

WARM SRAM: A Novel Scheme to Reduce Static Leakage Energy in SRAM Arrays

Mahadevan Gomathisankaran^{1,*} and Akhilesh Tyagi²

¹Intel India, #28 Cubbon Road, Bangalore 560003, India

²Department of ECpE, Iowa State University, Ames, IA 50011, USA

(Received: 14 June 2006; Accepted: 26 October 2006)

The increasing sub-threshold leakage current levels with newer technology nodes has been identified by ITRS¹ as one of the major fundamental problems faced by the semiconductor industry. Concurrently, the expected performance improvement and functionality integration expectations drive the continued reduction in feature size. This results in ever-increasing power per unit area and the accompanying problem of heat removal and cooling.² Portable battery-powered applications, fuelled by pervasive and embedded computing, in the last few years have seen a tremendous growth and have reached a point where battery power can't be increased further.³ This raises the computational throughput per watt target for the future technology nodes. SRAM arrays which are used widely as a system component, such as caches and register files, in both high-performance and portable systems, are getting to be dominant power consumers because of their large capacity and area. Hence any reduction in cache energy can result in considerable overall power reduction. In this paper, we propose a novel circuit technique using depletion mode devices, to reduce the static energy of SRAM array in an on-chip cache by 90% without any performance impact.

Keywords: Leakage Energy, SRAM, On-Chip Cache.

1. INTRODUCTION

Static subthreshold leakage has emerged as one of the major impediments in CMOS scaling. The magnitude of the problem is reflected in the fact that the leakage current per unit transistor width is expected to increase a 1000–10000 fold in going from 180 nm technology node to 70 nm technology node. This appears to pose a problem only for state holding circuits such as memory arrays that are idle for extended periods until one considers the following. In the same period, 180 nm to 70 nm technology node, the on (drive) current is expected to stay constant at 750 $\mu\text{A}/\mu\text{m}$ in order to sustain the historical speed advantage. In the 180 nm technology node, the ratio of on current to leakage current per unit transistor width is approximately 7.5×10^6 . This ratio is projected to shrink to 750 in 70 nm technology node and to 75 in 32 nm technology node! Some of the implications of this projection are as follows. In 32 nm node, the leakage current is expected to play a major role in the design of some of the high fanout, computing logic as well. Consider a bus with 30 clients being driven by one client. The twenty-nine *off*

transistors (drivers) from the inactive bus clients are able to leak about one half (38%) of the *on* drive current of the driving client! Similar concerns will surface for logic blocks such as decoders, multiplexors, and gate arrays. This argues that we need to develop low-leakage design styles even for computing logic with high activity rate.

This paper proposes a CMOS design style with low leakage characteristics. This design style belongs to a family of CMOS design styles that we name *warmup CMOS*. These logic styles warmup initially to charge some nodes up to a certain potential. The steady state energy savings are delivered in the “warmed-up” state with an initial energy cost for the warmup. The specific warmup CMOS design style presented in this paper relies upon depletion mode transistors to provide proper biasing to reduce the leakage. Hence we name it dep-warmup-CMOS. During the active phase of logic, the depletion mode transistors are transparent (Fig. 1). However, during the inactive phase, the depletion mode transistors leak enough charge to bias the transistors to a point favorable with respect to the leakage current. A low leakage current equilibrium state is forced in the inactive phase. We describe the dep-warmup CMOS design style in Section 2.

An immediate application of dep-warmup CMOS logic design style is in static memory arrays characterized

*Author to whom correspondence should be addressed.
 Email: mahadevan.gomathisankaran@intel.com

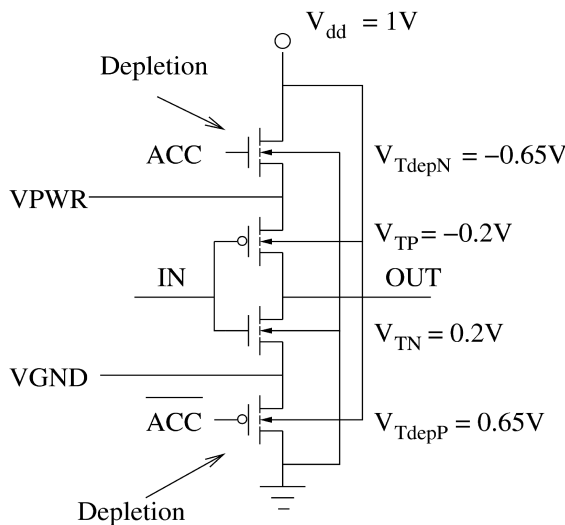


Fig. 1. Circuit with depletion device.

by caches and register files. Microprocessors attain significant performance improvement by increasing the size and associativity of on-chip caches. For example, Intel’s latest processor family, Centrino¹⁹ has a 1 MB L2 cache on-chip. Both *dynamic* switching energy, and *static* subthreshold leakage current induced energy of on-chip caches are already significant factors in over-all power consumption of the processors. The static leakage energy would overwhelm the dynamic energy for these caches with the expected 1000–10000 fold increase in leakage current in the 70 nm technology node.

This is why the static leakage power reduction is one of the most important considerations in both high-performance and low standby power circuits’ design. As the device size shrinks, to maintain the same *on* current, threshold voltage V_t has to be reduced. This causes subthreshold current to increase exponentially. Table I compares 0.18 μm technology and 70 nm technology with respect to *on* and *off* currents. This illustrates the magnitude of the problem.

As we had argued earlier, with feature size reduction, the static energy component of the on-chip caches constitutes a sizeable fraction total processor energy. Circuit techniques such as DVS,¹⁴ ABB-MTCMOS,¹⁰ Gated- V_{dd} ,¹¹ have been proposed to reduce the leakage energy in the caches. Control algorithms deploying these techniques estimate the active footprint in the cache to selectively power it. This results in performance penalty as well as in energy penalty in switching the cache lines back and forth from active to dormant state.

Gated- V_{dd} ¹¹ technique interposes a high-threshold transistor between the circuit and one of the power supply

Table I. Leakage current trend.

Technology	W_{min} (nm)	L_{eff} (nm)	V_t (V)	I_{off} (nA/ μm)
TSMC [7] 0.18 μm	200	100	0.35	0.05
BPTM [4] 70 nm	70	38	0.20	53.96

rails. Typically a gated- V_{dd} SRAM cell will have an NMOS selectively connecting the cell to the ground rail. When the *active* signal is asserted, SRAM cell operates normally, but when *active* is de-asserted, the cell is disconnected from the ground, and the state contained within the cell is lost. The activation transistor and the control mechanism for *active* can be shared by all cells within a cache line to minimize the extra area needed by the control transistor. Thus this technique reduces the leakage current of a normal threshold transistor to effectively the leakage current of the high-threshold control transistor. For 0.18 μm technology with V_{dd} as 1 V and *control* transistor with V_{dd} as 0.4 V, 97% of the static leakage energy could be saved. But the only and the main disadvantage in this method is that the contents of the cache are lost. Hence the control algorithm should be smart enough to switch off the lines only when not needed.

In ABB-MTCMOS,¹⁰ the static-leakage current is reduced by dynamically raising the transistor threshold voltage, typically by modulating the back-gate bias voltage. In a typical circuit employing ABB-MTCMOS, during normal operation, when *active* is asserted, the SRAM is connected to V_{dd} and ground and back-gate voltages are set to the appropriate power rails. When *active* is de-asserted the PMOS wells are biased using an alternate power supply, V_{dd+} , at a higher voltage level than the source terminals. And NMOS transistor wells are grounded while their sources are increased to higher voltage level through two diode drops. Thus all the transistors in SRAM experience higher threshold voltages and corresponding drop in leakage current. The drawbacks of this method are, higher energy/delay per transition (due to higher capacitance switching of power lines, both V_{dd} and gnd, N-Wells, and P-Wells consume more energy and takes more time), higher V_{dd} offsets the leakage power savings.

