different V_{DD}'s. We observed that as V_{DD} scales down, the SNM decreases as well. Although a low-voltage cache has lower SNM compared to a high voltage one, the noise level that it experiences is also lower since all signals within the cache are at a lower V_{DD}. As the noise level is a strong function of the signal voltage level, the static noise immunity of a low voltage cache in VOSCH is better

Cache	Frequency (GHz)				
	1	2	10	20	100
V_{DD}=1.0V	415	215	50	29	12.2
V_{DD}=0.6V	275	144	40	27	14.4
V_{DD}=0.55V	235	126	38	25.8	14.6
V_{DD}=0.5V	194	107	36	25.3	15.1
Drowsy Cache V_{DD}(SRAM cell)=0.3V V_{DD}(others)=1.0V	8	5.7	2.9	2.2	1.4

Table 3. Dynamic noise analysis—coupling capacitance required for flipping an SRAM Cell.

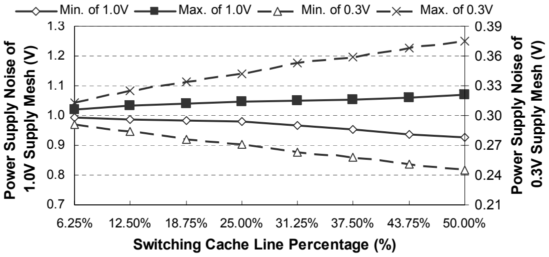


Figure 2. Power supply noise incurred by switching cache lines to drowsy mode.

quantified by the ratio of SNM to the respective V_{DD}. The third line of Table 2 shows that a low voltage cache operating at 0.6V, 0.55V or 0.5V has a higher (relative) static noise immunity compared to a cache operating at 1.0V. For completeness, we also show the normalized SNM for V_{DD} $= 0.3V$. At such a low voltage, which is very close to the threshold voltage of 0.2V, the static noise margin degrades significantly. Hence, we do not consider such a low V_{DD}.

To investigate the dynamic noise immunity for low voltage caches, we introduce a coupling capacitance to the storage node and applied a full swing transition at the input of the capacitor. Table 3 lists the smallest coupling capacitance required to flip the cell for various transition times. In general, low voltage caches are less immune to dynamic noise compared to a high voltage one. However, the coupling capacitance required for flipping a low voltage SRAM cell even in the high-frequency range is more than 10fF, which is much higher than the typical coupling capacitance at the 70nm technology node. On the other hand, it takes very small coupling capacitances to cause a cell to flip for a drowsy cache due to the uneven voltage scaling of SRAM cells (low V_{DD}) and bitlines and wordlines (high V_{DD}).

Dynamic voltage scaling may induce severe power supply noise. To analyze this, we consider a drowsy cache hierarchy configuration, called `D2K-D0`, where the drowsy 32K L1 data and instruction caches each has a period of 2000 cycles and the 4M L2 cache stays drowsy all the time [5]. In other words, after every 2000 cycles, all L1 lines that have been awaken during the previous period are put back into the drowsy mode while L2 lines are immediately put back into the drowsy mode after access.

Figures 2 shows the worst-case power supply noise induced on both 1.0V and 0.3V power supply meshes for a drowsy cache. The simulation shows that when 37.5% of cache lines switched, the voltage drop on the 0.3V supply network can be as high as 20% of the supply voltage, which is way higher than the industry tolerance of 10% [4]. Based on SimpleScalar simulations of `D2K-D0`, we observe that an average of 40% L1 cache lines are awaken in a 2000-cycle period. The power supply noise can potentially cripple the system. Of course, this problem can be alleviated by various schemes. One possible way is to cycle through the cache one row at a time and put only one cache row to drowsy mode in one (or a few) clock cycle(s). Such a scheme would achieve the effect of putting a row to sleep after an update window. However, this is beyond the scope of this study.

