Controlling Leakage Power with the Replacement Policy in Slumberous Caches

Nasir Mohyuddin, Rashed Bhatti, Michel Dubois
Department of Electrical-Engineering Systems
University of Southern California
Los Angeles, CA 90089-2560
{nasir.mohyuddin,rashed.bhatti}@usc.edu,dubois@paris.usc.edu

Abstract

As technology scales down at an exponential rate, leakage power is fast becoming the dominant component of the total power budget. A large share of the total leakage power is dissipated in the cache hierarchy. To reduce cache leakage, individual cache lines can be kept in drowsy mode, a low voltage, low leakage state. Every cache access may then result in dynamic power consumption and performance penalties. A trade-off between the amount of leakage power saved on one hand, and the impact on dynamic power and performance on the other hand must be reached.

To affect this trade-off, we introduce "slumberous caches" in which the power level of cache lines is controlled with the cache replacement policy. In a slumberous cache, cache lines are maintained at different power save modes which we call "tranquility levels", which depend on their order of replacement priorities.

We evaluate the effectiveness of this idea in the context of PLRU, LRU and MRR (Modified Random replacement) cache replacement algorithms. We explore various assignment of the tranquility levels to lines and compare overall power and performance impacts. As technology scales down, the dynamic power and performance penalties required to energize slumberous cache lines drops drastically while the leakage power savings remain roughly steady.

1.0 INTRODUCTION

Traditionally, computer architects have mostly be concerned about performance, cost and reliability. Power considerations were secondary. Moreover computer architects are used to ignore and abstract the technology level details of their design. In recent years, this situation has dramatically changed and power is becoming one of the primary design parameters at both architecture and physical design levels. Several factors have contributed to this trend. Perhaps the primary driving factor has been the remarkable success and growth of the class of personal computing devices (portable desktops, audio- and video-based multimedia products) and wireless communications systems (personal digital assistants and personal communicators), which demand high-speed computation and complex functionality with low power consumption. In high-end machines, power dissipation and its effect on temperature, cooling and performance are becoming the major limiting factor to feature size and frequency scaling.

There are two types of power dissipated in a chip: dynamic power and static power. Dynamic power is incurred whenever the state of a circuit changes, whereas static power is dissipated (*leaked*) in each and every circuit, all the time, independently of its changes of state. The International Technology Roadmap for Semiconductors (ITRS) produced by the Semiconductor Industry Association predicts that leakage current Ioff will double with each generation for both high-performance (low threshold voltage Vt, high leakage) and low-power (high Vt, low leakage) transistors [13].

Different techniques apply to dynamic and static power reduction. Static power is the focus of this paper. Static power is also often referred to as *leakage* power.

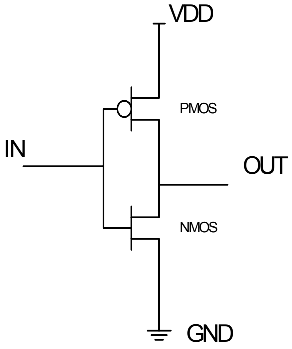

FIGURE 1. Schematic view of a basic CMOS inverter

To understand leakage power one can look at the basic structure of a CMOS inverter, shown in Figure 1. The three major sources of leakage are sub-threshold leakage, substrate leakage, and leakage through gate oxide [12]. In Figure 2 we model the sources of leakage by diodes. Sub threshold leakage is due to diode D2 and D4. When the input of the inverter is low the output is high and the reverse biased diode D2 causes sub threshold leakage. Conversely, when the input is high and output is low the reverse biased diode D4 causes sub threshold leakage.

Diode D3 between the power supply VDD and ground GND is responsible for the substrate leakage. The overall substrate leakage is proportional to the dimensions and number of devices grown in the n-wells over the p-substrate. Since the substrate is lightly doped leakage through the substrate is very small as compare to sub threshold leakage.

The leakage through the gate oxide (Diodes D1 and D5) is also very small. The ways to reduce substrate leakage and gate oxide leakage are mostly technology level techniques such as twin tub and SOI technologies.

Sub-threshold leakage is currently the largest of these three components, and is bound to increase in future fabrication technologies as threshold voltages are scaled down [7]. In this paper, we focus on sub-threshold leakage. We ignore gate oxide and substrate leakages and the techniques proposed in this paper do not address these leakages.

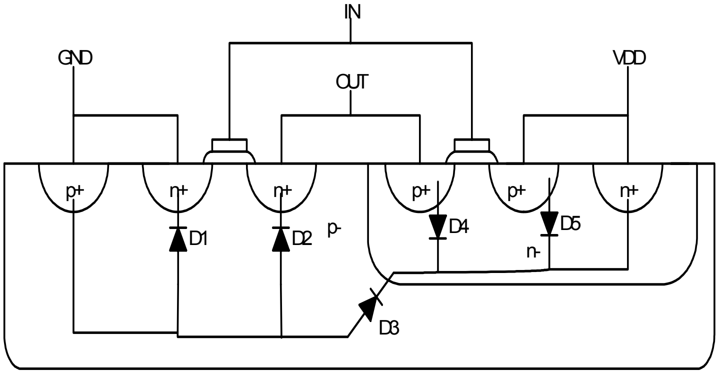

FIGURE 2. A cross-sectional view of the basic CMOS inverter with different N and P type regions inferring diodes.

Effective leakage power reduction techniques are based on switching SRAM memory cells to low leakage mode when they are not accessed. However, whenever a cell in lower leakage power mode is accessed, power levels may change, which result in dynamic power consumption and performance penalties. A trade-off between the amount of leakage power saved on one hand, and the impact on dynamic power and performance on the other hand must be reached.

To affect this trade-off in the context of the cache hierarchy, we introduce "slumberous caches" in which the power level of set-associative cache lines is controlled with the cache replacement policy. The replacement policy is useful in set-associative caches to improve the hit rate of the cache because it exploits the locality property of memory accesses. This same locality property can be exploited to optimize the trade-off between static power, dynamic power and performance. In a slumberous cache, cache lines are maintained at different power save modes which we call "tranquility levels". The lines in each set of a slumberous cache are maintained at tranquility levels which depend on their order of replacement priorities.

The effectiveness of this idea is first evaluated in the context of PLRU a common cache replacement algorithm. Then it is extended to couple of other replacement algorithms. We explore various schemes for the tranquility levels assigned to lines and compare overall power and performance impacts. As technology scales down, the dynamic power and performance penalties required to energize slumberous cache lines drops drastically while the leakage power savings remain roughly steady.

In the next section we overview several well-known techniques of leakage power reduction architectural level. We propose a low leakage design scheme keeping the future technologies in view in the section 5. In the subsequent sections we explain our exploratory approach. The results of the simulations are shown in the section 6. Finally we discuss the proposed idea in the light of the empirical results and future possible extension to this idea before we conclude.

2.0 RELATED SCHEMES TO REDUCE LEAKAGE POWER

Several ideas have been explored to reduce leakage power at the architectural level in microprocessors. All of these leakage power saving schemes rely on some changes at the circuit level to cut off power [16] to cache lines or to switch them to reduced (drowsy) voltage levels. When power to a line is cut off, the information is lost, a backup copy must exist in the hierarchy, and the next access to the line causes a miss. Drowsy voltage levels are such that the information is not lost, but the line must first be energized to full voltage before it can be accessed, which results in dynamic power and performance penalties.

M. Martonosi et.al [1] proposed the Cache Decay scheme. By invalidating and "turning off" cache lines when they hold data not likely to be reused leakage power can be saved. Success relies on accurately predicting the cache line dead periods, i.e. the periods when a cache line is sitting idle and is useless, only consuming static power. A cache line is turned off if a preset number of cycles (called “decay interval”) have passed since the cache line’s last access. This results in 70% leakage energy reduction. An adaptive scheme that chooses the best decay interval for each generation of a cache line on the fly is also proposed. Problem with this scheme is that early shut off of a cache line will increase the miss rate consequently affecting overall performance and incurring dynamic power.

A compiler based strategy to reduce leakage energy was proposed by W. Zhang et al. for instruction caches [9]. Their scheme is based on marking the last usage of instructions by a special instruction which turns off the cache line. To limit the frequency of these special instructions, the authors turn off instructions at the loop granularity level. At the exit from a loop that will not be visited again, the cache lines are turned off.

The concept of resizable cache was proposed by Babak Falsafi et al [5]. This method exploits the fact that cache utilization varies from application to application and also within an application. So, statically or dynamically varying the cache size by turning off unused cache portions can save lot of static energy. Two different schemes were used to vary the cache size. “Selective-ways” changes the cache set associativity and “Selective-sets” changes the cache set sizes according to cache usage. Static resizing is done across entire applications and dynamic resizing changes the cache size on demand during execution.

Dynamic threshold modulation [15] using MTCMOS applied at cache line granularity in which the threshold voltage of the transistors in the SRAM cell is dynamically increased when the cell is set to sleep mode by raising the source-to-body voltage of the transistors in the circuit. This higher Vt reduces the leakage current while allowing the memory cell to maintain its state even in sleep mode

All the schemes describe so far rely on shutting down parts of the cache. Our approach is based on the idea of drowsy caches [3]. Drowsy Cache lines are never completely shut down. Every cache line can be in two voltage levels: full rail voltage and drowsy mode voltage. In drowsy mode the supply voltage of the cache line is lowered to the minimum possible level without corrupting the data. A drowsy bit is used to select the mode between full rail voltage and drowsy voltage mode. All cache lines are put in drowsy mode at regular intervals of 2000 cycles. Results for Spec2000 benchmarks suggest that, for most of the benchmarks, 90% of the cache lines can be kept in drowsy mode. With wake-up penalties for a drowsy cache line of no more than one cycle, the authors results show that the total leakage energy was reduced by an average of 71% when tags were always awake and by an average of 76% using the drowsy tag scheme, with modest performance impact. In the same vein of work, drowsy instruction caches [4] a cache bank prediction scheme to predict which bank of the instruction cache to put in drowsy mode and which to turn on.

3.0 CONTROLLING LEAKAGE IN SLUMBEROUS CACHES

In slumberous caches we propose to reduce the leakage power by controlling the voltage levels of lines in the same cache set with the replacement policy.

3.1 Tranquility Levels

We consider set-associative caches with at least two priority levels for replacement within a set (Thus our approach is not suitable for random replacement policies or direct-mapped caches.) In an n-way cache, the cache lines are ranked with respect to their priority of replacement, P1,..., Pn. The line with priority level Pn is always selected for replacement. We dynamically assign different voltage levels to the cache lines at different priority levels, based on the information kept in the replacement policy state bits. These voltage levels are called *tranquility levels*, and various schemes are possible in general to assign tranquility levels (T1,...Tn) to replacement priorities (P1, P2,.Pn). T1 is the highest voltage level and Tn is the lowest voltage level. The lowest possible supply voltage must be greater than 200mV above the threshold voltage in order to avoid that ambient noise flips some bits of the line.

Figure 3 shows a simple circuit to switch a cache line from one tranquility level to another for a 4-way set associative cache. The four power rails remain energized with the different voltage levels. The replacement policy state bits control the transistors feeding the voltage level to the line. Multiple switching transistors can be distributed along the power rails of the cache lines to avoid current bottlenecks. In all schemes considered in this paper, cache tags and state bits are always at full rail voltage so that no clock cycle is wasted in waking them up. This framework allows for the design of *slumberous caches*, in which leakage power is controlled by the replacement policy.

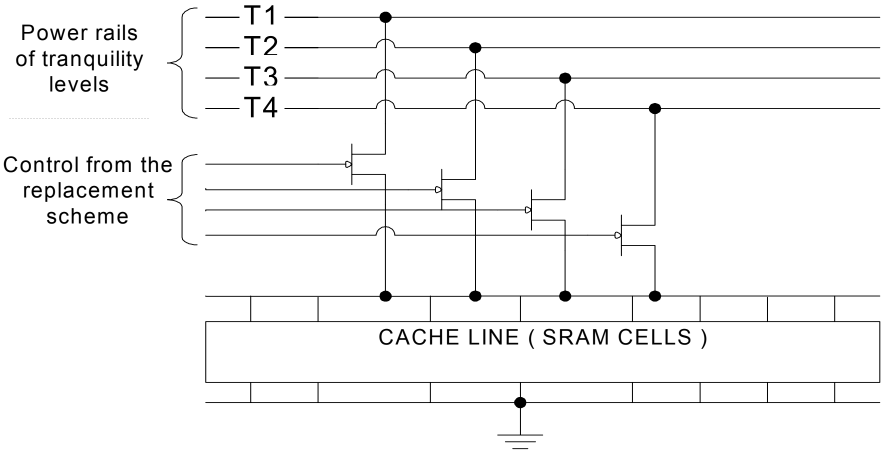
Power rails of tranquility levels
T1
T2
T3
T4
Control from the replacement scheme
CACHE LINE (SRAM CELLS)

FIGURE 3. Control circuitry to implement proposed scheme

3.2 Maximum Leakage Power Savings

The leakage power per byte without any power saving scheme is given by the following equation

$$P_{leakage} = 8 \bullet (I_{leakage} \bullet V_{dd})$$

Where $I_{leakage}$ is the leakage current per bit and V_{dd} is the full rail voltage.

