Speed and Power Scaling of SRAM's

Bharadwaj S. Amrutur and Mark A. Horowitz

***Abstract*—Simple models for the delay, power, and area of a static random access memory (SRAM) are used to determine the optimal organizations for a SRAM and study the scaling of their speed and power with size and technology. The delay is found to increase by about one gate delay for every doubling of the RAM size up to 1 Mb, beyond which the interconnect delay becomes an increasingly significant fraction of the total delay. With technology scaling, the nonscaling of threshold mismatches in the sense amplifiers is found to significantly impact the total delay in generations of 0.1 μm and below.**

***Index Terms*—Delay scaling, power scaling, scaling, speed scaling, static random access memory (SRAM), technology scaling.**

I. Introduction

HIGH-PERFORMANCE large-capacity SRAM's are a crucial component in the memory hierarchy of modern computing systems. This paper analyzes the scaling of delay and power of SRAM's with size and technology. SRAM design requires a balancing act between delay, area, and power consumption. The circuit styles for the decoders and the sense amps, transistor sizing of these circuits, interconnect sizing, and partitioning of the SRAM array can all be used as a tradeoff for these parameters. Exploring this large design space using conventional SPICE circuit simulation would be extremely time-consuming and, hence, simplified analytical models are very valuable. Such models not only help in designing SRAM's for the current generation, but can also be used to forecast trends for the future. Analytical models for delay, area, and energy have been developed separately by a number of authors [2]–[5]. Wada *et al.* [2] and Wilton and Jouppi [3] develop delay models for the decoder and the bit line path and use it to explore the impact of various cache organizations on the access time. Evans and Franzon develop analytical models in [4] and [5] for the energy consumption of a SRAM as a function of its organization. This paper extends the delay models of [2] and combines them with the energy and area models for the SRAM. The delay models are modified to include the effects of interconnect resistance and more complex partitioning schemes. We allow for multilevel hierarchical structures for the bit line and data line muxes [10], [11], which is an additional degree of freedom in the organization not considered by [2]–[5]. The models are then used to estimate the delay, area, and energy of SRAM's of various capacities and in different technology generations.

With technology shrinking by a factor of 2 every 18 months, two effects stand out: the interconnect is getting worse compared to the transistor and the threshold mismatches between transistors are not scaling with the supply voltage [15], [21]. One expects both these effects to have a significant influence on SRAM's, since SRAM's require information to be broadcast globally across the whole array, and part of the signal path within the array uses small-signal swings followed by sense amplification. The paper investigates both these effects with the aid of the analytical models.

We first review the organization of a typical SRAM and point out the essential features which influence its delay, area, and energy in Section II. To keep the analysis tractable, we make certain simplifying assumptions and discuss the main ones in Section III. An extensive list of all the other assumptions is provided in the Appendix. Using these assumptions, we then develop models for delay, area, and energy for the key components of the SRAM. We then apply these models to estimate the delay and power for SRAM's and investigate the scaling trends with densities and technology in Section IV.

II. SRAM Overview

Fig. 1 shows the typical architecture of an SRAM. The SRAM access path can be broken down into two components: the decoder, which is the portion from the address input to the word line, and the output mux, which is the portion from the cells to the output. In this paper, we focus on the read access as it determines the critical timing for the SRAM. For the read access, the address input is decoded to activate a specific word line. The decoder typically employs the divided word line structure [8] shown in Fig. 1, where part of the address is decoded to activate the horizontal global word line and the remaining address bits activate the vertical block select line. The intersection of these two activates the local word line. The cells connected to this word line transfer their data onto the bit lines. Data from a subset of bit lines is routed by the column mux into the sense amplifiers which amplify and drive it onto the data lines. Signals from the data lines are further amplified by the global sense amplifiers and finally driven out of the array. Energy dissipation in an SRAM has three components: 1) the dynamic energy to switch the capacitance in the decoders, bit lines, data lines and other control signals within the array; 2) the energy of the sense amplifiers; and 3) the energy loss due to the leakage currents.

Typically, a large array is partitioned into a number of identically sized subarrays (referred to as macros in this paper), each of which stores a part of the accessed word, called the subword, and all of which are activated simultaneously to access

Manuscript received February 3, 1999; revised October 1, 1999. This work was supported by the Advanced Research Projects Agency under Contract J-FBI-92-194 and by Fujitsu Ltd.

B. S. Amrutur was with the Center for Integrated Systems, Stanford University, Stanford, CA 94305 USA. He is now with Agilent Technologies, Palo Alto, CA 94304-1392 USA.

M. A. Horowitz is with the Center for Integrated Systems, Stanford University, Stanford, CA 94305 USA.

Publisher Item Identifier S 0018-9200(00)00948-3.

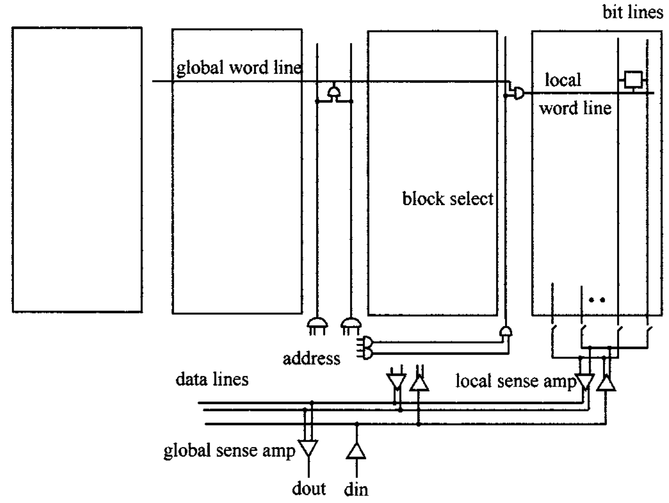


Fig. 1. SRAM access path.

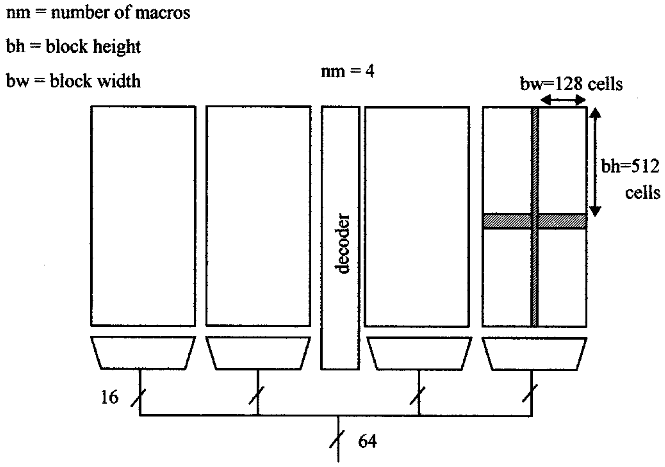
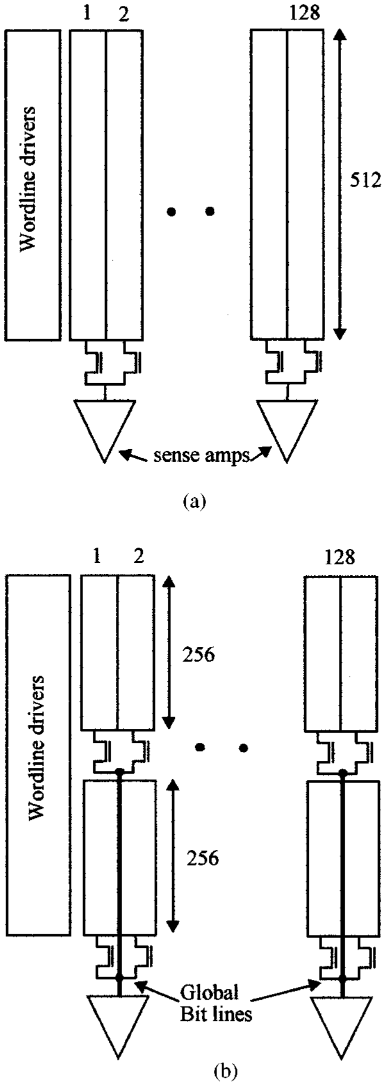


Fig. 2. Array partitioning example.