DVS¹⁴ scales the voltage of (power line) cell to approximately 1.5 times V_{dd} in standby mode. The state of the memory cell is also maintained at this voltage level, hence the state is not lost. In a typical sub-micron process (70 nm) leakage current increases exponentially with supply voltage. Hence by reducing the supply voltage to an optimum value (knee point of the curve), two-fold reduction (both voltage and current) of the leakage power is achieved. But the memory cell in standby mode cannot be accessed (read or written). Hence, access latency increases whenever a memory cell in *drowsy* state is to be accessed. The capacitance of power rail is considerably less than the capacitance of N-wells and P-wells. A transition from *drowsy* to normal state is therefore faster when compared with ABB-MTCMOS. Another drawback of this method is reduction in the ‘noise margin’ of the SRAM cell inverters. The voltage at which an inverter transits its state is given by,

$$V_M = (r(V_{dd} - |V_{tp}|) + V_{tn}) / (1 + r)$$

where, $r = \sqrt{\beta_p / \beta_n}$

Hence the V_M moves very close to one of the two states, making the SRAM very susceptible to noise. With this DVS scheme and a simple policy of switching all cache lines in L1 to *drowsy* mode every 2000–4000 cycles, achieves, on average, 71% static energy reduction. This translates into 50% overall cache energy reduction for 70 nm process. This clearly indicates that the static energy forms a considerable fraction of overall cache energy.

Power consumption in any digital integrated circuit, is given by the equation,

$$P_{\text{total}} = I_o V_{\text{dd}} + \alpha C V_{\text{dd}}^2 f \quad (1)$$

where, (I_o) is the leakage current, which is governed by the diode equation $I_s(e^{qV/kT} - 1)$, V_{dd} is the power supply voltage, α is the average switching activity factor, C is the total capacitance of the circuit, and f is the frequency of operation. The first term in the equation corresponds to the leakage power and the second term corresponds to the dynamic switching power. With the reduction in feature sizes, V_{dd} has also decreased, forcing a reduction in the threshold voltage V_t of the transistors. Thus the leakage current (I_o) which depends on V_t , through the diode equation (presented in the preceding discussion), increases.¹⁵

A more elaborate expression for the sub-threshold leakage current is given by [Ref. [16], p. 201],

$$I_{\text{sub}} = A \exp\left(\frac{q}{n'kT}(V_g - V_s - V_{\text{th0}} - \gamma'V_s + \eta V_{\text{ds}})\right) B \quad (2)$$

where,

$$A = \mu_0 C_{\text{ox}} \frac{W_{\text{eff}}}{L_{\text{eff}}} \left(\frac{kT}{q}\right)^2 e^{1.8}$$

$$B = 1 - \exp\left(\frac{-qV_{\text{ds}}}{kT}\right)$$

The leakage power decreases exponentially with respect to V_{ds} due to the Drain Induced Barrier Leakage effect.¹⁶ This fact has been used by earlier researchers in DVS¹⁴ and in Row-by-Row Dynamic V_{dd} Control (RRDV) Scheme.¹⁸ We propose a novel technique using *depletion mode devices* to achieve a leakage reduction of 90% in SRAM arrays without any additional control mechanism. We call our SRAM as *warm sram* for the reasons explained in Section 3.

2. PROPOSED CIRCUIT TECHNIQUE

2.1. Circuit Operation

In order to achieve leakage reduction we should be able to dynamically control the voltages V_{gs} , V_{ds} , and V_s . We use depletion mode devices to achieve this. A depletion device works in same way as the enhancement mode device, except that the device is *ON* even when the V_{gs} is zero. To understand how depletion mode devices can

help in leakage current reduction, consider Figure 1. Let, V_{TdepN} , V_{TdepP} , V_{TN} , and V_{TP} be the threshold voltages of the devices, such that

$$V_{\text{TdepN}} < 0, V_{\text{TN}}; \quad V_{\text{TdepP}} > 0, V_{\text{TP}}$$

When the access signal *ACC* is *HIGH*, both the depletion devices are *ON* hence the virtual power and ground nodes $V_{\text{PWR}} = V_{\text{dd}}$ and $V_{\text{GND}} = \text{GND}$. The circuit is in normal operation mode. When *ACC* is made *LOW* both depletion mode devices are switched-off and their V_{gs} is heavily reverse biased, i.e., for depletion NMOS $V_{\text{gs}} = -V_{\text{dd}}$ and for Depletion PMOS $V_{\text{gs}} = V_{\text{dd}}$. With $V_{\text{ds}} = 0$ both of these depletion mode devices will be in deep sub-threshold region, hence will have very minimal leakage. One of the two enhancement mode devices will be leaking depending on the *OUT* state. Hence the charge stored in V_{PWR} node starts getting accumulated in the node V_{GND} . As V_{PWR} decreases, (and simultaneously V_{GND} increases), the depletion mode NMOS (and PMOS) current increases. Note that the leakage of charge from V_{PWR} to V_{GND} reduces the V_{ds} and increases the V_{sb} for the enhancement mode devices. The leakage current in the enhancement mode device reduces exponentially as the V_{sb} reduces, V_{gs} is reverse biased and the V_{sb} increases. Thus all the three factors in Eq. (2) come in to play and reduce the leakage current dramatically. The circuit reaches a stable point when the current supplied by depletion mode device is equal to the leakage current of the enhancement devices. If the threshold voltages are carefully chosen this equilibrium can be achieved at deep sub-threshold region of depletion mode device, hence reducing the leakage current in the circuit dramatically.

Intuitively, we can see that this equilibrium is reached when V_{PWR} is closer to $-V_{\text{TdepN}}$ and V_{GND} is closer to and $V_{\text{dd}} - V_{\text{TdepP}}$. Thus V_{ds} (consider *OUT* as *HIGH* and hence $V_{\text{out}} = V_{\text{PWR}}$) across the NMOS enhancement transistor will be $V_{\text{TdepP}} - V_{\text{TdepN}} - V_{\text{dd}}$. In order to differentiate $V_{\text{out}}^{\text{high}}$ from $V_{\text{out}}^{\text{high}}$, this $V_{\text{ds}} = V_{\text{TdepP}} - V_{\text{TdepN}} - V_{\text{dd}} > 0$, resulting in $V_{\text{TdepN}} - V_{\text{TdepP}} - V_{\text{dd}}$. The equilibrium condition of the circuit can be found by solving Eq. (2), substituting appropriate threshold voltages and equating all the currents. To have at-least a voltage difference of 0.3 V between *HIGH* and *LOW* and for $V_{\text{dd}} = 1$ V, the above equilibrium condition requires (for the case $|V_{\text{TdepN}}| = |V_{\text{TdepP}}|$), $|V_{\text{TdepN}}| = 0.65$ V. We used 70 nm technology model files provided by BPTM⁴ to perform the HSPICE⁵ simulation. We chose the following threshold voltages for the HSPICE simulation: $|V_{\text{TN}}| = |V_{\text{TP}}| = 0.2$ V; $|V_{\text{TdepN}}| = |V_{\text{TdepP}}| = 0.65$ V; $V_{\text{dd}} = 1$ V. The steady state currents and voltages of different nodes are as given in Table II. In these measurements, the depletion mode transistors were opened (*ACC* made 1) for 1 ns and then closed. All the three signals, i.e., *IN*, *ACC* and were driven by minimum sized inverters. Hence, they all have the same delay, rise and fall times.

Table II. Steady state response of circuit in Figure 1.

