Low Leakage Bit-Line Sram Design Architectures

Ratan Kumar S.V., Dept. of ECE, Assistant Professor, RGM CET, Nandyal, saneratankumar@gmail.com
Mr. S. Kashif Hussain, Asst. Prof in ECE, RGM CET, Nandyal, kashif1919@gmail.com
Dr. S. Govinda Rajulu, Dept. of ECE, Professor, RGM CET, Nandyal, rajulusg09@gmail.com

Abstract: In high performance Systems-on-Chip, leakage power consumption has become comparable to the dynamic component, and its relevance increases as technology scales. These trends are even more evident for memory devices, for two main reasons. First, memories have historically been designed with performance as the primary figure of merit; therefore, they are intrinsically non-power-efficient structures. Second, memories are accessed in small chunks, thus leaving the vast majority of the memory cells unaccessed for a large fraction of the time. In this paper, we present an overview of the techniques proposed both in the academic and in the industrial domain for minimizing leakage power, and in particular, the sub threshold component, in SRAMs. The surveyed solutions range from cell-level techniques to architectural solutions suitable to system-level design[15].

Keywords: memory, Systems-on-Chip, leakage power consumption

I. INTRODUCTION
The emergence of static power consumption in CMOS devices has been one of the first adverse effects of technology scaling. Roughly, when feature size broke the 100 nm barrier, the CMOS transistor ceased to be a virtually ideal switch consuming power only when changing state. While static (or leakage) power affects all kinds of CMOS circuits, it is particularly critical for SRAMs, for two main reasons. First, leakage power is proportional to the total number of transistors on chip. As reported in the ITRS roadmap, transistors devoted to memory structures in a typical microprocessor based system is about 70% today and it is expected to rise to 80% in the near future [1,15]. Another reason is related to the temperature dependence of some sources of leakage power. SRAMs are highly optimized structures resulting in very high density: Typical SRAM cells have areas in the order of 0.1 m. Such a high density, coupled with large power consumption, translates into an increase of temperature, which in turn affects leakage current (and, in particular, the sub threshold component) exponentially. For these reasons, there has been a wide spectrum of research on the reduction of leakage power for SRAMs, at various abstraction levels, from optimized cells structures to alternative memory architectures. The purpose of this survey is to present an exhaustive review of such methods, and to provide a systematic classification and qualitative assessment of the various solutions proposed in the literature[15].

II. OVERVIEW
A. Sources of Leakage Consumption in SRAMs[15]:
Leakage power in a CMOS transistor originates from several sources, corresponding to various leakage currents flowing in the device [2]. The list mostly comprises currents that are present when the channel is non-conducting (off state): Sub threshold leakage, gate-induced drain leakage, and depletion punch-through leakage. Two other relevant sources of leakage exist independent of the conduction state of the channel: Gate tunneling leakage through bulk, source and drain (usually regarded as a single current), and p-n junction leakage (from both source and drain). The latter has various sub-sources, and it is dominated by the band-to-band (BTBT) tunneling effect. Fig. 1 summarizes the flow of such currents in the transistor’s schematic. Most of the leakage sources can be considered as parasitic effects and are indeed negligible. In [3], the authors show that three are the currents that should be considered for power analysis: Sub-threshold, gate and junction leakage. Their projections on predictive technologies indicate that the three sources will have comparable magnitude already in the 32 nm node. However, technologies
that are currently in use do not seem to support this projection. Device materials technology has managed to keep gate leakage under control thanks to the adoption of high-k dielectrics, which helped to move forward in time the point in which gate leakage will become a limiting issue. Band-to-band tunneling currents, conversely, seem to be critical only for strongly reverse-biased devices; although reverse-biasing is a common strategy in low-power design, it is used selectively and with moderate amount of bias for performance reasons. Based on these considerations, our survey addresses only the reduction of the most relevant component of the various leakage currents, that is, sub-threshold leakage[15].

![Image of 6T SRAM Cell](Fig. 2. Reference 6T SRAM Cell[15].)

B. Sub-Threshold Leakage in a SRAM Cell[15]
In the following, for ease of reference we will use the structure of a conventional 6T SRAM cell depicted in Fig. 2, and in particular the labeling of the transistors shown in the figure: P1/N1 and P2/N2 denote p and n transistors of the bitcell, and N3/N4 the two nMOS access transistors.