the complete word. The macros can be thought of as independent RAM’s, except that they might share parts of the decoder. Each macro is further subdivided into a number of blocks with the accessed subword residing completely within the block. In this paper, a block denotes an array of cells which are framed by the local word line drivers and the local sense amps and other column circuitry at their periphery. At the top level, any partitioning can be captured by three variables: the number of macros (nm) which comprise the array, the block width (bw) and block height (bh) of each of the subblocks which make up a macro. Fig. 2 shows an example of partitioning a 1024 × 1024 array of cells of a 1-Mb SRAM for a 64-bit access. The array is broken into four macros, all of which are accessed simultaneously, each providing 16 bit of the accessed word. Each macro is further subdivided into four blocks of 512 rows and 128 columns and with one of the blocks furnishing the 16-bit subword. When the block height is very large, it can be further partitioned to form multilevel bit line hierarchies by using additional layers of metal. In general, the multiplexor hierarchy can be constructed in a large number of ways ($2^{r-1} * 2^c$ mux designs are possible for a $2^r \times 2^{c+k}$ block with 2^r number of rows, 2^c number of columns, and an access width of 2^k bit). Fig. 3 shows two possible designs for the 512 × 128 block. The schematic shows only the nMOS pass gates for a single-ended bit line to reduce the clutter in the figure, while the real multiplexor would use CMOS pass gates for differential bit lines to allow for reads and writes. Fig. 3(a) shows the single-level mux design, where two adjacent columns with 512 cells each are multiplexed into a single sense amplifier. Fig. 3(b) shows a two level structure in which the first level multiplexes two 256 high columns, the output of which are multiplexed in the second level to form the global bit lines, feeding into the sense amplifiers. Similarly hierarchical muxing can also be done in the data line mux. This paper includes such multilevel mux hierarchies in the analysis.

Partitioning of the RAM incurs area overhead at the boundaries of the partitions. For example, a partition which dissects the bit lines requires sense amps, precharge, and write buffers to be inserted at the boundary. Partitions which dissect the word lines require the use of word line drivers at the boundary, and multilevel bit line muxes require space to be allocated for the mux transistors. Since the RAM area determines the lengths of



(a)

(b)

Fig. 3. Bit line mux hierarchies in a 512 × 128 block: (a) single-level mux and (b) two-level mux.

the global wires in the decoder and output mux, it directly influences their delay and energy. Hence, we estimate area as an integral part of the analysis. The next section details the assumptions made for the analysis and describes the models developed for the decoder and the output mux.

TABLE I
FEATURES OF THE BASE 0.25-μm TECHNOLOGY

Parameter	Value	Comments
C_g	1.29 fF/μm	Unit gate capacitance of an inverter (p-width = 2 x n-width)
C_j	1.72 fF/μm	Unit parasitic junction capacitance of an inverter
R_g	4.4 KΩ-μm	Unit output Resistance of an inverter
τ_{fo4}	90pS@2.5V	Delay of an inverter driving a load four times its size.
M1 width	0.45μm	Minimum width of metal 1
M1 aspect ratio	1.5	Aspect ratio (height/width) of metal 1 cross section.
M1 cap	0.21 fF/μm	Capacitance of metal1
M1 res	0.1 Ω/μm	Resistance of metal1, (aluminum)

TABLE II
TECHNOLOGY SCALING OF SOME PARAMETERS

L (μm)	C_g (fF/μm)	τ_{fo4} (pS)	Supply (V)	M1 min. Width (μm)	M1 aspect ratio	M1 cap (fF/μm)	M1 res, type (Ω/μm)
0.25	1.29	90	2.5	0.45	1.5	0.21	0.1, al
0.18	1.29	65	1.8	0.324	1.77	0.23	0.12, cu
0.10	1.29	36	1.2	0.18	2.37	0.29	0.29, cu
0.07	1.29	25	0.9	0.126	2.83	0.33	0.49, cu

III. MODELING OF THE SRAM

In order to explore the large SRAM design space in a tractable manner, we make some simplifying assumptions about certain aspects of the design. We outline and justify the key assumptions in the next subsection and list all the assumptions in the appendix. We then develop simple analytical models for delay, area and power for the various SRAM components and the verify these against HSPICE circuit simulations. These models are then used to explore the performance of a large range of SRAM organizations of various sizes and in different technology generations to determine optimal configurations and scaling trends and these results are discussed in the following section.

A. Assumptions

The base technology used for this analysis is a 0.25 μm CMOS process and the relevant process details are shown in Table I. A convenient process independent unit of length called λ is used to describe geometric parameters in the paper. λ is equal to half the minimum feature size for any technology. We assume that all the device speeds and dimensions scale linearly with the feature size. The supply scales as in [1] and the wires scale as in [17] with copper metallurgy from 0.18-μm generation onwards. The key features for four different generations used for the analysis are shown in Table II.

Mizuno *et al.* show in [21] that the dominant source of threshold variations in closely spaced transistors in deep submicrometer geometries is the random fluctuations of the channel dopant concentrations. They also show that this portion of the threshold mismatch remains constant with process scaling (see also [15]). So we assume a constant mismatch of 50 mV in the thresholds of the input differential pair in the sense amplifiers, irrespective of the process generation.

We model the delay and energy of the RAM core and ignore the external loads as they are a constant independent of the internal organization. Since the read access results in the critical timing path for the RAM, only the delay and power consumption of the read operation is modeled.

We assume a static circuit style for all the gates in the RAM to simplify the modeling task. The pMOS portion of the gate is sized to yield the same delay as the nMOS portion, and hence the gate can be characterized by a single size parameter. Since high-speed SRAM's commonly skew the transistor sizes in the decoder gates to improve the critical path, we will quantify its impact on our delay analysis.

There is a large diversity in the circuits used for the sense amplifiers. In this paper, we will assume a latch style amplifier which consists of a pair of cross-coupled gain stages which are activated by a clock [7], [13], [19], [22]. In these structures, the amplifier delay is proportional to the logarithm of the required voltage gain [18]; hence, if the sense clock timing is well controlled, they lead to the fastest implementations. They also consume very low power since they are inherently clocked. We will assume that the sense clock timing is perfectly controlled but will quantify the impact of nonideal sense clock generation on our delay estimates.

When the number of wiring levels is limited, space has to be allocated in between blocks to route the data lines. This significantly adds to the overall memory area, especially when the block size is very small. Since the number of available wiring levels has been steadily growing [1], we will assume in this paper that the data lines can be routed vertically over the array if required. Thus, extra routing space for a horizontal bus is required only once at the bottom of the array.

Transistor sizing offers another technique to tradeoff delay, area, and power. In this paper, we assume that the gates in the access path are sized to give minimum delay to simplify the analysis. Hence, the fanout of each logic gate is chosen to yield a delay of that of a fanout of four loaded inverters. While this assumption does not affect minimum delay solutions, it causes the low-energy and low-area solutions to be suboptimal.

A simple RC model is used for the logic gates [23]. Since the gate is sized to have equal rising and falling delays, a single-size parameter w, which is the size of the nMOS transistor in an equivalent inverter having the same output resistance, is used to represent the gate. Let C_g be the input capacitance per unit width and R_g be the output resistance for a unit width of an inverter. Then the output resistance of the gate (and the equivalent inverter), of size w, is $R_o = R_g/w$. The input capacitance of the gate is $C_g * 3w * le$, where le is the logical effort of the gate and captures the relative input capacitance of the gate with respect to the inverter (whose pMOS size is $2w$), due to the logical function it implements (Sutherland and Sproull in [6]). The delay of a logic gate of size w, driving a load C_L through a wire of resistance R_w and capacitance C_w (Fig. 4), is estimated as in (1) by using the simple approximation proposed by Elmore in [9]. Here p is the intrinsic

$$D = \underbrace{R_o \cdot (C_L + C_w) + p}_{\text{gate delay}} + \underbrace{R_w \cdot \left(\frac{C_w}{2} + C_L\right)}_{\text{wire delay}} \quad (1)$$

delay of the gate due to its drain junction capacitance. In an energy-efficient SRAM design, the dynamic power dominates the total power dissipation, and so we only model the dynamic energy required to switch the capacitances in the decoder and the output mux. We will next discuss in detail the models for the decoder and the output mux.

