

# Memory Modeling for System Synthesis

Sari L. Coumeri and Donald E. Thomas

Carnegie Mellon University

Pittsburgh, PA 15213

{scoumeri, thomas}@ece.cmu.edu

## 1. ABSTRACT

**We present our methodology for developing models of on-chip SRAM memory organizations. The models were created to enable the quick evaluation of energy, area, and performance of different memory configurations considered during synthesis. The models are defined in terms of parameters, such as size and mode of operation, which are known at synthesis time. Our methodology does not require knowledge of the underlying memory circuitry and provides models with average percentage errors within 8%. We found that only 10 different memories from a large span of possible memory sizes are needed to obtain reasonably accurate models, with average errors within 15%. We further use these models to evaluate different low power memory organizations and have seen energy reductions of up to 88%. In this paper we present our modeling methodology, discuss the important aspects in developing the models, and show results of using the models in evaluating low power memory organizations.**

## 2. INTRODUCTION

Power consumption of digital systems has become a critical design parameter. Extending battery life in portable applications and reducing cooling requirements in higher transistor density applications make power reduction a crucial consideration during digital system design.

An important class of digital systems include applications, such as video image processing and speech recognition, which are extremely memory-intensive. In such systems, a significant amount of power is consumed during memory accesses. Thus, utilizing low-power memory organizations can greatly reduce the overall power consumption of the system.

This work targets on-chip memories created by memory module generators in which there are many possible memory organizations in terms of size, architecture, technology, etc. To utilize low power memory configurations during synthesis, we need models to quickly evaluate memory energy, area, and performance. These models need to be in terms of parameters, such as size, organization, and mode of operation, which are known during synthesis time as opposed to lower level parameters such as extracted

capacitance and resistance values.

In the past few years, various different memory models have been presented. Itoh [6] and Kamble [7] have presented analytical models of memory power. Ko [8] did a measurement-based characterization in which the power of a few different memories were measured. Evans [5] compared five different approaches for modeling the energy of SRAMs and used the models to analyze different internal architectures. Ogawa [12] used circuit reduction techniques for faster characterization of power and delay of SRAMs. Chinosi [3] developed a technique for the automatic characterization of memory power for different modes of operations for a certain sized memory. Landman [9] used a simulation and model fitting approach to develop power models in terms of the number of words and the bit width.

Our models were developed to predict energy, delay, and area across the different possible sizes and organizations produced by memory module generators and for different modes of memory operations. Our modeling uses a simulation-based approach which enables the development of black box models. Unlike analytical models, simulation-based approaches do not require detailed knowledge of the underlying circuits, just basic input/output timing information which can be provided by the memory generator. The models are in terms of high-level parameters and can be easily used during synthesis. Our approach is similar to [9] but is generalized enough to handle more complex memory organizations and more modes of memory operation with higher accuracy.

The focus of this paper is not just on our modeling methodology, but on defining the parameters and simulations necessary in building accurate models quickly and easily and showing how a synthesis tool can take advantage of these models in creating low power memory configurations.

## 3. EXPERIMENTAL METHODOLOGY

Our experimental methodology is as follows. First, memories are generated using Cascade's Epoch memory module generator [4]. Next, test vectors are automatically generated and SPICE files are modified to prepare them for simulations. Then, Avant!'s Star-Sim, fast circuit simulator, is used to simulate for energy and delay [15]. From the simulation data, models of memory energy, delay, and area are developed using linear regression with the S-Plus statistical package [14]. Finally, the models are validated to ensure they are statistically sound.

### 3.1 Generated Memories

Cascade's basic asynchronous SRAMs with chip and output enables were used. To generate a memory in Cascade, the number of address lines, the number of words, and the bit

width are specified. Additionally, the number of bits per column (BPC), which gives you control over the aspect ratio of the memory, can be specified. The legal BPC values in Cascade are 1, 2, 4, 8, and 16. Therefore, a unique memory size is not defined just by the number of words and bit width. It is defined by the number of rows, number of columns, and bit width; where:

