Power Modeling of SDRAMs

Ajay Joshi, Sriram Sambamurthy, Saket Kumar, and Lizy John
{ajoshi, sambamur, skumar, ljohn}@ece.utexas.edu

***Abstract*—We present a model for estimating the power consumption of SDRAM at an architectural level. The approach is based on identifying the various operating states for a typical SDRAM, and using the knowledge of current drawn by the memory chip, and fraction of the time spent in each state to estimate the total energy consumption. This model is integrated into Wattch, a simulator for architectural power analysis, and the impact of memory on system energy consumption is analyzed for various programs in the SPEC95 CPU Benchmark suite and some media benchmarks. We further apply this model to study the energy-performance trade-off in designing memory hierarchies. In this paper we discuss the important aspects of our memory power model, and present the results of applying the model to memory hierarchy design space exploration.**

Index Terms—**memory, power modeling, SDRAM, cycle simulator, energy-performance trade-off**

I. INTRODUCTION

Power consumption has become a first class design constraint. Extending battery life in portable devices, and reducing thermal cost in high-density transistor applications make power reduction a crucial consideration during design of digital systems.

Emerging workloads include applications such as multimedia, video image processing, and speech recognition, which are extremely memory-intensive. These applications consume a considerable amount of power during memory accesses. Thus, developing techniques to reduce memory power consumption can greatly reduce the system power consumption for these classes of applications.

The graph in Figure 1 gives a rough estimate of the range of total integration costs (thermal and power delivery) associated with supporting additional cooling requirements as the power rating of a chip is increased. It can be seen that when a microprocessor is in the 35-40 W range, the cost of each additional Watt increases above $1/W per chip. An interesting observation from the graph is that the power cost of DRAM is on a steeper curve than that of the microprocessor. This is primarily because the spatial layout of the system chassis is such that memory modules are harder to cool. Therefore, reducing the memory power consumption is essential to keep the system cost down. Modeling memory at an architectural level is one of the first steps in analyzing the impact of memory on system power consumption and developing various memory energy management schemes.

Currently, Synchronous Dynamic Random Access Memory (SDRAM) is the most popular main memory type in mobile and high-end computers. Almost every semiconductor manufacturer produces some variants of SDRAM. This work therefore targets SDRAM type of memories, and develops a power model that can be used to estimate the memory energy consumption at an architectural level.

The rest of this report is organized as follows: The next section discusses the techniques that have been previously used to model memory power. Section III gives an overview of the SDRAM architecture and provides a detailed description of our proposed SDRAM power model. Section IV analyzes the memory power model to find the distribution of energy consumption in various operating states of the SDRAM, and discusses the impact of memory on system power consumption. Section V presents the results of an experiment that applies the developed SDRAM power model to study the energy-performance trade-offs involved in designing memory hierarchies.

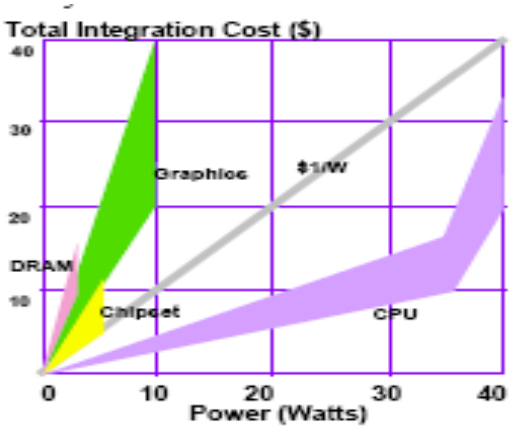

Figure 1: Thermal and power delivery cost in a personal computer (Courtesy [12])

II. PREVIOUS WORK

This section discusses the various approaches that have been previously used to develop power models for memory systems. For the sake of clarity, the modeling techniques have been classified into three categories: Conceptual, Empirical, and Eclectic.

In conceptual approaches the memory structure is assumed to be completely transparent. Models are derived from an

abstraction of the detailed implementation. Joo et al., [8] developed an energy simulator based on cycle-accurate energy measurement and state-machine based precise characterization of SDRAM main memory systems. This approach gives accurate results, but the price paid in terms of simulation time is extremely high.

The second approach, empirical modeling, treats the internal implementation of SDRAM memory details as completely hidden. The models are built from observations of the instance behavior of the SDRAM using statistical techniques. Landman et al., [11] used a simulation and model fitting approach to develop power models in terms of the number of words and the bit width. Coumeri et al., [3] created models for different memory configurations and used regression techniques to develop a model for memory energy consumption in terms of various design parameters. This approach requires commercial CAD tools that are not easily available in academia.

The third approach, eclectic modeling, is a mixture of the conceptual and empirical modeling methodologies. It requires some information of the internal structure of SDRAM memory and some empirical data. Vijaykrishnan et al., [13] proposed a power model for RAMBUS RAM by developing a state transition diagram and estimating the power consumed in each state.