B. Decoder

The decoder has two components: the row decoder which activates the word lines and the column decoder which sets the switches in the bit line and data line mux. Since the row decoder lies in the critical path of the RAM access, we model its delay, while the energy of both the row and column decoders is modeled. The decoder critical path is modeled by a string of three chains of logic gates each comprised of NAND gates and inverters, with the chains connected together by RC sections. The entire decode path is driven by an inverter at the input which has a minimum size of 16λ for pMOS and 8λ for nMOS. Fig. 5 sketches the critical path of a 12–4096 decoder from the address input, through the chains of the predecoder, the global word driver, and the local word driver. The global and local word driver chains consist of one 2-input NAND gate followed by inverters, since using a fanin 2 structure for these two chains minimizes both the delay and power of the decoder. The predecoder chain is made of a collection of 2 or 3 input NAND gates and inverters, to obtain the desired fanin for the decode path with the minimum logical effort (see [6] for a table of NAND gate compositions which result in the minimum logical effort implementation of the AND function). Since the local word drivers are located at regular intervals along the global word line, their loading is taken into consideration by distributing their input capacitance all along the global word line wire. Since the slowest predecode wire is the one which has all its global word drivers located at its extreme end, the decoder critical path model lumps all the input capacitance of the global word drivers at the end of the predecode wire. The delay of each stage in the decode path is then computed using the simple delay formula shown in (2). Each stage is sized to have the delay of a fanout-of-4 inverter to minimize the delay. When wire resistance of the predecode and the global word lines are not negligible (r_p, r_g in Fig. 5), then extra buffering will be required in the global and local word driver chains to reduce the impact of gate loading of these chains on their respective resistive inputs. The optimum number of buffer stages is easily found in a few iterations by computing the decode delay with various numbers of buffer stages at these two locations.

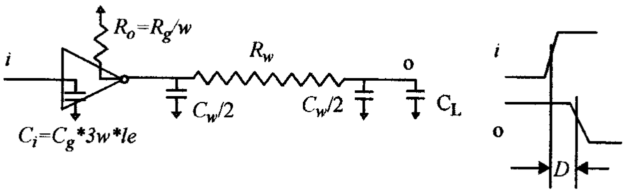


Fig. 4. Delay of a logic gate driving a load through an RC line.

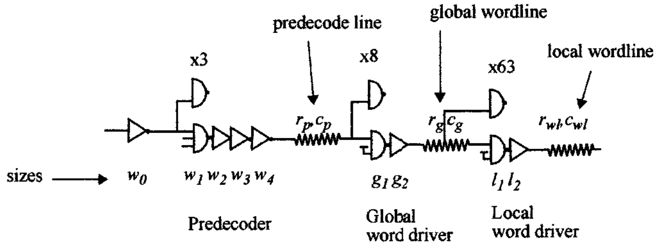


Fig. 5. Model of the critical path of a 12–4096 row decoder.

The decoder delay with fanout-of-4 sizing rule is summarized in (2) and is the sum of the extrinsic delays of each gate in the path (each of which is equal to the extrinsic delay of the fanout-of-4 inverter, τ_{fo4x}), their intrinsic delays, and the wire delays (here N is the number of gates in the path, p_i is the parasitic delay of gate i). Let G be the loading due to inputs of all the global

$$D_{\text{decoder}} = N \cdot \tau_{fo4x} + \sum_{i=0}^{N} p_i + D_{\text{wire}} \quad (2)$$

word drivers connected to the predecode wire. For the slowest predecode wire, all these gates are driven at the extreme end of the wire resulting in the predecode wire delay being $r_p * (c_p/2 + G)$. Let L be the loading due to the inputs of all the local word drivers connected to a global word line. In a real SRAM, the local word drivers are uniformly spaced at discrete points along the global word line, but we will model its capacitance as being uniformly distributed across the entire wire, making the global word line have a net capacitance of $c_g + L$. To minimize the wire delay, the global word lines are driven from the center of the wire, in effect driving two segments in parallel, each having a resistance of $r_g/2$ and capacitance of $(c_g + L)/2$. The global word line wire delay then is $0.5 * r_g/2 * (c_g + L)/2$. If the local word line is also driven from the center of the wire segment, its delay is given as $0.5 * r_{wl}/2 * c_{wl}/2$. The net wire delay is summarized in (3). The estimated

$$D_{\text{wire}} = \underset{\substack{\text{predecode}\\ \text{wire}}}{r_p \cdot (c_p/2 + G)} + \underset{\substack{\text{global word}\\ \text{line}}}{r_g \cdot (c_g + L)/8} + \underset{\substack{\text{local word}\\ \text{line}}}{r_{wl} \cdot c_{wl}/8} \quad (3)$$

delays for the row decoder in four different SRAM's are within 9% of HSPICE simulated delays (Fig. 6).

Since bit line delay depends on the local word line rise time, we estimate the edge rate at the end of the local word line. From circuit simulations, the rise time was found to be 1.4 times the delay of the final stage of the word driver and is summarized in (4). Since the final stage

$$\text{rise time} = (\tau_{fo4} + r_{wl} \cdot c_{wl}/8) \times 1.4 \quad (4)$$

is sized to have a fanout of 4, the total delay of the stage is the sum of a fanout-of-4 inverter delay (τ_{fo4}) and the RC delay of the local word line ($r_{wl} * c_{wl}/8$, assuming that the word drivers drive the local word line from the center of the line).

The gate and wire capacitances in the signal path are added up to give an estimate of the decoder energy. Decoder area is calculated by estimating the area of the local and global word drivers and the area required for the predecode wires. The area of the word drivers is modeled as a linear function of the total device widths inside the drivers (Fig. 7). The constants for this function (24.05 and 497) have been obtained by fitting it to the areas obtained from the layout of six different word drivers [13], [22] and have units of λ^2, where λ is half the minimum feature size of the technology. The total device width within the driver is estimated to be 1.25 times the size of the final buffer as the active area of the predriver can approximated to be a quarter of the final inverter when fanout-4 sizing is used for the gates. The area for running the vertical predecode and block select wires (Fig. 1) is also added to the total decode area. As an example, the increase in the SRAM array width due to the 12–4096 decoder of Fig. 5 is accounted for by the areas for 64 local word drivers, 1 global word driver, and vertical wiring tracks for 16 predecode wires and 64 block select wires.

C. Output Mux

The output mux consists of the bit line mux which routes the cell data into the sense amplifiers, and the data line mux which routes data from the sense amplifiers to the output. Since the signal levels in both these muxes are small (~100 mV), the input signal source for both these muxes can be considered as ideal current sources.

The degradation of the delay through a RC network for a current source input is different from that for a voltage source input. Consider an ideal current source driving a RC π network as shown in Fig. 8(a). The voltage waveforms of the nodes 1 and 3 are sketched in Fig. 8(b) along with the waveform when the resistance is 0 (dashed line). The time constant τ_{RC} of the network is evaluated as in (5) and is easily generalized for an arbitrary RC chain as the sum of the product of each resistance with a capacitance which is obtained by considering all the downstream capacitance lumped together, in series with all the upstream capacitance lumped together. In steady state ($t \gg \tau_{\mathrm{RC}}$), nodes 1, 2, and 3 slew at the same rate, and the delay to obtain a swing of V at node 3 can be approximated by (6), which is the delay when there is no resistance plus the time constant of the network. This formula is used for estimating the delay of both the bit line and data line muxes

