Low-leakage SRAM Design with Dual V_t Transistors

Behnam Amelifard
University of Southern California
Los Angeles, CA
(213) 740-9481
amelifar@usc.edu

Farzan Fallah
Fujitsu Labs of America
Sunnyvale, CA
(408) 530-4544
farzan@fla.fujitsu.com

Massoud Pedram
University of Southern California
Los Angeles, CA
(213) 740-4458
pedram@ceng.usc.edu

Abstract - *This paper presents a method based on dual threshold voltage assignment to reduce the leakage power dissipation of SRAMs while maintaining their performance. The proposed method is based on the observation that the read and write delays of a memory cell in an SRAM block depend on the physical distance of the cell from the sense amplifier and the decoder. The key idea is thus to realize and deploy different types of six-transistor SRAM cells corresponding to different threshold voltage assignments for individual transistors in the cell. Unlike other techniques for low-leakage SRAM design, the proposed technique incurs no area or delay overhead. In addition, it results only in a slight change in the SRAM design flow. Finally, it improves the static noise margin under process variations. Experimental results show that this technique can reduce the leakage-power dissipation of a 64Kb SRAM by more than 35%.*

1. Introduction

Due to technology scaling, reducing the leakage power dissipation has become one of the most important criteria in the design of VLSI systems. The leakage power dissipation is roughly proportional to the active area of a circuit. In many processors, caches occupy about 50% of the chip area [1]. Therefore, the static power dissipation of a cache is one of the key components of power dissipation in microprocessors. A number of researchers have addressed this problem. In [3], the authors proposed using asymmetric SRAM cells to reduce the leakage. Their method takes advantage of the fact that in ordinary programs most of the bits in data and instruction caches are zero. By including the device-level optimization into circuit-level techniques, reference [4] presented a forward body-biasing technique for active and standby leakage power reduction in cache memories. In [5], the authors proposed a dynamic V_t technique to reduce the leakage power in SRAMs. In their method, the threshold voltage of the transistors of each cache line is controlled separately by using body biasing. In [6], the use of a diode-connected PMOS bias transistor to control the virtual ground has been proposed.

Although many techniques have been proposed to address the problem of low-leakage SRAM design, most of them result in hardware overhead and hence increase chip's area and reduce the manufacturing yield. In this paper we present a method for low-power SRAM design based on using different types of cells with different threshold voltage assignments. The idea is that due to the non-zero delay of interconnects different cells in a memory array have different read and write delays. Therefore, the leakage power consumption can be reduced by using a high threshold voltage for some transistors. This technique has four main advantages over previous techniques:

- It does not have any hardware overhead,
- It does not have any delay overhead,
- It does not need a drastic change in the SRAM design flow, and
- It improves the static noise margin under process variation.

The remainder of this paper is organized as follows. In Section 2 the structure of an SRAM block is discussed. In Section 3 our idea for reducing the leakage power dissipation is presented. Section 4 shows the experimental results, while Section 5 concludes the paper.

2. SRAM Design

A typical SRAM block, shown in Figure 1, consists of cell arrays, address decoders, column multiplexers, sense amplifiers, I/O, and a control circuitry. The functionality of the control circuit is to generate internal signals of the SRAM. In the following, the functionality and design of other components are briefly discussed.

A. SRAM Cell

Figure 2 shows a 6-transistor (6T) SRAM cell. The bit value stored in the cell is preserved as long as the cell is connected to a supply voltage whose value is greater than the Data Retention Voltage (DRV) [7]. This feature, which is due to the presence of cross-coupled inverters inside the 6T SRAM, holds independent of the amount of leakage current.

Traditionally all cells used in an SRAM block are identical (i.e., the transistors with the same name in two different cells have the same width and threshold voltage) which results in identical leakage characteristic for all cells. However, as we shall demonstrate later, by using non-identical cells, yet with the same layout footprint, one can realize more power efficient designs.

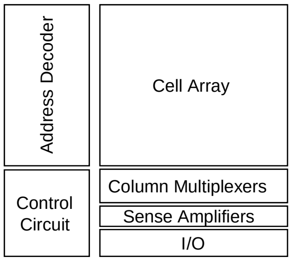


Figure 1. An SRAM block.

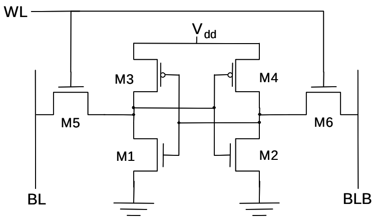


Figure 2. A 6T SRAM cell.