For an n-way set-associative slumberous cache the maximum leakage power saving per byte is

$$P_{saving} = 8 \bullet \left[V_{dd} \bullet I_{leakage} \angle (1/n) \sum_{i=1}^{n} V_n I_n \right]$$

where *Vn* and *In* are the voltage level and the per-bit leakage current of the tranquility level of each line in a set. *n* is the number of ways of associativity. This maximum savings only depend on the technology, the number of ways, and the tranquility levels. However, to reap the benefits of this maximum power savings we need to solve several problems and trade-offs.

3.3 Leakage Control Schemes

The simplest scheme is to keep all lines in a set at the same tranquility level, independently of the replacement priorities and, if needed, to wake up the line every time it is accessed. The schemes with only one tranquility level will be called TL1 (Tranquility Level One) and will use -Ti to indicate the voltage of that tranquility level. Hence all one tranquility level schemes will be denoted as TL1-Ti. TL1-T1 is the scheme that does not have any leakage savings, as all lines will be at the full rail voltage at all times and TL-Tn will have maximum savings as all lines will be at the lowest tranquility level. TL1-T1 will not have any performance impact and TL1-Tn will have worst performance impact as worst wake-up penalties are incurred every time a line is accessed. TL1-Tn will also incur dynamic power each time a cache line is accessed. So a trade-off is needed to be made between leakage power savings and performance impact.

To improve this trade-off, we exploit the fact that the MRU (Most Recently Used) line is very likely to be referenced over and over again. Our evaluations show that for different replacement policies, on the average more than 94% of data hits are to the MRU line. Thus we can keep the MRU line at T1, while keeping all the other lines in the set at one tranquility level, T2,T3,...or Tn. This will reduce the leakage power savings but at the same time it will reduce the performance loss and the dynamic energy needed to wake-up lines. Hence we have TL2 (Two Tranquility Levels) schemes. As for TL2 schemes one tranquility level is T1 by default, so we indicate two tranquility level schemes as TL2-Ti where Ti is the tranquility level employed for non-MRU lines i.e. T2, T3,... Tn. TL2-T2 means a scheme that has two tranquility levels, T1 for MRU and T2 for all non-MRU lines. Similarly we can have TL2-T3, TL2-T4, ... TL2-Tn.

Finally, more than two tranquility levels can also be used hence we will have TL3, TL4, ...TLn, where n is the number of ways in a set associative cache. For TLn each priority level P1, P2,.Pn will be associated with a different level of tranquility T1, T2,...,Tn (respectively). We have considered linearly distributed voltages for tranquility levels between the lowest possible operating voltage (deepest tranquility state) i.e. Vt + 200mv and the full power supply voltage (wake up state). Other distributions are possible, but are beyond the scope of this paper.

4.0 TECHNOLOGICAL FINDINGS TO EVALUATE SLUMBEROUS CACHES

We have done many evaluations, both technological and architectural, to evaluate the trade-offs between leakage power, dynamic power and performance in the design of slumberous caches. Technological evaluations will be discussed in this section.

4.1 Leakage Power of Different Tranquility Levels

An hspice deck was setup with a standard SRAM cell to measure the leakage power of one SRAM cell shown in Figure 4. We simulated the cell over present and future technologies using presently available and Berkeley predictive technology (BPT) models [2] for simulations.

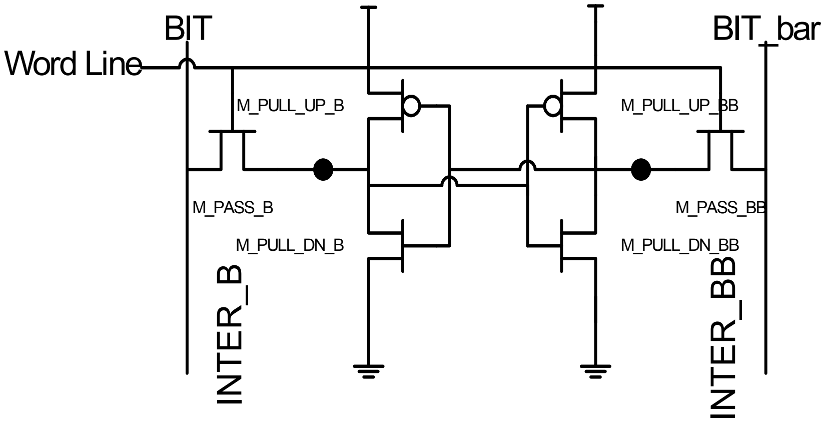

FIGURE 4. Standard SRAM Cell.

Next we establish the minimum voltage level that guarantees a consistent state of the memory. A simple noise analysis suggests that minimum voltage should be approximately 200mV above Vth, the threshold voltage. Vth for each different technology is determined through simulations using BPT models [2], Table 1 shows the operating supply voltages and threshold voltages suggested by our simulations.

TABLE 1. Supply and threshold voltages for different technologies

Technology	Supply (V)	Vth (V)
130nm	1.3	0.596
100nm	1.1	0.546
70nm	0.9	0.394

We used full rail V_{dd} level for T1 and Vth + 200mv for T4 level. Power save Voltage levels for T2 and T3 are selected by linear interpolation between T1 and T4. We have run extensive simulations over different technologies to verify the correct operation of an SRAM cell at different tranquility levels and the outcome is shown in Table 2. The leakage current increases significantly as the threshold voltage decreases with technology scaling. Also leakage current decreases with the voltage for different tranquility levels within a technology.

TABLE 2. Leakage current of one cell at different power save levels using BPT models [2]

Technology	Operating voltage / steady state leakage current per bit for different tranquility levels used							
	T 1		T 2		T 3		T 4	
	Voltage (V)	Current (nA)	Voltage (V)	Current (nA)	Voltage (V)	Current (nA)	Voltage (V)	Current (nA)
130nm	1.3	0.948	1.1	0.673	0.9	0.550	0.7	0.475
100nm	1.1	2.522	0.95	1.818	0.8	1.481	0.65	1.292
70nm	0.9	8.949	0.8	7.321	0.7	6.340	0.6	5.655

4.2 Dynamic Power Costs

Table 3 shows the energy required to switch between various tranquility levels in different technologies.

TABLE 3. Energy per transition per byte between different tranquility levels.

	Energy per transition (joules)					
Technology	T1<->T2	T1<->T3	T1<->T4	T2<->T3	T2<->T4	T3<->T4
130nm	8.6819E-15	3.4728E-14	7.8137E-14	8.6819E-15	3.4728E-14	8.6819E-15
100nm	3.3922E-15	1.3569E-14	3.0530E-14	3.3922E-15	1.3569E-14	3.3922E-15
70nm	1.0967E-15	4.3870E-15	9.8707E-15	1.0967E-15	4.3870E-15	1.0967E-15

To derive the expression for dynamic power consumption we start with a basic circuit of forced switching of a capacitor through an energy dissipating switching device as shown in the Figure 5. Theoretically no energy is dissipated by a capacitor in switching from one voltage level to another; the energy is dissipated in the non ideal resistive switching device.

To analyze we start from abnitio:

Let C be the capacitance of the capacitor

V_i : Initial voltage of capacitor

V_f : Final voltage of the capacitor

Instantaneous current through capacitor during switching is $I_c = C\frac{\delta V_c}{\delta t}$.

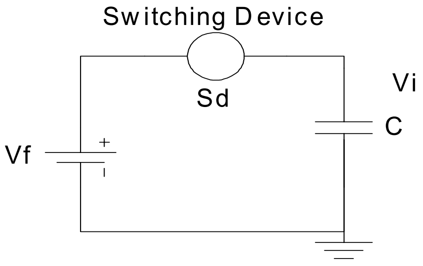

FIGURE 5. Energy dissipation through a switching device

$I_c = I_s$(Current through switching device)

The power dissipation in the switching device: $P_s = V_s I_s$

Plugging in the I$_c$ we have

$$P_s = V_s \times C\frac{\delta v_c}{\delta t_s} => P_s \times dt_s = V_s \times Cdv_c$$

Writing the voltage across the switching device (Vs) in terms of voltage across capacitor (Vc)

$$V_s = V_f \angle V_c => P_s \times dt_s = (V_f \angle V_c) \times Cdv_c$$

Integrating above equation for the switching time Ts, during which $V_c = V_i \Rightarrow \mathrm{V}_f$

$$\begin{aligned}\int_0^{T_s} P_s \times dt &= C \times \int_{V_i}^{V_f} (V_f \angle V_c) dv_c \\ &= CV_f \int_{V_i}^{V_f} dv_c \angle C \int_{V_i}^{V_f} V_c dv_c \\ &= CV_f(V_f \angle V_i) \angle \frac{C}{2}(V_f^2 \angle V^2{}_i) \\ &= C(V_f \angle V_i)\left[V_f \angle \frac{1}{2}(V_f + V_i)\right] \\ &= C(V_f \angle V_i)\left[\frac{1}{2}(V_f \angle V_i)\right] \\ &= \frac{C}{2}(V_f \angle V_i)^2\end{aligned}$$

$$=> \quad E = \frac{1}{2}C\Delta V^2$$

Hence when a cache line with total line capacitance of *C*, is switched from a tranquility level *Ti to* a tranquility level *Tf* the energy dissipated is given as : $E_{dynamic} = \frac{1}{2} \bullet C|(Vi \angle Vf)|^2$

The total amount of dynamic energy depends on the replacement policy, the benchmarks, and the levels of tranquility. We must consider the effect of benchmarks to evaluate dynamic power costs.

4.3 Performance Costs

Transitions between tranquility levels come at a cost in terms of performance. To determine the exact wake-up time, we have run simulations to measure the time needed to wake-up an SRAM cell from each tranquility level to the full power mode. Figure 6 shows the simulated curves obtained by switching power rails of an SRAM cell for 70nm technology. We switch lines from different tranquility level voltages to full rail voltage (from left, 1st we switch from T2 then from T3 and finally from T4). Time is measured for each transition and compared to the clock periods proposed by Agarwal et. al. [14] for the corresponding technology.

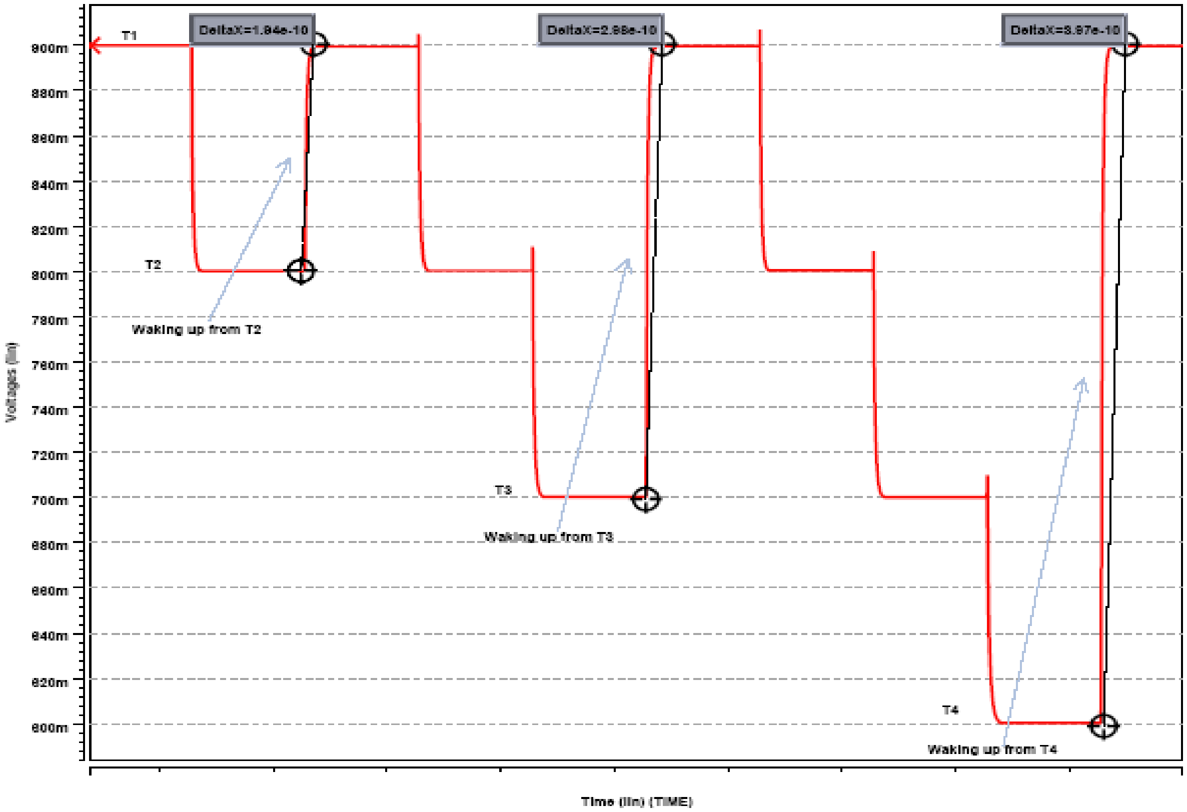

FIGURE 6. The power wake up cycle from different levels of tranquility in the 70nm technology

8FO4 clock frequencies for the considered technologies as suggested by Agarwal et. al. [14] are given in Table 4. Table 5 shows wake-up penalties in terms of cycles. Our hspice simulations using the predictive technologies [2] revealed that the wake up penalty from T2 level is 1 cycle and is 2 cycles from both T3 and T4. We observed that the trend is towards increasing wake-up penalty as also discussed by Agarwal et. al. [14] that cache access time is not scaling as fast as clock. So for future technologies the wake-up penalty from lowest level of tranquility will be 3 cycles or even more.