$$D + \frac{(C_1 + C_2 + C_3) \cdot V}{I} + \tau_{\mathrm{RC}}. \tag{6}$$

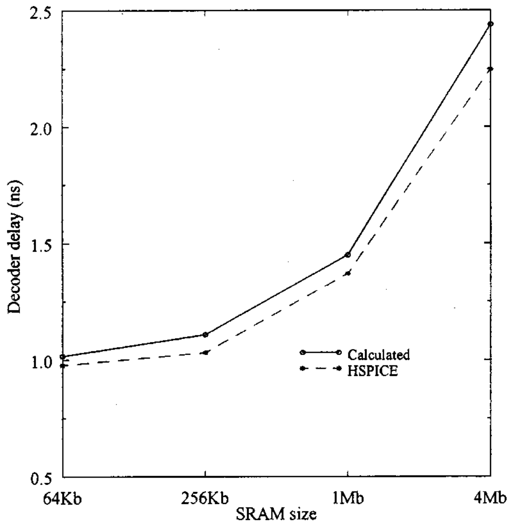
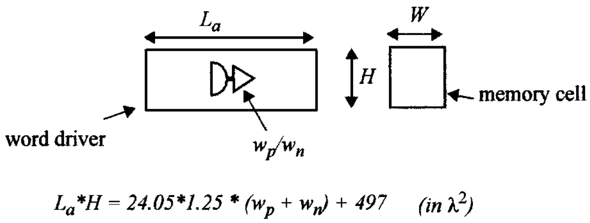
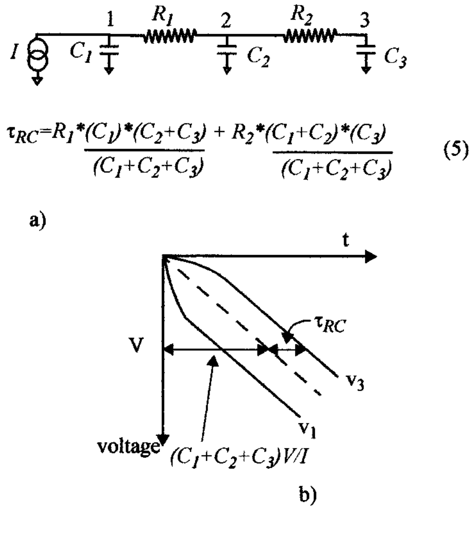


Fig. 6. Comparison of estimated and HSPICE simulated delay for row decoders.



L_a
W
H
memory cell
word driver
w_p/w_n
$L_a{*}H = 24.05{*}1.25 * (w_p + w_n) + 497 \quad (in\ \lambda^2)$

Fig. 7. Area estimation for the word drivers. The constants have been obtained by an empirical fit on areas from actual layouts.



1
R_1
2
R_2
3
I
C_1
C_2
C_3
$\tau_{RC}=R_1{*}\frac{(C_1){*}(C_2+C_3)}{(C_1+C_2+C_3)} + R_2{*}\frac{(C_1+C_2){*}(C_3)}{(C_1+C_2+C_3)}$
(5)
a)
t
τ_{RC}
V
v_3
v_1
voltage
$(C_1+C_2+C_3)V/I$
b)

Fig. 8. (a) Current source driving a RC π network and (b) sketch of the node waveforms.

A single-level bit line mux is shown in Fig. 9 and is modeled as an ideal current source driving a RC network as in Fig. 8. Local and global bit line wires and the mux switches contribute to the capacitances and resistances in the network. The bit line delay to obtain a signal swing of δv by (6) is the sum of the delay to generate the voltage swing with no resistance and the time constant of the RC network (7). Long local word lines can have slow rise times because of the line resistance. Since the rise time affects the cell delay, we need to include it in the delay model. The effect of the rise time (T_r) can be captured by adding an additional term to the delay equation which is proportional to it [3]. The proportionality constant z depends on the ratio of the threshold voltage of the access device in the cell to the supply voltage, and we find it from simulations to be about 0.3 for a wide range of block widths. The RC time constant τ_{RC} in the bit line delay equation is estimated as in (5)

$$D_b = (c_b + j_{\mathrm{mux}} \cdot (k+1) + c_s) \cdot \frac{\delta v}{I_{\mathrm{cell}}} + z \cdot T_r + \tau_{\mathrm{RC}} \tag{7}$$

c_b bit line capacitance;
$j_{\rm mux}$ unit junction capacitance of the mux switch;
k number of columns multiplexed into a single sense amplifier;
c_s input capacitance of the sense amplifier;
δv voltage swing at the input of the sense amplifier;
$I_{\rm cell}$ memory cell current;
T_r local word line rise time;
z proportionality constant determined from HSPICE;
$\tau_{\rm RC}$ time constant of the bit line RC network.

Fig. 10 graphs the estimated and HSPICE measured delay through the local word driver and the resistive word line and bit line, up to the input of the sense amps. The estimated delay is within 2.4% of the HSPICE delay when the bit line height is at least 32 rows for both short word lines (16 columns) and long word lines (1024 columns).

The sense amplifier buffer chain is shown in Fig. 11 and consists of the basic cross-coupled latch followed by a chain of inverters and a pair of nMOS drivers [12], [22]. The latch converts the small swing input signal to a full swing CMOS signal and is used for both the local and global sense amplifiers. In the case of the local sense amplifiers, the latch output is buffered by the inverter chain and driven onto the gates of the output nMOS drivers. These nMOS transistors create a small swing voltage signal at their outputs by discharging previously precharged data lines (analogous to the memory cell discharging the precharged bit lines). The delay of the sense amplifier structure is the sum of the delay of the latch amplifier τ_s and the delay to buffer and drive the outputs. τ_s is proportional to the logarithm of the desired voltage gain and the loading of the amplifier outputs [18]. For a gain of about 20 with only the self-loading of the amplifier, τ_s is found to be about $2\tau_{fo4}$ by both calculations and circuit simulations. If we assume that all the transistors of the latch are scaled in the same proportion, then its output resistance and input capacitance can be expressed as simple functions of the size of the cross-coupled nMOS in the latch, w_s, as shown in Fig. 11. The nMOS drivers are modeled as current sources, with their current output proportional to their size w_n. As in the decoders, optimal sizes $w_s, w_1, \cdots, w_n$ are determined to minimize the total output mux delay. Equation (8) captures the relevant portions of the output mux delay needed for doing this optimization and is the sum of the delays of the bit line mux, the latch senseamp, the buffers, and the nMOS drivers

$$T = D_b(w_s) + \tau_s + \frac{R_s \cdot 3 \cdot C_g \cdot w_1}{w_s} + \frac{R_g \cdot 3 \cdot C_g \cdot w_2}{w_1} \cdots \frac{R_g \cdot C_g \cdot w_n}{w_{n-1}} + \frac{C \cdot \delta v}{I_n \cdot w_n} + \text{other constants} \tag{8}$$

$$D_b(w_s) \approx \frac{G_s \cdot w_s \cdot \delta v}{I_{cell}} \tag{9}$$

where

τ_s $= 2\tau_{fo4}$; amplification delay of the latch senseamp;
G_s $= 3.8\,\mathrm{fF}/\lambda$: senseamp input capacitance unit width in 0.25-µm process;

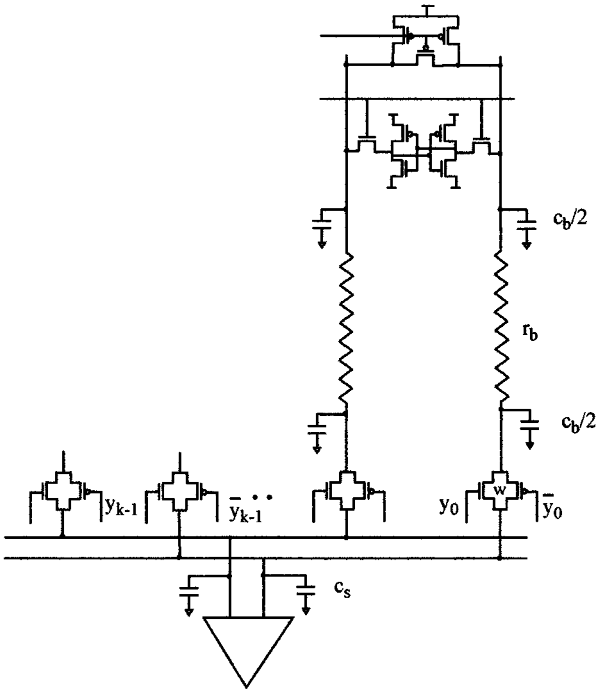


Fig. 9. Schematic of a single-level bit line structure.