$$\text{Rows} = \text{Words} / \text{BPC} \quad (1)$$

$$\text{Columns} = \text{Bitwidth} / \text{BPC} \quad (2)$$

$$\text{RowAddressLines} = \text{ceil}(\lg(\text{Rows})) \quad (3)$$

$$\text{ColumnAddressLines} = \lg(\text{BPC}) \quad (4)$$

The number of rows can range from 4-256, the number of columns from 1-256, and the bit width from 1-256. By varying the number of rows, columns, and bit width there are 62,992 different possible legal memory sizes. It is obviously impossible to simulate all possibilities, so a subset must be chosen. Twenty-five different basic Cascade SRAMs were generated in .6u technology with a 3.3V supply. The largest and smallest size memories were included and the rest were chosen randomly. The subset of chosen memories was examined to ensure a good variation in the number of rows, number of columns, bit width, number of row and column address lines, BPC, and total number of storage bits in the memory.

## 3.2 Memory Simulations

After obtaining the SPICE files from the generated memories, Star-Sim simulations were run to measure energy and delay. During these runs data was collected for the various modes of the memories. Since an entire memory was simulated at once as opposed to separate simulations for the different pieces of the memories, creating the test vectors for the simulations was easy. Knowledge of the memories' internal circuitry was not required, only the I/O timing information supplied by Cascade. These simulations are necessary because the delay estimates provided by Cascade are overly conservative and the power estimates do not account for different memory modes of operation.

### 3.2.1 Energy Simulations

The average energy per operation was measured. Read and write energy were treated separately. The energy was also measured while the chip and output enable lines were toggling and for different levels of switching activity on the address lines.

The hierarchical SPICE netlist was instrumented to separately measure the energy of the different memory components. The separate components in the memories are the address transition detection (ATD) logic, memory cells, chip enable multiplexors, row and column decoders, precharge logic, senseamps, and extra buffers.

With a write operation, there were two additional parameters to consider. Once the address changes at the start of a write cycle, the write enable line must remain high for the address setup time. Next, the write enable line is lowered during which time the data is written. Finally, the write enable line is raised for the address hold time before the

address changes to again start the next cycle.

Due to static power dissipation, the amount of time the write enable is held low affects the energy. Additionally, the amount of time the write enable signal remains high before it is lowered can impact the energy. Since these are asynchronous memories, address transition detection (ATD) logic is used to detect a change on the address lines and start a memory access. If the write enable line remains high longer than the required address setup time, a memory read will occur before a write, thus, resulting in additional energy.

In synchronous designs there are different ways to generate the write enable signal from the clock, each of which results in different address setup and write enable low times. Therefore, including these parameters in the models of memory energy is important.

### 3.2.2 Delay Simulations

The worst case delay for a memory operation was measured. The read time (the address changing to the data appearing on the output), the write bit time (write enable going low to the data being written to the memory cells), and the write out time (the write enable going low to the data appearing on the output) were measured. Cascade specified values for hold and setup times were used.

Delays for when the chip enable is activated and with and without a capacitive load were measured as well. The rise and fall times of the four physical corners of the memory were measured and the worst delay for each was taken.

## 3.3 Developing Memory Models

Three categories of memory models were developed from the simulations: area, delay, and energy. All the models are linear equations in terms of parameters known during synthesis. For area, there are width and height models. For delay, there are read, write bit, write out, setup, and hold time models. For energy, there are distinct models for read and write operations.

Each energy model is composed of separate models for the components of the memory (ATD, senseamps, etc.). The sum of the individual component models forms the total energy read and write models. Having separate models for the different components of the energy enables us to develop more accurate models and gain more insight into the energy trade-offs of the generated memories.

Table 1 summarizes the parameters used in all of the models. The size parameters are used for all of the models. The others parameters relating to the mode of operation are used for both delay and energy. CE, OE, and RW are all Boolean variables which indicate whether or not the specified action is occurring.

The models were developed using stepwise linear regression in the S-Plus statistical package. The initial models were the specified variables defined in Table 1. The stepwise regression improves the initial model by iteratively adding and deleting terms. It can consider multiple interaction

terms. For example, since the number of rows and number of columns are both variables specified in the model, it can consider adding the term  $Rows \cdot Cols$  to the model. It adds and deletes terms based upon the AIC criteria [1] which tries to improve the coefficient of multiple determination,  $R^2$ , without overfitting the model. It adds terms which nontrivially contribute to the model and removes useless terms which do not.