TABLE 4. 8FO4 clock frequencies used for our simulations.

Technology (nm)	8FO4 Clock (GHz)	Cycle time (nSec)
130nm	2.67	0.37
100nm	3.47	0.29
70nm	4.96	0.20

TABLE 5. Wake up penalties in terms of cycles

	Wake-up penalty (cycles)			
Technology	T1	T2	T3	T4
130nm	0	1	2	2
100nm	0	1	2	2
70nm	0	1	2	2

4.4 Maximum Possible Leakage Power Savings in Slumberous Caches

In this section we ignore any dynamic power costs of switching from one tranquility level to another tranquility level, to get an upper bound on the leakage power saving that can be achieved by different schemes discussed in Section 3.3. An propound for any leaking power scheme employing DVS (Dynamic Voltage Scaling) is obtained by keeping all cache lines at lowest possible voltage level at all times, i.e. TL1-T4 for a 4-way set associative cache. All hits and misses will wake-up one cache line, but immediately put it back to T4 level.In this section a 4-way set associative cache is considered. Upper bounds for various slumberous cache schemes viz TL1-T4, TL4 (all 4 replacement priority levels have separate tranquility level, 4 levels in this case) TL2-T2, TL2-T3 and TL2-T4 are calculated. Table 6 shows the upper bounds for average leakage power saved per byte for above mentioned schemes. Table 7 and Figure 7 show the same information in terms of the % savings (relative to total leakage power). The maximum savings only depends on the technology, the number of cache ways, and the tranquility levels. They are independent of the replacement policy (because, at anytime, the same number of lines are at any one tranquility level). They do not include the dynamic power needed to switch between tranquility levels (that’s why we call them *maximum*.)

TABLE 6. Average leakage power dissipated per byte and maximum leakage power savings per byte

Technology	Average Static Power dissipated per Byte (nW)	Average Static Power Saved per Byte (nW)				
		TL1-T4	TL4	TL2-T2	TL2-T3	TL2-T4
130nm	9.86	7.20	4.26	2.95	4.42	5.40
100nm	22.19	15.48	9.14	6.28	9.54	11.61
70nm	64.43	37.29	20.95	13.18	21.70	27.97

TABLE 7. Maximum percent leakage power savings.

Technology	TL1-T4	TL4	TL2-T2	TL2-T3	TL2-T4
130nm	73.00%	43.18%	29.92%	44.85%	54.75%
100nm	69.73%	41.19%	28.31%	42.97%	52.30%
70nm	57.87%	32.51%	20.46%	33.67%	43.40%

Of all approaches, TL1-T4, which keeps all lines at the minimum power levels, yields the best reduction of leakage power. TL2-T4 is second, TL2-T2 is last, and TL4 is in-between. These observations should be obvious, given that leakage power savings depends on tranquility levels. However, one must also contend with dynamic power and performance penalties before making a final judgment.

.

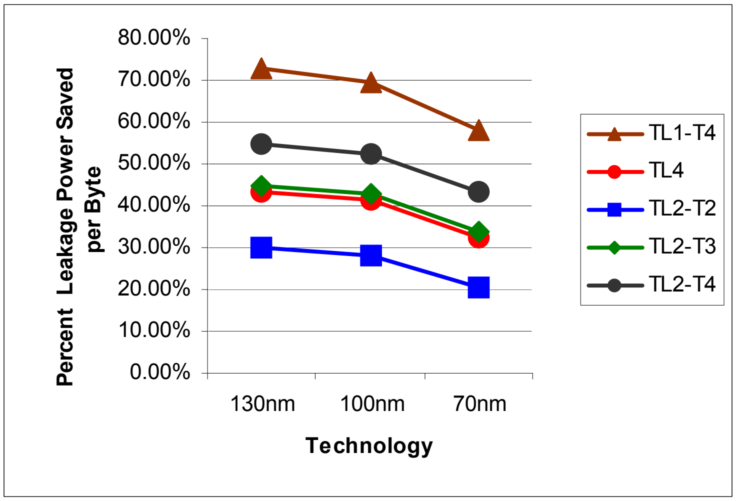

FIGURE 7. Maximum percent leakage power savings for various schemes

Though leakage energy savings increases exponentially with technology scaling as we will see in Section 6.0, percent leakage savings decrease as technology scales, Figure 7. This decrease in percent saving is because the difference between T1and Tn levels is reduced with technology scaling. for technologies considered it reduces from 0.6V to 0.3V i.e. 50% reduction! So ultimately state destroying leakage energy saving techniques seem to survive. Using replacement priority information for state destroying leakage saving is beyond the scope of this paper.

5.0 Architectural Simulations

Whereas the leakage power savings are independent of the benchmarks, we must run architectural simulations to understand the dynamic power and performance implications of each power savings schemes.

5.1 Simplescalar Simulations

We modified the simplescalar code to provide the required statistics for the calculation of the average power dissipation by the L1 data cache for different Spec2000 benchmark programs. Table 8 gives the processor model used for our simulations.

TABLE 8. Baseline microprocessor Simulation Model

Instruction Cache	16k 2-way set-associative, 32 byte blocks, 1 cycle latency
Data Cache	8k 4-way set-associative, 32 byte blocks, 1cycle latency
Unified L2 Cache	1Meg 4-way set-associative, 64 byte blocks, 20 cycle latency
Memory	100 cycle round trip access
Out-of-Order Issue	out-of-order issue of up to 4 instructions per cycle, 128 entry re-order buffer
Architecture Registers	32 integer, 32 floating point
Functional Units	4-integer ALU, 2-load/store units, 4-FP ALUs, 1-integer MULT/DIV, 1-FP MULT/DIV

We have used single sample Simpoints [13] of 100-million instruction each for selected spec2000 programs The resulting Simpoints are given in Table 9.

TABLE 9. The single simpoints for simulations of the Spec2000 benchmarks

Spec2000 Benchmarks	gzip	gcc	mcf	parser	vpr	bzip2	twolf	equake	art
Single Simpoint	814	960	369	1030	1722	184	11	5496	42

6.0 EVALUATIONS OF SLUMBEROUS CACHES FOR FAMOUS CACHE REPLACEMENT ALGORITHMs

The concept of slumberous caches can be applied to various replacement policies with at least two priority levels, we considered three replacement policies namely LRU (Least Recently Used) PLRU (Pseudo LRU), and MRR (Modified Random Replacement). We have concentrated on the design of the L1 data cache in the Pentium 4, an 8k 4-way set associative cache with 32-byte lines.

6.1 Pseudo LRU (PLRU)

For completeness we review the PLRU policy. PLRU approximates LRU. LRU is difficult to maintain in wide caches because of the complexity of updating the state bits to keep track of replacement priorities. To implement PLRU in a 4-way cache, we need three state bits called Bit_0, Bit_1 and Bit_2. Table 10 shows the cache line priority levels for different combinations of these three bits. Line at P1 is the MRU line and line at P4 is the line to replace.

TABLE 10. Cache-line replacement priorities for different combinations of Bit_0, Bit_1 and Bit_2 in PLRU

Bit_2	Bit_1	Bit_0	Line_0	Line_1	Line_2	Line_3
0	0	0	P4	P3	P2	P1
0	0	1	P2	P1	P4	P3
0	1	0	P3	P4	P2	P1
0	1	1	P1	P2	P4	P3
1	0	0	P4	P3	P1	P2
1	0	1	P2	P1	P3	P4
1	1	0	P3	P4	P1	P2
1	1	1	P1	P2	P3	P4

Figure 8 shows how the state bits are used to select a victim line. Bit_0 selects between two groups of cache lines, group_0 (line_0 and line1) or group_1 (line_2 and line_3). Bit_1 selects between line_0 and line_1 and Bit_2 selects between line_2 and line_3. If Bit_0 is zero we don't care about Bit_2 and Bit_1 decides which of line_0 or line_1 is replaced. Similarly if Bit_0 is 1 then we don't care about Bit_1 and Bit_2 selects between line_2 and line_3. When a cache line is referenced we change the state of the state bits e.g. if line_0 is accessed we set Bit_0 to 1 so that the next victim will be in group_1 and also we set Bit_1 to 1 so that next time when group_0 is selected line_1 will be selected for replacement.

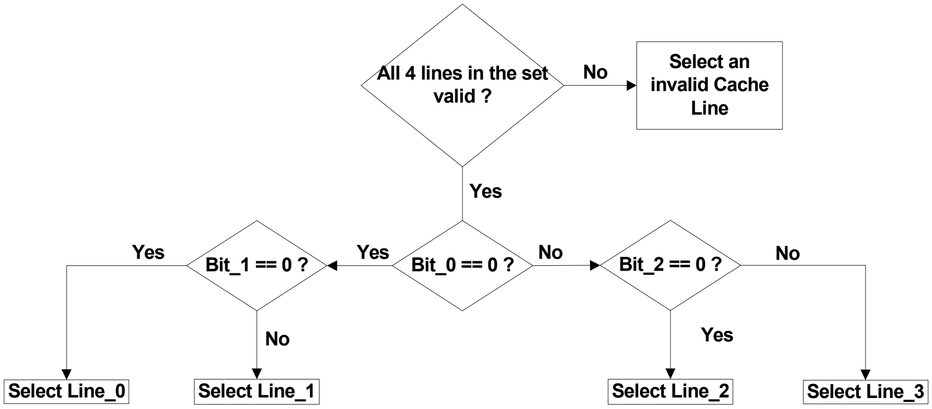

FIGURE 8. PLRU implementation for 4-way caches

Figure 9 shows the priority level transitions of cache lines on an access to a set. For example, if a hit occurs at a cache line whose priority level is P3, then its priority goes to P1, the priority of the line previously at P4 goes to P2, the line at P1 goes to P3 and the line at P2 goes to P4. These priority level transitions are dictated by PLRU and result in various tranquility level transitions, depending on the control scheme employed.

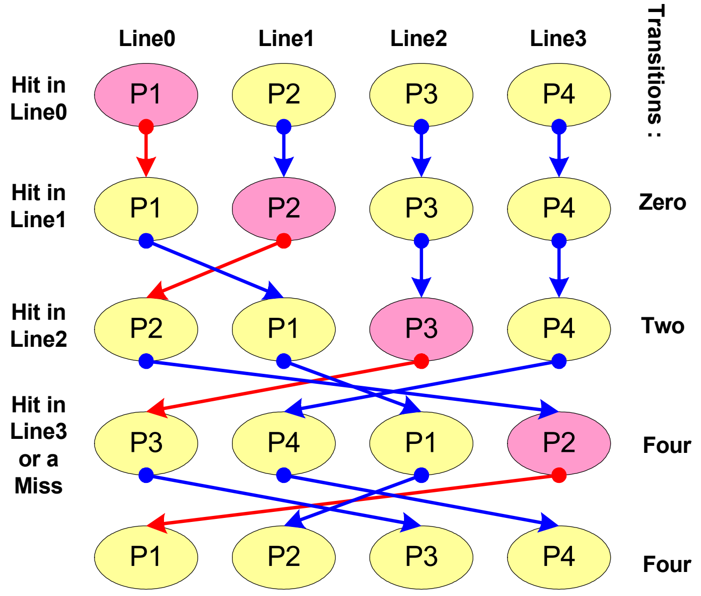

FIGURE 9. How hits at different Priority levels affect the whole cache under PLRU replacement policy.

Leakage saving using PLRU algorithm is evaluated for schemes described in Section 4.4

6.1.1 Dynamic Power Penalties

Figure 10 shows the dynamic power required per byte to save leakage power in L1 data cache for TL4 under PLRU.