There are two dominant leakage paths in a 6T SRAM cell: 1) V_{dd} to ground paths inside the SRAM cell and 2) the bit line to ground paths through the pass transistors. To reduce the first type of leakage, the threshold voltages of the pull-down NMOS transistors and/or pull-up PMOS transistors may be increased, whereas to lower the second type of leakage, the threshold voltages of the pull-down NMOS transistors and/or pass transistors can be increased.

B. Cell Array

Usually there is more than one cell array in an SRAM circuit. Figure 1 shows only one of them. The size of the cell array depends on both performance and density requirements. Generally speaking, as technology shrinks, cell arrays are moving from tall to wide structures [2, 13]. However, since using wider arrays needs more circuitry for column multiplexers and sense amplifiers, in cases where a large area overhead is intolerable (e.g., large L3 caches), the number of rows can be still high [8, 9].

C. Address Decoder

Although the logical function of an address decoder is very simple, in practice designing it is complicated because the address decoder needs to interface with the core array cells and pitch matching with the core array can be difficult [10]. To overcome the pitch-matching problem and reduce the effect of wire's capacitance on the delay of the decoder, the address decoder is often broken into two pieces. The first piece, called *predecoder*, is placed before the long decoder wires and the second part, *row decoder*, which usually consists of a single NAND gate and buffers for driving the word-line capacitance, is pitch-matched and placed next to each row as shown in Figure 3 [10].

D. Column Multiplexers and Sense Amplifiers

Column multiplexing is inevitable in most SRAM designs because it reduces the number of rows in the cell array and as a result increases the speed. Since bit or bit-bar line is discharged about $200mV$ during a read operation, a sense amplifier is used to sense a small voltage difference and generate a digital value.

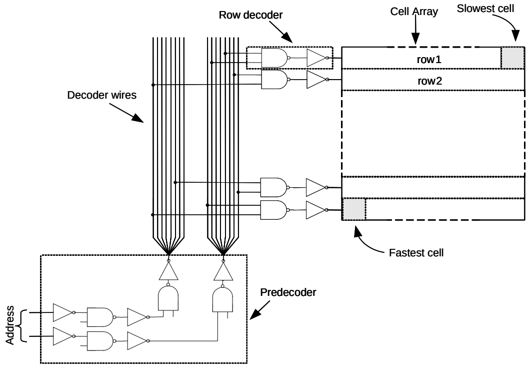


Figure 3. An SRAM block with its decoder.

3. Hybrid Cell SRAM

Due to the non-zero delay of the interconnects of the address decoder, word-lines, bit-lines, and the column multiplexer, read and write delay of cells in an SRAM block are different. Simulations show that for a typical SRAM block, the read time of the closest cell to the address decoder and the column multiplexer is 5-10% less than that for the furthest cell (cf. Figure 3.) This gives an opportunity to reduce the leakage power consumption of the SRAM by increasing the threshold voltage of some of the transistors in SRAM cells.

On the other hand, due to the delay of sense amplifiers and output buffers in a read path, the write delay of a cell is lower than its read delay. Considering the fact that increasing the threshold voltage of the PMOS transistors in a 6T SRAM cell increases the write delay, without having much effect on its read delay, one may try to reduce the leakage power by increasing the threshold voltage of the PMOS transistors as long as the write time is below some target value.

It is known that each additional threshold voltage needs one more mask layer in the fabrication process, which results in increasing the fabrication cost [11]. At the same time, there are studies that show the benefit of having more than two threshold voltages is small [11]. As a result, in many cases, only two threshold voltages are utilized. Therefore, we concentrate on the problem of low-leakage SRAM design when only two threshold voltages are available. However, it is possible to extend the results to handle more than two threshold voltages.

A. SRAM Cell Configurations

To reduce the leakage power consumption of a cell, the threshold voltage of all or some of the transistors of the cell can be increased. If the threshold voltage of all transistors within a cell is increased, the leakage reduction is the highest; however, since this scenario has the worst effect on the read delay of the cell, the number of cells that can be replaced is low. Thus, we consider other configurations that have smaller leakage reductions due to their lower delay overheads.

Unlike [3], we use a symmetric cell configuration, which means the symmetric transistors within a cell have the same threshold voltages. Thus, there are eight different possibilities for assigning high and low threshold voltages to the transistors within a cell. These configurations are shown in Table 1. Here it is assumed that the threshold voltage of each transistor in the SRAM cell can be adjusted independent of other threshold voltages by changing the channel doping, which is deemed to be a safe assumption because in an SRAM cell the channels of the transistors are not too close to each other [14]. However, in the Simulation Results Section it will be shown that even if only one threshold voltage is used inside the cell, power reduction can be considerable.