III. SDRAM Power Model – Our Approach

III (a) SDRAM Architecture

Dynamic Random Access Memories (DRAM) can be classified into synchronous and asynchronous RAMs. SDRAMs buffer data, address, and control signals in order to separate the timing of the memory cells from the timing of the controlling processor. This way, the data and control interfaces of the memory can be configured for a pipelined operation.

Figure 2 shows the architecture of a typical SDRAM. A SDRAM chip consists of a multitude of memory cells that each hold one bit of information. Typically four to thirty two bits form the smallest accessible piece of information that can be addressed. These memory cells are arranged in a two-dimensional array consisting of rows and columns. Information can be accessed using the row and column index in the two dimensional matrix. Every address delivered to the SDRAM from the memory controller is broken down into a row and a column address within the chip. These amplifiers form a considerable part of the total cost of the SDRAM chip, the memory arrays are usually symmetric. The memory array and the sense amplifiers are together referred to as a memory-bank. For the purpose of our model, we assume that SDRAM chip consists of two memory banks. The functionality of a read operation involves decoding of the row address, activation of the memory row, decoding the column address, selecting the column entry and transferring the data to the output buffer, and precharging of the accessed memory bank. The last step is necessary because accesses to the memory cells of a SDRAM through the sense amplifiers are destructive. The precharge operation is required to reconstruct the contents of the memory cell based on the information in the sense amplifiers. The write functionality is similar except that the data is transferred from the input buffer to the sense amplifiers, and then to the memory cells.

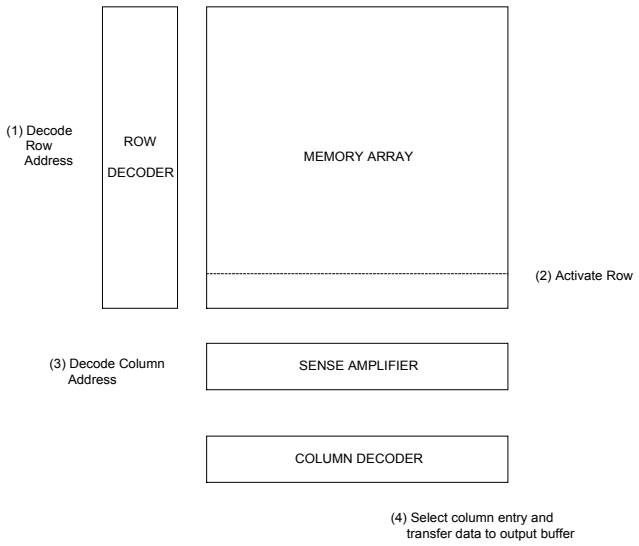

Figure 2: Architecture of SDRAM, and typical sequence during a READ operation

III (b) SDRAM Power Model

Our approach towards formulating a power model for SDRAM involves identifying the various operating modes in an SDRAM chip. The energy consumed in each state will then be computed using the knowledge of the current drawn by the chip when operating in that state, and the fraction of time spent in that state. The typical values of these currents are obtained from the datasheets of various SDRAM manufacturers. This model is then incorporated into Wattch [1], an architectural simulator built on top of SimpleScalar. SimpleScalar simulator is instrumented to obtain the number of main memory read and write requests issued by the program being simulated. Using this information, it is possible to calculate the fraction of the total number of simulation cycles spent in an operating state. This information, along with the power consumed in each state, can then be used to estimate the total energy consumed in that state. The sum of the energy dissipated in all the states is the total memory energy consumed by the application being simulated.