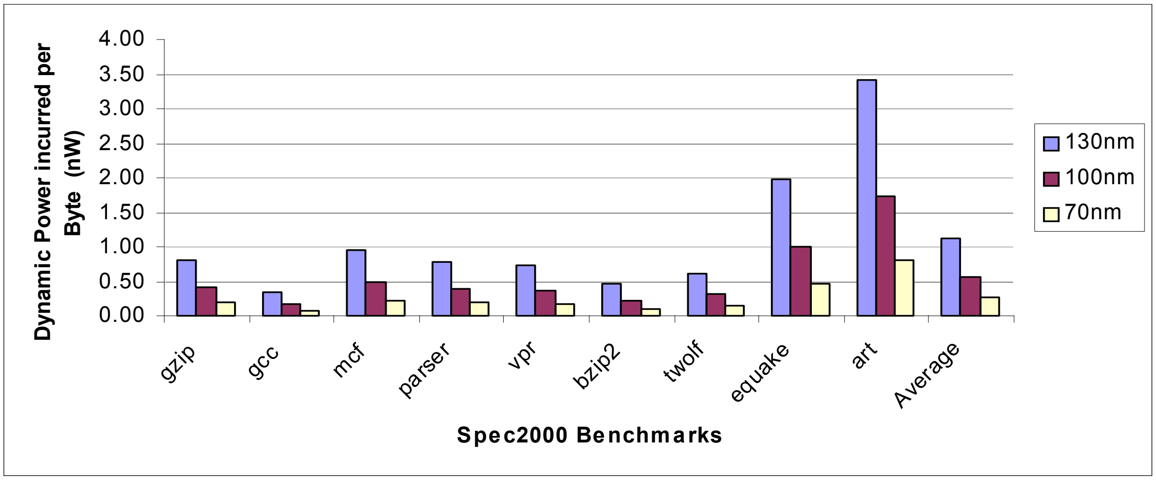

FIGURE 10. Dynamic power incurred per byte for L1 data cache for TL4 under PLRU policy for different benchmarks

This power varies in a wide range across the benchmarks and depends upon the number of hits in P2-P4 levels and number of misses. The dynamic power costs are different for various benchmarks under TL2-T4 as compared to TL4 see Figure 11, the reason is increased dynamic costs for hits in P2, P3 and misses. On the average dynamic power cost is doubled. Dynamic power is significantly reduced as technology scales. This is because the range of voltage levels between T1 and T4 is reduced as technology scales and the dynamic power cost is inversely proportional to the square of the voltage difference.

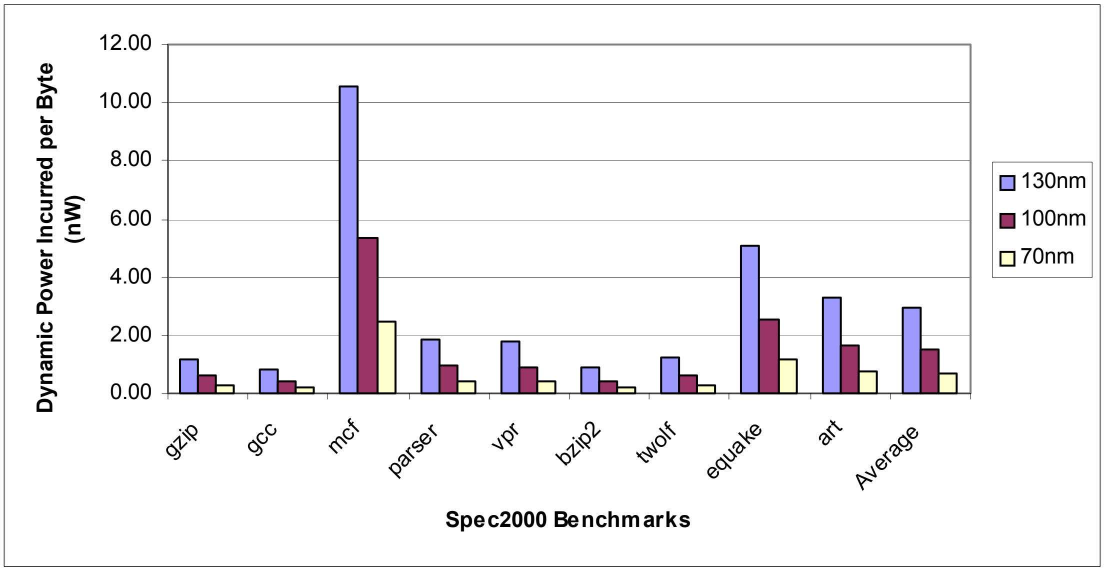

FIGURE 11. Dynamic power incurred per byte for L1 data cache for TL2-T4 under PLRU policy for different benchmarks

In Table 11 and Figure 12 we compare the amount of average dynamic power consumed by various schemes to save leakage power. The dynamic power is the average dynamic power across all benchmarks, obtained by summing up all the dynamic energy needed for all the benchmarks and dividing the sum by the total execution time. In all cases, dynamic power is by far the worse in TL1-T4, although the gap closes quickly with scaled-down technologies. The curves can be explained by the voltage difference between drowsy and full rail levels in the different schemes.

TABLE 11. Average dynamic power costs for various schemes under PLRU

	TL1-T4	TL4	TL2-T2	TL2-T3	TL2-T4
130nm	20.07	1.12	0.33	1.31	2.96
100nm	10.20	0.57	0.17	0.67	1.50
70nm	4.71	0.26	0.08	0.31	0.69

It is clear that we need to consider both the effects of leakage power and dynamic power caused by the leakage power scheme in our evaluations.

The net power savings is:

$$\text{Net Saving} = \text{Leakage Power Saved - Dynamic Power incurred}$$

Figure 13 shows the Net Savings for different benchmarks for TL4 scheme under PLRU. We see that, as technology scales, net power savings become independent of the benchmark because the dynamic power becomes negligible and the static power saved is independent of the benchmark.

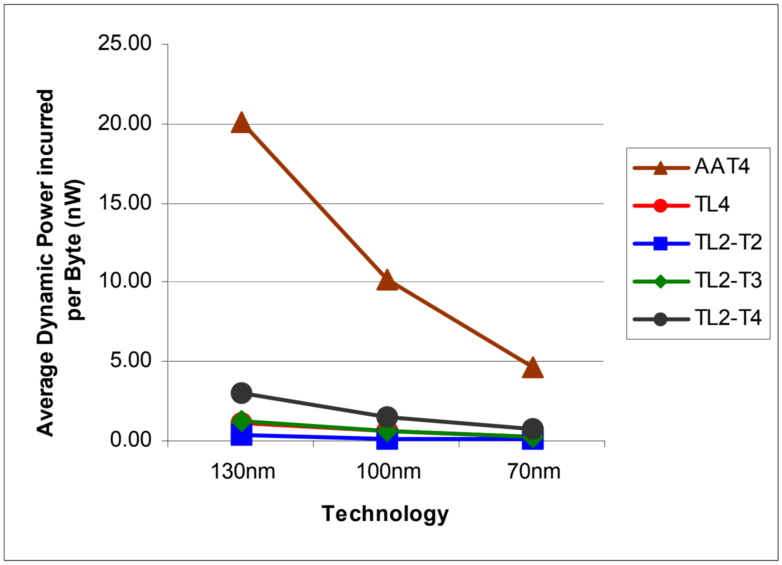

FIGURE 12. Average dynamic power incurred per byte for L1 data cache for different power schemes under PLRU

Whereas the Net Savings increases exponentially with scaled down technology, it is important to look at the percent savings, as the leakage power also grows exponentially with the technology.

The percent leakage power savings is: $\%\ \text{Net Saving} = (\text{Net Saving} / \text{Total Leakage Power}) \times 100$

The %Net Savings for TL4 under PLRU are shown in Figure 14, across the benchmarks. It shows that the percent savings remains steady across technologies, and also becomes independent of the benchmark as technology scales down.

I

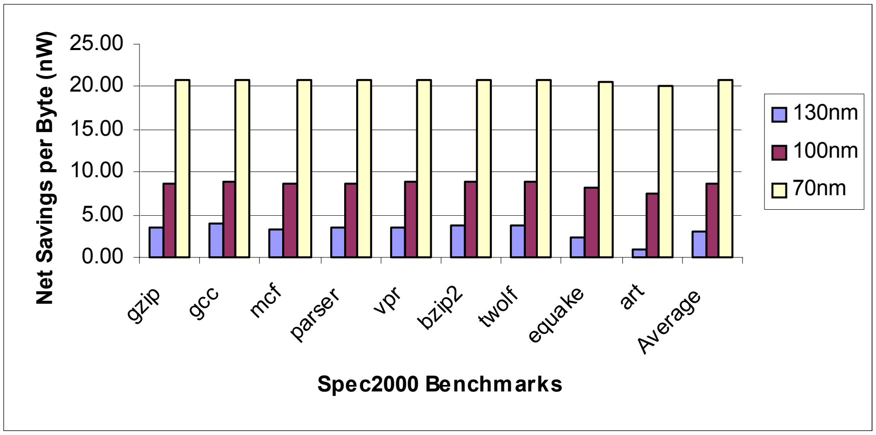

FIGURE 13. Net Savings per byte for L1 data cache for different benchmarks for TL4 uner PLRU

.

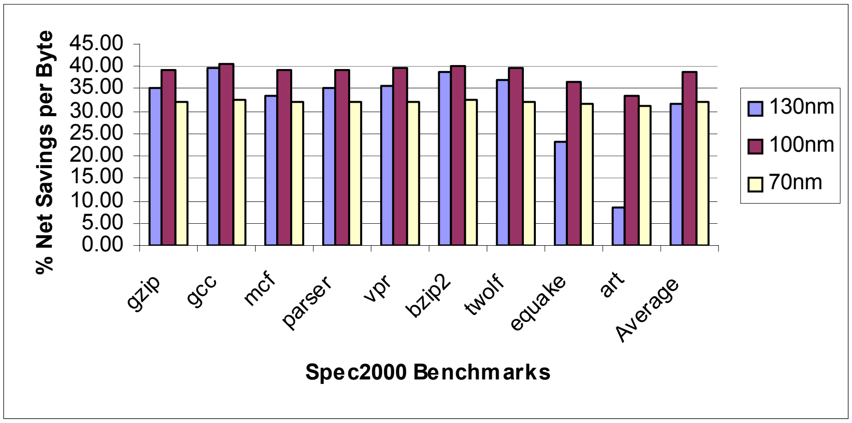

FIGURE 14. % Net Savings per byte for L1 data cache for TL4 scheme under PLRU policy

Figure 15 shows the average net leakage power saved per byte across all benchmarks. Regardless of the power scheme, Net Savings increases exponentially with technology in all cases. Because of the explosive increase of leakage power and the rapid drop in dynamic power with technology scaling, the net gains obtainable by TL1-T4 (which means all lines at T4) are on a steeper upwards curve than those for any of the schemes governed by the replacement policy. We observe that TL2-T4 give the most savings among schemes dictated by the replacement policy and that, for the most advanced technology we have looked at, its savings are roughly equal to those of TL1-T4.

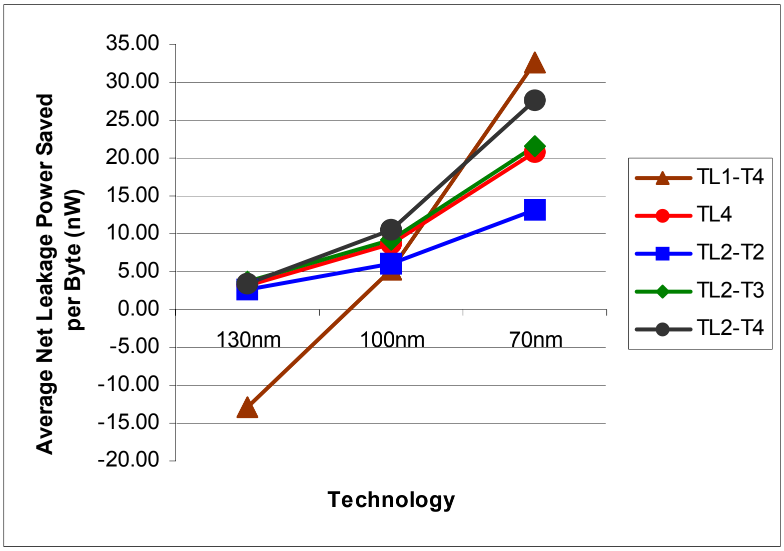

FIGURE 15. Average Net Leakage power saved per byte for L1 data cache under various schemes over future technologies

Finaly average net leakage power saving is compared to the maximum saving compouted in Section 4.0, all savings are shown as percent of TL1-T4 savings of Section 4.0 refer Figure 16.

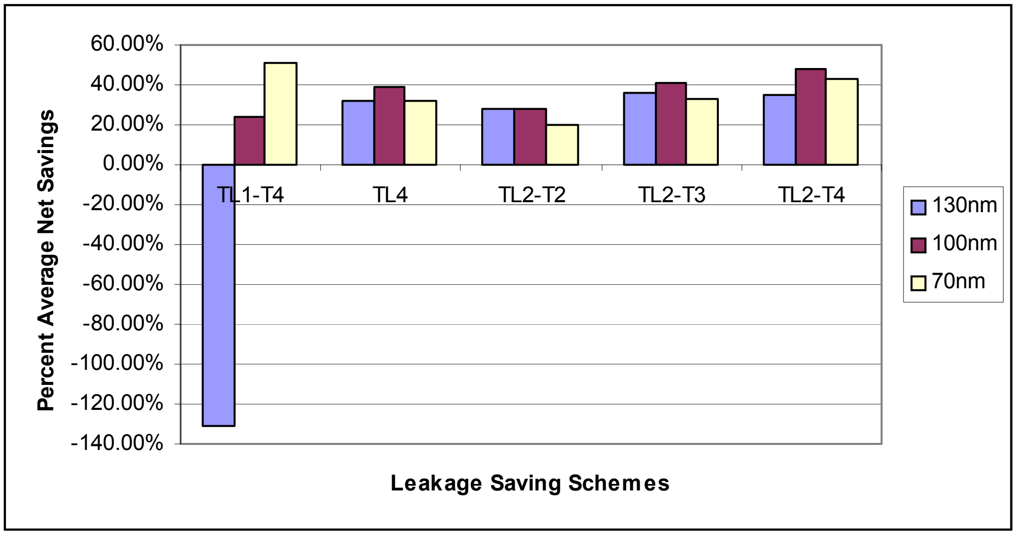

FIGURE 16. Percent Average Net Leakage power saved per byte for L1 data cachefor various schemes under PLRU

6.1.2 Performance Penalties

Figure 17 shows the average increase in L1 cache hit access time (in %) taken over all the benchmarks. We do not show the case of TL1-T4, as TL1-T4 increases hit latency by 200% and would obfuscate the comparison between the schemes driven by the replacement policy, if we included it.