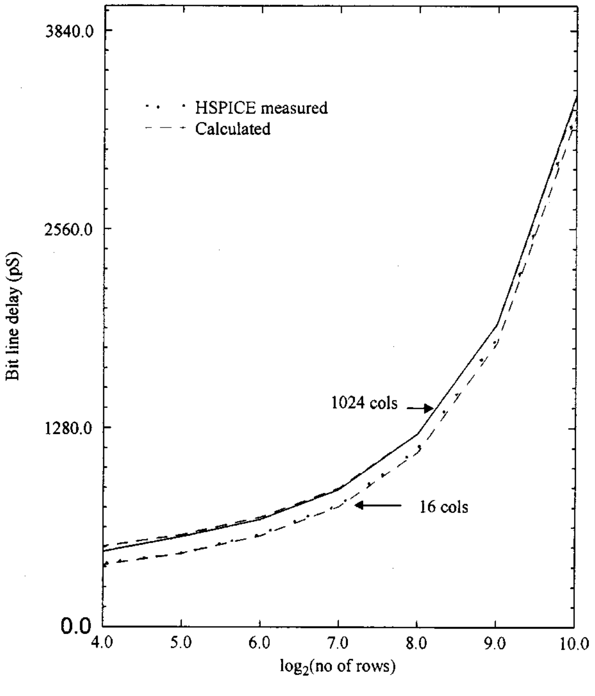


Fig. 10. Bit line delay versus column height; 0.25 μm, 1.8 V, and four columns multiplexing.

R_s =36 kΩ-λ senseamp output resistance per unit width;
w_s size of senseamp;
R_g, C_g output resistance and input capacitance per unit width of a 2 : 1 inverter;
I_n 37.5 μA/λ: current per unit width of nMOS;
C capacitance of the data line mux.

To simplify the procedure for finding the optimal sizes, impact of the latch senseamp size on the bit line mux time constant is ignored and only the cell delay is considered (9). Similarly, we ignore the effect of the nMOS junction capacitance on the data line RC time constant. Both these factors have little influence on the optimal sizing, but we include them for the final delay calculations. The minimum delay through the sense amp structure occurs when each term in (8) is equal to the extrinsic delay of a fanout-of-4 loaded inverter. The delay of the global sense amp is estimated in a similar fashion, except that the buffering delay to drive the output load is not considered in this analysis.

With technology scaling if the transistor threshold mismatch in the sense amplifier does not scale, then the delay of the output mux has a component which does not scale. This component is the delay of the memory cell to generate the bit line swing of δv_m, which is the input offset voltage of the sense amplifier. Hence, for delay estimations in future technologies, we keep this component a constant.

For low-power operation, the signals on high-capacitance nodes like the bit lines and the data lines are clamped to have small voltage swings [22]. Bit lines are clamped by pulsing the word lines, resulting in a total signal swing of about $2 * \delta v$ (the data lines are clamped in an analogous fashion to have similar signal swings). Hence, the energy of the bit line and data line mux is computed as $C * V_{dd} * 2 * \delta v$, where C is the capacitance on the line and includes the wire, junction, and input gate capacitances and V_{dd} is the supply voltage. The energy of a unit-sized sense amp is obtained from simulations to be 12 fJ/ $\lambda$for the 0.25-μm process and it is scaled up by w_s to obtain the sense amp energy.

The area of the switches in the bit line mux and the circuitry of the sense amplifier, precharge, and write drivers add to the vertical area of the SRAM array (Fig. 12). We base the area estimates of these components on data from a previous design [13]. Since the write driver, precharge, and mux transistors are not optimized, we add a fixed overhead of 4, 1, and 2 memory cells, respectively. The area of the local sense amps is modeled as a linear function of the total device width within the sense amp. The parameters to the model are obtained by fitting it to the data obtained from five different designs [13], [22] and is shown in Fig. 12. The total device width within the sense amp structure is itself estimated from the size parameters w_s, w_n, and w_p. The sum of all the device widths within the latch is estimated as $w_s * 8.7$, where the factor of 8.7 is obtained for the latch design in [13]. With fanout-of-4 sizing, the active area of the buffers prior to each nMOS output driver is no more than 1/3 of the driver width w_n. Hence, the active area of two nMOS drivers and their respective buffers is given by $2 * w_n * 1.33$.

We will next describe the results obtained by using these models to analyze many RAM organizations of different sizes in various technology generations.

IV. Analysis Results

We enumerate all the RAM organizations and estimate the area, delay, and energy of each using the simple models described previously. This allows us to determine the optimal organizations which minimize a weighted objective function of delay, area, and energy

$$\min\,(1 - \beta - \alpha) \cdot \text{Delay} + \alpha \cdot \text{Area} + \beta \cdot \text{Energy}. \tag{10}$$

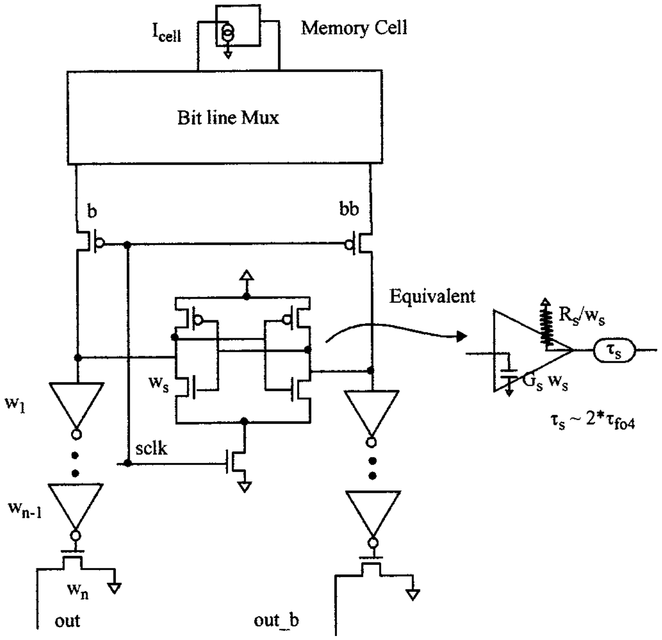


Fig. 11. Local sense amplifier structure.

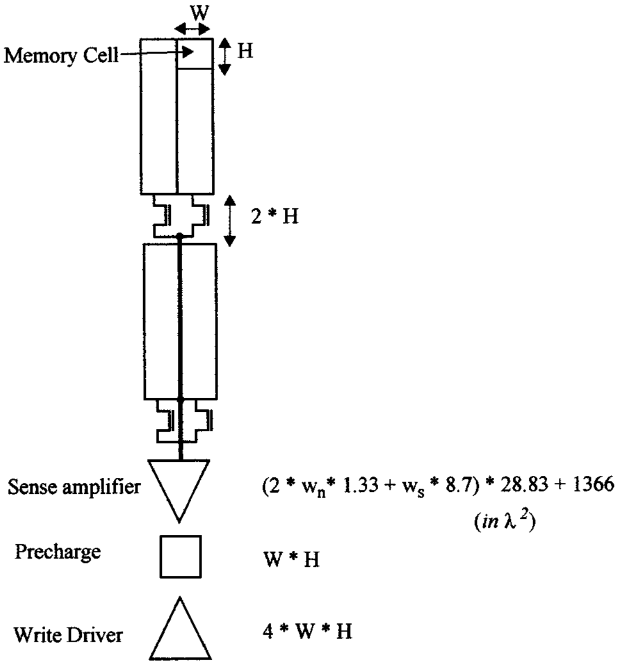


Fig. 12. Area estimation of the output mux.

The tradeoff curves are also obtained between these by varying the weight values α and β between 0 and 1.

Fig. 13 plots the delay of SRAM's organized for minimum delay ($\alpha = \beta = 0$ in (10)), with and without wire resistance, for sizes from 64 kb to 16 Mb with an access width of 64 bit, in the 0.25-μm technology. The delay of the SRAM without wire resistance is about $15\tau_{fo4}$ for a 64-kb design and is proportional to the log of the capacity as observed in [2]. The delay increases by about $1.2\tau_{fo4}$ for every doubling of the RAM size and can be understood in terms of the delay scaling of the row decoder and the output path. The delays for both of these are also plotted in the same graph and are almost equal in an optimally organized

SRAM. In the case of the row decoder, each address bit selects half the array, and, hence, the loading seen by the address bit is proportional to $S/2$, where S is the total number of bits in the array. With the fanout-4 sizing rule, the number of stages in the decoder will be proportional to the logarithm to base 4 of the total load, with each stage having approximately the delay of one τ_{fo4}. Hence, each doubling in number of bits adds about half a τ_{fo4} delay. In the case of the output path, the wire capacitance in the data line mux increases by about 1.4 for every doubling of the size, since it is proportional to the perimeter of the array, and, hence, the delay of the local sense amps increases by about $0.25\tau_{fo4}$. The remaining increase comes about due to the doubling of the multiplexor size for the bit line and the data line mux and its exact value depends on the unit drain junction capacitance and the unit saturation current of the memory cell and the nMOS output drivers.

