High Performance Memory Mode Control for HDTV Decoders

Seong-Il Park, Yongseok Yi, Student Member, IEEE and In-Cheol Park, Senior Member, IEEE

Abstract — *To increase the bandwidth of synchronous memories that are widely adopted for HDTV decoder systems, a predictive mode control scheme is proposed in this paper. Memory latency and energy consumption can be reduced by effectively managing the states of banks. The local access history of each bank is considered to predict the memory mode. In a HDTV decoder system, experimental results show that the proposed scheme reduces the memory latency and the energy consumption by 18.8% and 23.3%, respectively, over the conventional scheme that always keeps the memory in idle state. A hardware architecture and its VLSI implementation are also presented*[1].

Index Terms — HDTV decoder, History-based prediction, memory controller, memory performance, synchronous memory.

I. INTRODUCTION

In many applications such as portable wireless devices and multimedia systems, several factors such as increased system complexity, time-to-market pressure, cost effectiveness, and various functionality requirements have made the trend of system-on-a-chip (SoC) design indispensable [1][2][3]. In general, SoC devices are connected to off-chip memories that feed instructions and data to the programmable processors and temporarily store data to be transferred between functional blocks. As the SoC integrates more functional blocks and needs higher performance to carry out ever increasing tasks, high data bandwidth is required to meet a given system specification.

High definition television (HDTV) decoders have been integrated into a single chip to exploit the merits of SoC as illustrated in **Fig. 1** [4][5][6][7]. The HDTV decoder SoC consists of a system parser, a video decoder, an audio decoder, a display controller, and peripheral interfaces. The HDTV decoder SoC uses off-chip memory to buffer the MPEG-2 bitstream and temporarily store data to be decoded and displayed. Since high memory bandwidth is required to deal with large amount of video data in a given time specification, synchronous memories such as Synchronous DRAM (SDRAM) and Rambus DRAM are widely used to increase data transfer speed, reduce clock cycle time, and ease synchronous design [5][8][9].

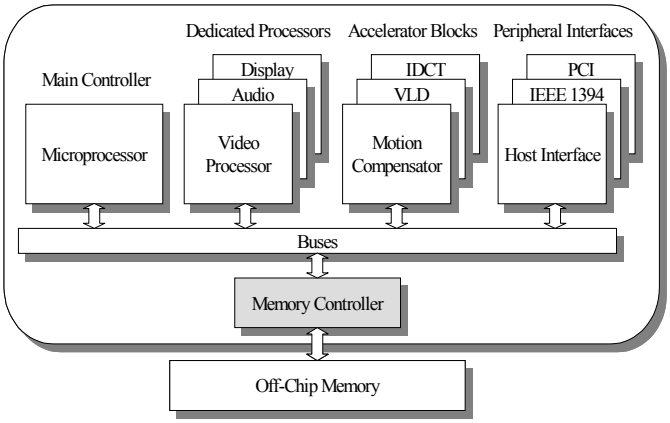

Fig. 1. HDTV decoder SoC containing programmable processors and functional blocks.

Several architectural features developed to alleviate memory latency enable the synchronous memories to meet the bandwidth requirement [10][11]. The features are based on the fact that all the cells along a word line are latched to sense amplifiers when the row is selected and activated, and can be reused without additional row-activation and precharge as long as the row addresses of successive accesses in the corresponding bank are identical. The row-active state can be used to reduce the latency and the power consumption of memory operations if the successive memory access refers to the same row in the same bank (a page hit). However, if the row address differs from the previous one (a page miss), additional cycles that cannot be hidden are needed for a precharge and a row-activation, resulting in performance degradation. Therefore, to increase memory performance, the memory controller has to control the operation mode by efficiently predicting whether the next memory reference will be a page hit or not.

Several optimizations have been proposed to reduce page misses by statically scheduling the address sequence in memory and controlling the memory operation mode [9][12][13]. Those techniques are successfully applied to image and video processing applications, in which memory access patterns are relatively regular enough to be known in advance. In the HDTV decoder system, however, several functional blocks and processors are connected to the external memory through a shared bus, and as a result, memory access patterns become irregular. The irregularity is caused by the motion compensation and the mixed memory accesses of several functional blocks.