For the TL4 scheme we see approximately a 7% average increase in hit latency over a cache with no leakage power scheme. The only scheme better than TL4 is TL2-T2 but this scheme has the least leakage savings and does not trend well. TL2-T4 results in 12% increase in hit latency and an increasing trend in wake-up penalties suggest that TL2-T4 may not scale well w.r.t. performance whereas TL4 will continue to save leakage power with little performance impact

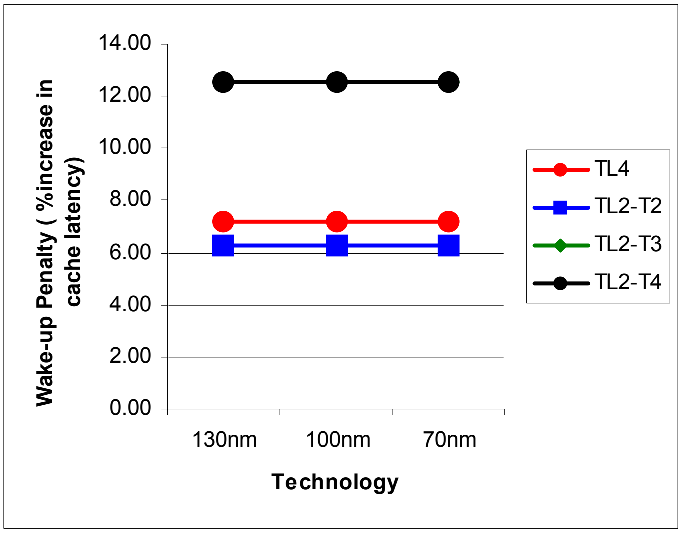

FIGURE 17. Percent increase in L1 access time for hits for various schemes under PLRU

6.2 LRU (Least Recently Used) Policy

Though LRU is considered hard to implement but some real world systems do use LRU e.g. UltraSPARC IV used LRU for L2 cache [17], hence we evaluate it in a similar way we did PLRU.

There can be many possible implementations of LRU for a 4-way set associative cache. For the sake of discussion consider an implementation that employs 2 bits per way to assign replacement priority to different lines. Initially all bits will be reset, hence at same priority level. Table 12 shows the replacement priority assignment to different bit combinations.

TABLE 12. Replacement priority assignment for LRU

Bit Combination	Replacement Priority
00	P1
01	P2
10	P3
11	P4

Table 13 shows how replacement priorities for different cache lines change corresponding to hits in different ways within a set.

TABLE 13. Cache-line replacement priorities for hits in different cache line LRU

Hit @	Line_0	Line_1	Line_2	Line_3
Line_0	P1	P2	P3	P4
Line_1	P2	P1	P3	P4
Line_3	P3	P2	P4	P1
Line_1	P3	P1	P4	P2
Line_0	P1	P2	P4	P3
Line_2	P2	P3	P1	P4
Line_3	P3	P4	P2	P1
Line_2	P3	P4	P1	P2

Figure 18 shows the priority level transitions of cache lines on an access to a set under LRU. For example, if a hit occurs at a cache line whose priority level is P3, then its priority goes to P1, the priority of the line previously at P4 will remain unchanged, the line previously at P1 goes to P2 and the line at P2 goes to P3. These priority level transitions are dictated by LRU and result in various tranquility level transitions, depending on the control scheme employed

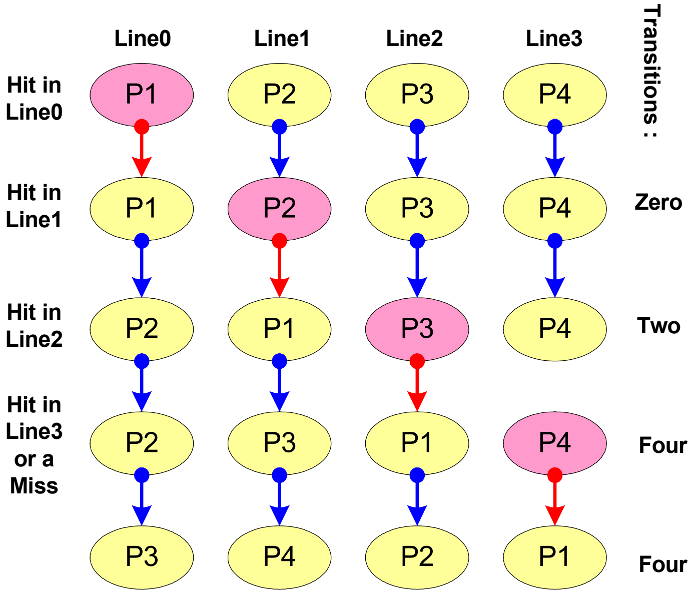

FIGURE 18. How hits at different Priority levels affect the whole cache under LRU replacement policy.

Similar to PLRU evaluations, various LRU schemes are denoted as TL2-T2, TL2-T3, TL2-T4 and TL4.

6.2.1 Dynamic Power Penalties

Figure 19 shows the dynamic power required per byte to save leakage power in L1 data cache for TL4.

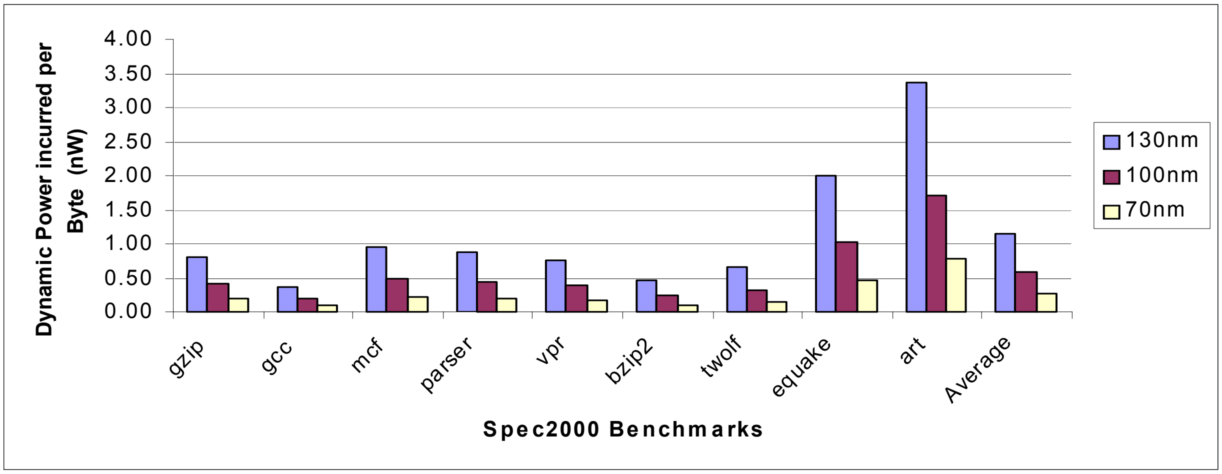

FIGURE 19. Dynamic power incurred per byte for L1 data cache for TL4 scheme under LRU policy for different benchmarks

Although Figure 19 looks very similar to Figure 10, in reality dynamic power costs are different for same benchmark for the two replacement algorithms considered.Table 14 compares TL4 schemes under LRU and PLRU with respect to dynamic power consumption. It is interesting to observe that dynamic power incurred is always less for PLRU except for art, that has miss rate of almost 50%. This confirms the intuition that more complex algorithms are in general more power hungry. In both cases dynamic power is significantly reduced as technology scales. Hence for 70nm technology both policies on the average consume the same power.

TABLE 14. Dynamic power costs for different benchmarks for TL4 schemes under PLRU and LRU

	LRU	PLRU	LRU	PLRU	LRU	PLRU
	130nm	**130nm**	**100nm**	**100nm**	**70nm**	**70nm**
gzip	0.8050	0.8027	0.4089	0.4077	0.1889	0.1883
gcc	0.3654	0.3527	0.1856	0.1791	0.0857	0.0827
mcf	0.9570	0.9411	0.4861	0.4780	0.2245	0.2208
parser	0.8727	0.7920	0.4433	0.4023	0.2047	0.1858
vpr	0.7508	0.7253	0.3814	0.3684	0.1761	0.1702
bzip2	0.4665	0.4543	0.2369	0.2307	0.1094	0.1066
twolf	0.6467	0.6182	0.3285	0.3140	0.1517	0.1450
equake	1.9952	1.9833	1.0134	1.0074	0.4681	0.4653
art	3.3741	3.4094	1.7138	1.7318	0.7916	0.7999
Average	**1.1370**	**1.1199**	**0.5775**	**0.5688**	**0.2668**	**0.2627**

In Table 15 the amount of average dynamic power consumed by various schemes to save leakage power is shown for LRU policy. These values have the same trend as in Table 11 but are a little bit more.

TABLE 15. Average dynamic power costs for various schemes under LRU

	TL1-T4	TL4	TL2-T2	TL2-T3	TL2-T4
130nm	**20.51**	**1.14**	**0.34**	**1.35**	**3.03**
100nm	**10.42**	**0.58**	**0.17**	**0.68**	**1.54**
70nm	**4.81**	**0.27**	**0.08**	**0.32**	**0.71**

Net saving for LRU are calculated similar to the way it was calculated for PLRU.

Figure 20 shows the Net Savings for different benchmarks in the case of TL4 under LRU and across various benchmarks. The observation is the same i.e. as technology scales, net power savings become independent of the benchmark because the dynamic power becomes negligible and the static power saved is independent of the benchmark.

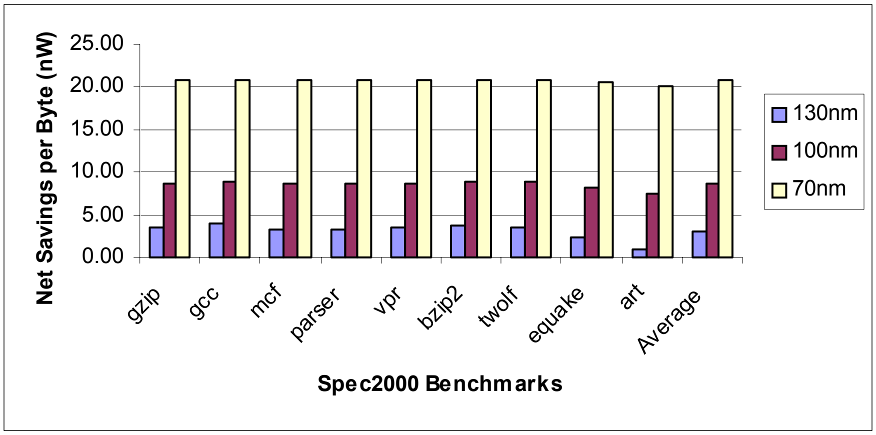

FIGURE 20. Net Savings per byte for L1 data cache for different benchmarks for TL4 under LRU policy

The % Net Savings for TL4 under RLU across different benchmarks are shown in Figure 21,. It shows that the percent savings remains steady across technologies, and also becomes independent of the benchmark as technology scales down.

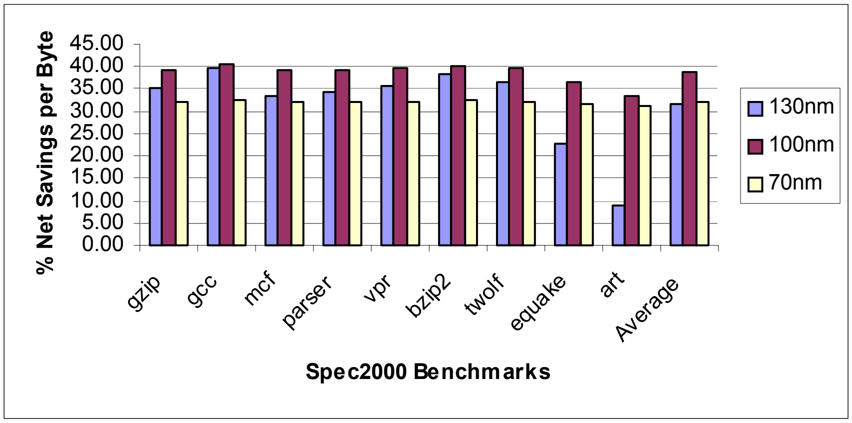

FIGURE 21. % Net Savings per byte for L1 data cache for different benchmarks for TL4 under LRU policy

Figure 22 shows the average net leakage power saved per byte across all benchmarks. Regardless of the power scheme, Net Savings increases exponentially with technology in all cases. Because of the explosive increase of leakage power and the rapid drop in dynamic power with technology scaling, the net gains obtainable by TL1-T4 (which means all lines at T4) are on a steeper upwards curve than those for any of the schemes governed by the replacement policy. It is observed that TL2-T4 gave the most savings among schemes dictated by the replacement policy and that, for the most advanced technology we have looked at, its savings are roughly equal to those of TL1-T4.