The final curve in Fig. 13 is the SRAM delay with wire resistance. The global wires for this curve are assumed to have a width of 10λ (7.5 Ω/mm). Since the wire RC delay grows as the length of the wire, the wire delay for global wires in the SRAM scales as the size of the SRAM and becomes dominant for large-sized SRAM's.

Wire width optimization can be done to reduce the impact of interconnect delay. Fig. 14 shows the total delay for the 4-Mb SRAM for two different wire widths in four different technology generations. It is assumed that the metallization in 0.18 μm and below is in copper. The lowest curve plots the delay when the wire resistance is assumed to be zero. Since the threshold voltage mismatch remains constant with technology scaling, the bit line and data line signal swing do not scale in proportion to the supply voltage, and, hence, their delays will get worse relative to the rest of the RAM. As seen in the figure, the delay of the RAM increases by about $2.2\tau_{fo4}$ for the 0.1 μm and by $3.6\tau_{fo4}$ for the 0.07 μm, when interconnect delay is ignored. The second curve adds the round-trip signal delays around the access path assuming a speed of light propagation of 1 mm/6.6 pS and gives the lower bound for interconnect delay. The speed of light interconnect delay is about $1.75\tau_{fo4}$ for the 4-Mb SRAM, independent of the technology and doubles for every quadrupling of RAM size. The two curves above it graph the delay with wire resistance being nonzero for two different wire widths of 8 and 10λ. Significant reduction in wire delay is possible when fat wires are used for global wiring. Going from 0.25 μm with aluminum wiring to 0.18-μm copper wiring essentially leaves the delays (in terms of τ_{fo4}) unchanged, but, with further shrinks of the design, the delay for any particular wire width worsens, since the wire RC delay does not scale as well as the gate delay. However, by widening the wires in subsequent shrinks, it is possible to maintain the same delay (in terms of τ_{fo4}) across process generations. A wire width of 110λ brings the delay within a τ_{fo4} of the speed of light limit at the 0.25- and 0.18-μm generations, while wider wires are needed in the 0.1- and 0.07-μm generations. The larger pitch requirements for these fat wires can be easily supported when the divided word line structure in the decoders and column multiplexing in the bit lines are used.

We will next look at some ways in which the performance of actual SRAM implementations might differ from those predicted by the previous curves. Large SRAM's typically incorporate some form of row redundancy circuits in the decode path. This usually takes the form of a series pass transistor in the local word driver and will cause the delay curves to shift up by about 1/2 to $1\tau_{fo4}$. Fanouts larger than 4 in the word line driver, commonly done to reduce area, will also shift the delay curves up by about 1/2 to $1\tau_{fo4}$. High-performance SRAM's do not use static circuit style in the entire decode path but skew the gates in the predecoders and the global word drivers to favor a fast word line activation [7], [19], causing the delay curves to shift down. In order to estimate the speed improvements possible by skewing, let us first consider a chain of inverters which are skewed to such an extreme that the input signal is connected either to the nMOS or the pMOS gate and not to both as in Fig. 15. We assume that the complementary MOSFET is present, but its gate is deactivated (and will be activated in a separate reset phase in an actual implementation), and it merely adds to the self loading of the gate. Under these assumptions and the parameters from Table I, the average optimal fanout in the skewed chain is about 5 and the delay of a skewed gate is about 70% that of a nonskewed gate. In the case of the decoder, the local word drivers are not skewed typically due to the excessive area overhead incurred for the resetting circuitry. If the predecoder and the global word driver are skewed, then the delay of the decoder in the 64-kb RAM reduces to about $6\tau_{fo4}$ instead of the $8\tau_{fo4}$ for the static implementation. Furthermore, with every doubling of the RAM

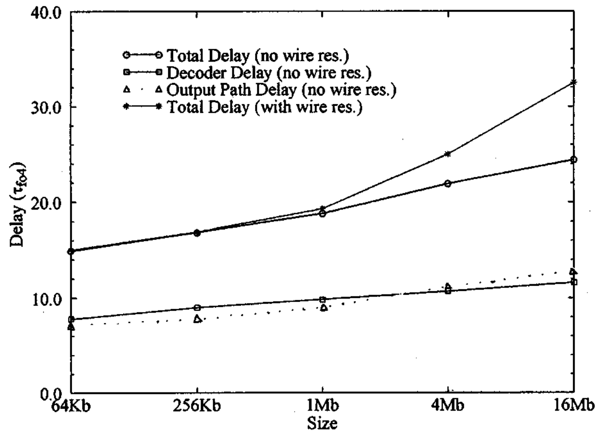


Fig. 13. Delay scaling with size in the 0.25-μm process.

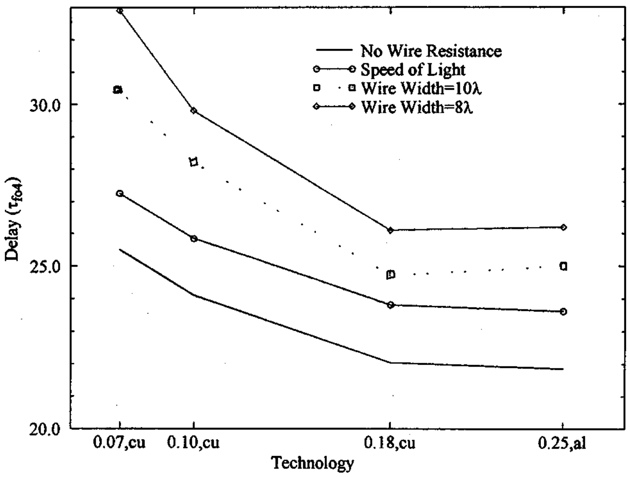


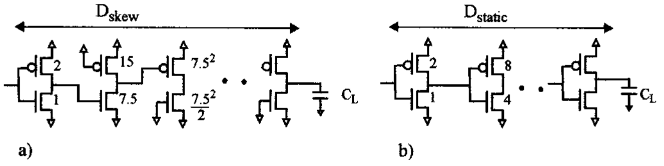
Fig. 14. Delay versus technology for different wire widths for a 4-Mb SRAM.

size, the decoder delay will increase by about $0.3\tau_{fo4}$ instead of $0.5\tau_{fo4}$ for the static case. Finally, the sense clock for the local sense amplifiers is usually padded with extra delay to enable operation over a wide range of process conditions [7], [22] which incurs an additional delay of up to $1\tau_{fo4}$, when bit lines are short. Thus, when all these effects are combined, the SRAM delay curve will shift up by about $1\tau_{fo4}$ in Fig. 13.

Partitioning allows for a tradeoff between delay, area, and power. Tradeoff curves can be obtained by solving (10), with various values for the parameters α and β. When β equals zero, the delay-area tradeoff is obtained and the curve for a 4-Mb SRAM in the 0.25-μm process is shown in Fig. 16. Any point on this curve represents the lowest area achievable via RAM reorganization for the corresponding delay. Starting from a minimum delay design which is finely partitioned, significant improvements in the area is possible by reducing the amount of partitioning and incurring a small delay penalty, while subsequent reduction in partitioning results in decreasing improvements in area for increasing delay penalty. Partitioning parameters for three points A, B, and C are shown in the figure. Points A and B are in the sweet spot of the curve, with A being about 22% slower and 22% smaller area and B being 14% slower and 20% smaller area when compared to the fastest implementation.