IN (V)	OUT (V)	V_{PWR} (V)	V_{GND} (V)	I_{off} (pA)
0.0	0.949	0.949	0.148	10
1.0	0.052	0.852	0.052	01

2.2. Leakage Reduction and Performance Impact

Table I shows the leakage current of a minimum sized transistor to be $(53.96 \text{ nA}/\mu\text{m}) \times 70 \text{ nm}$ which is 3.77 nA . Compared to 10 pA leakage current from our scheme in Table II, we have achieved a leakage current reduction of 377 times, but with a penalty in performance. The performance impact of the extra NMOS access transistor in the charging path and PMOS access transistor in the discharging path could be high. Since these devices are depletion mode devices, this impact can be managed to a great extent. Various delay parameters are listed in Table III. There is a 54.5% increase in the average propagation delay which may not suit high-speed logic circuits. We can reduce the delay by several means. The easiest one is to make the ACC signal rise time slope higher (faster) compared to the input signal slope, or equivalently pre-raise ACC earlier than the input transitions. In this inverter, just by making the ACC rise 10 times faster than the input, we reduced the propagation delay penalty to 18%. But the increase in fall time, which is limited by the PMOS, is still 76%. We can increase the width of the depletion mode transistor, which unfortunately will also increase the leakage current. This is one of basic limitations of this circuit design style. The other limitation is the energy spent in switching V_{PWR} node. Note that whenever the input is 1 and $ACC = 1$, the virtual power node is pulled up to V_{dd} . However, during the inactivity period ($ACC = 0$), the virtual power node leaks some charge to the virtual ground node settling at a voltage equaling approximately $-V_{TdepN}$, which is about 0.65 V in our design. A complete energy estimate needs to take this switching of the V_{PWR} node between V_{dd} and $-V_{TdepN}$ into consideration. The V_{PWR} node has capacitance $C_{eq} = 2C_{diff}$. Consider a 0.3 V swing between V_{dd} and $-V_{TdepN}$. The extra energy required then will be:

$$\text{Extra switching energy} = \varepsilon = 0.3C_{diff}J$$

If the circuit is idle for Δ_t time, on average, after each active period, the switching will pay-off only if,

$$\Delta_t \geq C_{diff}7.9 \times 10^7$$

Table III. Performance impact of circuit in Figure 1.

	t_{pLH} (ps)	t_{pLH} (ps)	t_r (ps)	t_f (ps)
Base	16.8	10.54	33.63	17.31
New	25.9	16.32	40.72	30.89
%Inc	54.2	54.80	21.10	78.50

This is because from Table I, the 70 nm leakage current is $53.96 \text{ nA}/\mu\text{m}$. With $W_{min} = 70 \text{ nm}$, this gives the leakage current as $(0.070 \mu\text{m}) \times (53.96 \text{ nA}/\mu\text{m}) = 3.77 \text{ nA}$. The leakage energy over time Δ_t will equal $3.77 \times 10^{-9} \times \Delta_t$. For the leakage energy to exceed the extra switching energy of V_{PWR} , $3.77 \times 10^{-9} \times \Delta_t > 0.3C_{diff}$, which leads to $\Delta_t \geq C_{diff}7.9 \times 10^7$. With 70 nm technology and minimum width transistors, C_{diff} typically equals 0.1 fF hence $\Delta_t \geq 7.9 \text{ ns}$. For typical high performance circuits, with clock cycle time as 0.2 ns , we get around 40 cycles as the idle time. Hence the dep-warmup CMOS circuit technique can be applied if the average idle time between active periods is far greater than this *break-even* window. The immediate application will be in SRAM arrays, which occupy more than 50% of the typical microprocessor's area and consumes more than 50% of the energy.

2.3. Cross Coupled Inverter

The circuit in Figure 1 is not truly *regenerative* in idle mode, i.e., as the devices are cascaded, the high output V_H^i of i th stage is less than V_H^{i-1} (of $(i-1)$ st stage). Similarly the low output V_L increases. But the V_H doesn't go lower than $|V_{TdepN}|$ and V_L doesn't increase beyond $V_{dd} - |V_{TdepP}|$. To find the leakage reduction in that case we can simply cross-couple the inverters and study the circuit. This will be the minimum leakage reduction possible through this method as V_{PWR} will be equal to V_H and V_{GND} will equal V_L . In this scenario, one of the factors in leakage reduction, reverse biased V_{gs} , doesn't exist. Leakage current for this cross-coupled inverter estimated using HSPICE⁵ is 515 pA . And $V_{PWR} = HIGH = 742 \text{ mV}$, $V_{GND} = LOW = 225 \text{ mV}$. Even though the leakage current increases by 25 times (per inverter) when compared to circuit in Figure 1, we still have achieved a reduction of 12–15 times when compared to the original inverter. Moreover, the high and low levels are only 500 mV apart, and hence require less switching energy. Further reduction in leakage current is possible if multiple cross-coupled inverters share the same depletion device pair.

2.4. Other Applications

The circuit in Figure 1 is well-suited for clocked circuits, where the activity in the circuit is in sync with a clock. Another possible circuit design style with this principle to reduce leakage energy is shown in Figure 2. This circuit also has the charging and discharging timing characteristics of Table III, except that either only charging or discharging path is active at any point of time. However, this design style requires each variable to be present in both original and complemented forms. Hence this style will be useful in differential signal design styles such as DCVS. In Ref. [20] Johnson et al. use stacking of transistors to reduce the leakage current. Our method can also be modelled similarly. In the case of low-power circuits, static

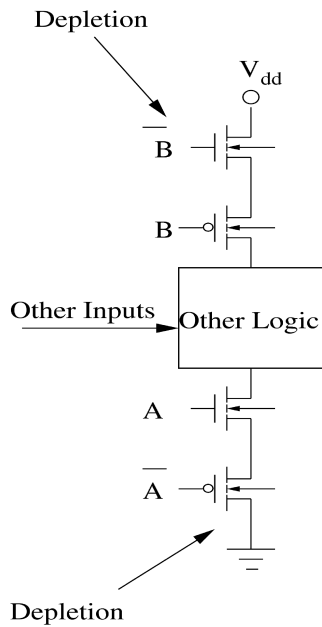


Fig. 2. Circuit technique to reduce static power in low-power circuits.

power is a major limitation in not moving towards smaller technologies. The proposed technique can then be used to reduce the static power with smaller technologies, hence achieving greater performance when compared to larger technologies without any increase in static power.

3. REDUCING STATIC ENERGY IN ON-CHIP CACHES

As we saw in Section 2, the immediate application of the *dep-warmup-CMOS* is in SRAM arrays. Hence, in this section, we study its application in caches. The latest processors have two or three levels of caches, namely L0, L1, and L2. L0 is closest to the CPU and L2 is closest to the main memory. To improve the performance, L2 caches are now made on-chip and sized upto 1 MB. In contrast, L1 cache sizes are typically 32 KB to 64 KB. L1 miss rates are on the average less than 2%, hence L2 will be very infrequently accessed when compared to L1. Given this, we validate our circuit design technique with L1 cache, which is in the critical path. If the scheme works with L1, we can infer that it will work with L2 as well (L2 has longer idle periods and is less critical for processor performance).

We use CACTI 3.0,¹³ as the base model to evaluate the performance impact. Cacti uses the underlying cache architecture shown in Figure 3, which explains the general cache architecture for an N -way set associative cache. The *address decoder* decodes the incoming address and selects the appropriate *set* or a row by driving one wordline in the data array as well as the corresponding wordline in the tag array. Every wordline corresponds to a *set* in the cache and contains as many bitlines as the number of bits in the *set*. Thus in an N -way set associative cache with block

size B bytes, each wordline would drive $N \times B \times 8$ cells. To reduce the switching power and the driving capacity of bitline drivers, each bitline is made to transit from V_{dd} (1 V in 70 nm process) to V_{bitpre} (bitline precharge voltage, 0.7 V in 70 nm process). Moreover, *sense amplifiers* are used to amplify this difference to produce the output data bits. Each sense amplifier monitors a pair of bitlines and detects a differential change. Determination of whether bitline or its complement goes low allows the sense amplifier to infer the memory cell's contents. When a sense amplifier is shared among several pairs of bitlines, a multiplexor is inserted before the sense amps. These multiplexors are driven by the *address decoder*. All the bits from a *set* in an N -way set associative cache are read in parallel. These bits are logically grouped into N ways. To select only one of the ways, the address, which was read (from the tag array) concurrently with the data, is compared with the input address. Only the data from the matching way is sent to the CPU. Thus N comparators are needed for an N -way associative cache. Other techniques such as sub-arrays, block tiling, sub-banking are used to reduce the cache access time. However, in all the cache architectures, all the N -ways are still accessed concurrently and only the matching way is forwarded.