**Table 1: Parameters Used In Models**

Abbr.	Parameter	Models Used In
Rows	# of Rows	energy, delay, area
Cols	# of Columns	energy, delay, area
BW	Bit Width (width of data word)	energy, delay, area
Addr	# of Address Lines	energy, delay, area
R.Addr	# of Row Address Lines	energy, delay, area
C.Addr	# of Column Address Lines	energy, delay, area
Sw	# of Addr Lines Switching per access	energy
R.Sw	# of Row Addr Lines Switching per access	energy
C.Sw	# of Column Addr Lines Switching per access	energy
CE	Chip Enable signal toggling (0, 1)	energy, delay
OE	Output Enable signal toggling (0,1)	energy
Cap	Capacitive Load (fF)	delay
WTL	Time Write Enable Held Low (ns)	write energy
RW	Extra read before write (0,1)	write energy

Using stepwise regression in our modeling methodology allows us to develop accurate models quickly and easily. It automatically determines which parameters are important to the models and finds the interactions between the independent variables. Without stepwise regression we would have to specify the form of equation which is difficult to do with a large number of parameters and would require detailed knowledge of the underlying memory circuitry to determine the interaction between the variables.

### 3.4 Model Validation

Table 2 shows the statistical data for the developed models. The second and third columns have the statistics for the model-building data set which are the coefficient of multiple determination,  $R^2$ , and the residual standard error for each of the models. The area models had the best fits followed by the energy and delay models.

Simulations for 25 additional memories were run to build a validation data set. The statistics for this set, shown in columns four and five, include the square of the correlation between the measured and predicted values,  $r^2$ , and the square root of the mean squared prediction error,  $\sqrt{MSPR}$ . These values can be compared to the  $R^2$  and the residual standard error of the model-building data set to measure our models' predictive ability. The predictions for the energy and area models are very accurate. The accuracy drops slightly for the delay models.

**Table 2: Statistical Data**

Model	Model-Building Data Set		Validation Data Set		Avg  %err
	$R^2$	Residual St. Error	$r^2$	$\sqrt{MSPR}$	
Height ( $\mu$ )	.9998	15.81	.9998	14.99	1.5%
Width ( $\mu$ )	.9999	9.59	.9994	16.94	1.8%
Write Bit Time (s)	.9755	9.78e-11	.8920	1.99e-10	6.1%
Write Out Time (s)	.9762	1.06e-10	.9318	1.69e-10	5.5%
Read Time (s)	.9703	1.50e-10	.9059	2.26e-10	3.4%
Read Energy (J)	.9962	1.57e-11	.9879	2.31e-11	8.4%
Write Energy (J)	.9990	3.40e-11	.9864	4.77e-11	13%
Weighted Read Energy (J)	.9953	1.75e-11	.9855	2.68e-11	7.8%
Weighted Write Energy (J)	.9975	5.32e-11	.9866	4.60e-11	6.1%
Simple Read Energy	.9831	2.79e-11	.9424	4.76e-11	19%
Simple Write Energy	.8340	3.87e-10	.6939	5.91e-10	69%
Non-Comp. Read Energy	.9959	1.40e-11	.9810	2.68e-11	9.6%
Non-Comp. Write Energy	.9963	5.84e-11	.9761	6.13e-11	10%

The last column in the table shows the average absolute percentage error for all of the simulated memories. This is calculated by the equation:

$$|\%err| = |MeasuredEnergy - PredictedEnergy| / MeasuredEnergy \quad (5)$$

The average percentage error is fairly low but jumps to 13% for the write energy. The problem occurs because there is more than a 500x difference in write energy between the largest and smallest data points. The extremely small memories have energy values smaller than the standard error of the equations and therefore, can end up with percentage errors larger than 100%. To account for this problem, each data point,  $i$ , was given the following weight:

$$Weight_i = \max(Energy) / Energy_i \quad (6)$$

( $\max(Energy)$  is the maximum energy for all the data points and  $Energy_i$  is the energy for data point,  $i$ .) A weighted stepwise regression was done for the read and write energy. Rows 8 and 9 show the weighted regression results. This weighting boosts the importance of the smaller energy data points and improves the average absolute percentage error. The improvement was less than 1% for the read energy. However, the write energy absolute percentage error was cut in half.