While it is important to devise an “active-asleep” policy that does not put strain on the power supply network for DVS, VOSCH is inherently power-supply friendly. For a low voltage cache of VOSCH, only a cache row would be switching at any given time. Consequently, the power supply noise on the corresponding single V_{DD} supply network is less than 2% of the supply voltage. VOSCH is evidently a simpler and more robust alternative to DVS. As we shall see in the next section, it is also very effective in energy conservation.

4. VOSCH vs. Drowsy

We used SimpleScalar 3.0 [18], in particular, `sim-outorder`, to obtain the throughput performance and energy results for VOSCH and DVS cache design. Recall that we use cache configurations that are similar to the Intel Core 2 architecture: a 32 Kbyte, 8-way, 64 byte block level one instruction cache and a level one data cache of the same configuration and a 4 Mbyte, 16-way, 64 byte block L2 unified cache. We also considered 2 Mbyte, 1 Mbyte, and 512 Kbyte L2 unified cache in our study. As we observed similar trends in the performance and energy among these different cache sizes, we shall focus on presenting the results of cache configurations with a 4 Mbyte L2 cache in the sequel. In the simulations, we used the ‘compressed’ instruction cache of SimpleScalar which implies that instructions are assumed to be 32 bits long. For the purpose of energy calculation, we assumed that the processor is clocked at 3 GHz. We evaluate the

	Configuration H-H		**Configuration D2K-D0**		**Configuration H-D0**		**Configuration H-L**	
	IPC	**Tot. Ener. (J)**	**Norm. Perf.**	**Norm. Ener.**	**Norm. Perf.**	**Norm. Ener.**	**Norm. Perf.**	**Norm. Ener.**
164.gzip	1.67	12.13	1.02	0.49	1.01	0.49	1.01	0.50
168.wupwise	1.21	119.04	1.00	0.43	1.00	0.44	1.00	0.46
171.swim	0.96	592.07	1.01	0.37	1.00	0.38	1.00	0.40
172.mgrid	1.42	6.33	1.01	0.44	1.00	0.44	1.00	0.47
173.applu	0.86	134.07	1.01	0.40	1.01	0.40	1.01	0.41
175.vpr	1.36	4.37	1.04	0.48	1.03	0.48	1.03	0.49
176.gcc	1.44	23.60	1.03	0.49	1.02	0.50	1.02	0.50
177.mesa	1.70	359.13	1.00	0.47	1.00	0.48	1.00	0.50
178.galgel	1.88	21.64	1.03	0.50	1.02	0.51	1.02	0.51
179.art	1.19	104.57	1.07	0.47	1.05	0.47	1.05	0.46
181.mcf	0.65	17.78	1.03	0.38	1.02	0.38	1.02	0.39
183.equake	1.38	9.31	1.01	0.44	1.00	0.45	1.00	0.47
186.crafty	1.61	24.38	1.02	0.49	1.01	0.49	1.01	0.51
187.facerec	1.52	25.09	1.01	0.47	1.01	0.47	1.01	0.48
188.ammp	0.17	79.88	1.01	0.29	1.01	0.30	1.01	0.31
189.lucas	1.45	18.86	1.01	0.47	1.01	0.47	1.01	0.49
191.fma3d	1.57	0.35	1.03	0.49	1.03	0.49	1.03	0.51
197.parser	1.28	34.67	1.02	0.46	1.01	0.46	1.01	0.48
200.sixtrack	1.92	1.43	1.00	0.49	1.00	0.50	1.00	0.51
252.eon	1.47	2.65	1.01	0.49	1.00	0.49	1.00	0.51
253.perlbmk	1.43	11.79	1.01	0.48	1.00	0.48	1.00	0.50
254.gap	1.10	66.83	1.02	0.43	1.01	0.44	1.01	0.45
255.vortex	1.51	128.75	1.02	0.48	1.01	0.48	1.01	0.50
256.bzip2	1.72	29.40	1.01	0.48	1.00	0.48	1.00	0.50
300.twolf	0.96	6.19	1.02	0.44	1.01	0.44	1.01	0.46
301.apsi	1.34	89.82	1.02	0.46	1.01	0.46	1.01	0.47
em3d	0.97	2.37	1.04	0.44	1.03	0.44	1.03	0.45
tsp	0.68	188.28	1.00	0.36	1.00	0.37	1.00	0.40
Norm. Ave. (4M L2)	1	1	1.02	0.45	1.01	0.45	1.01	0.47
Norm. Ave. (2M L2)	1.06	0.67	1.08	0.38	1.07	0.39	1.07	0.38
Norm. Ave. (1M L2)	1.20	0.53	1.23	0.36	1.22	0.36	1.22	0.34
Norm. Ave. (512K L2)	1.36	0.56	1.40	0.37	1.38	0.37	1.38	0.36