Cacti uses sub-arrays to reduce the bit-line and word-line delays. For a 32 KB 4-way cache with 32 B block size, Cacti finds the optimal configuration as 8 sub-arrays with each having 128 rows of 256 bits. The timing parameters for various components in a 32 KB, 32 B, 4-way, 1 RW port, 70 nm cache are given in Table IV. The cache access time for this configuration is 556.1 ps, which includes the data output driver delay of 76.2 ps. From Table IV it is clear that data array delay does not form the critical path in cache access timing. A 32 bit processor with this cache configuration requires a tag array of 19 KBits, which is just 7.4% of the data array size. Therefore, we apply our method of static leakage reduction only to data array, which is not in the critical path (increase in data array delay will not impact the cache access time), and consumes more than 92% of the static power in the caches.

We can use a depletion device pair per SRAM cell to reduce the leakage power as described in Section 2. However, this will increase the SRAM cell size, and hence the area increase will out-weigh the leakage power reduction. An alternative is to share the depletion mode transistors with multiple SRAM cells. The wordline access signal can be used to control the depletion devices since it already encodes the active periods of a SRAM cell. Hence, no additional control signals need be generated. There are two ways to share the depletion or the *voltage clamp* devices. As we saw in Table III the fall time t_f is approximately 4 times the rise time t_r . We decided to increase the depletion PMOS width by a factor of 4. This leads to approximately equal t_r and t_f . For an SRAM cell which is a cross-coupled inverter, write time is not the bottleneck

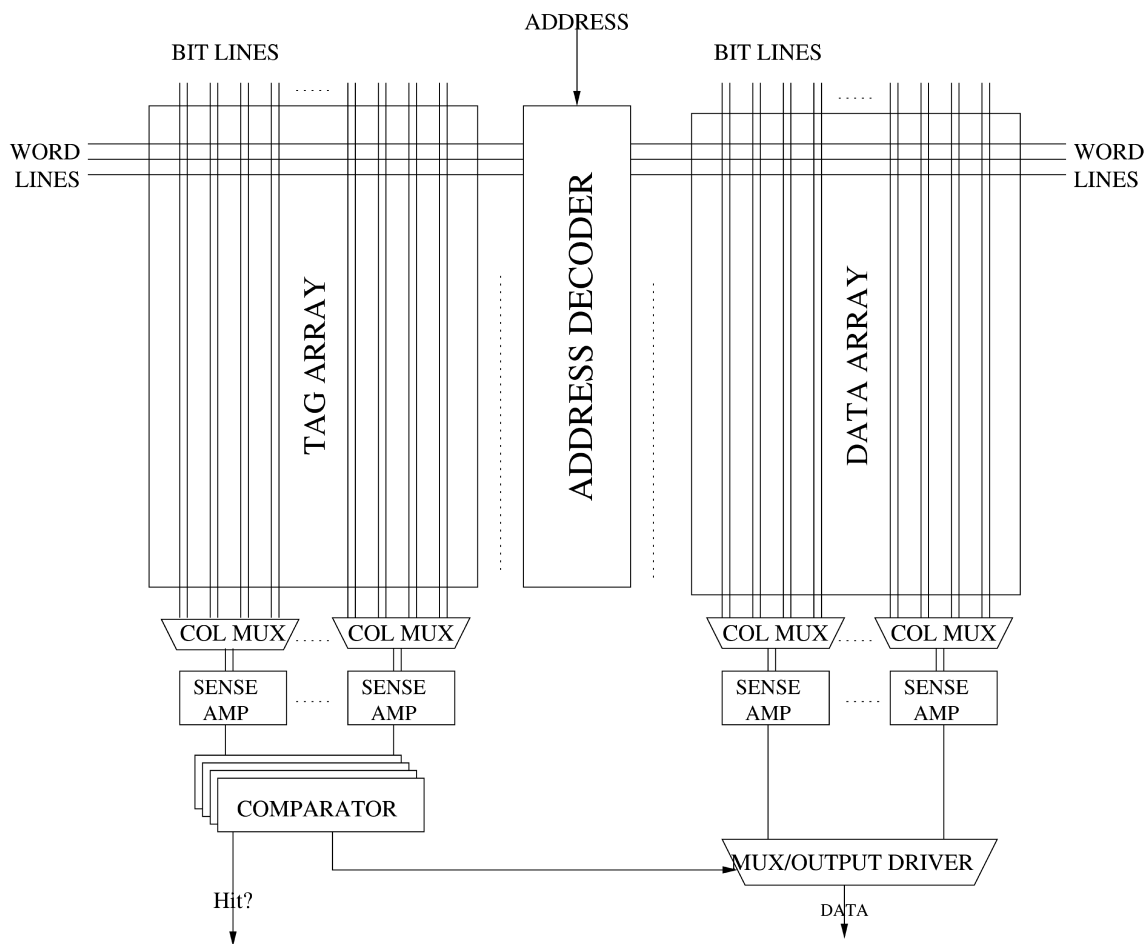


Fig. 3. General cache architecture of an n -way set associative cache.

as the cross-coupling effect aids in state transition. The read time is impacted only by the discharge path. Hence increasing depletion PMOS width alone is justified. From Table IV data, the data array delay is 85.2% of the tag array delay and data bitline delay is 2.73% of the data array delay. Hence upto 6 times increase in bitline delay will still not increase the overall cache access time. We try to exploit this fact in deciding the number of *voltage clamp* devices and their sizes. From Table III, we can share one depletion PMOS with 4 SRAM cells resulting in 4 to 5 times increase in the fall time t_f . Based on our earlier calculation of t_r and t_f , we can share one depletion NMOS

Table IV. Cache access timing for a 32 KB, 4-way, 32 B,1 RW port, 1 sub-bank cache as given by CACTI.

	Data array delay (ps)	Tag array delay (ps)
Decoder	208.572	099.410
Wordline	115.975	044.415
Bitline	011.765	011.898
Senseamp	072.625	044.625
Compare	—	112.912
Mux driver	—	150.077
Sel inverter	—	016.612
Total	408.936	479.949

with 16 SRAM cells. In our configuration, each wordline has 256 SRAM cells. For every group of 16 SRAM cells, one *voltage clamping* depletion device pair is used as shown in Figure 4. This dep-warmup CMOS configuration is compared with the base wordline configuration of 256 SRAM cells. The basic SRAM cell we use in HSPICE simulation is shown in Figure 5. We use low V_t devices

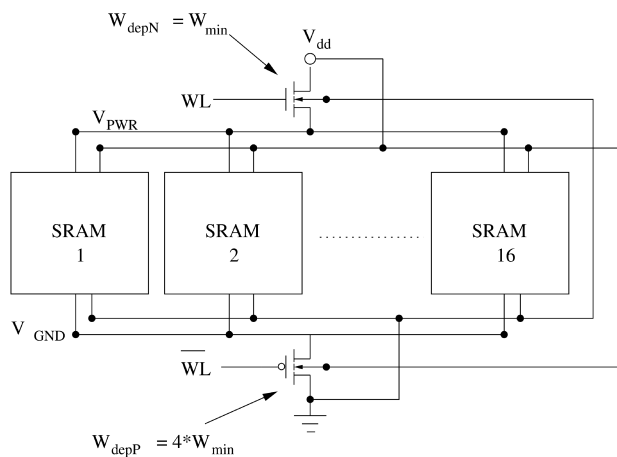


Fig. 4. Warm SRAM wordline.

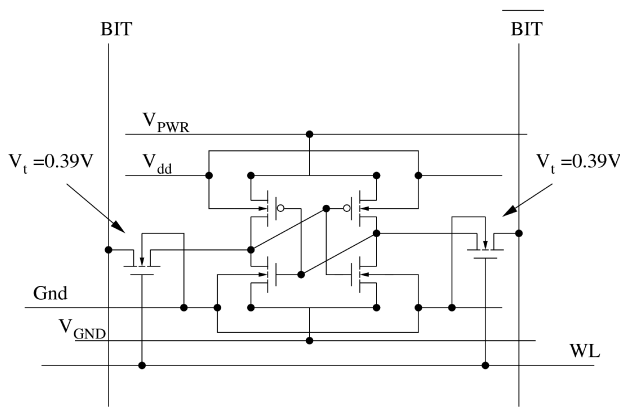


Fig. 5. Basic SRAM cell used in simulation.

in the cross-coupled inverters and high V_t devices for the access transistors, in order to reduce the leakage due to the access transistors. Except for the access transistors, all the other transistors have $|V_t| = 0.2$ V.