The configuration ordering is such that, excluding C0 which is the original configuration, the leakage current saving of other configurations is monotonically decreasing as we move from C1 toward C7. Figure 4 shows the leakage current saving of different configurations versus the value of the high threshold voltage. The numbers in Figure 4 and the following ones are obtained using a 180nm CMOS technology with 1.8V for the supply voltage and 0.37V for the low threshold voltage at 100^0C.

Each of these configurations has different effects on read and write delays of cells. Figures 5 and 6 show the increase in read and write delays for each configuration for different values of the high threshold voltage. It can be seen that the increase in read delay for some configurations (e.g., C1 and C3) is very high, whereas it is very small for some other configurations (e.g., C6). It can also be seen from Figure 6 that not all configurations increase the write time; In fact, C4 and C7 decrease the write time.

Table 1. Possible configurations for high threshold voltage assignment

Config.	High threshold transistors
C0	None
C1	M1, M2, M3, M4, M5, M6
C2	M3, M4, M5, M6
C3	M1, M2, M5, M6
C4	M1, M2, M3, M4
C5	M5, M6
C6	M3, M4
C7	M1, M2

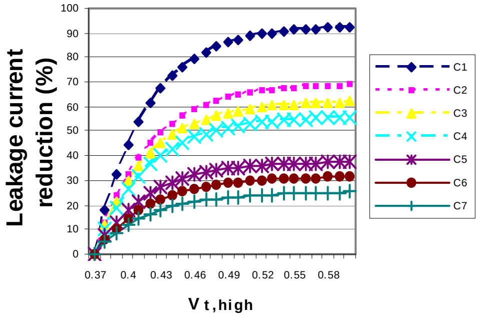


Figure 4: Leakage current reduction for each configuration.

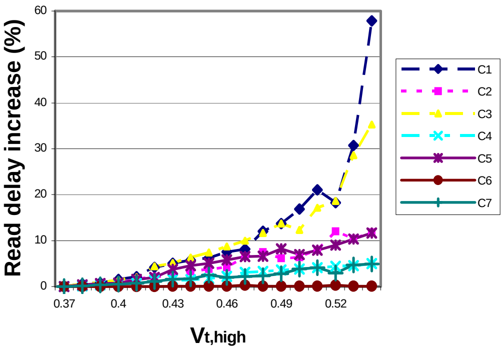


Figure 5: Read delay increase for each configuration.

B. Noise Margin

The static noise margin (SNM) of a CMOS SRAM cell is defined as the minimum DC noise voltage necessary to flip the state of a cell [12]. SRAM cells are especially sensitive to noise during a read operation because the "0" storage node rises to a voltage higher than ground due to a voltage division along the pull-down NMOS transistor and the pass transistor; if this voltage is high enough, it can change the cell's value. In general, it is expected that by using high threshold transistors in the SRAM cells, the static noise margin would increase. We measure the SNM of each configuration under two scenarios: nominal condition and process variation.

B.1 Static Noise Margin under Nominal Conditions

Simulation results shown in Figure 7 confirm that for all configurations except C6 (i.e., when only PMOS transistors are high threshold), the nominal SNM is more than that of C0 and improves with increasing the high threshold voltage. In C6, however, the SNM is slightly less than that of C0 and degrades with increasing the high threshold voltage.

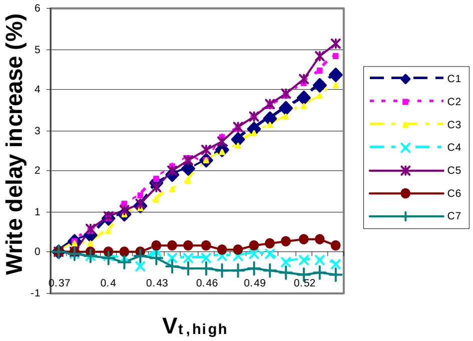


Figure 6: Write delay increase for each Configuration.

B.2 Static Noise Margin under Process Variation

Technology scaling has made the process variation one of the concerns of designers. As the minimum size transistors are typically used in SRAM cells to achieve a compact design [2], SRAMs are very sensitive to process variation. To measure the static noise margin of each configuration under process variation, we used the Monte Carlo simulation technique. For each configuration we used 400 vectors of six threshold voltages, where each threshold voltage has a Gaussian distribution with its mean equal to the nominal value and its $3\sigma/\mu$ equal to 0.15.