Of the various organization parameters, the RAM delay is most sensitive to the block height, and fast access times are obtained by using smaller block heights. Fig. 17 shows the delay and area for a 4-Mb SRAM for various block heights, while using optimal values for the remaining organization parameters. Small block heights reduce the delay of the bit lines but increase the delay of the global wires since the RAM area increases due to the overhead of bit line partitioning. For very large block heights, the slow bit line delay limits the access time. Hence, an optimum block height exists and is 32 rows for the example above. Increasing the block height to 128 rows incurs a delay penalty of about 8% while the area can be reduced by 7.6%, illustrating the area—delay tradeoff that are possible via partitioning.

By setting α equal to 0 in (10), one can obtain the delay-energy tradeoff through partitioning, with no constraints on the area, and is shown in Fig. 18. The unit used on the left-hand vertical axis is the energy consumed to switch the gate of a $16\lambda/32\lambda$-sized inverter (Eunit =72 fJ). Partitioning allows for a large tradeoff between energy and delay as noted in [4] and [5]. The figure also indicates the optimal degree of column multiplexing (cm) and the block height (bh) required to obtain the corresponding delay and energy for some of the points. We find that, for low-energy solutions, the column multiplexing is one, i.e., the block width is equal to the access width, since this enables only the minimum number of bit line columns to switch. Since we do sizing optimization to minimize delay, the final transistors in the output of the local sense amps become large and consequently have a large capacitance associated with their drain junction capacitance. Hence, in the low-energy designs, it is advantageous to have large block heights, as noted in [4] and [5], since this allows most of the muxing to be done in the bit line mux where the junction capacitances from the memory cell's access transistor are very small compared to the junction capacitances in the data line mux. We also find that the energy consumption in optimally organized SRAM's can be expressed as a sum of two components. One is independent of the capacity, depends only on the access width, and is due to the local word line, the precharge signal, local and global sense amps, etc. The other component scales as the square root of the capacity, as observed in [4] and [5], is related to the power dissipation in the global wires and the decoders.



$$D_{skew} = 0.31 * \tau_{fo4} + 0.62 * D_{static} \qquad (10)$$

Fig. 15. Optimal sizing for (a) extremely skewed and (b) statically sized inverters.

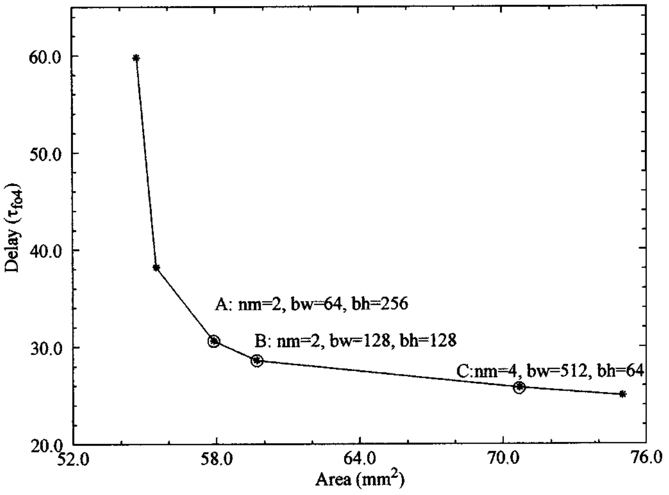


Fig. 16. Delay versus area for a 4-Mb SRAM in the 0.25-μm process.

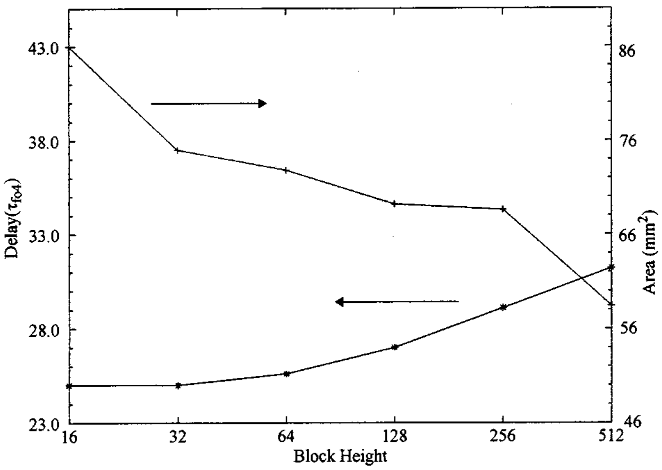


Fig. 17. Delay and area versus block height for a 4-Mb SRAM in a 0.25-μm process.

V. Conclusions

Analytical models for delay, area, and energy allow one to explore a range of design possibilities in a very short span of

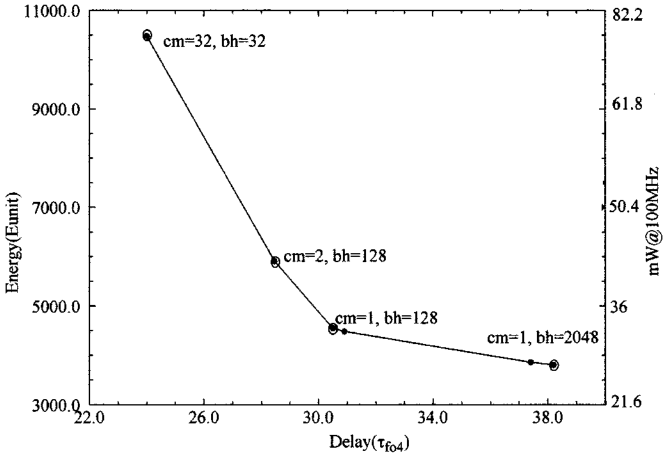


Fig. 18. Energy versus delay for a 4-Mb SRAM in a 0.25-μm process.

time. These models are used to study the impact of SRAM partitioning and it is found that a substantial tradeoff between area, delay, and energy can be obtained via the choice of SRAM organization.

The models are also used to predict the scaling trends of delay with capacity and process technology. The delay of SRAM can be broken into two components; one is due to the transistors in the technology (gate delay) and the other is due to the interconnect (wire delay). The gate delay increases by about $1.2\tau_{fo4}$ for every doubling of the RAM size, starting with $15\tau_{fo4}$ for a 64-kb RAM, when a static circuit style is used to design the decoders.

Nonscaling of threshold mismatches with process scaling causes the signal swings in the bit lines and data lines also not to scale, leading to an increase in the gate delay of an SRAM across technology generations. For an optimally organized 4-Mb SRAM, the increase in delay is about $2\tau_{fo4}$ in the 0.1-μm and $3.6\tau_{fo4}$ in the 0.07-μm generations and is worse for other organizations. This delay increase for most SRAM organizations can be mitigated by using more hierarchical designs for the bit line and data line paths and using offset compensation techniques such as those used in [10] and [20].

The wire delay starts becoming important for RAM's beyond the 1-Mb generation. Across process shrinks, the wire delay becomes worse and wire redesign has to be done to keep the wire delay in the same proportion to the gate delay. A divided word line structure for the decoders and column muxing for the bit line path opens up enough space over the array for using fat wires, and these can be used to control the wire delay for 4-Mb and smaller designs across process shrinks. The wire delay is lower bounded by speed of light, which is about $1.75\tau_{fo4}$ for the 4 –b SRAM, and doubles with every quadrupling of capacity. Thus, for high-performance RAM designs at the 16-Mb and higher level, the RAM architecture needs to be changed to use routing of address and data (see, for example, [14]), instead of the current approach where the signals are broadcast globally across the array. Wire delay is also directly proportional to the cell area, and, hence, cell designs with smaller area will win out for large RAM's, even if the cells are weaker. Thus, the DRAM cell, multivalued cells, TFT-based cells, and other novel cell designs will be worth investigating for designing future high-performance high-capacity RAM's.

Appendix

We list all the assumptions not covered in Section III in this appendix.

A. Technology

The base technology is assumed to be a 0.25-μm CMOS process and the relevant process details are shown in Table I. The key features for four different generations are shown in Table II. Copper metallurgy is assumed from the 0.18-μm generation onwards.

Higher level metals are designed as "fat" wires: their heights are also scaled along with their widths to yield a larger cross section, but the heights are increased only by the square root of the factor of increase of the widths [17]. For example, a higher level metal layer with twice the minimum width of the metal 1 layer has a height which is 1.4 times the metal 1 height, thus resulting in a resistance which is a factor of 3 smaller than the metal 1 resistance. We assume that the wiring pitch is twice the wire width for all the global wires.