3.1. Leakage Reduction

The most relevant parameters of the proposed dep-warmup CMOS cache is the leakage reduction achieved by this circuit and the steady state voltages of the nodes. The *current* measurements presented in Table V are for a single SRAM cell. The steady state values were measured after a *bit* was written into the cell. When compared to the values presented in Section 2 for the cross-coupled inverter case, we see that V_H has decreased, i.e., moved closer to $|V_{TdepN}|$. This is because we have shared one depletion mode NMOS with 16 SRAM cells, hence the leakage current in the circuit has to decrease. All the 16 SRAM cells have to share the leakage current supplied by one depletion mode NMOS. V_L has also moved closer to $V_{dd} - |V_{TdepP}|$ but not as much as V_H . This is because we have increased the width of depletion mode PMOS by 3 times over the minimum. Hence it could supply more leakage current.

We can reduce the leakage current further by increasing the sharing but that will affect the transition delay and hence impact the performance. From Table V it is clear that we have achieved more than 23 times static power reduction. However, we need to assess the performance impact as well. The factor which will contribute to performance reduction is the delay in charging up V_{PWR} node from 0.686 V to 1.0 V. In the following two sub-sections we will analyze the performance impact on *write* and *read* operations.

Table V. Steady state values of a WARM SRAM cell.

Param	Base	Warm SRAM
I_L (μ A)	6250	262
$V(BIT)$ (V)	1.0	0.686
$V(BIT-bar)$ (V)	0.0	0.252

Table VI. Transient analysis parameters.

Param	Value	Param	Value
WL t_r and t_f	100 ps	Base t_r	47.0 ps
WL t_r and t_f	10 ps	Base t_f	22.0 ps
WL pulse width	200 ps	Warm SRAM t_r	50.1 ps
V_{bitpre}	0.5 V	Warm SRAM t_f	00.0 ps

3.2. Performance Impact on Write Operation

We performed HSPICE simulations on a subarray of size 128 rows by 256 columns (as given by Cacti) to study the impact on *read* and *write* times. Each bit and bitbar line was connected with a transmission gate to the bit value to be written, and with a PMOS transistor to precharge it to V_{bitpre} for read operation. The wordline enable signal was generated and characterized with the delay estimated by Cacti. The values of the different parameters are given in Table VI. There are two assumptions involved in WL and signal delay parameters. First, the delay of WL is not affected by the addition of $16W_{min}$ depletion mode NMOS transistors. This assumption is valid as WL is already driving $512C_g$ of memory cell access transistors, and an addition of $16C_g$ does not increase the delay by more than 3% (or the driver could be sized to absorb this delay). Second, signal is generated from WL and since it is driving only $64C_g$, its delay can be made one tenth of the WL delay.

Table VI gives the transition delay values for both the base configuration and the *warm sram* configuration. These delays are measured from the point WL signal is completely *ON*. Figure 6 displays the bit transition

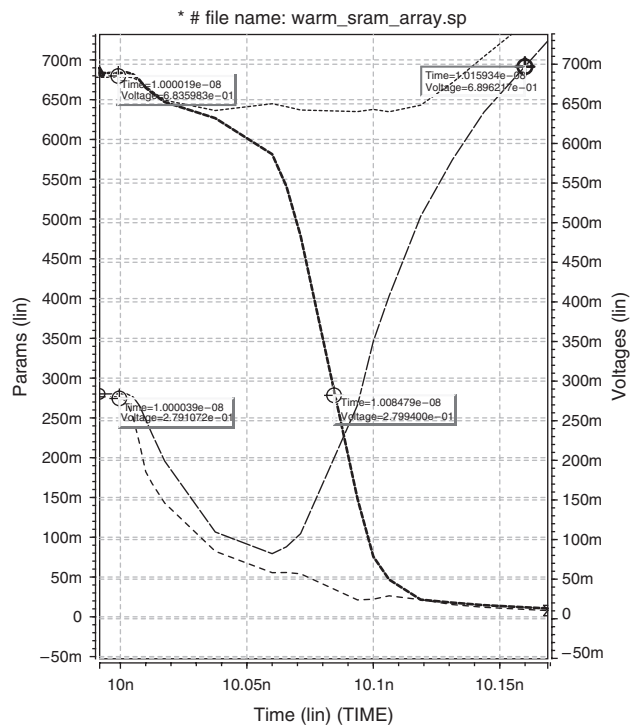


Fig. 6. Transient analysis of write operation.

responses for the *warm sram*. It is clear from these values that the write operation is not getting affected by the presence of depletion mode devices. Firstly we are getting the advantage of depletion PMOS getting switched on ten times faster than the access transistors. Therefore V_{GND} transits to zero even before access transistors are completely opened allowing bit to become zero with zero delay. Secondly since the bit transits from a non-zero initial value to V_{H} , the peak current requirement for the transition is smaller. Thus the single depletion NMOS transistor is able to supply the required current for 32 inverter transitions. This fact also illustrates that the proposed circuit uses less energy for bit transitions.

However, the *warm sram* has one disadvantage in terms of energy, i.e., every time a bit is written, whether the bit is transitioning or not, it has to raise the V_{PWR} node from V_{H} (≈ 700 mV) to V_{dd} . This also applies to the output node of an inverter which is in V_{H} state. Since V_{PWR} node has capacitance $518C_{\text{diff}}$ (per wordline) and the output node of an inverter has capacitance $768C_{\text{diff}}$ (per wordline), this energy will approximately equal $327.9C_{\text{diff}}$. For 70 nm technology, C_{diff} is in the range of 0.11 fF. Hence the extra energy spent will be approximately 36.07 fJ. Note that this extra energy is paid only when the memory cell bit does not change state due to this write. This is because the normal write scheme will not consume any energy in such a case, however warm SRAM will still need 0.14 fJ per bit (where state does not change). Hence, the extra write energy is proportional to the number of bits that do not change state. We calculated the write energy by integrating the input current. This energy is parameterized by the number of bit-transitions (bits changing state). The results are shown in Table VII. The first column of the table specifies the number of cells changing state. This table shows that *warm sram* consumes less energy than the conventional circuit even with 128 bits changing state (out of 256 in a single row/block). The break-even point is somewhere closer to the point where 70 bits change state. For a write with less than 70 bits changing state, the constant energy to raise V_{PWR} and V_{out} nodes for the bits not changing state dominates. This results in higher energy consumption for warm SRAM than for the conventional SRAM. Whenever a cache block is replaced, if we assume a uniform distribution of bit values, we will get on the average 128 bit transitions. For cache access writes, this depends on the access width. The *warm sram* will be spending at most 36.07 fJ extra energy for any access causing less than 64 bits to

transit. When compared to the dynamic energy per cache access, which is estimated by Cacti as 0.3 nJ, this extra energy is very insignificant. Hence we can safely assume that it has little effect on the overall dynamic energy.

3.3. Performance Impact on Read Operation

The most critical operation in cache is read operation which occurs twice as often as writes. It is critical because load latency cannot be hidden. The instructions waiting on read results often stall. Whereas, the store (write) latency can be easily hidden using write buffers. No instructions in the immediate vicinity depend on the outcome of the write operation. The tag array access is, however, in the critical path in the cache read, as we observed from Table IV. Hence we can exploit this slack to make the data array path slower without impacting the same cache access time. This is the basis for the chosen depletion mode transistor widths. If we have to reduce the delay any further, we can reduce the degree of sharing, trading it with increase in leakage current. As per Cacti, bitline delay is defined as the delay between the time at which wordline enable is ON and voltage difference between bit and bitbar becomes 100 mV. Cacti uses a precharge voltage of 0.7 V for the 70 nm technology and estimates the bitline delay to be 11.7 ps. We varied the precharge voltage from 0.7 V to 0.5 V in 50 mV steps to study its influence on bitline delay for both the base circuit and the *warm sram* circuit. The results are shown in Table VIII. The difference in Cacti's estimation and our results can be attributed to the high- V_{t} access transistors. Since we are interested in estimating the leakage savings in *warm sram*, we used high- V_{t} ($=0.39$ V) access transistors in all the other reported simulation results. We used a pulse width of 200 ps for the wordline signal in our simulations. *Warm sram* bitline delay for 0.7 V and 0.65 V precharge voltages are greater than this 200 ps pulse width for the wordline. Hence, the bitlines did not achieve the 100 mV difference. Furthermore, since 0.5 V closely matches with Cacti's estimation of the bitline delay, we use 0.5 V as our precharge voltage for all the further simulations. From Table VIII, it is clear that bitline delay increase for precharge voltage of 0.5 V and 0.55 V does not increase both *cache access time* as well as the *wave pipelined cycle time*. Thus *warm sram* does not have any performance impact on cache access time. The read operation for a 0.5 V precharge voltage is shown in Figure 7.