Table 4. Simulation results for different cache power configurations.

Energy Policy	Size of Filter	Victim?	4MB L2		2MB L2		1MB L2		512KB L2	
			Norm. Perf	Norm. Ener	Norm. Perf	Norm. Ener	Norm. Perf	Norm. Ener	Norm. Perf	Norm. Ener
H-L	0	N	1.01	0.44	1.03	0.35	1.16	0.31	1.29	0.32
	2 Kbyte	N	1.00	0.25	1.03	0.16	1.15	0.12	1.28	0.13
		Y	0.98	0.25	1.00	0.17	1.12	0.13	1.25	0.14
	4 Kbyte	N	0.98	0.26	1.01	0.18	1.13	0.14	1.26	0.15
		Y	0.97	0.28	0.99	0.19	1.11	0.15	1.24	0.16
H-XL	0	N	1.02	0.39	1.04	0.32	1.12	0.29	1.33	0.30
	2 Kbyte	N	1.02	0.20	1.04	0.14	1.18	0.11	1.32	0.11
		Y	0.99	0.21	1.01	0.14	1.15	0.12	1.29	0.12
	4 Kbyte	N	1.00	0.22	1.02	0.15	1.15	0.13	1.30	0.13
		Y	0.98	0.23	1.00	0.16	1.16	0.13	1.28	0.15
L-XL	0	N	1.07	0.21	1.09	0.14	1.24	0.11	1.42	0.12
	2 Kbyte	N	1.05	0.19	1.07	0.11	1.21	0.08	1.40	0.09
		Y	1.01	0.19	1.03	0.12	1.17	0.10	1.34	0.10
	4 Kbyte	N	1.02	0.20	1.04	0.13	1.18	0.10	1.36	0.11
		Y	1.00	0.22	1.02	0.15	1.16	0.12	1.33	0.13
L-XXL	0	N	1.07	0.19	1.10	0.13	1.24	0.10	1.46	0.11
	2 Kbyte	N	1.05	0.16	1.08	0.10	1.22	0.07	1.41	0.08
		Y	1.01	0.17	1.03	0.11	1.18	0.09	1.36	0.10
	4 Kbyte	N	1.02	0.18	1.04	0.12	1.19	0.10	1.37	0.10
		Y	1.00	0.19	1.02	0.14	1.17	0.11	1.35	0.12
XL-XXL	0	N	1.14	0.17	1.17	0.10	1.34	0.07	1.51	0.08
	2 Kbyte	N	1.08	0.16	1.10	0.10	1.25	0.07	1.41	0.08
		Y	1.03	0.17	1.05	0.11	1.20	0.09	1.35	0.09
	4 Kbyte	N	1.04	0.18	1.06	0.12	1.21	0.09	1.36	0.10
		Y	1.01	0.19	1.03	0.13	1.18	0.11	1.33	0.12

Table 5. The impact of adding a filter cache and a 4-entry victim cache to VOSCH.

following four energy policies for the caches:

1. H-H: This is the baseline policy, where both L1 instruction and data caches as well the L2 cache are powered at 1.0V. The instruction and data L1 cache latencies are assumed to be 2 cycles (see Table 1 for the access time) and the unified L2 cache latency is assumed to be 10 cycles: 6 cycles of cache access and 4 cycles of bus access.