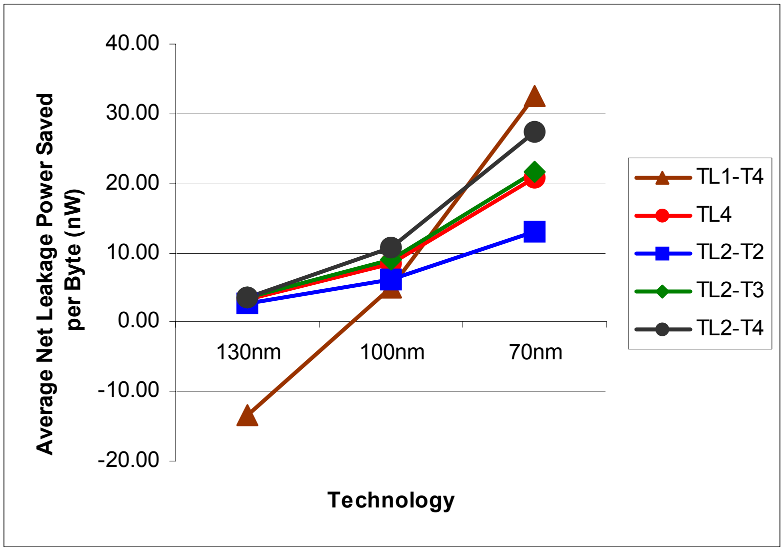

FIGURE 22. Average Net Leakage power saved per byte for L1 data cache under various schemes over future technologies under LRU policy

Figure 23 compares average net leakage power saving as percent of TL1-T4 savings of Section 4.0.

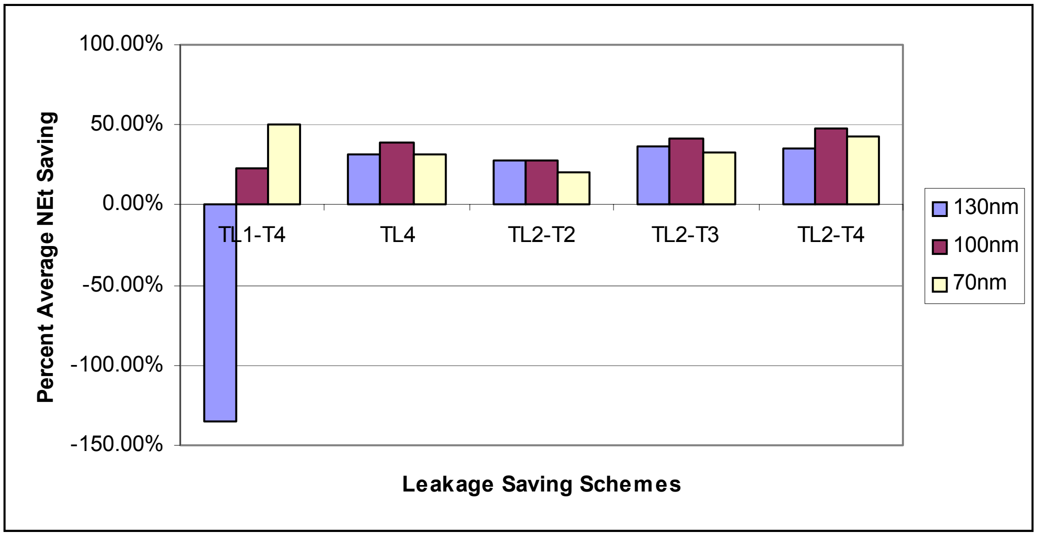

FIGURE 23. Percent Average Net Leakage power saved per byte for L1 data cachefor various schemes under LRU

6.2.2 Performance Penalties

Figure 24 shows the average increase in L1 cache hit access time (in %) taken over all the benchmarks. TL1-T4 is not shown, as TL1-T4 increases hit latency by 200% and would obfuscate the comparison between the schemes driven by the replacement policy, if we included it.

For theTL4 scheme we see approximately 8% average increase in hit latency over a cache with no leakage power scheme. The only scheme better than TL4 is TL2-T2 but this scheme has the least leakage savings and does not trend well. TL2-T4 results in 14% increase in hit latency and an increasing trend in wake-up penalties suggest that TL2-T4 may not scale well w.r.t. performance whereas LRU4 will continue to save leakage power with little performance impact.

Table 16 compares various power saving schemes for PLRU and LRU, where as performance impact means percent increase in the hit latency of L1 data cache. In all cases the performance impact of LRU is more than that of PLRU. The average miss rate, for considered benchmarks for LRU (9.40%) is less than that of PLRU (9.62%), hence this difference in performance is because LRU has 2.6% more hits in non MRU cache lines compared to PLRU.

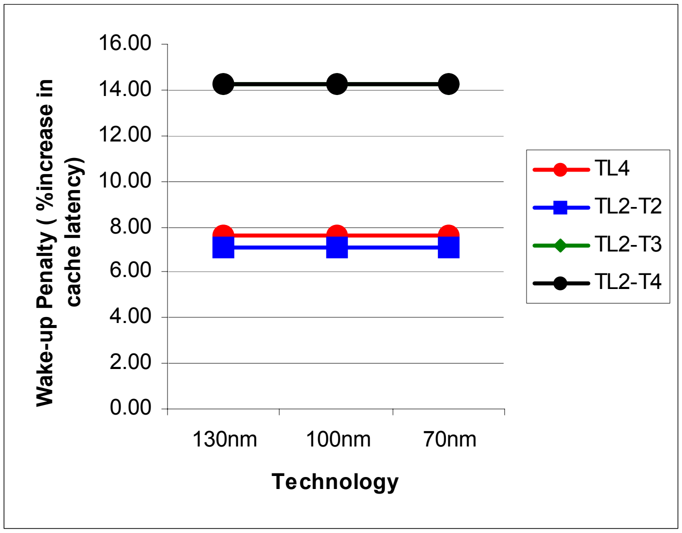

FIGURE 24. Percent increase in L1 access time for hits for various schemes under LRU

TABLE 16. Comparing performance impact of various power saving schemes under LRU and PLRU

Power Saving Scheme	Performance impact (PLRU)	Performance impact (LRU)
TL4	7.20%	7.57%
TL2-T2	6.26%	7.11%

TABLE 16. Comparing performance impact of various power saving schemes under LRU and PLRU

TL2-T3	12.52%	14.23%
TL2-T4	12.52%	14.23%

6.3 MRR (Modified Random Replacement) Policy

It was mentioned in Section 3.0 that slumberous cache idea can be applied only to the replacement algorithms that have at least two priority levels. Random replacement algorithm is used in real world systems e.g. Sun's Niagara [18], to make it fit for slumberous cache idea, MRU information is added to random replacement algorithm and it is named Modified Random Replacement. MRR has two priority levels i.e. MRU and Non-MRU. For 4-way set associative cache, three ways will be at Non-MRU priority level and one at MRU level, hence a single bit is needed to differentiate between the priority levels.

TABLE 17. Replacement priority assignment for MRR

Bit	Replacement Priority
0	P2
1	P1

Table 18 shows how replacement priorities for different cache lines change corresponding to hits in different ways within a set.

TABLE 18. Cache-line replacement priorities for hits in different cache line under MRR

Hit @	Line_0	Line_1	Line_2	Line_3
Line_0	P1	P2	P2	P2
Line_1	P2	P1	P2	P2
Line_3	P2	P2	P2	P1
Line_3	P2	P2	P2	P1
Line_0	P1	P2	P2	P2
Line_2	P2	P2	P1	P2
Line_3	P2	P2	P2	P1
Line_2	P2	P2	P1	P2

Figure 25 shows the priority level transitions of cache lines on an access to a set under MRR. For example, if a hit occurs at a cache line whose priority level is P2, then its priority goes to P1, the priority of the line previously at P1 goes to P2. These priority level transitions are dictated by MRR and result in various tranquility level transitions, depending on the control scheme employed.

MRR performance compared to LRU and PLRU is shown in Table 19. As far as IPC is considered, MRR is a little better than PLRU and a little worse than LRU.

TABLE 19. Comparing IPCs of various replacement algorithms

	IPC		
	MRR	PLRU	LRU
gzip	2.02	1.99	2.04
gcc	0.53	0.52	0.52
mcf	0.86	0.86	0.87
parser	2.06	2.08	2.12
vpr	0.90	0.90	0.91
bzip2	0.68	0.67	0.68
twolf	1.45	1.38	1.47
equake	0.99	1.00	0.99
art	2.00	1.43	1.44
Average	1.19	1.17	1.20

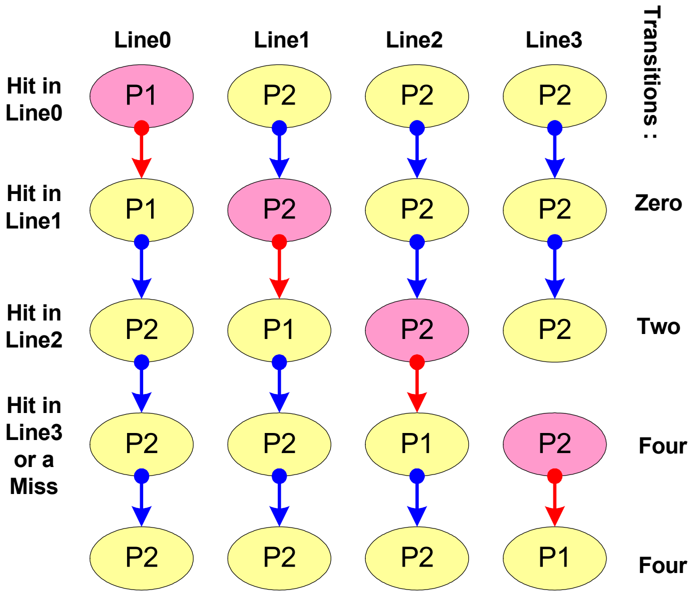

FIGURE 25. How hits at different Priority levels affect the whole cache under MRR replacement policy.

TL4 is not possible for MRR, so similar to the evaluations for PLRU and LRU, schemes TL2-T2, TL2T3 and TL2-T4 are considered.

6.3.1 Dynamic Power Penalties

Figure 26 shows the dynamic power required per byte to save leakage power in L1 data cache for TL2-T4 under

MRR.

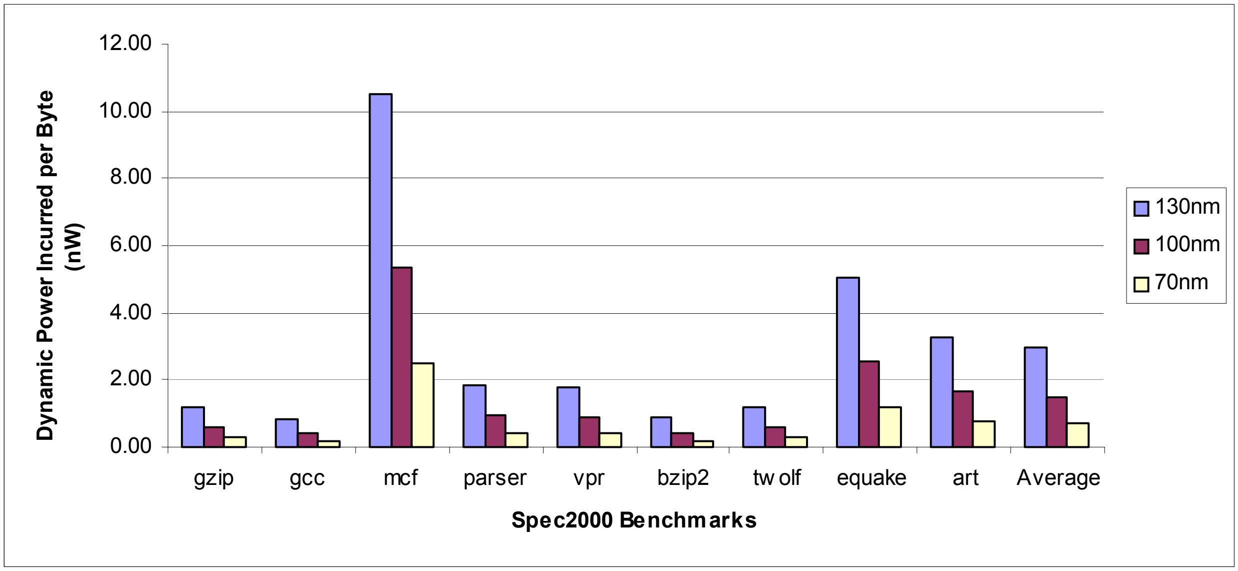

FIGURE 26. Dynamic power incurred per byte for L1 data cache for TL2-T4 under MRR policy for different benchmarks

Table 20 compares TL2-T4 scheme for PLRU, LRU and MRR replacement policies with respect to dynamic power consumption. It is interesting to observe that in all cases MRR is equally good as PLRU. LRU being more complex to implement is more power hungry. In all cases dynamic power is significantly reduced as technology scales. Hence for 70nm technology all cases incur almost the same dynamic power cost.