Sub-threshold leakage occurs whenever a transistor is off and is non-zero. Therefore, leakage can occur either inside the bit cell or on the access transistors paths, as shown in Fig. 2: Solid arrows denote internal or cell leakage, whereas dashed arrows denote bitline leakage. Which transistor is actually leaking in a cell depends on (i) the value stored, (ii) the logic level of the word line, and (iii) the type of operation (i.e., the value of the bit lines). Because of the symmetric structure of the cell, whatever the value stored, the word line value and the operation, there will be some leaking transistors in the cell.

C. Taxonomy of Approaches[15]
Fig. 3 shows a taxonomy of the leakage reduction techniques for SRAMs. A first level of classification distinguishes between approaches that target bit line leakage versus those that reduce leakage in the cell array. The former give fewer degrees of freedom (leakage is due to the access transistors only), and the possible solutions are based on the design of the bit lines and/or the word lines. Reduction of leakage in the cell array offers more alternatives: We categorized the methods based on whether they target active or standby leakage. It is essential to underline here that our distinction between active and standby leakage is somehow different from the one often used in the literature. By active leakage we literally mean “leakage in the active state”, that is, when the cell is used (read or written); therefore, these techniques aim at reducing intrinsic leakage of the bit cell. Conversely, by standby leakage we mean “leakage in the standby state”; thus, reduction techniques addressing standby leakage include all solutions in which the memory features some low-leakage state into which some portions (of variable size) of the memory block can be put during inactive phases. In the case of active leakage, solutions are limited to schemes that propose customized design of the memory cell. In the case of standby leakage, methods under this category are assimilated to different implementations of dynamic power management. In other words, a given unit of power management is identified, and based on some criterion related to the access pattern to the unit, the latter is selectively put into a low-leakage state. The granularity of the unit is quite variable, and it may range from a single cell (cell-based power management) to more or less large portions of the memory array (coarse-grain power management). In the latter category, the vast majority of the approaches refer to caches, which represent the typical embodiment of SRAMs at the architectural level. For this reason, we will term these category cache dynamic power management so as to characterize it more precisely. For those solutions, as the granularity of the unit increases, the policy used to drive the transition between the two states and the architectural implications become more important. Given the wide scope of these approaches, Section IV.B presents a further and more specific sub-classification[15].

III. BITLINE LEAKAGE REDUCTION TECHNIQUES[15]
Although bit line leakage is less relevant in magnitude compared to cell leakage, it is nevertheless very important because it also affects the reliability of the memory device. Fig. 4 shows the worst-case scenario for a single-column portion of a SRAM. The accessed cell "1" stores a “1” while all the others store a “0”. When the cell is read, the bit line leakage current (i.e., the current absorbed by the cells whose WL is at “0”) becomes the noise against the cell current (i.e., the current injected by the accessed cell to sustain the bit line pre-charge), thus inducing a sensible voltage drop on the bit line. Under this condition, the sense amplifier needs more time to detect the input variation causing more read latency, or, if the leakage approaches the current of the accessed cell, it may provide a faulty output. While in old CMOS technologies the empirical design rule of using an on-off current ratio larger than 10 (e.g., by limiting the height of the SRAM column) was conservative enough to guarantee functionality, with the advent of leakage-dominated technologies new dedicated design
techniques have become essential. In the sequel, we review the most effective design solutions to reduce bit line leakage current[15].

**Fig. 3.** Classification of Sub Threshold Leakage Reduction Techniques for SRAMs[15].

They can be broadly classified into approaches that re-design the bit line and techniques that re-design the word line. Methods in the first category rely on direct leakage mitigation or on the assignment of appropriate pre-charging values. By leakage mitigation methods we refer to techniques that try to shield the sense amplifier from the leakage current. Two main strategies can be grouped under this definition: Bit line leakage compensation (BLC) and bit line voltage calibration (X-calibration). The pre-charging value assignment techniques, conversely, include those methods which operate on the pre-charging phase and assign to bitlines special, leakage-aware values. We will present two techniques: the self-reversed bias (SRB), and the floating bit line. Within the second category (word line design), we discuss two types of techniques: Those based on word line value assignment, (Negative Word line) and those which use leakage-aware modified access structures (8-Transistor cell). Table I summarizes the bit line leakage reduction techniques that are surveyed next[15].