2. D2K-D0: This is the drowsy cache policy outlined in Section 3.3. The L1 and L2 caches respectively incur an additional one and two cycle latencies in waking up a line [5].

3. H-D0: The L1 caches are powered at 1.0V, but the L2 cache stays drowsy all the time.

4. H-L: A conservative VOSCH: the L1 and L2 caches are powered at 1.0V and 0.6V, respectively. The latency of the low-V_{DD} L2 cache is 15 cycles (11 and 4 cycles for cache and bus access, respectively).

In this set of experiments, we used the default `sim-outorder` settings for all the other parts of the processor. We used the entire SPEC 2000 suite [19] as well as `tsp` and `em3d` from the pointer-intensive Olden benchmark suite [14] for our experiments. In order to reduce the simulation time, we used the reduced input set for SPEC 2000 [20].

The results of the simulation, i.e., the number of simulated cycles and the total (dynamic and leakage) energy of the L1 and L2 caches, are given in Table 4. The entries for D2K-D0, H-D0, and H-L in the tables are normalized (denoted by ‘Norm.’). The entries for normalized energy are obtained by dividing by the corresponding values of the baseline, i.e. H-H., and the entries for normalized performance are obtained the other way round. In both cases, a normalized entry that is < 1.0 represents an improvement.

In line with the results reported in [5], the average performance slowdown for the D2K-D0 policy is slightly more than 1% of the baseline H-H policy. The energy consumed by the drowsy cache design is only 42% of that of H-H. Interestingly, the H-D0 and H-L policies exhibit similar performance slowdown and energy reduction as D2K-D0. In other words, for overall energy reduction, it is not necessary to apply dynamic voltage scaling, in particular drowsy cache techniques, to L1 caches. This is because the L1 caches are accessed very frequently. As a result, the dynamic energy consumed by L1 is relatively high. In fact, an average of 40% cache lines have to be awaken in an update window of 2000 cycles, which may induce significant noise on a power supply network.

	2K Filter	4K Filter	Victim
Access energy (nJ)	0.1009	0.2004	0.0696
Leakage power (W)	0.0270	0.0540	0.0146
Access time (ns)	0.3205	0.3205	0.2347

Table 6. Power and delay numbers of filter and victim cache powered at V_{DD} = 1V.

5. Aggressive VOSCH designs

We further investigated more aggressive VOSCH designs. Specifically, we consider a L1 powered at 0.6V with an access time of 3 cycles ('`L-?`' in Table 5), as well as at 0.55V and an access time of 4 cycles ('`XL-?`'). We also considered a 0.55V L2 with a 17 cycle access time ('`?-XL`') and a 0.5V L2 with a 20 cycle access time ('`?-XXL`'). Table 5 showed that the overall energy consumption can be reduced further. However, this comes with performance degradation.

We found that performance can be recovered by the insertion of an additional memory hierarchy into VOSCH consisting of *filter caches* [10]. A filter cache is a small, direct mapped cache added to the front of the memory hierarchy before L1. In effect, it deepens the cache hierarchy by one level. A filter cache has the following performance and energy impact:

- Since it is small and direct-mapped, it is faster to access (see Table 6). If an application exhibits good spatial and temporal locality, most of its memory requests can be satisfied by the filter cache.

- By 'filtering' the memory references, the filter cache reduces visit to L1. This translates to lesser activities in L1 and therefore lowering the dynamic energy consumption.