A dynamic memory mode control scheme [14] has been proposed to manage the memory operation mode according to runtime behavior of memory access patterns. The state of

[1] This work was supported in part by the IDEC and the MICROS centers.

The authors are with the Division of Electrical Engineering, Korea Advanced Institute of Science and Technology, 373-1, Guseong-dong, Yuseong-gu, Daejeon, 305-701, Republic of Korea (e-mail: sipark@ics.kaist.ac.kr; yslee@ics.kaist.ac.kr; icpark@ee.kaist.ac.kr).

Contributed Paper
Manuscript received June 23, 2003

SDRAM is changed from idle to row-active state if a memory access leads to a page hit and sustains the row-active state until the number of the successive page misses exceeds a threshold value. The dynamic scheme is effective if in-row accesses are dominant. However, if in-row accesses are not dominant and the pattern of memory accesses is irregular, frequent mode transitions lead to many overhead cycles needed for precharges and row-activations.

In this paper, we propose a new dynamic memory mode control scheme to reduce memory latency by predicting the next operation mode. The prediction is based on the history of memory references. SDRAM is used to show the effectiveness of the proposed control scheme.

The rest of the paper is organized as follows. Section II gives a brief background on the architecture, the bank states, and the operations of SDRAM. We describe the proposed dynamic memory mode control scheme in Section III. In Section IV, experimental methodology and results are presented. VLSI architecture and implementation are presented in Section V. Finally, conclusions are made in Section VI.

II. BACKGROUND ON SDRAM

Fig. 2 shows a simplified block diagram of SDRAM, which consists of four independent banks. The four banks share address buffers and I/O buffers, while each bank has its own row decoders, column decoders, sense amplifiers, and a memory array. The state of resources of a bank is maintained independently.

Each bank has two stable states that are idle and row-active states as shown in **Fig. 3**. The idle state is entered by the precharge operation. The state transition from idle to row-active is made by the row-activation operation. Column access operations do not change the state of the bank. Thus the bank is in row-active state as long as the precharge operation is not performed.

The operation mode of SDRAM is controlled by a memory controller that translates a read/write request into a sequence of memory commands. Three major operations of SDRAM are as follows:

- Row-activation: The bank and the row where the data are accessed are selected and activated. Then, all the cells along a word (row) line of the bank are latched to the corresponding sense amplifiers. The bank is in row-active state after completing the operation.
- Column access: The column access operation selects and accesses a column of the activated row. A number of words equal to the burst length are read out from the sense amplifiers to the I/O buffers, one word per clock.
- Precharge: By the precharge operation, the sense amplifiers are precharged and the bank of the SDRAM is made to stay in idle state. A row-activation command can be issued when the state of the corresponding bank is idle.

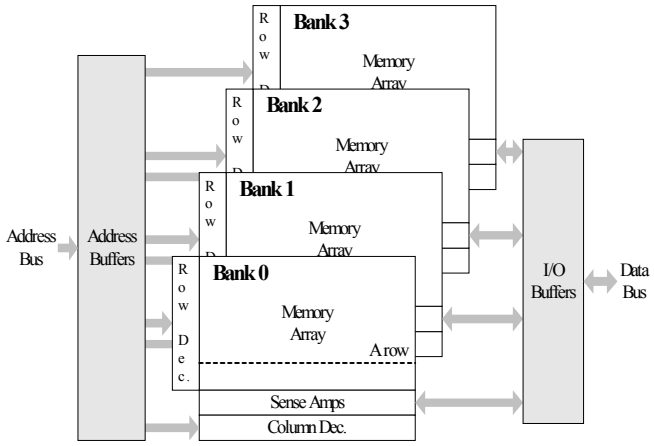

Fig. 2. Block diagram of SDRAM.