Table VII. Write energy comparison.

No of bits	Energy (fJ)		Peak current (mA)	
	Base	Warm SRAM	Base	Warm SRAM
256	320	144	5.53	0.997
192	240	132	4.14	0.930
128	160	118	2.75	0.840
64	80	99	1.36	0.735

Table VIII. Bitline delay as a function of precharge voltage.

V_{bitpre} (V)	Base (ps)	Warm SRAM (ps)	%Inc
0.70	39.8	>200	X
0.65	30.6	>200	X
0.60	24.1	181.4	652.7
0.55	21.2	129.3	509.9
0.50	20.5	113.6	454.1

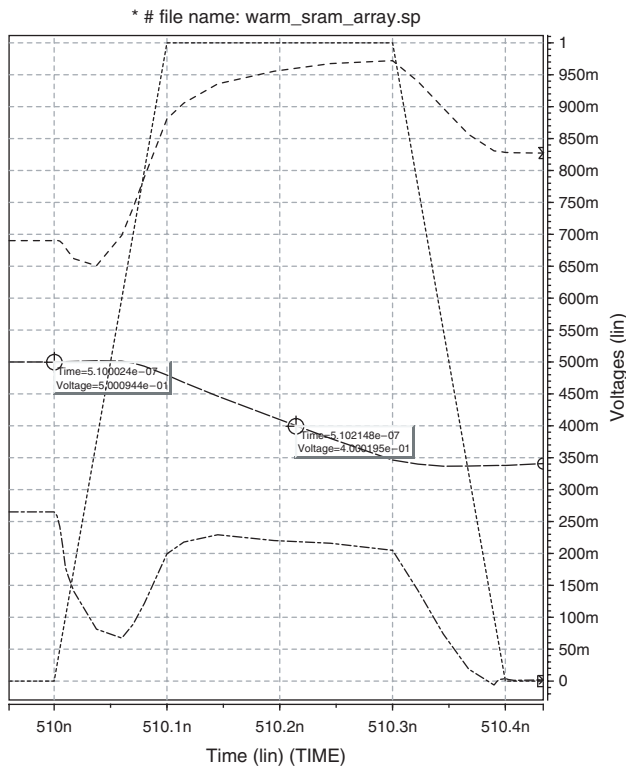


Fig. 7. Transient analysis of read operation.

As is the case with write, read operation also requires extra energy to charge the nodes to V_{dd} . This can be avoided if we don't switch on depletion NMOS for read operation. This requires additional control logic and a separate driver for depletion NMOS transistors. This may not pay off since the extra energy spent for charging V_{PWR} and output node is at most 36 fJ (based on our prior estimation). This potential energy saving is insignificant when compared to per access dynamic energy of the cache. This energy estimate is worst case since the V_{PWR} node acts like a capacitor and is discharged by the leakage current in the circuit. The time the node takes to reach the steady state depends on the leakage current magnitude. It takes more than 200 ns to leak down to the steady state value from V_{dd} at room temperature ($T = 25^\circ\text{C}$) as shown in Figure 8. If the cache line was accessed again within this 200 ns period, we will spend less energy than 36 fJ on charging V_{PWR} and output node to V_{dd} . We calculated the energy used per word line access for different inter-access time intervals. The results are shown in Table IX. These results are very close to our estimates. This also shows that if two accesses happen within 25 ns of each other, we don't spend any extra energy compared to the base SRAM cell design. The reason why the warm SRAM energy is less than the base circuit at 25 ns inter-access

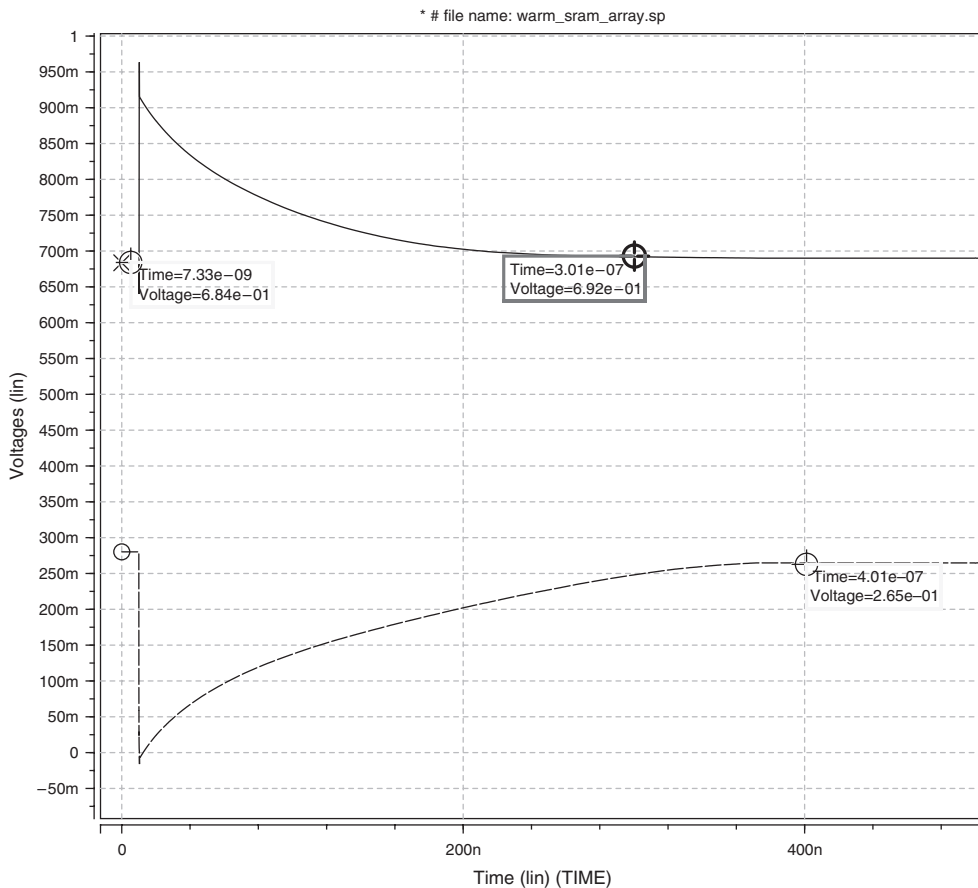


Fig. 8. Discharging of V_{PWR} node and charging of V_{GND} node.

Table IX. Read energy with respect to time after an access.

Base read energy: 25.92 fJ		
Time (ns)	Energy (fJ)	Extra energy (fJ)
25	23.99	-1.93
50	33.86	7.94
75	41.56	15.64
100	47.22	21.30
125	51.38	25.46
150	55.27	29.35
175	57.45	31.53
200	59.44	33.52
300	59.44	33.52

gap is as follows. During the read operation, the potential of the SRAM cell node which is pulling the bitline rises by a small amount over GND . The discharging NMOS transistor cannot do with V_{ds} at 0 V. This causes a short circuit path from V_{dd} to ground. Moreover, this effect is higher in the case of base circuit as it is discharging more current.