B. Architecture

The SRAM is synchronous, i.e., a clock starts off the access, though the results can be easily extended to asynchronous SRAM's, by adding the power and delay to generate the address transition detection (ATD) signal.

An embedded SRAM structure is assumed, viz., all the data bits of the accessed word come out of the memory core in close physical proximity to each other (Fig. 1), unlike in stand-alone SRAM's, where the data IO port locations are optimized for external pad connections. Since this optimization adds a constant offset to the delay and power of the SRAM core, the conclusions of this study are applicable even to stand-alone SRAM's.

The RAM cell size used for the analysis is $19.2\lambda \times 32.8\lambda$, as in [7], and the cell area is typical of high-density six-transistor CMOS layouts.

C. Circuit Style

The RAM is designed for high-speed operation with low-power pulsed techniques which reduce energy loss without affecting speed, as discussed in [22]. The local word lines are pulsed to control the bit line swings and small swings are used in the data lines to reduce power. Since these techniques do not affect the speed of the RAM, our analysis results pertaining to delay scaling are applicable to any speed-optimized SRAM design.

A latch-style sense amplifier (Fig. 11) with perfect timing control is assumed for the sense amplifier as this consumes the least power and is the fastest. Hence, our analysis results will be of relevance to both high-speed and low-power SRAM's. For the 0.25-μm process, the optimal input swing which minimizes the sense amp delay is found from simulations to be 100 mV, of which 50 mV is the input offset.

The transistors in the bit line mux have a fixed size of 16λ and those in the data line mux are sized to be 50λ wide to simplify the analysis. Circuit simulations indicate that the RAM delay is only weakly sensitive to the sizes of these transistors.

D. Energy Modeling

The swings in the bit lines and IO lines are limited for low-power operation. While ideally they should be limited to be exactly that required for optimum detection by the sense amps, in practical designs, there is some slack in how tightly they can be controlled [22] and hence are assumed to be twice the optimum signal swing. Thus, for the 0.25-μm process, these swing by about 200 mV since the optimal swing for the sense amps is about 100 mV.

Acknowledgment

The authors wish to thank Dr. V. De of Intel Corporation for pointing out the impact of threshold nonscaling on the total delay. They also gratefully acknowledge the invaluable comments from the members of the *mhstudents* in Stanford.

References

[1] "1997 National technology roadmap for semiconductor,".

[2] T. Wada, S. Rajan, and S. A. Przybylski, "An analytical access time model for on-chip cache memories," *IEEE J. Solid-State Circuits*, vol. 27, pp. 1147–1156, Aug. 1992.

[3] S. J. E. Wilton and N. P. Jouppi, "An enhanced access and cycle time model for on-chip caches,", WRL Research Report 93/5, June 1994.

[4] R. J. Evans and P. D. Franzon, "Energy consumption modeling and optimization for SRAM's," *IEEE J. Solid-State Circuits*, vol. 30, pp. 571–579, May 1995.

[5] R. J. Evans, "Energy consumption modeling and optimization for SRAM's," Ph.D. dissertation, Dept. of Electrical and Computer Engineering, North Carolina State Univ., July 1993.

[6] I. E. Sutherland and R. F. Sproull, "Logical effort: Designing for speed on the back of an envelope,", Advanced Research in VLSI, 1991.

[7] H. Nambu *et al.*, "A 1.8ns access, 550MHz 4.5Mb CMOS SRAM," in *ISSCC Dig. Tech. Papers*, Feb. 1998, pp. 360–361.

[8] M. Yoshimoto *et al.*, "A 64kb full CMOS RAM with divided wordline structure," in *ISSCC Dig. Tech. Papers*, Feb. 1983, pp. 58–59.

[9] W. C. Elmore, "The transient response of damped linear networks with particular regard to wideband amplifiers," *J. Appl. Phys.*, vol. 19, pp. 55–63, 1948.

[10] K. Seno *et al.*, "A 9-ns 16-Mb CMOS SRAM with offset-compensated current sense amplifier," *IEEE J. Solid State Circuits*, vol. 28, Nov. 1993.

[11] K. Osada *et al.*, "A 2 ns access, 285MHz, two-port cache macro using double global bit-line pairs," in *ISSCC Dig. Tech. Papers*, Feb. 1997, pp. 402–403.

[12] M. Matsumiya, "A 15-ns 16-Mb CMOS SRAM with interdigitated bit-line architecture," *IEEE J. Solid-State Circuits*, vol. 27, pp. 1497–1503, November 1992.

[13] T. Mori *et al.*, "A 1V 0.9mW at 100MHz 2kx16b SRAM utilizing a half-swing pulsed-decoder and write-bus architecture in 0.25mm Dual-Vt CMOS," in *ISSCC Dig. Tech. Papers*, Feb. 1998, pp. 354–355.

[14] T. Higuchi *et al.*, "A 500MHz synchronous pipelined 1Mbit CMOS SRAM," (in Japanese), , Tech. Rep. IEICE, May 1996.

[15] J. D. Meindl *et al.*, "The impact of stochastic dopant and interconnect distributions on gigascale integration," in *1997 IEEE Int. Solid-State Circuits Conf., Dig. Tech. Papers*, pp. 232–233.

[16] G. A. Saihalasz, "Performance trends in high-end processors," in *Proc. IEEE*, vol. 83, Jan. 1995.

[17] H. B. Bakoglu and J. D. Meindl, "Optimal interconnection circuits for VLSI," *IEEE Trans. Electron Devices*, vol. ED-32, pp. 903–909, May 1985.

[18] C. L. Portmann *et al.*, "Metastability in CMOS library elements in reduced supply and technology scaled applications," *IEEE J. Solid-State Circuits*, vol. 30, pp. 39–46, Jan. 1995.

[19] T. Chappell *et al.*, "A 2-ns cycle, 3.8-ns access 512-Kb CMOS ECL SRAM with fully pipelined architecture," *IEEE J. Solid-State Circuits*, vol. 26, pp. 1577–1585, Nov. 1991.

[20] K. Ishibashi *et al.*, "A 6-ns 4-Mb CMOS SRAM with offset-voltage-insensitive current sense amplifiers," *IEEE J. Solid-State Circuits*, vol. 30, Apr. 1995.

[21] T. Mizuno *et al.*, "Experimental study of threshold voltage fluctuation due to statistical variation of channel dopant number in MOSFET's," *IEEE Trans. Electron Devices*, vol. 41, pp. 2216–2221, Nov. 1994.

[22] B. S. Amrutur and M. A. Horowitz, "A replica technique for wordline and sense control in low-power SRAM's," *IEEE J. Solid-State Circuits*, vol. 33, pp. 1208–1219, Aug. 1998.

[23] N. C. Li, G. L. Haviland, and A. A. Tuszynski, "CMOS tapered buffer," *IEEE J. Solid-State Circuits*, vol. 25, pp. 1005–1008, Aug. 1990.

Bharadwaj S. Amrutur received the B.Tech. degree in computer science and engineering from Indian Institute of Technology, Bombay, in 1990 and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1994 and 1999, respectively.

He is currently a Member of Technical Staff with Agilent Technologies, Palo Alto, CA, where he is working on high-speed I/O.

Mark A. Horowitz received the B.S. and M.S. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1978, and the Ph.D. degree from Stanford University, Stanford, CA, in 1984.

He is the Yahoo Founders Professor of Electrical Engineering and Computer Science at Stanford University. His research area is in digital system design, and he has led a number of processor designs including MIPS-X, one of the first processors to include an on-chip instruction cache, TORCH, a statically-scheduled, superscalar processor and FLASH, a flexible DSM machine. He has also worked in a number of other chip design areas including high-speed memory design, high-bandwidth interfaces, and fast floating point. In 1990, he took leave from Stanford to help start Rambus, Inc., a company designing high-bandwidth memory interface technology. His current research includes multiprocessor design, low-power circuits, memory design, and high-speed links

Dr. Horowitz is the recipient of a 1985 Presidential Young Investigator Award, and an IBM Faculty Development Award, as well as the 1993 Best Paper Award at the International Solid State Circuits Conference.