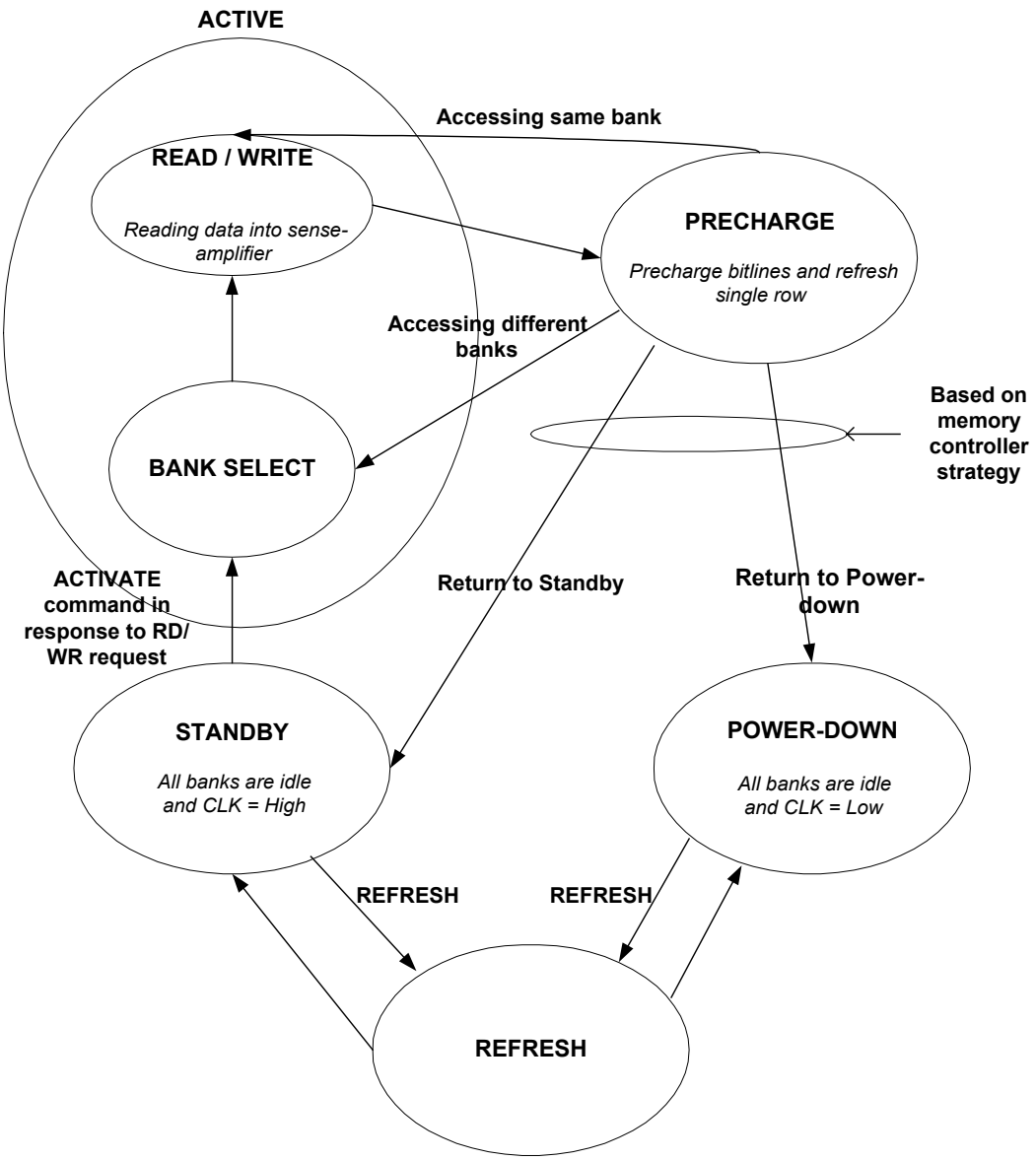

Figure 3: Transition diagram showing the various operating states in a SDRAM

Figure 3 shows a state transition diagram in which the states refer to various operating modes of the SDRAM, and the transitions between them is caused due to the issuing of various commands by the memory controller. Following are the different states/operating modes illustrated in the transition diagram:

Active: When data is stored in the sense-amplifier, the SDRAM is said to be in the *active* state. In this state, the SDRAM module is ready for receiving the row and column packets and can transition immediately to read or write state. This state internally consists of two sub-states – *selecting and activating the correct bank to read/write*, and *reading/writing from the target row*. In order to receive row and column packets, both the row and column multiplexer receivers have to be active. The resynchronization time for this mode is the least, and the energy consumption the highest.

Standby: In this state, the column multiplexers are disabled resulting in significant reduction in energy consumption compared to the *active* state. The resynchronization time

for this state is typically one or two memory cycles. In this state the memory clock, CLK, is high and both the banks are idle. On receiving an ACTIVATE command in this state, in response to a read or write request, the SDRAM will transition to the *active* state.

Power-down: In this state, the periodic clock and the synchronization circuitry are shut down resulting in another order of magnitude saving in energy. The resynchronization time is also significantly higher (typically thousands of cycles). On receiving an ACTIVATE command in this state, in response to a read or write request, the SDRAM will transition to the *active* state.

Precharge: After the memory cells of the SDRAM have been accessed through the sense-amplifiers, the charge is lost and cannot be reconstructed by the cell itself. Hence, a precharge operation is needed to reconstruct the cell contents based on the current information that is held in the sense amplifiers. After every read or write operation, the memory controller initiates a precharge operation. After the precharge operation, the SDRAM may immediately move into *read/write* sub-state of the *active* state if the same bank is being accessed, otherwise a different bank needs to be selected. If there are no pending read or write operations after the precharge operation has been completed, the SDRAM state may transition into s*tandby, power-down* or *active* states, depending on the strategy used by the memory controller. In this model we assume that the memory controller strategy is such that it transitions the SDRAM into *standby* state if there are no pending read or write requests. This is a realistic assumption because this is typically the strategy adopted by memory controller due the benefit of low energy consumption and small synchronization time in *standby* state.