3.4. Architecture/Program Level Energy Estimation

In the preceding sections we saw how *warm sram* can reduce the leakage energy without impacting the read/write access times of a cache. In this section, we evaluate the static leakage energy savings for a hypothetical 32 KB, 4-way cache. The extra access energy for warm SRAM to charge V_{PWR} and output up to V_{dd} is accounted for in a cycle by cycle microarchitecture simulator, SimpleScalar version 3.0.⁸ The average number of cache accesses are captured by SimpleScalar. We executed SPEC2000⁶ integer benchmarks on this simulator. Suleyman Sair et al.,⁹ did analyze the memory behavior of SPEC2000 benchmarks and found the 500 million instruction window which closely matches with the characteristics of the whole application. We use these instruction windows, as given in Table XI in our simulation. We use SimpleScalar 3.0 with the configuration shown in Table X to collect the statistics.

In order to calculate the net energy savings, we need to account for the extra energy incurred by this method. We have so far seen one source of extra energy, i.e., energy spent in bringing *warm sram* to normal state. This energy penalty will be significant only if the accesses occur more than 50 ns apart. Table XII lists the percentage of L1 cache accesses within 50 cycle window and 100 cycle window. From Table XII it is evident that more than 80% of the accesses are occur within 100 cycles of another access. Using this fact and the extra energy figures presented in Table IX, we can estimate the extra energy per access.

Table X. SimpleScalar simulation parameters.

Parameter	Value
Processor	Alpha out of order 4 way issue
L1-DCache	32 KB, 4 way, 32 B line, 1 cycle access latency
L1-ICache	16 KB, 1 way, 32 B line, 1 cycle access latency
L2-Unified	256 KB, 4 way, 64 B line, 8 cycle access latency

Table XI. Instruction window for SPEC2000 benchmarks.

Benchmark	#Insts skipped (M)
gzip	40
vpr-place	190
vpr-route	1150
gcc	1000
mcf	4000
crafty	10
parser	250
eon	3
perlbmk	500
gap	65
vortex	0.5
bzip2	200
twolf	400

These numbers depend on the cycle time. We use both 0.2 ns and 0.5 ns as the cycle times to illustrate the energy savings. In both the cases, 100 cycles is less than 50 ns hence we penalize only the accesses occurring after more than 100 cycles from another access with 33.52 fJ. We chose 0.2 ns because it is the wave pipelined cache cycle time estimated by Cacti and 0.5 ns because it approximates the cache access time.

Another source of extra energy is the access control signals generation for the depletion NMOS and PMOS. Since depletion NMOS is driven by the wordline WL signal, we will not incur any extra energy for that. However, we have to generate and this will be approximately (1/8)th of the wordline driver energy (as this signal has to drive only $64C_g$, (1/8)th the capacitance driven by WL). From Cacti, the wordline energy is 158.95 fJ. Hence, we assess 19.87 fJ extra energy per access to generate signal. We take this into account as well while calculating the energy savings. Table XIII lists various benchmarks and the number of execution cycles for 500 Million instructions and the corresponding energy savings.

The energy saving is calculated with the leakage currents given in Table V. For a 32 KB cache, this gives us the base leakage 1.638 mW while the *warm sram* leakage power as 68 μ W. Using these figures and taking the energy penalty incurred by our method into consideration, we achieve a 90% reduction in leakage energy on average

Table XII. Access percentage with respect to time.

Benchmark	50 cycles	100 cycles
crafty	59.73	9.15
eon	77.91	6.06
gcc	77.85	5.47
twolf	70.40	6.46
gzip	79.73	5.61
bzip	86.92	4.90
mcf	68.47	11.02
perlbmk	77.32	3.37
parser	75.18	7.36
vpr	69.59	7.81
Avg	74.31	6.721

Table XIII. Net energy savings.

Program	Execution cycles	Memory access	Energy penalty per access (μJ)	%Net saving (0.2 ns/cyc)	%Net saving (0.5 ns/cyc)
crafty	396782412	195828079	5.93	91.28	94.02
eon	350714953	240118536	6.06	90.57	93.74
gcc	393784461	223031723	5.68	91.45	94.09
twolf	444314516	172189507	4.76	92.58	94.54
gzip	277336702	169725136	4.21	91.22	94.00
bzip	269543836	185471790	4.19	91.10	93.95
mcf	487390086	195632037	5.23	92.57	94.54
perlbnk	346674071	216796572	5.71	90.82	93.84
parser	326925643	190878110	4.91	91.26	94.01
vpr	421717636	185474202	5.09	92.16	94.37
Avg	371518431.60	197514569.20	5.18	91.50	94.11

in a 32 KB data array (for both 0.2 and 0.5 ns cycle times) without impacting the cache access time. Table XIII lists the energy savings for various SPEC2000 integer benchmarks for both 0.2 ns cycle time and 0.5 ns cycle time.

DRG cache,²¹ which uses gate-grounded transistor to reduce sub-threshold leakage current in SRAM cell, achieves 26–27% of energy savings with a performance penalty of 5.4% in 70 nm technology. Thus our WARM SRAM achieves better leakage reduction with lesser performance impact.

4. MODEL VALIDITY

In all our HSPICE simulations, we have assumed that *enhancement mode* model file with its threshold voltage modified to suit the *depletion device* will be able to model the depletion device's characteristics. We justify the validity of this assumption in this section. We discuss only the depletion NMOS case, however the discussion is equally applicable to depletion PMOS as well. Note that existing CMOS processes typically do not support depletion mode devices since it requires an extra mask. However processes evolve to support the devices prevalent in a period. If depwarmup-CMOS style gains favour with designers, the processes will normally evolve to accommodate it. It is not an issue of technical feasibility.

Depletion NMOS is formed by implanting donor atoms in the substrate. The threshold voltage of such a device is given by Eq. (3) [Ref. [17], p. 238].

$$V_T = V_{T0} + \gamma_1 \left(\sqrt{\phi_{bi} + V_{sb}} - \sqrt{\phi_{bi}} \right) \quad (3)$$

Where,

$$V_{T0} = V_{FB} + \phi_{bi} - \frac{qN_d d_1}{C_{ox}} \left(1 + \frac{d_1 C_{ox}}{2[\epsilonpsilon]_s} \right) + \gamma_1 (\sqrt{\phi_{bi}})$$

$$\gamma_1 = \left(1 + \frac{d_1 C_{ox}}{[\epsilonpsilon]_s} \right) \gamma$$

V_{FB} → Flat band voltage

ϕ_{bi} → Built in potential ($\phi_{Fp} - \phi_{Fn}$)

γ → Body effect coefficient of the unimplanted substrate

d_1 → Implantation depth

N_d → Donor concentration

Note that a depletion mode transistor has higher body effect compared to enhancement mode devices. Hence N_d and d_1 should be varied to get the required device characteristics. The first point of consideration is that the depletion NMOS should get cut-off when $V_{sb} = 0.65$ V and $V_{gs} = 0$ V. Hence by equating 3 to -0.65 V, we can solve for N_d and d_1 . The second point of consideration is that when $V_{gs} = 1$ V the gate should have gain comparable to what is predicted by the enhancement mode model. If these two operating points can be verified then our circuit will yield a similar result if we use proper depletion model for the devices.

There are four regions of operation in a depletion device,

- (1) Cut-off—device is completely depleted at the source end.
- (2) Surface depletion—surface is depleted but the buried channel exists conducting the current.
- (3) Surface accumulation—as V_{gs} increases beyond threshold, inversion occurs and carriers are accumulated on the surface.
- (4) Surface accumulation/depletion—as V_{db} or V_{sd} increase beyond a certain value, it depletes the channel on one of the sides making the device behave like a saturated enhancement mode device.

Our operating points should be in cut-off and surface accumulation regions. The device enters surface accumulation region if $V_{gs} > V_N$ where $V_N = V_{FB} + \phi_{bi}$. The gain of the device in surface depletion region is given by $\beta = \mu_b C_{ox} / (1 + \sigma)$, where

$$\sigma = \frac{C_{ox} d_1}{[\epsilonpsilon]_s} \left(\frac{C_{ox} d_1}{2[\epsilonpsilon]_s} + 1 \right)$$

Even though μ_b , bulk mobility, is typically larger than the surface mobility, the factor $(1 + \sigma)$ tends to reduce

Table XIV. Process parameters for depletion NMOS.