The simulations show that the distribution of SNM of each configuration under process variation can be approximated by a Gaussian distribution. The mean and standard deviation of SNM for different configurations are shown in Table 2. Here the important value is μ-3σ and as one can see in the table, all configurations have a better μ-3σ than C0.

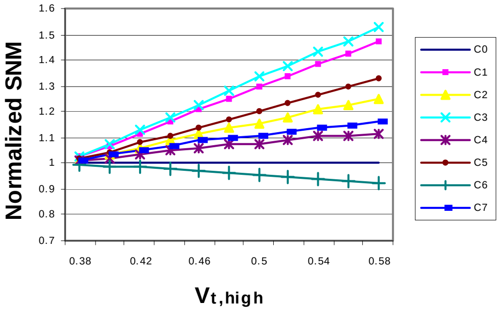


Figure 7: SNM for different configurations under the nominal condition.

Table 2. Mean and standard deviation of SNM for different configurations ($V_{t,high}$=0.47V).

Config.	μ	σ	μ-3σ
C0	0.255	0.0217	0.190
C1	0.312	0.0235	0.242
C2	0.287	0.0224	0.220
C3	0.318	0.0187	0.262
C4	0.269	0.0175	0.217
C5	0.293	0.0165	0.244
C6	0.245	0.0143	0.202
C7	0.276	0.0268	0.196

C. Hybrid Cell Assignment

To design a hybrid-cell SRAM, we need to find out the slowest read and write delay starting with all low-V_t SRAM cells (C0 case.) Next, since C1 results in the highest leakage reduction among all configurations, we replace as many C0 cells as possible with C1 cells in such a way that the access delay of the replaced cells will not be larger than the slowest access delay. After that, we try to replace the remaining C0 cells with C2, C3, C4, C5, C6, and C7.

Figure 8 shows the pseudo-code of the hybrid cell assignment. *rownum* and *colnum* are the number of rows and columns of the SRAM, respectively. Moreover, v_{tH_lb} and v_{tH_ub} are the lower and upper bounds of the high V_t value, respectively. The fastest cell is denoted by index [0, 0], while the slowest one is denoted by index [*colnum*-1, *rownum*-1]. Subroutines *ReadDelay(col, row, config)* and *WriteDelay(col, row, config)* return the read and write delays of cell [*col*, *row*] when configuration *config* is used. If cell [*col, row*] fails working with configuration *config*, then all cells [*i*, *j*], where $i \geq col$ and $j \geq row$ fail with the same configuration. Therefore, a large number of cells can be pruned as soon as a cell fails working for a given configuration. In the pseudo-code, *flag*[*config*][*col*, *row*] is a flag that specifies if *cell*[*col*, *row*] can work with configuration *config*.

Initially all flags are set to 1. In the next section, it will be shown that this algorithm, despite its simplicity, results in a significant power reduction in SRAM blocks.

To speed up the process, instead of checking for possible replacement on each single SRAM cell, one can select $n \times n$ blocks and do the checking for the slowest cell in the block. If the slowest cell passed the delay test, the whole block is replaced; otherwise, next configuration or block is examined. Here n is a multiple of two and it is clear that choosing a larger number for n decreases the design time, but degrades the result.

```
Hybrid-Cell-Assignment (rownum, colnum, $v_{tH\_lb}$, $v_{tH\_ub}$)
Begin
  $T_{max}$=ReadDelay (colnum-1, rownum-1, C0)
  For $v_{t,high}$= $v_{tH\_lb}$ to $v_{tH\_ub}$
    For config=C1 to C7
      For (0≤col<colnum, 0≤row<rownum)
        flag[config][col,row]=1;
    For col=0 to colnum-1
      For row=0 to rownum-1
        For config=C1 to C7
          If (flag[config][col,row] ==1)
            If (ReadDelay(col,row,config)<$T_{max}$
                && WriteDelay(col,row,config)<$T_{max}$)
              Replace cell[col][row] with config; Break;
            Else
              For (i≥col, j≥row)
                flag[config][i,j]=0;
End
```

Figure 8. Pseudo-code for the hybrid cell assignment.