Refresh: The memory cells in an SDRAM slowly lose charge due to leakage effects. The charge on these memory cells therefore needs to be restored periodically. The refresh interval is typically 64 ms. The process of restoring the charge on the memory cells is called refreshing. Thus the memory controller issues a REFRESH command to the SDRAM after every refresh interval. The refresh operation is normally evenly distributed over time. When calculating the power consumption in this state, we assume that the SDRAM device is in precharge power-down state at all times except when the actual REFRESH command is executed.

The following terminology is used when formulating equations for computing the energy consumed in the various SDRAM operating states mentioned above:

num_sim_cycles – This refers to the total number of simulation cycles required to complete execution of the target application.

mem_freq – This is the memory clock frequency that is typically 133 MHz for SDRAMs.

proc_freq – This is the frequency of operation of the processor and we assume this to be 600 MHz.

num_mem_cycles – This is equal to
(*num_sim_cycles* * *mem_freq) / (proc_freq)*

num_mem_reads – This is the total number of read access made to the main memory by the application being simulated.

num_mem_writes - This is the total number of write accesses made to the main memory by the application being simulated.

read_latency – This is the delay between issuing of the read request by the processor and the availability of data.

write_latency - This is the delay between issuing of the write request by the processor and the data being written into SDRAM.

num_read_cycles – This is computed as:
num_mem_reads * *read_latency*

num_write_cycles – This is computed as:
num_mem_writes * *write_latency*

num_idle_cycles – Total number of memory cycles during which the SDRAM is in *standby* state. This can be computed as:
num_mem_cycles – num_read_cycles – num_write cycles

standby_time – This is the time spend in *standby* state and is equal to *num_idle_cycles* / *mem_freq*.

I_{DD2} - This is the current drawn by the SDRAM chip when in *standby* mode. In this state the all banks are idle, clock enable, CKE, is high, and CLK is set to $CLK_{min,}$ and the address and other control inputs change once per clock cycle.

V_{DD} – This is the supply voltage to the SDRAM chip.

I_{DD0} – This is the operating current drawn by the SDRAM chip when it is in *active* and *precharge* operating modes.

t_{RC} – This is the time between the issue of successive ACTIVATE commands to the same bank.

n_{ACT} - This is defined as the number of memory clock cycles between successive ACTIVATE cycles

I_{DD3} – This is the current drawn by the SDRAM chip when in active state, just after the bank has been selected.

I_{DD4} – This is the current drawn by the SDRAM chip when reading or writing data.

refresh_inverval – This is the interval between two successive refresh operations.

$t_{REFRESH}$ – This is the duration of one refresh operation.

num_of_data_pins – This is the total number of data pins present on the SDRAM memory chip.

III (c) SDRAM Energy Calculations

The total energy dissipated (E_{TOT}) in a SDRAM is the sum of energy dissipated when in *standby* state (E_{STBY}), *refresh* state (E_{REF}), *read* state (E_{READ}), and *write* state (E_{WRITE}).

$$E_{TOT} = E_{STBY} + E_{REF} + E_{READ} + E_{WRITE} \quad \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

Standby Energy (E_{STBY})

When SDRAM is in *standby* state it draws a current of I_{DD2} and hence the energy consumed in *standby* state is given by:

$$E_{STBY} = I_{DD2} * V_{DD} * standby\_time \quad \ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Write Energy (E_{WRITE})

In order to accomplish to write data, a row must first be selected using an ACTIVATE command, by decoding the address into row and column addresses. For every ACTIVATE command, there is a corresponding PRECHARGE command. The ACTIVATE command opens a row for read or write operation, and the PRECHARGE command closes the row. Hence the ACTIVATE and PRECHARGE commands are always paired together.

After the ACTIVATE command is issued, a large current is drawn till the command/address is decoded and the row data is transferred from the memory array to the sense-amplifiers. After this is completed, the SDRAM is maintained in the *active* state and draws a current of I_{DD3} till the PRECHARGE command is issued,. Therefore the base energy consumed during a single write operation, E_{Active_Base}, is given by:

$$E_{Active\_Base} = I_{DD3} * V_{DD} * write\_latency \quad \ldots\ldots\ldots\ldots\ldots\ldots (3)$$

The PRECHARGE command restores the data from the sense-amplifiers into the memory array and resets the bank for the next ACTIVATE command. Once this transition is completed, the device is returned to *precharge* state. The I_{DD0} current value is the average current required for the device operation. Therefore, to calculate the energy consumed by the ACTIVATE-PRECHARGE command pair, $E_{Active-Precharge}$, I_{DD3} must be subtracted from the total operating current I_{DD0}. This energy is given by

$$E_{Active-Precharge} = (I_{DD0} - I_{DD3N}) * V_{DD} * write\_latency \quad \ldots\ldots\ldots\ldots\ldots\ldots (4)$$