γ_I	d_I (10^{-10} m)	σ	N_d (10^{-18} cm $^{-3}$)	V_{T0} (V)	V_N (mV)
1.5γ	24.21	0.625	28.2	-0.6786	-37.06
2.0γ	48.41	1.5	14.23	-0.6881	-54.84
3.0γ	100	5	5.667	-0.7084	-78.78

the gain in surface depletion region when compared to the enhancement mode device. Once surface accumulation occurs the variations of channel charge with V_{gs} occur at the surface, and, thus, the gain is determined by the surface mobility and oxide thickness only. In other words the gain is comparable to the enhancement mode device. One problem with depletion device could be that if the implantation depth is too high then controlling the buried channel just with gate voltage will not be possible as the surface inversion will occur before pinch-off condition could be reached. This can be overcome if the depth is not made very large or with high V_{sb} the buried channel could be depleted. In our circuit since both drain and source are always above 0.65 V this condition can never occur.

The process parameters for the 70 nm technology⁴ we are considering are,

$$N_{sub} \rightarrow \text{Substrate doping concentration} = 6 \times 10^{16} \text{ cm}^{-3}$$

$$N_{ch} \rightarrow \text{Channel doping concentration} = 1.2 \times 10^{18} \text{ cm}^{-3}$$

$$X_j \rightarrow \text{Junction depth} = 299.99 \times 10^{-10} \text{ m}$$

$$t_{ox} \rightarrow \text{Oxide thickness} = 16 \times 10^{-10} \text{ m}$$

$$\gamma \rightarrow \text{Body effect coefficient of bulk} = 6.563 \times 10^{-2}$$

We can solve Eq. (3) for various values for γ_I i.e., d_I for the condition $V_T|_{V_{sb}} = -0.65$ V. We can then choose the solution closest to our assumptions. The various values of N_d and σ for various d_I values are listed in Table XIV.

Note that there are various possible solutions but we should choose the one with the lowest σ . Depending on the process and the maximum N_d allowed, the device has to be scaled to take into account the decrease in the gain. In all these cases, note that $V_N < 0$ hence both V_{gs} and V_{gd} are $> V_N$. This places the device in surface accumulation region when switched on. Hence, our assumption of validating the *warm sram* circuit technique with enhancement device models is justified. With proper depletion mode models, we can verify other characteristics of the device such as subthreshold leakage characteristics. However, to within a first order approximation, our circuit can achieve the claimed static energy reduction.

5. CONCLUSION

The static leakage energy is one of the biggest challenges facing the semiconductor industry in the near to intermediate term future (3–10 years). The static energy consumption grows exponentially with reduction in feature size.

On-chip caches occupy a major fraction of the processor's area. This holds both for high performance and for low power embedded processors. Since static leakage energy constitutes a large fraction of cache energy, and hence of total processor energy, on-chip caches form a good target for static energy reduction techniques. We proposed, modelled, and verified a new CMOS design style, warmup CMOS. These devices operate best when warmed up like an engine, where some of the key nodes have attained a certain potential. The warming-up cost is paid only once for each activity period, but the savings more than offset the initial warmup cost. The specific version of warmup CMOS presented in this paper is based on depletion mode devices—dep-warmup CMOS. We presented a SRAM cell design in dep-warmup CMOS and its block level implementation. The detailed SPICE simulations estimate the static leakage energy savings for L1 caches at more than 90% without any affect on the performance. We are further investigating other warmup CMOS design styles and their application to broader logic blocks.

Acknowledgments: The authors would like to acknowledge support from NSF Grant CCR 0209078 and AFRL through contract number F33615-02-C-1238.

References

1. International Technology Roadmap for Semiconductors, 2001 edn., Executive Summary. <http://public.itrs.net>
2. J. M. C. Stork, Technology leverage for ultra-low power information systems. *Proc. IEEE* (1995), Vol. 83.
3. T. Bell, Incredible shrinking computers. *IEEE Spectrum* (1991), pp. 37–43.
4. Berkeley predictive technology model. <http://www-device.eecs.berkeley.edu>.
5. Star-HSPICE 2001.4 Avant! Corporation.
6. <http://www.specbench.org/osg/cpu2000/>
7. Taiwan Semiconductor Manufacturing Company Ltd. <http://www.tsmc.com>
8. Doug Burger and Todd M. Austin, The SimpleScalar tool set, version 2.0. Computer Sciences Department Technical Report #1342. University of Wisconsin-Madison (1997).
9. Suleyman Sair and Mark Charney, Memory Behavior of the SPEC2000 Benchmark Suite. IBM Research Report (2000).
10. K. Nii, H. Makino, Y. Tujihashi, C. Morishima, Y. Hayakawa, H. Nunogami, T. Arakawa, and H. Hamano, A low power SRAM using auto-backgate-controlled MT-CMOS. *International Symposium on Low Power Electronics and Design* (1998), pp. 293–298.
11. M. Powell, S. H. Yang, B. Falsafi, K. Roy, and T. Vijaykumar, Gated- V_{dd} : A circuit technique to reduce leakage in deep-submicron cache memories. *International Symposium on Low Power Electronics and Design* (2000), pp. 90–95.
12. John L. Hennessy and David A. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufman, CA (2003).
13. Steven J. E. Wilton and Norman P. Jouppi, An enhanced access and cycle time model for on-chip caches. WRL Research Technical Report 93/5 (1994).
14. K. Flautner, N. S. Kim, S. Martin, D. Blaauw, and T. Mudge, Drowsy caches: Simple techniques for reducing leakage power. *Proceedings of 29th Annual International Symposium on Computer Architecture*, IEEE CS Press (2002), pp. 148–157.

15. J. A. Butts and G. S. Sohi, A static power model for architects. *Proceedings of 33rd Annual International Symposium on Microarchitecture* (2000).
16. Kaushik Roy and Sharat C. Prasad, *Low-Power CMOS VLSI Circuit Design*, John Wiley and Sons, Inc. (2000).
17. Yaniis Tsividis, *Operation and Modeling of The MOS Transistor*, WCB/McGraw-Hill (1999).
18. Kaouichi Kanda, Takayuki Miyazaki, Min Kyeong Sik, Hiroshi Kawaguchi, and Takayasu Sakurai, Two Orders of Magnitude Leakage Power Reduction of Low Voltage SRAM's by Row-by-Row Dynamic V_{dd} Control (RRDV) Scheme. <http://www.intel.com>
19. <http://www.intel.com>
20. M. C. Johnson, K. Roy, and D. Somasekhar, A model for leakage control by transistor stacking. Technical Report TR-ECE 97-12, Purdue University, Department of ECE.
21. Amit Agarwal, Hai Li, and Kaushik Roy, DRG-Cache: A data retention gated-ground cache for low power. *Proceedings of Design Automation Conference* (2002), pp. 473–478.

Mahadevan Gomathisankaran

Mahadevan Gomathisankaran received B.E. in *Electronics and Communication Engineering* (1998) from the *Regional Engineering College (now known as National Institute of Technology)*, Trichy, He received Ph.D. in *Computer Engineering* from *Iowa State University, Ames* in 2006. He is with *Intel (Corporate Technology Group)* working in *Trusted Platforms Lab*. His research interests are *secure and DRM architectures, low power design, and logic design*.

Akhilesh Tyagi

Akhilesh Tyagi received B.E. (Honors) in *Electrical and Electronics Engineering* from (1981) *Birla Institute of Technology and Science, Pilani* followed by M. Tech. in *Computer Engineering* (1983) from *Indian Institute of Technology, New Delhi, India*. He received Ph.D. in *Computer Science* from *University of Washington, Seattle* in 1988. He was an assistant professor with the *Department of Computer Science at the University of North Carolina at Chapel Hill* from August of 1987 to June of 1993. Subsequent to that, he was with the *Department of Computer Science*, and is now with *Electrical and Computer Engineering Department at Iowa State University, Ames, Iowa*. His research interests include *VLSI complexity theory and low energy design, secure and DRM architectures and compilers*.