Rows 10 and 11 show results for simplified models. These models were developed doing a weighted linear regression using the equation from [9] as opposed to using stepwise regression. This equation, shown below, does not account for different aspect ratios within the memory or for different modes of operation.

$$Energy = C_0 + C_1 Words + C_2 BW + C_3 Words \cdot BW \quad (7)$$

The simple read model had fits and standard errors slightly worse than our model. However, the simple write model was inaccurate with residual and predicted errors approximately

an order of magnitude larger. The average percentage error was considerably larger for both the read and write models.

The last two rows of the table are the results for models created doing weighted stepwise regression for the total energy as opposed to separate componentized models for each portion of the memory (senseamps, ATD, etc.). The fits and standard errors were comparable for these models. However, the average percentage errors were worse.

#### 4. IMPORTANT FACTORS OF MODELS

Using our methodology, very accurate models of energy, area, and delay were created. However, running many memory simulations can be CPU intensive. The simulations ranged from a few minutes to a few days of CPU time, depending on the size of the memory. Therefore, to create accurate models quickly and easily, it is necessary to determine which factors are most important while developing the model.

##### 4.1 Parameters of Models

Table 3 shows the independent parameters used in each of the read energy models. TypeIII ANOVA (analysis of variance) tables [11] were examined to see how much each independent variable reduces the sum of square error in the model. The variables in the table are listed in order of importance (from highest to lowest variance).

The ANOVA tables for all of the components show that the most important variables to the models are the size parameters, followed by the address switching parameters, followed by the chip and output enable toggling.

**Table 3: Components of Read Energy**

Component	Variables in Model (Abbr. are defined in Table 1)	Avg. % of Read Energy ( $\pm$ St. Dev.)	Max %
Precharge	Cols, Rows	42 $\pm$ 10%	74.4%
Senseamps	BW, Rows, Cols, R.Addr	35 $\pm$ 15%	55.1%
ATD	Sw, Rows, Cols, BW, Addr, R.Addr	12 $\pm$ 10%	58.2%
Buffers	Cols, OE, CE, Addr, BW, C.Sw	5 $\pm$ 2%	16.3%
Row Decoder	Cols, Sw, BW, Rows, Addr	4 $\pm$ 2%	11.9%
Column Decoder	BW, C.Sw, R.Sw, Cols, Rows, R.Addr, CE, C.Addr	0.8 $\pm$ 2%	13.8%
Chip Enable Muxes	R.Sw, C.Sw, CE, Addr, BW, Cols	0.8 $\pm$ 0.8%	7.8%
Memory Cell	BW, Sw, Addr, R.Addr, CE, Cols	0.6 $\pm$ 0.5%	5.3%
Entire Model	Cols, BW, Rows, Sw, R.Addr, CE, Addr, OE, R.Sw, C.Sw, C.Addr	-----	---

Columns 3 and 4 in Table 3 show the average percentage and maximum percentage of read energy consumed in each of the memory components. This was calculated by using the models to make predictions on the 62,992 different Cascade memories. The precharge logic and senseamps consumed the largest average percentage of energy, consuming 42% and 35% respectively. The standard deviations for these averages are quite high. Therefore, the

distribution of the energy and the effects of the different parameters vary throughout the memory design space. The models for the average energy (in pJ) consumed during a read access in the precharge logic and senseamps are shown below:

$$Pre = 1.4 + 0.28Cols + 0.06Rows + 0.005Rows Cols \text{ pJ} \quad (8)$$

$$Sense = 1.6 + 0.12BW + 0.002Rows + 0.003Rows BW + 0.05Cols - 0.001Cols BW - 0.47RAddr + 0.14RAddr BW \text{ pJ} \quad (9)$$

Both the precharge and senseamp models are dependent solely on size parameters. The switching and chip and output enable toggling parameters are important parameters for the ATD and buffer models which consume much lower average percentages of energy. However, these components have higher maximum percentages of energy, 58% and 16% respectively. The switching parameters are significant in memory configurations with a large ratio of number of address lines to total bits of storage. Chip and output enable toggling parameters are important in memories with a low number of storage bits where the energy of the buffers is not overshadowed by the precharge and senseamp energy.