### SRAM DESIGN TECHNIQUES FOR BITLINE LEAKAGE REDUCTION[15]

<table>
<thead>
<tr>
<th>Bitline (BL) Design</th>
<th>Wordline (WL) Design</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compensation</td>
<td>[4]</td>
</tr>
<tr>
<td>Calibration</td>
<td>[5]</td>
</tr>
<tr>
<td>BL Biasing</td>
<td>[6]</td>
</tr>
<tr>
<td>Floating BL</td>
<td>[7]</td>
</tr>
<tr>
<td>Negative WL</td>
<td>[8][9]</td>
</tr>
<tr>
<td>8-T Cell</td>
<td>[10]</td>
</tr>
</tbody>
</table>

**A. Bitline Design[15]**

1) Leakage Compensation: The compensation method was first proposed in [4]. As shown in Fig. 5, it is based on a pre charge circuit which includes a Bit-Line Compensation (BLC) scheme. The idea is to inject into the bit line an amount of current equal to the leakage current, so that the leakage is compensated and then not sensed by the SA. Although intuitive, this approach requires the implementation of two tricky functions: i) Leakage detection and ii) current injection. According to the BLC structure described in [4], at the end of the pre-charging cycle the bit line leakage is measured by means of an additional capacitor, whose function is that of storing the amount of bit line leakage. The charge accumulated in the capacitor provides a potential, which is proportional to the detected leakage current. Such a potential is applied to the gate terminal of a dedicated pMOS transistor, which serves as a voltage-to-current converter. The resulting current is then injected into the bit line, thus balancing the bit line leakage and guaranteeing stable memory access. Although effective, this scheme present two severe limitations. First, the detection/injection structure, which is based on a dynamic current mirror, is susceptible to threshold voltage variations induced by the fabrication process. Second, the gate voltage of the pMOS transistor that decides the injected amount of the compensation current is extremely sensitive to the coupling noise generated on the bitline during the pre-charge phase[15].

2) Sensing Calibration[15]: In [5], the authors propose an orthogonal approach which is based on the concept of bit line voltage calibration, that is, where the voltage difference sensed by the SA is calibrated in order to take into account the effect of the bitline leakage current. The calibration process, performed by a dedicated circuitry placed at the input of the SA (see Fig. 6), consists of a two-step process: i) Generation of the equilibrium offset and ii) cancellation of the offset[15].
In the first phase, which starts after the pre-charge cycle, the pull-up transistors are turned-off and the bitlines are left floating. This triggers a leakage-induced discharge transient, at the end of which the voltage on the bitline BL reaches an equilibrium level which defines the equilibrium offset across the bitline pair. This offset reflects the amount of bitline leakage current and it represents the actual voltage noise sensed by the SA. Bringing the same voltage offset on the complement bitline BLB would make the reading operation insensitive to leakage. That is the goal of the second phase, during which the generated offset is flipped over the BLB line thanks to a crossing structure. The reversed offset is then stored in a coupling capacitor and made available at the input of the SA. The offset voltage is now present on both the SA’s inputs; thus the differential sensing results leakage-free[15].

Compared to the BLC scheme, the calibration technique can handle a higher bitline leakage current at the cost of an extra area overhead and a bitline loading capacitance, resulting in higher power consumption and longer access times.

3) Self-Reverse Bias Bitline[15]: The idea behind this technique is to invert the pre-charging value of the bitline so as to force a low-leakage state on the un-accessed cells [6], as shown in Fig. 7.

Unlike conventional SRAM, characterized by a dynamic pull-up, the bitline is now pre-discharged to GND through a nMOS pull-down transistor. This arrangement imposes a different role for the bit-cell: Oppositely to the normal operations, here the bitcell drives the pull-up of the bitline when a logic “1” is stored and sustains the bitline pre-discharge in the case of a logic “0”. The polarity of the access transistors is thus inverted (i.e., source terminal connected to the bitline). However, since n-transistors exhibit a self shut-off when used as pull-up devices, during the evaluation phase the charging process stops when the bitline saturates at , where is the threshold voltage of the nMOS access transistor. Adding a pMOS buster stage to the bitline allows complete full-swing transitions. This helps the SRB memory to maintain constant robustness and read/write stability as conventional SRAMs[15].

The main advantage of the SRB structure is that, after the pre-discharging phase, the access transistors’ leakage causes the bitline to charge up towards . This induces a negative reverse-bias under-drive voltage on the access transistors that, in turn, lowers the bitline leakage. Fig. 8 compares the access transistor operation of a conventional SRAM to the SRB scheme. Moreover, the body effect on the access transistors increases due to the under-drive, which further elevates their . Finally, the drain-to-source voltage for the SRB access transistors is reduced due to the elevated source voltage, which further lowers the drain-to-source leakage due to mitigated drain-
induced barrier lowering (DIBL).