TABLE 20. Dynamic power costs for different benchmarks for TL2-T4 scheme under PLRU, LRU and MRR

	PLRU	**LRU**	**MRR**	**PLRU**	**LRU**	**MRR**
	130nm	**130nm**	**130nm**	**70nm**	**70nm**	**70nm**
gzip	1.19	1.26	1.19	0.28	0.29	0.28
gcc	0.85	0.87	0.85	0.20	0.20	0.20
mcf	10.53	10.74	10.53	2.47	2.52	2.47
parser	1.86	2.05	1.86	0.44	0.48	0.44
vpr	1.78	1.82	1.78	0.42	0.43	0.42
bzip2	0.86	0.89	0.86	0.20	0.21	0.20
twolf	1.20	1.26	1.20	0.28	0.30	0.28
equake	5.05	5.16	5.05	1.18	1.21	1.18
art	3.28	3.25	3.28	0.77	0.76	0.77
Average	2.96	3.03	2.96	0.69	0.71	0.69

TABLE 21. Average dynamic power costs for various schemes under MRR

	TL1-T4	TL2-T2	TL2-T3	TL2-T4
130nm	20.51	0.33	1.31	2.96
100nm	10.42	0.17	0.67	1.50
70nm	4.81	0.08	0.31	0.69

In Table 21 the amount of average dynamic power consumed by various schemes under MRR policy is shown. Net savings for MRR are calculated like LRU and PLRU.

Figure 27 shows Net Savings for different benchmarks in the case of TL2R-T4 under MRR. The observation is the same i.e. as technology scales, net power savings become independent of the benchmark because the dynamic power becomes negligible and the static power saved is independent of the benchmark.

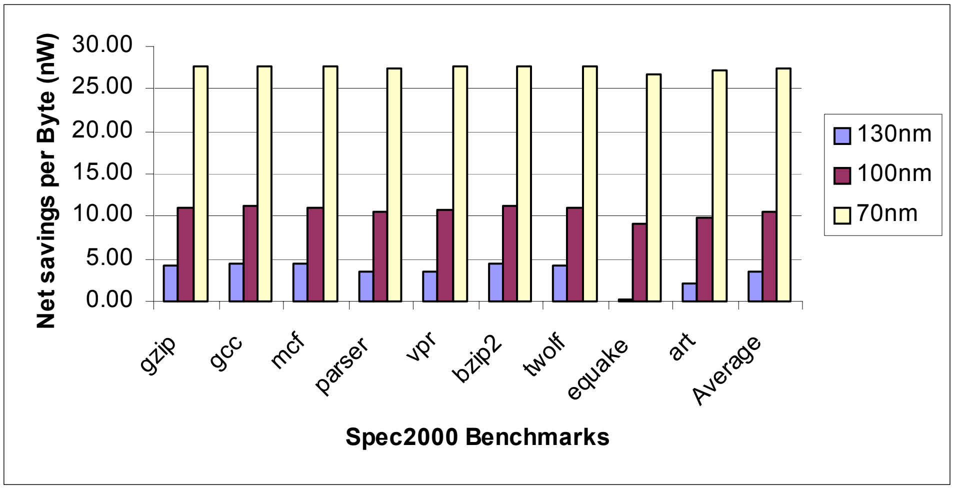

FIGURE 27. Net Savings per byte for L1 data cache for different benchmarks for TL2-T4 under MRR policy

The % Net Savings for TL2-T4 across different benchmarks are shown in Figure 28,. It shows that the percent savings remains steady across technologies, and also becomes independent of the benchmark as technology scales down.

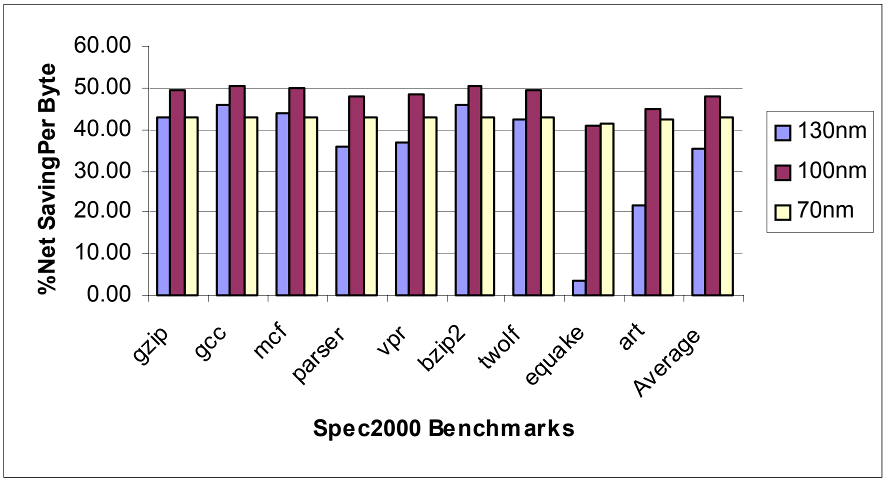

FIGURE 28. % Net Savings per byte for L1 data cache for different benchmarks for TL2-T4 scheme under MRR

Figure 29 shows the average net leakage power saved per byte across all benchmarks. Regardless of the power scheme, Net Savings increases exponentially with technology in all cases. Because of the explosive increase of leakage power and the rapid drop in dynamic power with technology scaling, the net gains obtainable by TL1-T4 (which means all lines at T4) are on a steeper upwards curve than those for any of the schemes governed by the replacement policy. It is observed that TL2-T4 gave the most savings among schemes dictated by the replacement policy and that, for the most advanced technology we have looked at, its savings are roughly equal to those of TL1-T4.

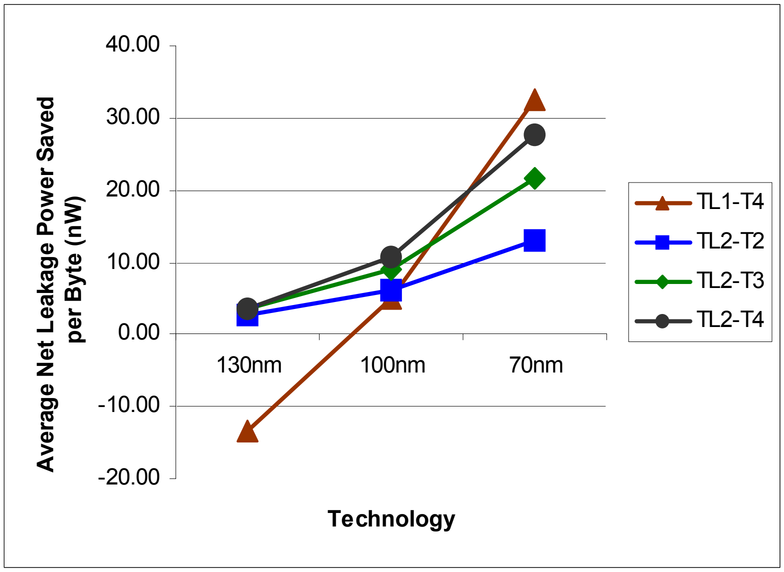

FIGURE 29. Average Net Leakage power saved per byte for L1 data cache under various schemes over future technologies under MRR policy

Finaly average net leakage power saving is compared to the maximum saving compouted in Section 4.0, all savings are shown as percent of TL1-T4 savings of Section 4.0 refer Figure 30.

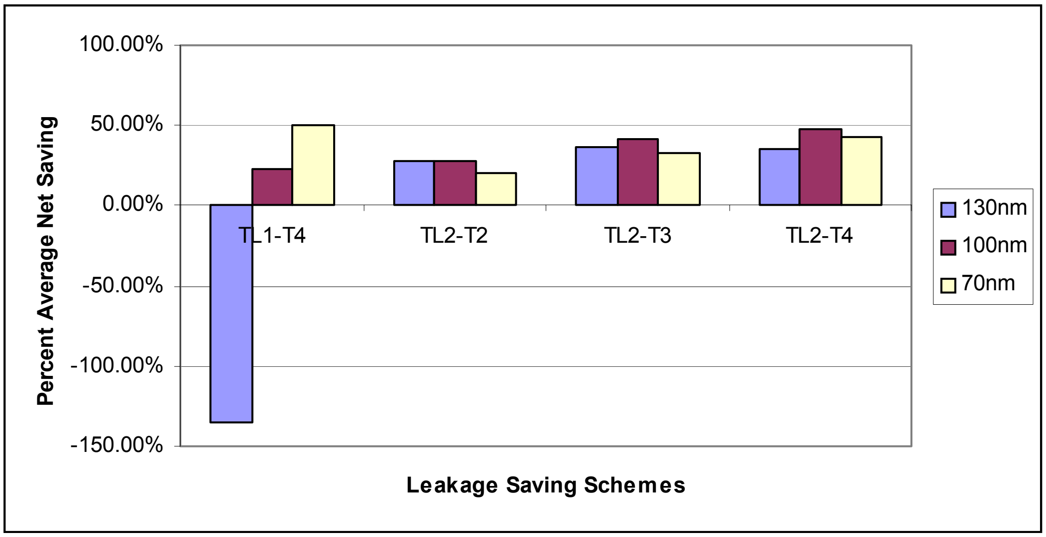

FIGURE 30. Percent Average Net Leakage power saved per byte for L1 data cachefor various schemes under MRR

6.3.2 Performance Penalties

Figure 31 shows the average increase in L1 cache hit access time (in %) taken over all the benchmarks. TL1-T4 is not shown, as TL1-T4 increases hit latency by 200% and would obfuscate the comparison between the schemes driven by the replacement policy, if we included it.

For the TL2-T4 scheme we see approximately 13% average increase in hit latency over a cache with no leakage power scheme. TL2-T2 and TL2-T3 both have the same performance impact of approximately 6%.

Table 22 compares various PLRU, LRU and MRR power saving schemes, where as performance impact means percent increase in the hit latency of L1 data cache. In all cases the performance impact of LRU is the maximum where as that of MRR is minimum.

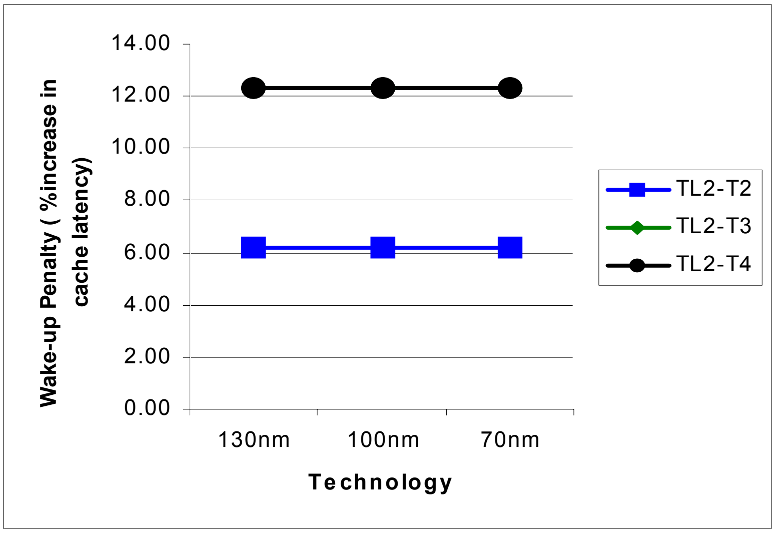

FIGURE 31. Percent increase in L1 access time for hits for various schemes under MRR

TABLE 22. Comparing performance impact of various power saving schemes under PLRU, LRU and MRR

Power Saving Scheme	Performance impact (PLRU)	Performance impact (LRU)	Performance impact (MRR)
TL4	7.20%	7.57%	NA
TL2-T2	6.26%	7.11%	6.17%
TL2-T3	12.52%	14.23%	12.33%
TL2-T4	12.52%	14.23%	12.33%

7.0 Best Leakage Control Scheme for L1 data Cache

So far 12 leakage control schemes based on the replacement algorithm were discussed. To find the best out of these a metric is devised called net leakage saving per percent increase in the hit latency of the L1 data cache. This metric will be referred to as **L**eakage **E**nergy **S**aving **M**etric (**LESM**) in the rest of the paper. As per Table 23 PLRU4 is the best scheme for all technologies considered. For 130nm technology PTL2-T2 and MRR-T2 are as good as PLRU4. PLRU4 is slumberous scheme in real sense, whereas others are drowsy compatible versions, using replacement policies. Hence slumberous caches are proved to be better than drowsy caches as technology scales down.