##### 4.2 Number of Memories in Data-Set

Since the size parameters are the most important to the models, the question to answer is how many different sized memories are needed to get accurate models? An experiment was conducted in which different read energy models were developed from subsets of the 25 model-building data set memories. For a certain sized subset, a weighted stepwise linear regression was run, and the rest of the data from the 50 simulated memories (model-building data set plus validation data set) were used as validation data.

Table 4 shows the statistical results of the subset models. There were four different sized subsets: 5, 10, 15, and 20. The sizes of the validation data sets for each of these were 45, 40, 35, and 30, respectively. For each of the subset sizes, 20 samples were run. Columns 3 and 4 show the average square of the correlation between the measured and predicted values,  $r^2$ , and the average square root of the mean squared prediction error,  $\sqrt{MSPR}$ . Column 5 shows the average of the average absolute percent error.

**Table 4: Models from Subsets of Data**

Subset Size	# of Samples	Avg. $r^2$ ( $\pm$ St. Dev.)	Avg. $\sqrt{MSPR}$ ( $\pm$ St. Dev.)	Avg Avg  %Err
5	20	.53 $\pm$ .31	6.60e-10 $\pm$ 1.88e-09	310%
10	20	.97 $\pm$ .02	3.22e-11 $\pm$ 1.57e-11	15.3%
15	20	.94 $\pm$ .16	4.60e-11 $\pm$ 6.47e-11	17.1%
15	19	.98 $\pm$ .01	3.17e-11 $\pm$ 9.14e-12	10.1%
20	20	.98 $\pm$ .02	3.14e-11 $\pm$ 1.04e-11	10.4%

The average predictions of the models based upon 5 memories are very poor. But, the predictions improve significantly with 10 memories in the data-set. The predictions drop slightly for the sample size of 15. This was

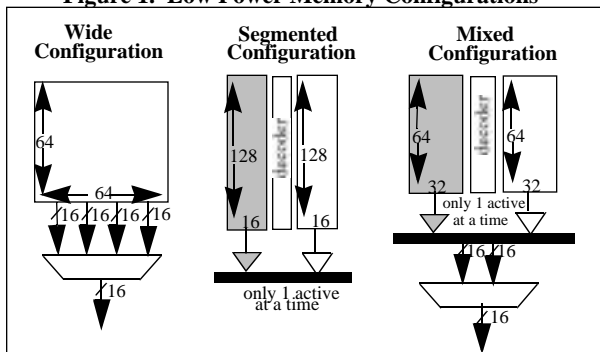
due to the fact that one of the samples was really poor with an  $r^2$  of .25. If the outlier is removed from the samples, the predictions improve over the 10 memory sample size. The predictions of the 20 size samples improves even further. With just 10 memories in the data-set fairly accurate models of read energy can be developed. However, some care must be taken to ensure that the parameters of the memories are well distributed. In the outlier sample, there were no memories with a small number of rows and large bit width. Therefore, the developed models were unable to predict accurately in this region of the memory space.

### 5. LOW POWER CONFIGURATIONS

Since the models of memory energy, area, and delay are in terms of high-level parameters, a synthesis system can use them to evaluate different memory configurations. Tools such as [13] can make use of such memory models during scheduling and allocation. The basic modeled memories can be combined to form low power memory configurations to be built by a synthesis system.

Some low power memory configurations are shown in Figure 1. The first configuration is a wide configuration in which multiple words are read from the memory at once and selected between by a multiplexor [10]. This configuration can be thought of like a fast-page mode except the page size is relatively small. The benefit of this configuration is the reduced number of accesses to the memory if the words are used right after each other. The next configuration is a segmented configuration in which the memory is broken up into smaller components [2]. Only one component is active at a time and is selected by a decoder. The benefit of this configuration is each access is to a smaller less capacitance memory. There can also be a mix of the two in which there are segmented wider memories. These configurations can be explored by a synthesis system considering energy/area/delay trade-offs.