We also experimented with the addition of two four-entry fully associative *victim caches* [8]. Victim caching were proposed to improve the performance of direct mapped caches. They were implemented in a number of processors including the HP PA-RISC processors. The tiny victim caches were effective in improving performance by a few percentage points. However, because they are accessed in every memory reference, they consume a non-trivial amount of energy. Together with 4 Kbyte filter caches, VOSCH (Figure 1d) recovered all performance lost while incurring less than 20% of the energy of the baseline—double the gains achieved by other voltage policies including dynamic voltage scaling (see Table 4). We also added filter and victim cache to `H-H`, `D2K-D0`, and `H-D0`, and the normalized performance and energy numbers are shown in Table 7. While improvements in performance and energy can also be observed, they are not as significant as in the case of VOSCH. The access time and energy as well as the leakage power used to compute the values in Tables 5 and 7 are shown in Table 6.

6. Conclusion

In this paper, we first re-examined dynamic voltage scaling in the light of the growing size of the cache hierarchy. At the same time, we also studied the issue of noise in such schemes. We find that the amount of energy that one can save in the L1 cache by means of dynamic voltage scaling is small compared to the total leakage power of the L2 cache. Perhaps the biggest obstacles to the success deployment of such dynamic voltage scaling schemes are (i) the low dynamic noise margin of drowsy SRAM cells, (ii) the high level of power supply noise injected by the active-to-drowsy mode switching of cache lines, and (iii) the high leakage energy due to logic components and dual-V_{DD} meshes. In response to these challenges, we proposed VOSCH, cache hierarchies in which each cache is powered by a single V_{DD} that may be different from the other caches in the hierarchy. VOSCH offers an attractive alternative to achieving similar level of energy reduction, while maintaining the same level of robustness as conventional cache design and incurring a lower area penalty.

We further investigated the use of filter and victim caching mechanisms to augment VOSCH. These extended VOSCH designs can achieve an energy saving above that of dynamic voltage scaling while attaining nearly the same performance as the baseline. In conclusion, we believe that VOSCH is a simple, robust and effective scheme for both static and dynamic energy conservation especially for the large cache hierarchies of today's microprocessors.

References

[1] A. Agarwal, H. Li, and K. Roy. DRG-cache: A data retention gated-ground cache for low power. In *DAC '02: Proc. of the 39th Design Automation Conference*, pages 473–478, 2002.

[2] T. Burd, T. Pering, A. Stratakos, and R. Brodersen. A dynamic voltage scaled microprocessor system. *IEEE J. of Solid-State Circuits*, 35(11):1571–1580, Nov 2000.

[3] Y. Cao, T. Sato, D. Sylvester, M. Orshansky, and C. Hu. New paradigm of predictive mosfet and interconnect modeling for early circuit design. In *Proc. of the IEEE Custom Integrated Circuits Conference*, pages 201–204, 2000. http://www-device.eecs.berkeley.edu/ ptm.

[4] H. H. Chen, J. S. Neely, M. F. Wang, and G. Co. On-chip decoupling capacitor optimization for noise and leakage reduction. In *The 16th Symp. on Integrated Circuits and System Design*, 2003.

Energy Policy	Size of Filter	Victim?	4MB L2		2MB L2		1MB L2		512KB L2	
			Norm. Perf	Norm. Ener	Norm. Perf	Norm. Ener	Norm. Perf	Norm. Ener	Norm. Perf	Norm. Ener
H-H	0	N	1	1	1.02	0.64	1.15	0.49	1.27	0.52
	2 Kbyte	N	1.00	0.81	1.02	0.44	1.14	0.30	1.26	0.33
		Y	0.97	0.80	0.99	0.45	1.11	0.30	1.23	0.33
	4 Kbyte	N	0.98	0.81	1.00	0.46	1.11	0.31	1.24	0.34
		Y	0.96	0.82	0.98	0.47	1.10	0.32	1.22	0.35
D2K-D0	0	N	1.01	0.41	1.03	0.34	1.17	0.32	1.30	0.33
	2 Kbyte	N	1.01	0.23	1.03	0.15	1.16	0.13	1.29	0.14
		Y	0.98	0.23	1.00	0.16	1.13	0.14	1.26	0.15
	4 Kbyte	N	0.99	0.24	1.01	0.16	1.13	0.14	1.27	0.15
		Y	0.98	0.25	1.00	0.18	1.12	0.16	1.25	0.17
H-D0	0	N	1.01	0.42	1.03	0.34	1.16	0.32	1.29	0.33
	2 Kbyte	N	1.00	0.23	1.03	0.16	1.15	0.13	1.28	0.14
		Y	0.98	0.24	1.00	0.17	1.12	0.14	1.24	0.15
	4 Kbyte	N	0.98	0.25	1.01	0.18	1.13	0.15	1.26	0.16
		Y	0.97	0.26	0.99	0.19	1.11	0.17	1.24	0.18