4) Floating Bitline: In [7], the authors propose a design technique that allows bitline floating by turning off the pre-charging transistors. The leakage current from the bit cells automatically biases the bitline to a midrail voltage that reduces the bitline leakage current. This current depends on the data pattern stored in the SRAM column. As a result, the amount of leakage current may change at each writing operation. This also impacts the effectiveness of the scheme: If all the cells store a “0”, the leakage currents will fully discharge the bitline BL(Fig. 9), while the BLB will be held high. If all the cells store a one, the BL will be held high and the BLB is discharged [15]. Typically, a mix of ones and zeros biases the bitline to a midrail voltage. Since bitline voltage floats to an undefined level, the SA must be disconnected from the local bitline to avoid other sources of power consumption. In Fig. 9, this is done through additional switches that isolate the global bitline [15].

5) Negative Wordline: It has been shown in [8] that applying negative voltage to inactive wordlines successfully cuts off the bitline leakage. A wordline under-drive, in fact, imposes a negative gate-to-source voltage on the access transistors, which in turn show a reduced subthreshold conductance. This method, however, has never been used alone on real SRAMs, because it suffers from degradation of device reliability since the oxide of the pass-gate is overstressed.

To solve these issues, the authors of [9] propose the use of statically lower voltages for both storage cells and bitlines. As illustrated in Fig. 10, the wordline drivers are supplied with a global supply voltage and a negative (hundreds of mV). The access transistors of the selected cell are then driven by a gate voltage equal to , while all the other unselected transistors are under-driven at . For the storage cells and the bitlines, instead, a supply voltage lower than the global is used, i.e. This ensures that gate-to-source and gate-to-drain voltages of any transistor do not exceed the voltage limit of , thus preserving the stability of the SRAM while minimizing the leakage power significantly. Needless to say, there is a dynamic power overhead for generating and differentiating the supply voltages in the SRAM layout. [15]

6) 8T Cell: In [10], the authors propose the use of a leakage equalized 8-transistor storage cell. As shown in Fig. 11, the new cell uses a modified access structure made of two additional access transistors, N3 and N4. The two transistors, which are permanently turned off, are sized so as to match the two standard access transistors N1 and N2, and serve as leakage compensation devices. Thanks to this symmetric structure,
Fig. 12. Basic Asymmetric SRAM Cell and its Improvements. [15]

the 8T memory cell injects identical leakage currents into the two bitline rails BL and BLB. In fact, after the pre-charge cycle, when both bitlines are charged up to equals, thus eliminating the differential offset voltage of the contending bitlines. Clearly, this structure assures a fully balanced bitline voltage offset only at the beginning of the read operation, i.e., immediately after the pre-charge cycle. As soon as the differential voltage on the bitline pair increases (due to the evaluation phase), the leakage equalization effect degrades. However, the SA operates in a small sensing windows at the beginning of the evaluation phase, when the leakage is fully balanced. Although no special timing and/or additional control structure in needed, the structure has about 40% area overhead, if compared to a conventional 6-transistor cell[15].

IV. DISCUSSION[15]

Since comparing the various solutions in terms of absolute leakage savings is clearly unfeasible, this section presents a qualitative comparison among the various solutions. Although serving as a summary of this work, such analysis provides useful design insights for SRAM architects and designers. The analysis ranks the various categories of approaches according to three abstract metrics, namely, Efficiency (i.e., tradeoff between leakage reduction and implementation overhead), Tolerance to Variability (i.e., robustness to process variability), and Scalability (i.e., how efficiency scales with technology scaling). For each class of solutions, a qualitative score (Poor, Moderate, Good) has been assigned (see Table IV). 

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Efficiency</th>
<th>Tolerance to Variability</th>
<th>Scalability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bitline Design</td>
<td>Poor</td>
<td>Poor</td>
<td>Good</td>
</tr>
<tr>
<td>Wordline Design</td>
<td>Poor</td>
<td>Moderate</td>
<td>Good</td>
</tr>
<tr>
<td>Bitcell Design</td>
<td>Good</td>
<td>Poor</td>
<td>Poor</td>
</tr>
<tr>
<td>DVS-based DPM</td>
<td>Moderate</td>
<td>Poor</td>
<td>Poor</td>
</tr>
<tr>
<td>PG-based DPM</td>
<td>Moderate</td>
<td>Good</td>
<td>Good</td>
</tr>
<tr>
<td>RBB-based DPM</td>
<td>Good</td>
<td>Moderate</td>
<td>Poor</td>
</tr>
</tbody>
</table>