Figure 1. Low Power Memory Configurations



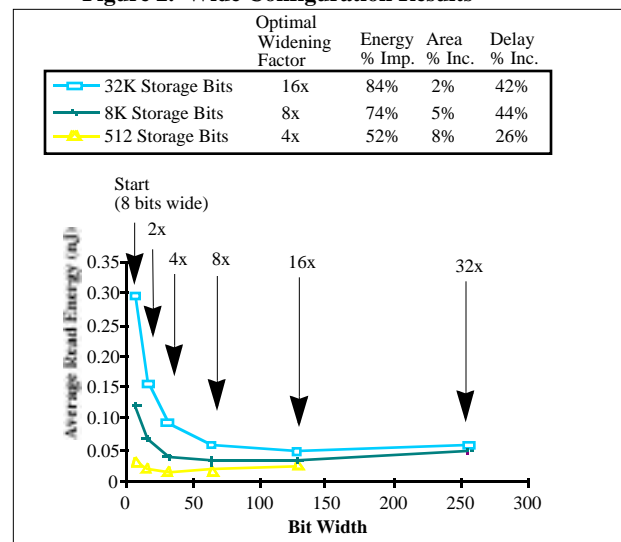
To optimize these configurations for a specific application a synthesis system would also need information about the memory access pattern for the application. For instance, when deciding whether or not to apply a wide configuration, the synthesis tool would need to know how many of the multiple accessed words would be used right after each other. The results in the next couple of figures are not for a specific application but are presented to illustrate potential

energy savings from these configurations for different sized memories. The assumption is made that each of the memories is accessed in binary order.

To evaluate these configurations models of additional multiplexor and decoder logic are needed. These models were created using our memory modeling methodology. However, these were simpler to develop since there are not many different modes of operation to consider.

Figure 2 shows the read energy results of applying a wider configuration to three different sized memories. When applying a wider configuration, the underlying number of storage bits of the memory remains constant. The graph starts with memories with a bit width of eight. Memories two times as wide internally have bit widths of sixteen and are reading out two words at once. Memories four times as wide internally have bit widths of thirty-two and are reading out four words at once. The assumption is a memory twice as wide is accessed 1/2 as often and a memory four times as wide is accessed 1/4 as often. This is a valid assumption when the memories are accessed in order.

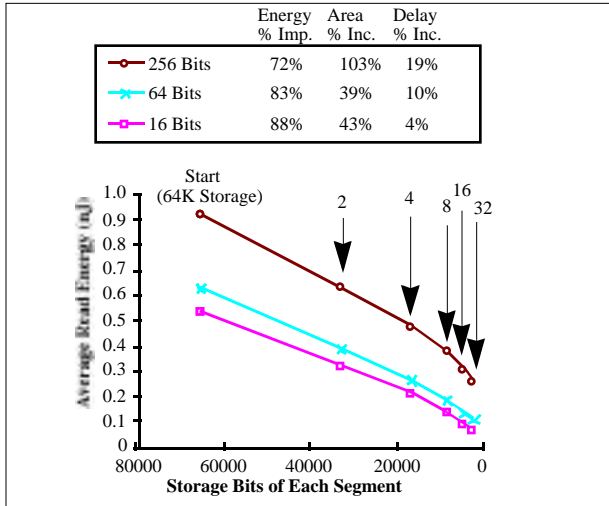
Figure 2. Wide Configuration Results



Initially the energy drops as the memories are widened. However, the energy starts increasing again as the memories get too wide. This is because the energy from the multiplexors is significant enough to offset the savings from the very wide memories. The figure also shows the optimal widening factor for each of the different sized memories, the percentage improvement from the highest energy to lowest energy configurations, and the area and delay penalties for this energy improvement. Area is not affected much by the wide configurations, but there is a significant impact to the delay. The 32K storage bit memory had an 84% improvement in energy due to widening with a 2% area penalty and 42% delay penalty. The optimal widening factor varies throughout the design space.

Figure 3 shows the read energy results of applying a segmented configuration to three memories with different

**Figure 3. Segmented Configuration Results**



bit widths. When applying the segmented configuration, the number and size of the segments vary while the bit width remains constant. The graphs start with 64K storage bits in each of the memories. When segmenting by two, each of the two memories contains 32K storage bits. When segmenting by four each of the four memories has 16K storage bits.