Table 7. The impact of adding a filter cache and a 4-entry victim cache to `H-H`, `D2K-D0`, and `H-D0`.

[5] K. Flautner, N. S. Kim, S. Martin, D. Blaauw, and T. Mudge. Drowsy caches: Simple techniques for reducing leakage power. In *ISCA '02: Proc. of the 29th International Symp. on Computer Architecture*, pages 148–157, 2002.

[6] Intel Corp. Intel Processor Numbers. http://www.intel.com/products/processor_number/chart/xeon.htm.

[7] The International Technology Roadmap for Semiconductors. http://www.itrs.net, 2005.

[8] N. P. Jouppi. Improving direct-mapped cache performance by the addition of a small fully-associative cache and prefetch buffers. In *ISCA '90: Proc. of the 17th International Symp. on Computer Architecture*, pages 364–373, 1990.

[9] N. S. Kim, K. Flautner, D. Blaauw, and T. Mudge. Single-VDD and single-VT super-drowsy techniques for low-leakage high-performance instruction caches. In *ISLPED '04: Proc. of the 2004 International Symp. on Low Power Electronics and Design*, pages 54–57, 2004.

[10] J. Kin, M. Gupta, and W. Mangione-Smith. Filtering memory references to increase energy efficiency. *IEEE Trans. on Computers*, 49(1):1–15, Jan 2000.

[11] H. Li, C.-Y. Cher, and K. Roy. Combined circuit and architectural level variable supply-voltage scaling for low power. *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, 13(5):564–576, May 2005.

[12] J. Lohstroh, E. Seevinck, and J. Groot. Worst-case noise margin criteria for logic circuits and their mathematical equivalence. *IEEE Journal on Solid-State Circuits*, SC-18:803–806, Dec 1983.

[13] S. Manne, A. Klauser, and D. Grunwald. Pipeline gating: Speculation control for energy reduction. In *ISCA '98: Proc. of the 25th International Symp. on Computer Architecture*, pages 132–141, 1998.

[14] The Olden Benchmark Page. http://www.cs.princeton.edu/ mcc/olden.html.

[15] S. Petit, J. Sahuquillo, J. M. Such, and D. Kaeli. Exploiting temporal locality in drowsy cache policies. In *CF '05: Proc. of the 2nd Conference on Computing Frontiers*, pages 371–377, 2005.

[16] M. Powell, S.-H. Yang, B. Falsafi, K. Roy, and T. N. Vijaykumar. Gated-Vdd: A circuit technique to reduce leakage in deep-submicron cache memories. In *ISLPED '00: Proc. of the 2000 International Symp. on Low Power Electronics and Design*, pages 90–95, 2000.

[17] P. Shivakumar and N. P. Jouppi. CACTI 3.0: An integrated cache timing, power, and area model. Technical report, HP Western Research Labs, 2001.

[18] SimpleScalar LLC. The SimpleScalar toolset. http://www.simplescalar.com.

[19] Standard Performance Evaluation Corp. SPEC CPU2000 V1.3. http://www.spec.org/cpu2000.

[20] University of Minnesota ARCTiC Labs. Minnespec: A new SPEC benchmark workload for simulation-based computer architecture research. http://www.arctic.umn.edu/minnespec/index.shtml.