4. Simulation Results

To study the efficiency of the proposed technique, a 400MHz, 64Kb SRAM with a 64-bit word has been designed and simulated using Cadence UltraSim in 180nm CMOS technology with 1.8V for the supply voltage and 0.37V for the low threshold voltage. The SRAM consists of two 256×128 cell arrays. For optimizing the delay of the decoder, the predecoding scheme has been used as described in Section III. The simulations results in this section are pre-layout. However, by modeling all local and global interconnects, including bit and bit-bar lines, word line, and decoder wires as distributed RC circuits, the accuracy of the simulations has been improved.

Table 3 shows the leakage power reduction achieved and the utilization of each configuration in the low-leakage SRAM for different values of the high threshold voltage. One interesting observation from Table 3 is that the leakage saving is not very sensitive to the threshold voltage value if the value is larger than 0.42V.

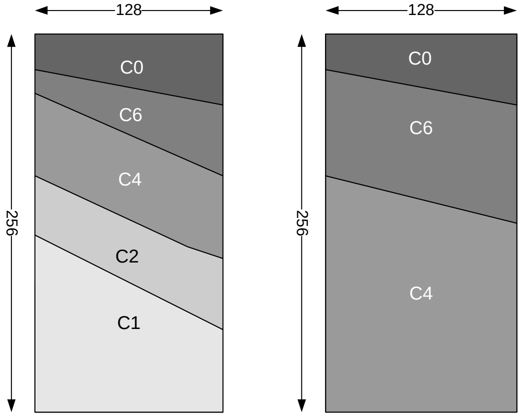
From Table 3 it can be seen that C3 is rarely used in the low-leakage SRAM. The reason is that the delay of this configuration is almost equal to the delay of C1, but its leakage is higher. Therefore, in most cases, C1 is used instead of C3. Similarly, from Figure 5 one can see that the increase in the read delay of C5 and C7 are almost equal to those of C2 and C4, respectively; so it is not surprising that in Table 3 most of the entries for C5 and C7 are zero. Figure 9 illustrates the approximate floor-planning of each type of cell for two values of the high threshold voltage (leakier areas are shown in darker gray.)

Table 3. The power reduction and the utilization of each configuration in the low-leakage SRAM.

$V_{t,high}$	Leakage Reduction (%)	Utilization of Each Configuration (%)							
		C0	*C1*	*C2*	*C3*	*C4*	*C5*	*C6*	*C7*
0.38	**14.64**	13.5	76.2	3.1	0.8	2.1	1.2	3.1	0
0.39	**25.00**	13.7	65.4	7.6	0	6.4	0	6.8	0
0.40	**31.84**	13.7	52.9	10.0	0	16.0	0	7.4	0
0.41	**36.48**	13.7	46.9	8.2	0	18.6	0	12.7	0
0.42	**38.77**	14.1	33.6	15.0	0	24.0	0	13.3	0
0.43	**36.55**	14.1	8.0	33.8	0	25.4	0	18.8	0
0.44	**34.74**	14.1	0.6	22.1	0.2	38.1	0	25.0	0
0.45	**35.06**	14.1	0	9.4	0.2	51.0	0	25.4	0
0.46	**34.66**	14.1	0	2.5	0	52.5	0	30.9	0
0.47	**35.30**	14.1	0	0	0	53.9	0	32.0	0
0.48	**35.14**	14.1	0	0	0	49.0	0	36.9	0
0.49	**34.54**	14.1	0	0	0	43.0	0	43.0	0
0.50	**33.61**	15.8	0	0	0	38.7	0	45.5	0
0.51	**33.36**	16.2	0	0	0	36.1	0	47.7	0

By observing the fact that C3, C5, and C7 are dominated by other configurations, they may be deleted from the set of candidate configurations, resulting in five configurations: C0, C1, C2, C4, and C6. If, for any reason, a designer likes to use a smaller number of configurations, the saving in the leakage will decrease. We have repeated our experiment for the case that only three configurations are utilized. In this case, the selection of suitable configurations depends on the value of $V_{t,high}$; for example, from Table 3 it can be seen that when $V_{t,high}$ is close to the low threshold voltage, the best configurations are C0, C1 and C2, while for higher value of $V_{t,high}$, C0, C4 and C6 ought to be used. Table 4 shows the configurations used in each case.