TABLE 23. Comparing all 12 leakage control schemes with respect to LESMs

	LESM		
	130nm	100nm	70nm
PLRU			
TL4	0.44	1.19	2.87
TL2-T2	0.44	0.99	2.10
TL2-T3	0.29	0.73	1.72
TL2-T4	0.28	0.85	2.20
LRU			
TL4	0.41	1.13	2.73
TL2-T2	0.38	0.87	1.85
TL2-T3	0.25	0.64	1.51
TL2-T4	0.24	0.75	1.93
MRR			
TL2-T2	0.44	1.00	2.13
TL2-T3	0.29	0.74	1.74
TL2-T4	0.28	0.86	2.23

8.0 Extension to Unified L2 Cache

To evaluate the leakage power savings schemes in the context of a unified L2 cache, a 4-way set-associative cache with PLRU replacement policy is considered. The block size is 64 bytes and cache size is 1Mbyte. Since the L2 cache is not accessed as frequently as L1 the dynamic power expanded in switching between the tranquility levels becomes negligible. Average dynamic power incurred for unified L2 cache is in the order of pWs per byte whereas, for L1 data caches, it was in the order of nWs. Thus dynamic power costs are negligible across all schemes, as show in Figure 32.When we compare net leakage power savings per byte in L2 for different schemes we see that TL1-T4 is always the best refer Figure 33 and is also trending up faster with technology.

L2 cache hit latency of the CPU model used is 20 cycles, so even if we put the whole L2 cache at T4 level at all times we will have only 10% increase in L2 latency. Impact on L2 cache latency has similar curve as that of L1 but is less than 2% in all cases refer Figure 34.

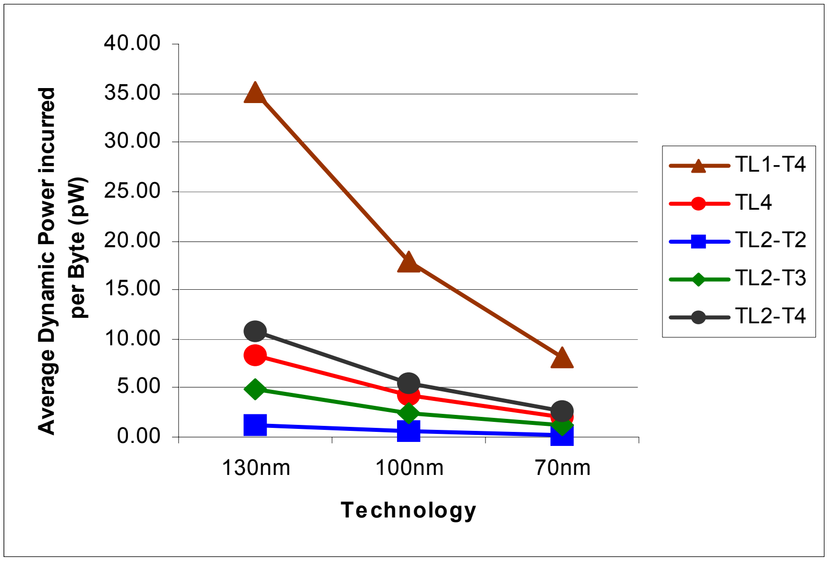

FIGURE 32. Average dynamic power incurred per byte for unified L2 cache for various power saving schemes over future technologies under PLRU

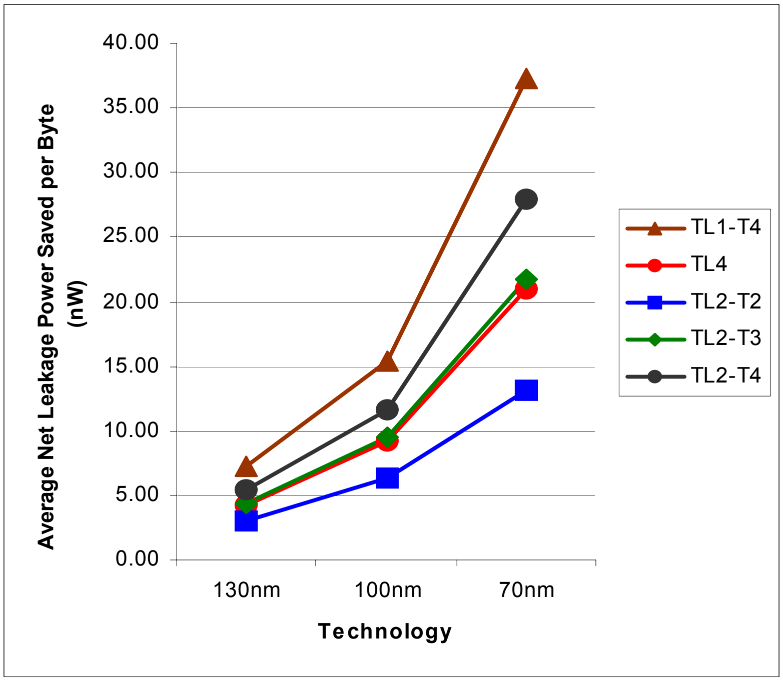

FIGURE 33. Average Net leakage power saved for unified L2 cache for various power saving schemes over future technologies under PLRU

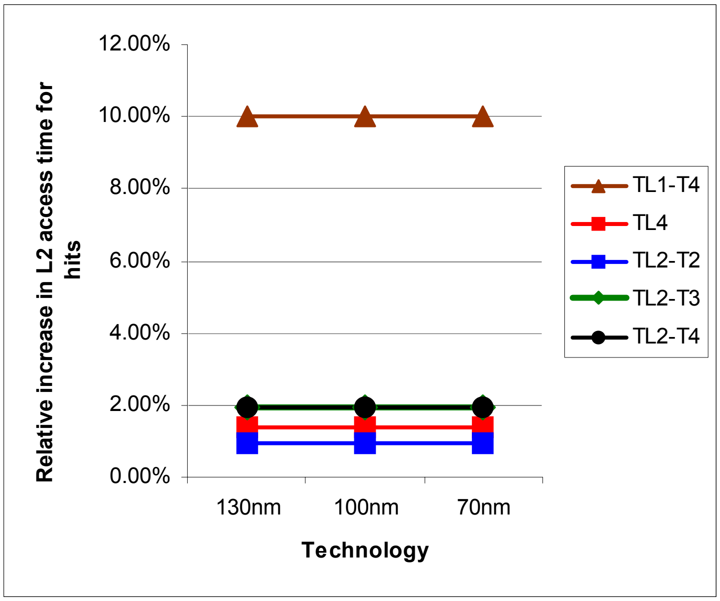

FIGURE 34. Percent increase in L2 access time for hits for various schemes under PLRU

9.0 Immunity to Soft Errors and Reliability of Slumberous Caches

Soft errors will be a main concern in future microprocessors [19] owing to miniature feature sizes and larger chip areas. As cache memories are occupying most of the chip's real state today, they have to be more reliable. Reducing voltage of cache lines makes them more vulnerable to soft error attacks, as mentioned by [21] soft error vulnerability increases exponentially with decreasing supply voltage. Further as pointed out by [20] MRU lines are more vulnerable to soft errors. Hence slumberous cache schemes that never put MRU lines into drowsy mode and gradually decrease the voltage of a cache line as its replacement priority is lowered, seem more promising keeping in view power, performance and reliability. A detailed evaluation of reliability of slumberous caches is beyond the scope of this paper.

10.0 Conclusions

From this paper it is established that huge leakage energy can be saved in future technologies, if some tranquility levels for the caches are selected and individual cache lines are switched to a tranquility level proportional to their frequency of utilization. Replacement policy can be used to discriminate between more frequently used and less frequently used cache lines to decide about which power save level they should be switched to. Our experimental results proved that on the average 45%-32% leakage power can be saved for 130nm-70nm technologies. Dynamic energy cost to implement the proposed scheme becomes negligible as technology scales and also as cache sizes increases for a fixed technology. So above mentioned percent savings are independent of the program execution and cache size used. The performance effect of this scheme is very less and it decreases towards no performance impact as technology scales.

Two priority levels schemes were considered to contrast with drowsy caches. Our scheme is similar to drowsy cache scheme in the way that we also reduce supply voltage to different cache lines. But drowsy cache scheme puts the entire cache to drowsy mode at some regular intervals in case of L1 data cache and for L1 instruction cache they introduced some way of bank prediction to put entire bank into drowsy mode. To mitigate the performance impact we never reduce the supply voltage of P1 priority level cache lines we only put P2-P4 levels to either multiple levels of tranquility in case of TL4 or to two levels of tranquility as the case of many two levels schemes discussed. For two level schemes different cases of assigning T2 or T3 or T4 voltage level to all three priority levels from P2-P4 were considered. Comparing all 12 schemes for L1 data cache, with respect to LESMs, showed TL4 under PLRU to be the best scheme, which also proves superiority of slumberous scheme over drowsy type schemes. On account of very less dynamic cost and very less performance impact TL1-T4 seems to be the best case for L2 caches i.e put whole L2 cache in deepest tranquility level and wake a cache line up only when needed and put them back to sleep in the very next cycle,

Another important thing to mention is that though percent leakage energy savings decrease as technology scales on account of decreasing voltage difference between different tranquility levels the absolute leakage energy saving increase 2-4 times (depending on the cache size and replacement policy) from 130nm to 70nm technology.

Although the replacement policy can very easily be used to decide that which cache line to be switched to which low power mode, but it is blind towards the age of any particular cache line, and is not completely power aware as it does not take care of the fact that for how long a cache line is inactive. The idea proposed by [10] at compiler level can be implemented by using some global and local counters to turn off a cache line that is not used for a certain minimum amount of time even for P1 priority level. The idea of slumberous caches with multi levels of tranquility can be combined with invalidation and turning off schemes to save a little bit more from total leakage power. Or even we can combine our idea with the drowsy cache idea of putting all cache lines to deepest tranquility level at some regular interval of time but once the priority of a cache line is reduced with aging effect we also reduce its voltage level in steps i.e. from T1 to T2 and from T2 to T3 and so on.

11.0 References

[1] S. Kaxiras, Z. Hu, and M. Martonosi. Cache decay: exploiting generational behavior to reduce cache leakage power. In ISCA 28, pages 240-251, May 2001.

[2] Berkeley predictive model. http://www-device.eecs.berkeley.edu

[3] K. Flautner, N. Kim, S. Martin, D. Blaauw, T. Mudge. Drowsy Caches: Simple techniques for reducing leakage power. In Proc. the 29th International Symposium on Computer Architecture, Anchorage, AK, May 2002.

[4] NS Kim et al., Drowsy Instruction Caches: Leakage Power Reduction using Dynamic Voltage Scaling and Cache Sub-bank Prediction”, Proc. Micro 2002

[5] Se-Hyun Yang, Michael D. Powell, Babak Falsafi, and T. N. Vijaykumar: Exploiting Choice in Resizable Cache Design to Optimize Deep-Submicron Processor Energy-Delay.In Eighth International Symposium on High-Performance Computer Architecture (HPCA'02) February 02 - 06, 2002

[6] D. Burger and T. Austin. The SimpleScalar tool set, version 2.0. Technical Report CS-TR-97-1342, University of Wisconsin, Madison, June 1997.

[7] J. A. Butts and G. S. Sohi. A static power model for architects. In MICRO-33, pages 191-201, December 2000.

[8] The Standard Performance Evaluation Corporation. WWW Site. http://www.spec.org, Dec. 2000.

[9] W. Zhang et al. Compiler-directed instruction cache leakage optimization. In Proc. the 35th Annual International Symposium on Microarchitecture, November 2002.

[10] H. Hanson et al. Static energy reduction techniques for microprocessor caches. In Proc. ICCD 2001, pages 276-83, Sept. 2001.

[11] The international technology roadmap for semiconductors. Semiconductor Industry Association, 2002. http://public.itrs.net/Files/2002Update/2002Update.htm

[12] Steven Dropsho, Volkan Kursun, David H. Albonesi, Sandhya Dwarkadas, Eby G. Friedman Managing Static Leakage Energy in Microprocessor Functional Units

[13] T. Sherwood, E. Perelman, G. Hammerley, and B. Calder. Automatically characterizing large-scale program behavior. In Proceedings of the International Conference on 10th International Conference on Architectural Support for Programming Languages and Operating Systems, Oct. 2002.

[14] V. Agarwal, M. S. Hrishikesh, S. W. Keckler, and D. Burger. Clock rate versus IPC: The end of the road for conventional microarchitectures. In Proceedings of the 27th Annual International Symposium on Computer Architecture, pages 248- 259, June 2000

[15] H. Hanson et al. Static energy reduction techniques for microprocessor caches. In Proc. ICCD 2001, pages 276–83, Sept. 2001.

[16] M. D. Powell et al. Gated-Vdd: A Circuit Technique to ReduceLeakage in Deep- Submicron Cache Memories. In ISLPED, 2000.

[17] http://www.sun.com/processors/whitepapers/us4_whitepaper.pdf.

[18] Poonacha Kongetira, Kathirgamar Aingaran,Kunle Olukotun, "Niagara: A 32-way Multithreaded Sparc Processor" Hot Chips 2005

[19] Wenbin Yao, Dongsheng Wang and Weimin Zheng, "A Fault-Tolerant Single-Chip Multiprocessor", ACSAC 2004.

[20] S.Kim and A.K.Somani, "Area Efficient Architectures for Information Integrity in Cache Memories," Proceedings of the 26th Annual International Symposium on Computer Architecture (ISCA), pages 246 – 255, May, 1999.

[21] V. Degalahal, N. Vijaykrishnan, and M. J. Irwin, "Analyzing soft errors in leakage optimized sram designs," in Sixteenth International Conference on VLSI Design, Jan. 2003.