The energy continues to drop as they are further segmented. This improvement comes with a significant area penalty but does not impact the delay as much. The 16 bit width memory had an energy improvement of 88% due to segmenting.

A synthesis system can use these models as well as the memory access pattern to determine the optimal widening and segmenting factors for a specific memory. There are powerful memory architectural trade-offs in terms of energy, area, and delay which can be made using these models.

## 6. CONCLUSIONS

We have presented our modeling methodology for memory energy, area, and delay. Our methodology provides an easy and accurate way to develop memory models without detailed knowledge of the underlying circuitry. The models developed using our technique had average percentage errors within 8%. Using a weighted stepwise linear regression technique to determine the form of the models reduced the standard error over an order of magnitude from a simplified model approach. We showed that the size parameters were the most important to consider while developing the models and that it is only necessary to simulate 10 different sized memories to obtain models with average errors within 15%.

Memory architectural decisions are capable of profoundly moving the power/area/delay characteristics of the design. Through such decisions we showed reductions of memory read energy of up to 88%. These complex decisions can be explored automatically within a synthesis system. Therefore, accurate and easy to develop memory models in terms of high-level parameters are necessary to explore a

rich set of energy, area, and delay trade-offs.

## 7. ACKNOWLEDGEMENTS

The authors would like to thank Michael Posner for his statistical expertise and Professor Herman Schmit for his insightful discussions. This research was partially funded by the Semiconductor Research Corporation contract number 068-048.

## 8. REFERENCES

- [1] J.M. Chambers and T.J. Hastie. *Statistical Models in S*, Pacific Grove, CA: Wadsworth and Brooks, 1992.
- [2] A. Chandrakasan and R.W. Brodersen. *Low Power Digital CMOS Design*, Kluwer Academic Publishers, 1995.
- [3] M. Chinosi, R. Zafalon, and C. Guardiani. "Automatic Characterization and Modeling of Power Consumption in Static RAM's," *International Symposium on Low-Power Electronics and Design*, August, 1998.
- [4] *Epoch User's Manual*. Cascade Design Automation Corporation, Bellevue, WA, 1996.
- [5] R.J. Evans and P.D. Franzon. "Energy Consumption Modeling and Optimization for SRAM's," *IEEE Journal of Solid-State Circuits*, Vol. 30, no. 5, pp. 571-9, May 1995.
- [6] K. Itoh, K. Sasaki, and Y. Nakagome. "Trends in Low-Power RAM Circuit Technologies," *Proceedings of the IEEE*, Vol. 83, no. 4, pp. 524-43, April 1995.
- [7] M.B. Kamble and K. Ghose. "Analytical Energy Dissipation Models for Low Power Caches," *International Symposium on Low Power Design*, pp. 143-8, 1997.
- [8] U. Ko and P.T. Balsara. "Characterization and Design of a Low-Power, High-Performance Cache Architecture," *International Symposium on VLSI Technology, Systems, and Applications*, pp. 235-8, 1995.
- [9] P.E. Landman and J.M. Rabaey. "Architectural Power Analysis: The Dual Bit Type Method," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 3, no. 2, pp. 173-87, June 1995.
- [10] D.B. Lidsky and J.M. Rabaey. "Low Power Design of Memory Intensive Functions Case Study: Vector Quantization," *Proceedings of IEEE Workshop on VLSI Signal Processing*, pp. 378-87, October 1994.
- [11] J. Neter, W. Wasserman, and M.H. Kutner. *Applied Linear Regression Models*, Homewood, IL: Irwin, Inc., 1989.
- [12] K. Ogawa, "PASTEL: A Parametrized Memory Characterization System," *Proceedings of DATE'98*, March 1998.
- [13] H. Schmit and D.E. Thomas, "Synthesis of Application-Specific Memory Designs," *IEEE Transactions on VLSI Systems*, Vol. 5, no. 1, pp. 101-11, March 1997.
- [14] *S-PLUS User's Manual*. StatSci, Seattle, WA, 1993.
- [15] *Star-Sim User's Guide*. Avant! Corporation, Fremont, CA, 1997.