Table 5 reports the power reduction of the SRAM block for different values of the high threshold voltage when only three different configurations are utilized. From this Table it can be seen that even when using only three configurations, the power reduction is high. In fcat, in most cases the power saving is more than 30%. Table 5 also reports power reduction of the SRAM block when all threshold voltages in an SRAM



(a) (b)

Figure 9: The approximate floor-planning of the cell array after hybrid cell assignment (a) $V_{t,high}$ =0.42V (b) $V_{t,high}$ =0.47

Table 4. Three suitable configurations for different values of $V_{t,high}$

Case	Configurations
$V_{t\ high} \leq 0.42$	C0, C1, C2
$V_{t,\ high} > 0.42$	C0, C4, C6

Table 5: Power reduction with three and two different configurations

$V_{t,high}$	Power Reduction (%)	
	Three Configurations	Two Configurations
0.38	14.05	13.82
0.39	23.03	21.21
0.4	26.57	23.34
0.41	28.33	25.09
0.42	27.33	20.55
0.43	31.13	5.39
0.44	32.21	0.52
0.45	33.95	0.10
0.46	34.38	0.0
0.47	35.3	0.0
0.48	35.14	0.0
0.49	34.54	0.0
0.5	33.61	0.0
0.51	33.36	0.0

cell are high or all are low i.e., only configurations C0 and C1 are used. It is seen that in this case more than 25% power reduction is achieved.

5. Conclusions

In this paper we presented a novel technique for low-leakage SRAM design. Our technique is based on the fact that due to the non-zero delay of interconnects of the address decoder, word-line, bit-line and the column multiplexer, cells of an SRAM have different access delays; thus, the threshold voltage of some transistors of cells can be increased without degrading the performance. By using eight different configurations for the SRAM cells, we have achieved a low-leakage SRAM without scarifying performance and area. Moreover, the simulations have shown that our technique improves the static noise margin under process variation. By applying this technique to a 64Kb SRAM, we have achieved more than 35% reduction in the leakage-power dissipation.

Acknowledgment

We would like to thank Tom Sidle, the VP of advanced CAD Technology at Fujitsu Laboratories of America for supporting this research.

References

[1] C. Molina et al, “Non redundant data cache,” in *Proc. ISLPED*, 2003, pp. 274-277.

[2] K. Zhang et al., “SRAM design on 65-nm CMOS technology with dynamic sleep transistor for leakage reduction,” *IEEE J. Solid-State Circuits*, vol. 40, no. 4, Apr. 2005, pp. 895-901.

[3] N. Azizi et al, “Low-leakage asymmetric-cell SRAM,” *IEEE Trans. on VLSI Systems*, vol. 11, no. 4, Aug. 2003, pp. 701-715.

[4] C. H. Kim et al, “A forward body-biased low-leakage SRAM cache: device, circuit and architecture considerations,” *IEEE Trans. on VLSI Systems*, vol. 13, no. 3, Mar. 2005, pp. 349-357.

[5] C. Kim and K. Roy, “Dynamic Vt SRAM: a leakage tolerant cache memory for low voltage microprocessor,” in *Proc. ISLPED*, 2002, pp. 251–254.

[6] A. Bhavnagarwala et al “A pico-joule class, 1 GHz, 32 kB 64 b DSP SRAM with self reversed bias,” in *Proc. Symp. VLSI Circuits*, 2003, pp. 251–252.

[7] H. Qin et al, “SRAM leakage suppression by minimizing standby supply voltage,” in *Proc. of ISQED*, 2004, pp.56-60.

[8] K. Zhang et al, "A 3-GHz 70Mb SRAM in 65nm CMOS technology with integrated column-based dynamic power supply," in *Proc. ISSCC*, 2005, pp. 474–475.

[9] D. Weiss et al, “The on-chip 3MB subarray based 3rd level cache on an Itanium microprocessor,” in *Proc. ISSCC*, 2002, pp. 112–113.

[10] A. Chandrakasan et al., *Design of High-Performance Microprocessor Circuits*. IEEE press, NJ, 2001.

[11]A. Sirvastava, “Simultaneous Vt selection and assignment for leakage optimization,” in *Proc. ISLPED*, 2003, pp. 146-151.

[12] Seevinck et al, "Static-Noise Margin Analysis of MOS SRAM Cells," *Journal of Solid-State Circuits*, Vol. SC- 22, No. 5, pp. 748-754, Oct. 1987.

[13] F. Hamzaoglu et al., “Dual Vt-SRAM cells with full-swing single-ended bit line sensing for high-performance on-chip cache in 0.13μm technology generation,” in *Proc. of ISLPED*, 2000, pp. 15–19.

[14] L. Wei et al., “Mixed-Vth (MVT) CMOS circuit design methodology for low power applications,” in *Proc. of DAC*, 1999, pp. 430-435.