Equation (3) is correct if the SDRAM is only used at the minimum t_{RC} specified in the datasheets. This is not true in most cases and the cycle is greater than the specified value of t_{RC}. In this case the operating current should be scaled as a ratio of the actual ACTIVATE-to-ACTIVATE duration to that for which the current is specified in the data sheets. Therefore the energy consumed by ACTIVATE-PRECHARGE pair, $E_{Active-Precharge}$, is given by:

$$E_{Active-Precharge} = (I_{DD0} - I_{DD3}) * (t_{RC} / (n_{ACT} * t_{CK})) * V_{DD} * write\_latency \quad \ldots\ldots\ldots\ldots\ldots\ldots (5)$$

Once a bank has been opened using the ACTIVATE command, the data can be written to the SDRAM. After the ACTIVATE command the DRAM draws a current of I_{DD3} till the data is written to the SDRAM. During the writing of data to the SDRAM, the current drawn by the chip is I_{DD4} . Therefore the additional energy dissipated during the actual writing of data, E_{WR}, is given by:

$$E_{WR} = (I_{DD4W} - I_{DD3N}) * (num\_write\_cycles / n_{ACT}) * V_{DD} * write\_latency \quad \ldots\ldots\ldots\ldots\ldots\ldots (6)$$

Therefore the energy consumed by a SDRAM during a single write operation is given by the sum of equations (3), (5), and (6)

$$E_{WRITE} = E_{Active\_Base} + E_{Active-Precharge} + E_{WR} \quad \ldots\ldots\ldots\ldots\ldots\ldots (7)$$

The total write energy can be computed by multiplying the energy consumed during a single write operation with the *num_mem_writes.*

Read Energy (E_{READ})

The Active-Base, Active-Precharge, and the additional energy dissipated during a read operation, is calculated similar to that during a write operation:

$$E_{Active\_Base} = I_{DD3} * V_{DD} * read\_latency \quad \ldots\ldots\ldots\ldots\ldots\ldots (8)$$

$$E_{Active-Precharge} = (I_{DD0} - I_{DD3}) * (t_{RC} / (n_{ACT} * t_{CK})) * V_{DD} * read\_latency \quad \ldots\ldots\ldots\ldots\ldots\ldots (9)$$

$E_{RD} = (I_{DD4} - I_{DD3}) * (num\_read\_cycles / n_{ACT}) * V_{DD} * read\_latency$ ……..(10)

However, during a read operation additional energy will be dissipated because the data bits being read need to be driven through the pins on the SDRAM chip. The current required to drive these pins is dependent on the termination scheme used by the particular implementation. The power dissipated at a single data pin to drive the data bit is given by the product of V_{OUT} and I_{OUT}, where V_{OUT} and I_{OUT} respectively refer to the voltage and current at the data pin. The total energy required to drive data through the data pins, E_{DQ}, is given by:

$E_{DQ} = V_{OUT} * I_{OUT} * (num\_of\_data\_pins) * (num\_read\_cycles / n_{ACT}) * read\_latency$ ……..(11)

Therefore the energy consumed for a single read operation is given by the sum of equations (9), (10), and (11).

$E_{READ} = E_{Active\_Base} + E_{Active\text{-}Precharge} + E_{RD} + E_{DQ}$ ……..(12)

The total read energy can be computed by multiplying the energy consumed during a single write operation with the *num_mem_reads*.

Refresh Energy (E_{REF})

In order to maintain the integrity of the data in the memory cells, the SDRAM needs to be periodically refreshed. This is due to the fact that the capacitor storing the charge in a memory cell loses its charge over a period of time and needs to be recharged. This refresh operation is normally distributed evenly over time. The total energy consumed due the periodic refresh operation is given by:

$E_{REF} = (I_{DD5} - I_{DD2}) * V_{DD} * (standby\_time / refresh\_interval) * t_{REFRESH}$ ...(13)

IV. ANALYSIS OF SDRAM POWER MODEL

Wattch is an architectural level power analysis tool built on top of SimpleScalar simulator. We integrated the SDRAM power model proposed in section III into Wattch, and instrumented the SimpleScalar simulator to obtain the *num_mem_reads*, and *num_mem_writes*. Programs in the SPEC95 CPU Benchmark suite and some media benchmarks were then simulated to obtain the energy consumed main memory (SDRAM) for various memory sizes, 64 Mb, 128 Mb, 256 Mb, and 512 Mb. In this section we describe an analysis of the impact of memory on the system power consumption. We also provide distribution of the energy consumed by various operating modes of the SDRAM.

Figure 4 shows a graph of energy dissipated in the main memory when executing programs from SPEC95 CPU Benchmark suit and media benchmarks for a system with memory sizes of 64 Mb, 128 M, 256 Mb, and 512 Mb.

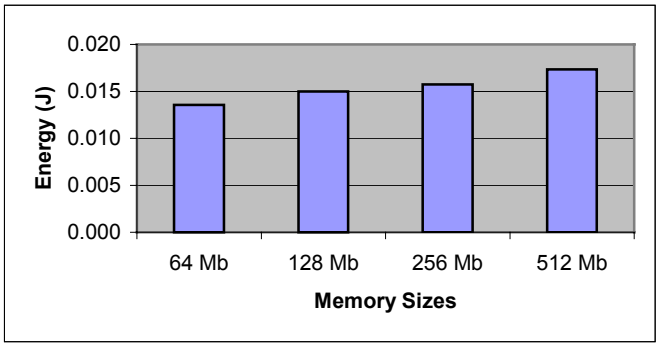

Figure 4: Main memory energy dissipation for different memory sizes (average of all the benchmarks)

The currents drawn in various states by the SDRAM chip increase as the memory size increases. The main memory energy consumption for the same set of programs increases almost linearly with increase in memory size. The additional energy dissipation is due to energy consumed in refreshing additional memory cells, and increase in the read and write access time for larger memory sizes.

Figure 5 shows the fraction of total memory energy consumed by various operating modes in the SDRAM.

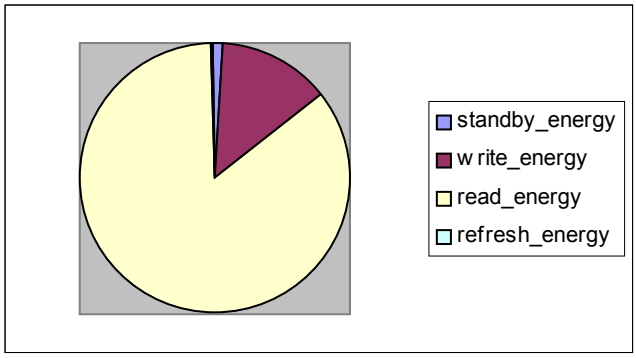

Figure 4: Breakdown of main memory power

We observe from the pie chart that the energy consumed during the read operations is the dominating factor in governing the entire memory energy consumption. This is due to the program characteristics of having more reads than writes, and the energy consumed when driving the read data bits through the data pins of the SDRAM chip. The energy consumed during the refresh and standby states

is extremely small. This justifies the strategy of the memory controller to transition to *standby* state when there are no pending read or write requests.

V. Energy-performance trade-off in designing memory hierarchies

In order to demonstrate an application of the SDRAM power model, we conducted an experiment to apply this model to the memory hierarchy design exploration.

As the complexity of the cache increases (higher-set-associativity, larger block size, and bigger cache size), the energy consumed by a cache access increases, and the main memory energy consumption reduces (due to increase in hit-rate). However, the cache access time increases and this may adversely affect the system performance. Therefore, the selection of the optimal cache and memory size during the design of a memory hierarchy involves making a trade-off between energy-performance for the given application domain. In this experiment we compare the memory hierarchy energy consumption (cache and main memory) with the system performance for various sizes and styles of data caches and different memory sizes.

V (a) Data Collection

We observed the energy consumption in the memory hierarchy and the system performance in terms of Instructions Per Cycle (IPC) of popular data cache designs over a range of sizes (4K, 8K, 16K, and 32K). Specifically we consider varying ranges of associativity (direct mapped, 2-way, 4-way, and 8-way) and block sizes (32 and 64 bytes) for each cache size. In this experiment we limit the investigation to data caches, but this work could easily be extended for instruction caches.

Programs in the SPEC95 CPU Benchmark suite and media benchmarks were simulated using the Wattch simulator for various cache sizes and styles mentioned above, and system main memory sizes of 64 Mb, 128 Mb, 256 Mb, and 512 Mb. The SimplesScalar simulator was configured as an 8-way superscalar, with 8 memory ports, and an average memory latency of 72 processor cycles. The L1 and L2 cache access latencies were computed using CACTI 3.2. The IPC and memory hierarchy energy consumption was averaged across all the benchmarks. The obtained results were then analyzed to understand the trade-off between optimizing memory hierarchies for energy consumption and speed.

V (b) Analysis of Results

Figure 6 shows a plot of the energy consumption and performance for various data cache sizes and main memory size of 128 Mb. As expected the IPC increases with increase in cache sizes. The memory hierarchy energy consumption is fairly constant for cache sizes up to 16KB. However, there is a sharp increase in memory hierarchy energy consumption when cache size is increased to 32KB.

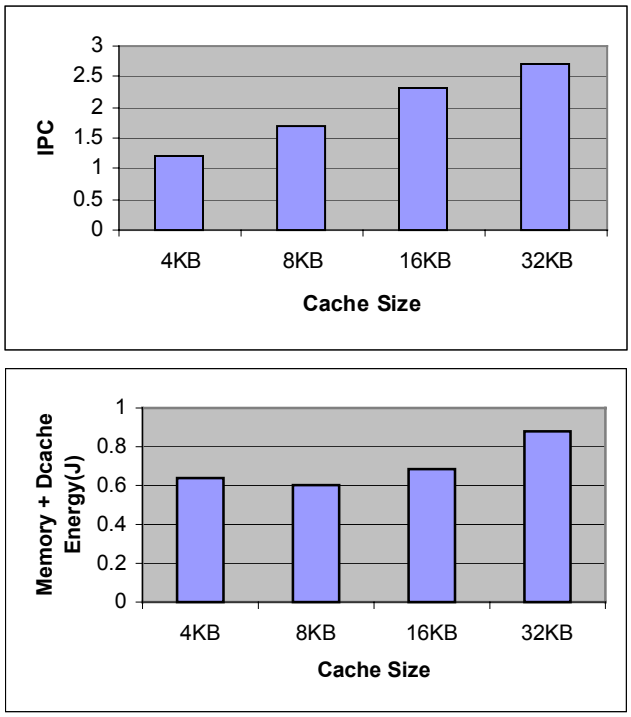

Figure 6: Performance and Energy for various cache sizes

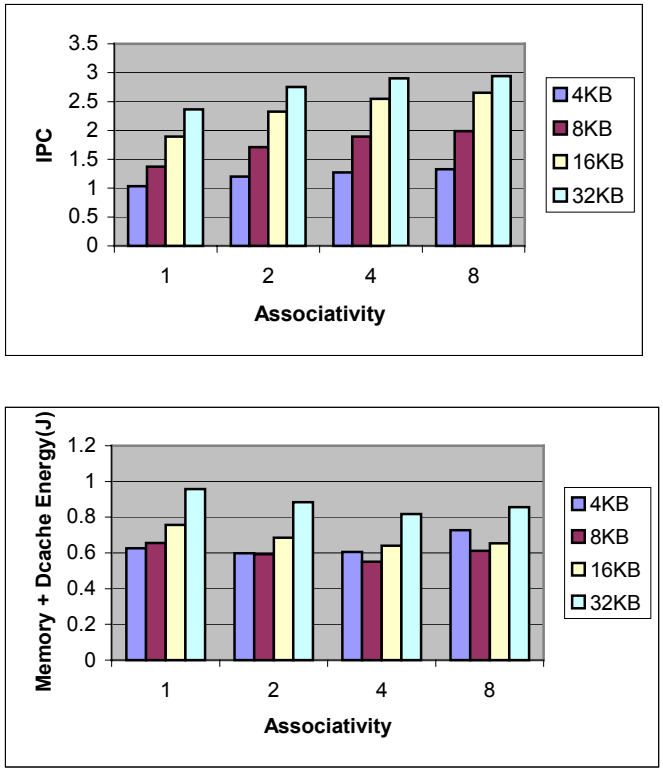

Figure 7: Performance and Energy for different associativity

Figure 7 shows a plot of the energy consumption and performance for various associativity of data cache and main memory size of 128 Mb. We observe that IPC increases with increase in associativity for all data cache sizes. However, this improvement in IPC saturates after associativity of 4. The energy consumption for all cache sizes is least for data cache associativity of 4.