TABLE IV

QUALITATIVE COMPARISON OF LEAKAGE REDUCTION STRATEGIES[15]

Efficiency: Bitline and wordline design strategies have limited effectiveness at a relatively high implementation cost and performance overhead. This is because the share of bitline leakage with respect to the entire budget is small. Concerning overheads, bitline techniques, and in particular those that are based on pre-charging, may induce sensible read/write latency degradation due to extra pre-charge cycles. Similarly, wordline design schemes, and in particular those based on special value assignment, are characterized by large dynamic power overhead for generating the special voltage levels. Conversely, bitcell design strategies are more effective, but they must carefully weight side effects such as area and cell stability. Solutions based on DPM can be considered, overall, the most efficient. All of them are very flexible since they can be applied at different levels of granularity Power gating (PG) is the most effective of all (it almost nullifies standby leakage power), but it requires a sleep transistor that may induce delay penalty in active mode; moreover, data are lost when the memory moves to the standby mode. Voltage scaling (DVS) has no overhead during the active periods, but it impairs stability during idle periods. Body voltage modulation (RBB) has the least overhead (access time and cell stability are unaffected since the bias is zero during active periods), with benefits comparable to those of DVS.

Tolerance to Variability: Sensitivity to parameter variations is another important metric. Bitline and wordline design schemes tend to be very susceptible to parametric variations: They rely on the compensation of bitline currents achieved by carefully-sized devices, which, however, due to variability, cannot be precisely designed as intended. Similar considerations apply to bitcell design strategies, which play with the threshold voltage of the internal transistors (known for being the most variable parameter). More tolerant are those approaches that assign special voltage values to the wordlines, even if parametric variations may reduce the resulting leakage savings. Concerning DPM, DVS is the most sensitive to process variations; due to variability of the threshold voltage, the margin between and the minimum retention voltage can become excessively small, with a negative impact on the stability of the data. Similar arguments apply to RBB. PG is the most robust architecture: Variations in the sleep transistor may have some performance impact, but its effectiveness in leakage reduction is only marginally sensitive to variability.

Scalability w.r.t. Technology: The main effect of technology scaling that impacts the proposed techniques is related to the reduction of the overdrive voltage, which has a significant impact on most of the leakage reduction techniques. A reduced gate overdrive, for instance, limits the voltage scaling that can be applied during data retention. This has negative impacts on DVS-based solutions. Similarly, having nominal closer to makes the control of intermediate implants quite difficult, thus making the use of asymmetric bitcells more challenging. The effects of CMOS scaling on RBB-based approaches is even more critical. These schemes become less
effective in nanoscale dimensions due to worsening of the body effect caused by shorter channel length. The efficacy of power-gating, on the other hand, does not show any evident limitation with CMOS scaling. The same applies to bitline and wordline design strategies, which however may require special design efforts due to difficulties in controlling the sensing of bitline leakage.

V. CONCLUSIONS AND FINAL REMARKS[15]

In this paper, we have provided an exhaustive overview of solutions for reducing leakage power in SRAMs. While all the proposed strategies share the same objective of leakage power reduction, they are quite diverse in many aspects.

Drawing a possible set of universal guidelines for SRAM designers and architects is not immediate. However, we can observe a couple of facts that allow a few suggestions that can have general value.

a) Exploit Orthogonality of Strategies: Although some techniques (e.g., bitline and wordline design) have a moderate impact in absolute terms, they are orthogonal to techniques that are based on DPM. The same consideration applies to the customized design of the bitcell. Therefore, whenever the redesign of the internals of the SRAM architecture is allowed, designer should try to apply such techniques to decrease the leakage cost of basic memory operations (bitline/wordline access and reads/writes).

b) Technology Matching: Although many techniques do not scale nicely with technology and/or scaling, designers should try to match the various techniques with the target technology. A technique may become less relevant in future technologies but might be the most suitable for the current ones.

A good example is body-biasing. Although it is expected to become less efficient for nodes beyond the 32 nm, it represents a good solution for 65 nm or 90 nm nodes. Therefore, especially in the embedded domain where technology scaling is limited by the integration of other types of technology (e.g., embedded FLASH memories) and mixed technologies on the same chip are not uncommon [36], body-bias still remains an efficient knob to control leakage.

REFERENCES:


