An Effective SDRAM Power Mode Management Scheme for Performance and Energy Sensitive Embedded Systems

Ning-Yaun Ker
Dept. of Electrical Engineering, National Cheng-Kung University
No.1 University Road, Tainan, Taiwan 70101, R.O.C
Tel:886-6-2757575-62400-722
dark@casmail.ee.ncku.edu.tw

Chung-Ho Chen
Dept. of Electrical Engineering, National Cheng-Kung University
No.1 University Road, Tainan, Taiwan 70101, R.O.C
Tel:886-6-2757575-62394
chchen@mail.ncku.edu.tw

Abstract- We present an effective power mode management scheme used in SDRAM memory controllers. The scheme employs a bus utilization monitoring mechanism to initiate proper operations of SDRAM chips. Our approach reduces energy consumption by actively switching memories to low-power mode at low bus utilization. At higher bus utilization, the scheme switches memories to open page mode to reduce precharge energy as well as program execution time. This bus utilization predictor reduces memory energy consumption without the expense of increasing program execution time. It achieved the performance level of open page policy by consuming 20% less of memory energy.

I. Introduction

Memory chips occupy a great portion of energy consumption in an embedded system. Several schemes have been proposed to reduce SDRAM power consumption using low-power-mode control [1-4]. These approaches predict the number of inter-access cycles [2]. The inter-access cycles are the idle cycles between transfers. If the inter-access time is long and stable enough, then the schemes will help. However, it takes time penalty when these low power control schemes predict a wrong number of inter-access cycles. In general, predictor based schemes reduce memory energy consumption at the expense of increasing program execution time.

To reduce memory energy use without increasing the program execution time, we developed an SDRAM power mode management scheme that uses a bus-utilization monitor to initiate low power mode operations as well as page mode selection. This scheme successfully reduces power consumption of SDRAM modules when the memories have many inter-access cycles. The scheme also reduces program execution time of the system when the SDRAM chips are accessed frequently.

The rest of the paper focuses on the architecture and design of the proposed scheme. Section II illustrates the background and observations from experiments. Section III describes the architecture of the proposed SDRAM controller, and Section IV shows the results of simulated system. This paper is concluded in Section V.

II. Preliminaries

We describe two bank activation policies and observations from the results of various predictor schemes.

A. Bank Active and Inactive Management Policy

The first policy for bank management is keeping-bank-active. The other is keeping-bank-inactive. They are also called open-page policy and close-page policy respectively [3]. In an open page policy, the selected memory row is opened as long as possible in order to save latency due to pre-charge and bank activation. A closed-page policy pre-charges an active memory row as soon as possible. An SDRAM controller could use one of the two policies; however, they can't be enabled at the same time.

The policy which keeping bank active results in a shorter latency when the row number of this access is the same with the previous one in the same bank. However, it takes more cycles than the keeping-bank-inactive policy when the row number is different. In an open-page policy, a hit reduces the transfer latency by changing state from the active-standby state to Read/Write state directly without passing the pre-charge and active states. As a result, this approach not only reduces transfer latency but also the operating current of the SDRAM modules.

We have implemented an integrated CPU/Memory controller system in Verilog HDL code and performed simulations for the above two policies. The experimental results are shown in Table 1. We find that the energy consumption and program execution time using open page policy is always less than that using close page policy when the hit rate is high. By comparing the case of "B-sort v.s. matrix-op" and "CRC32 v.s. Fib-seq", we observe that an SDRAM controller performs better when the hit rate is high for the same bus utilization rate (BU rate). By comparing the cases of "B-sort v.s. Seq-search", we can find that this SDRAM controller does not perform well when the hit rate is high in the case of low bus utilization rate. The results of Table 1 reveal that an SDRAM controller using open page policy is more beneficial when the hit rate and bus utilization rate are both higher.

0-7803-7659-5/03/$17.00 ©2003 IEEE.

Table 1. Comparisons between open page policy and close page policy.

Benchmarks	Hit rate	BU rate	Energy consumption	Program execution time
B-sort	81.40 %	37.8 %	93.48 %	92.57 %
matrix-op	65.48 %	37.2 %	96.87 %	96.21 %
CRC32	87.32 %	23.2 %	94.54 %	94.09 %
Seq-search	87.27 %	12.7 %	96.99 %	97.05 %
Fib-seq	98.39 %	25.5 %	92.95 %	92.16 %

1. BU: Memory bus utilization.
2. Energy consumption → (energy consumption of open page policy / energy consumption of close page policy) *100%
3. Program execution time → (program execution time of open page policy / program execution time of close page policy) *100 %.

B. Low Power Mode Control Schemes

If an SDRAM module has not been accessed for a while, it can transfer to a power down mode for saving energy [2]. An SDRAM controller with Constant-Threshold-Predictor (CTP) transfers an SDRAM module from inactive standby mode to power down mode based on a constant number of idle cycles measured by statistics or calculations. In a history-based predictor (HBP), the model estimates the inter-access time based on previous inter-access time. The controller directly transits to the best energy mode, and activates the module at the end of inter-access time. In an immediate-power-down predictor, the SDRAM module directly transits to power down mode whenever idleness occurs.

In our investigations, the SDRAM controller explores the above schemes. The results show that these predictor-based schemes are able to reduce a lot of energy consumption during idle periods. However, they have to pay extra latency when returning to active mode at the end of the idle periods. Is it worthy to gain the energy benefits at the expense of increasing program execution time?

Fig.1 shows the amount of reduced energy consumption of using various predictors. The reduced amount is compared with the mode when the SDRAM controller does not use any predictor for low energy mode operation. The constant threshold predictor counts different idles cycles before entering a low power mode. A smaller value such as CTP8 means that it enters a low power mode much more frequently than a larger value such as CTP64. We evaluate these power mode control schemes using random access load/store generator.

OP (Optimal predictor) is the optimal case where an SDRAM module goes into low power mode as soon as the access is completed and pays no penalty when returning back to the active mode. NOPD (No power down mode) is the baseline model where the SDRAM module does not enter any low power mode. Thus, NOPD consumes the largest amount of energy while OP consumes the least amount.

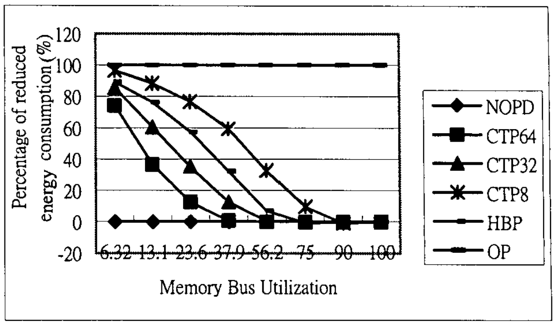

Fig. 1 Comparison of reduced energy consumption in different bus utilization. (Percentage of reduced energy consumption = (NOPD – CTP8) / (NOPD-OP) using CTP8 as an example.

The results in Fig.1 show that the CTP8 model reduces a lot of energy in low bus utilization. In general, these predictors are very effective in saving energy while the bus utilization is low. When the bus utilization is increased, these predictors become more and more inefficient. At higher utilization rate (above 60%), the energy saving rate diminishes rapidly. This is because there is less chance to enter power down mode given the constant value used in the predictor.

Fig. 2 shows the increased execution time in percentage for these predictors. We observed that SDRAM modules should enter low power mode as soon as possible when the bus utilization is low. When the bus utilization increases, frequent state transition does not gain much reduction in energy consumption. On the contrary, the program execution time is increased significantly. For instance, the CTP8 predictor is the most aggressive policy in this experiment; however, it has the highest overhead in terms of program execution time.

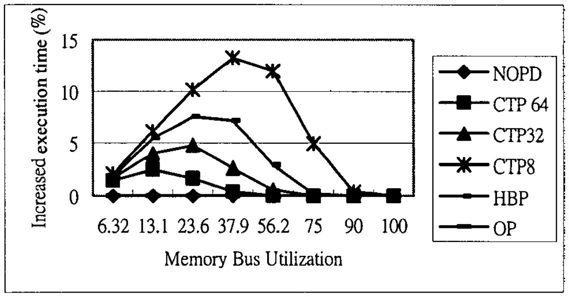

Fig. 2 Comparison of program execution time in different bus utilizations.

III. Bus-Utilization Monitor Scheme

The above experimental results indicate that

- A power mode control scheme performs better when the bus utilization is low.
- Open page policy architecture performs better when the bus utilization and hit rate are both high.
- The power mode control scheme reduces hit rate since the SDRAM banks are inactive.

Based on these observations, we have designed an SDRAM controller that operates in close page policy with a power mode control scheme when the bus utilization rate is low. The controller operates in open page mode when the bus utilization reaches a threshold value.

Fig. 3 shows the architecture of the SDRAM controller using bus-utilization monitor scheme. This architecture includes a bus monitor for the calculations of bus utilization rate, an idleness predictor for power mode control, a page number cache for the hit/miss information, a hit/miss signal generator, a latency counter for command latency, an auto refresh counter for auto refresh command, and an address decoder. The SDRAM controller computes the bus utilization every 128 cycles over a 512-cycle basis.

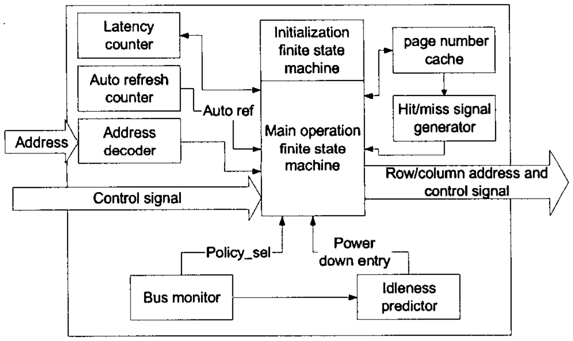

Fig. 3 Architecture of SDRAM controller using bus-utilization monitor scheme.

IV. Simulation Environment

A. Simulation platform

The simulation system uses an 8-bit RISC CPU as the host signal generator. The SDRAM controller and CPU are all written in Verilog RTL code.

The developed scheme was evaluated by running the benchmark programs on the RISC processor. Table 2 lists the specifications of the system used in the evaluation. Table 3 lists the characteristics of the current of the SDRAM module used in the evaluation.

Five testing programs are used: B-sort is a bubble sort operation program; matrix-op is a matrix operation program; CRC32 performs cyclic redundancy check operation; Seq-search is sequential search operation program; Fib-seq generates Fibonacci sequence. Table 4 shows the average bus utilization rate and load/store percentage of the benchmarks. The scheme changes policies based on the bus utilization rate.

Table 2. Simulation platform characteristics

8-bit RISC multi-cycle CPU with 100MHz clock rate.
Fully synchronous; all signals registered on positive edge of system clock.
Self Refresh and Adaptable Auto Refresh Modes 64ms , 4096-cycle refresh. CAS latency =3.
SDRAM data bus width = 8bits, address bus width = 11bits.
Column address width 10bit. (A10 = auto pre-charge enable , A9 = don't care)
Row address width 11bit. Bank width 1bit.

Table 3. Current characteristics

Parameter	Current
Operating Current: Active Mode.	90mA
Standby Current and Self refresh current:	2mA
Standby Current: Active Mode.	40mA
Operating Current: Burst Mode; Continuous burst.	85mA
Auto refresh current:	85mA

Table 4: Benchmark program characteristics

Test program	Bus utilization rate	Load/Store rate
B-sort	37.8%	43%
Matrix-op	37.2%	39.8%
Fib-seq	25.5%	15.1%
CRC32	23.2%	15.9%
Seq-search	12.7%	5.7%

B. Result of Simulation

The schemes evaluated include open page policy (op), closed page with immediate entering low power mode when idle, (cp_ald), closed page with history-based predictor (cp_hbp), close page with constant threshold predictor (cp_ctp8), open page with constant threshold predictor (op_ctp8), and bus utilization monitor scheme (bump). In bump scheme, when the bus utilization is below 20%, the controller operates in close page policy with immediate entering low power mode when idleness occurs. When the bus utilization is greater than 25%, the controller operates in open-page policy.

Fig. 4 shows the comparisons of energy consumption among the schemes investigated. The energy consumed by the cpp policy is used as a base. Fig. 5 compares the program execution time between the standard scheme (cpp) and the improved schemes (op, cp_ald, cp_hbp, cp_ctp8, op_ctp8, and bump). We set the energy consumption and execution time of the standard scheme (cpp) to be 100%.

The SDRAM controller using open page policy (op) reduced energy consumption by 5% and execution time by 5.6 %. The SDRAM controller using close page policy and immediately entering low power mode (cp_ald) reduced energy consumption by 50%. It performs much better than the others, however this reduction in memory energy is at the expense of the program execution time. Table 5 shows the weighted average of energy consumed and execution time of simulated programs. An SDRAM controller using cp_ald scheme reduces the energy consumption of SDRAM module by 50%; however, it also increases the system execution time by about 8.6% compared to ccp scheme.

The bus utilization monitor predictor (bump) possesses a well-balanced performance between program execution time and energy reduction rate. Compared with the open page policy (op), bump has achieved the performance level of high-speed access by using much less energy. By adding a small bus utilization monitoring logic into the controller, the bump scheme reduces energy consumption by 26% while maintaining the performance level that an open page policy can achieve. Notice that op_ctp8 and bump have about the same reduction rate in energy consumption; however, bump has less program execution time.

Table 5. Weighted average of energy consumption and program execution time.

	Average weighted energy consumption	Average weighted execution time
cpp	100	100
op	94.96	94.42
cp_ald	49.46	108.58
cp_hbp	61.03	105.19
cp_ctp8	71.32	104.85
op_ctp8	72.40	102.43
bump	73.92	98.06

V. Conclusion

We present an effective power mode control scheme that changes SDRAM bank access policy dynamically based on bus utilization. The scheme reduces both the energy and program execution time. The bus utilization predictor has achieved the performance level of high-speed open-page mode by using much less energy. Other close page-based predictor schemes have good results in memory energy reduction but at the expense of program execution time.

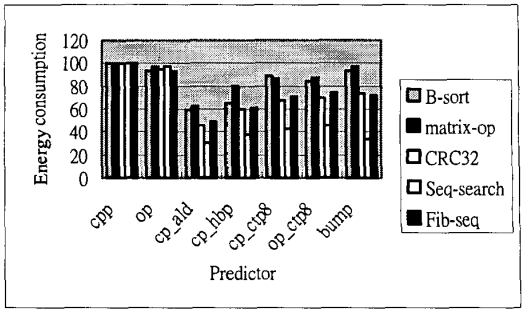

Fig. 4 Comparison of energy consumption using different predictors.

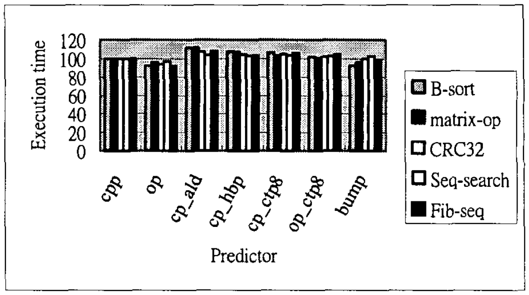

Fig. 5 Comparison of program execution time using different predictors.

Acknowledgments

The work of this paper is in part supported by the NSC-91-2213-E-006-083, Republic of China, Taiwan.

REFERENCES

[1] Miura, S. Ayukawa, K. Watanabe, T. "A dynamic-SDRAM-mode-control scheme for low-power systems with a 32-bit RISC CPU" International Symposium on Low Power Electronics and Design, 2001, pp. 358 – 363.

[2] Delaluz, V. Kandemir, M. Vijaykrishnan, N. Sivasubramaniam, A. Irwin, M.J. "Hardware and software techniques for controlling DRAM power modes," IEEE Transactions on Computers, Vol. 50, No. 11, 2001 pp. 1154 - 1173.

[3] Gries, M. "The impact of recent DRAM architectures on embedded systems performance", Proceedings of the 26th Euromicro Conference, Sept. 2000, pp. 282 – 289.

[4] Xiaobo Fan Ellis, C.S. Lebeck, A.R. "Memory controller policies for DRAM power management" International Symposium on Low Power Electronics and Design, 2001, pp. 129 – 134.