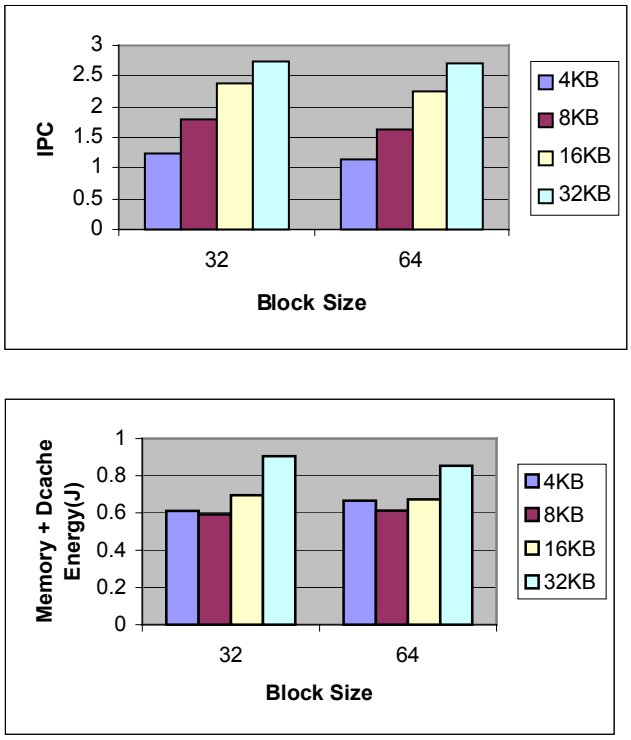

Figure 8: Performance and Energy for block sizes of 32 and 64 bytes

Figure 8 shows a plot of the energy consumption and performance for two different block sizes of the data cache and main memory size of 128 Mb. IPC slightly decreases when block size is increased from 32 to 64 bytes. However, the energy consumption for different cache sizes almost remains the same with increase in block size, except for cache size of 32 KB, where it slightly decreases.

From the above analysis we may conclude that a 4-way 16KB data cache with a block size of 32 bytes is the most optimal configuration for this class of workloads, if we make a trade-off between performance and energy consumed in memory hierarchy.

VI. Conclusions

In this paper we have proposed a power model for SDRAMs and integrated it with an existing architectural level power simulator. Since the model is based upon the various states the SDRAM transitions through, and the current it draws in these states, it is very close to the actual memory power consumption. We have demonstrated an application of this model to analyze the energy-performance trade-off in designing memory hierarchies.

VII. Future Work

We plan to validate the accuracy of this model by measuring the energy consumption on real hardware. We also plan to extend this model for SDRAM memory modules (DIMMs), which are widely used in high-end computers.

Acknowledgements

We would like to express our deep sense of gratitude to Dr. Lizy K. John for her guidance and invaluable suggestions throughout this project. We would also like to thank Madhavi Valluri for jump-starting this project by helping us with setting up the tools and the benchmarks, and answering all our queries.

References

[1] D. Brooks, V. Tiwari V., and M. Martonosi, "Wattch: A Framework for Architectural-Level Power Analysis and Optimizations", *in Proceedings of the 27th International Symposium on Computer Architecture, ISCA'00*, June 2000, pp. 83-94.

[2] D. Burger, and T.M. Austin, "The SimpleScalar Tool Set, version 2.0", *ACM SIGARCH Computer Architecture News*, Vol. 25, No. 3, June 1997

[3] Coumeri et al., "Memory Modeling for System Synthesis", *Proceedings of the 1998 International Symposium on Low Power Electronics and Design*, pp 179-184.

[4] V. Delaluz, M. Kandemir, N. Vijaykrishnan, A. Sivasubramaniam, and M.J.Irwin, "DRAM Energy Management Using Software and Hardware Directed Power Mode Control", *In Proceedings of HPCA 2001,* pp 159-170

[5] V. Delaluz, M. Kandemir, N. Vijaykrishnan, A. Sivasubramaniam, M.J. Irwin, "Hardware and software techniques for controlling DRAM power modes", *IEEE Transactions on Computers, Volume: 50*, *Issue: 11,* Nov. 2001, pp 1154-1173

[6] M. Gries, "A Survey of Synchronous RAM Architectures", Technical Report, TIK-Report No. 71, Computer Engineering and Networks

Laboratory, Swiss Federal Institute of Technology, Zurich, Switzerland, April 19, 1999

[7] P. Hicks, M. Walnock, and Robert Michael Owens, "Analysis of power consumption in memory hierarchies", *Proceedings of the 1997 International Symposium on Low Power Electronics and Design*, pp 239-242

[8] Y. Joo et al., "Energy Exploration and Reduction of SDRAM Memory Systems", *Proceedings of the 2002 Design Automation Conference,* pp 892-897

[9] N. Kim, T. Austin, T. Mudge, and D. Grunwald. "Challenges for architectural level power modeling", *in Power Aware Computing, (R. Melhem and R. Graybill eds.), Kluwer Academic Publishers: Boston, MA*, 2001

[10] N. Kirubanandan, A. Sivasubramaniam, N. Vijaykrishnan., ; Kandemir., M., Irwin, M.J., "Memory energy characterization and optimization for the SPEC2000 benchmarks", *Workload Characterization, 2001. WWC-4. 2001 IEEE International Workshop* on, 2 Dec. 2001 Page(s): 193 –201

[11] P.E. Landman and J.M. Rabaey, "Architectural Power Analysis: The Dual Bit Type Method", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, June 1995, Vol. 3, no. 2, pp. 173-87

[12] V. Tiwari et al., "Reducing Power in High-performance Microprocessors", *Design Automation Conference, 1998*, pp. 732-737

[13] N. Vijaykrishnan et al., Energy-driven Integrated Hardware-Software Optimizations using SimplePower, *in Proceedings of the 27th Annual International Symposium on Computer Architecture, ISCA'00*, June 2000, pp. 95-106

[14] Calculating Memory System Power for DDR SDRAM, http://www.micron.com/