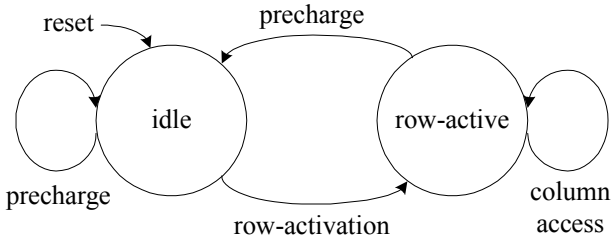

Fig. 3. State diagram for each bank.

III. HISTORY-BASED MEMORY MODE PREDICTION

The operation mode is controlled by commands generated by the memory controller. The read/write commands with the auto-precharge option change the memory to idle state after completing the corresponding operations as depicted in **Fig. 4**. As the precharge time (t_{RP}) can be overlapped with burst accesses or data transfer between the memory controller and the processor, the effective latency is the sum of the row-activation time (t_{RCD}) and the column select latency (t_{CL}). The read/write commands without the auto-precharge option maintain the memory in row-active state. If the successive access brings a page hit, the precharge and row-activation operations are not necessary. In this case, the effective latency can be reduced to t_{CL}. If the successive access leads to a page miss, a precharge, a row-activation, and a column select operation have to be performed, increasing the effective latency to $t_{RP} + t_{RCD} + t_{CL}$. Therefore, the memory mode must be controlled to stay in row-active state as long as possible and to minimize the number of page misses.

Although the address requested by the processor is random and unknown in advance, the principle of locality of memory reference [15] makes it possible to predict whether the successive access refers to the same row or not. Using the past history of memory references, we predict if the next access causes a page hit and control the memory mode according to the prediction. If the history predicts the successive access to refer to the same row, the memory controller makes the bank remain in row-active state. Otherwise, the bank is changed to idle state.

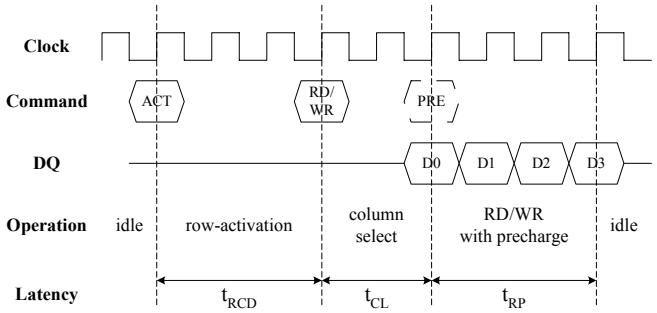

Fig. 4. Read/write operation with the auto-precharge option.

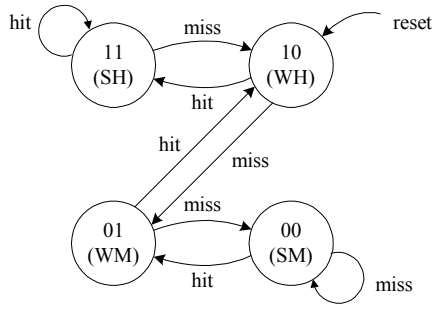

Fig. 5. State machine for storing page hit history information.

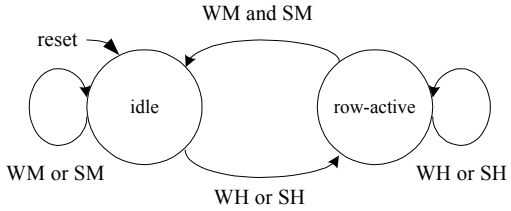

Fig. 6. State transition diagram based on the past reference history.

To store the past history of memory accesses, a state machine that can be built with a two-bit saturated up/down counter is used for each row (per-row counter), as shown in **Fig. 5**. The corresponding state machine is incremented on a page hit and decremented on a page miss after comparing the row and the bank addresses with those of the previous access.

For a pending memory access, the memory controller issues a command without the auto-precharge option if the state machine selected by the row and the bank addresses is in strongly hit (SH) or weakly hit (WH) state. If the state machine is in strongly miss (SM) or weakly miss (WM) state, a command with the auto-precharge option is issued. The state transition utilizing the past reference history is depicted in Fig. 6.

Although the per-row predictor can accurately reflect the behavior of memory references to the corresponding row, area overhead is considerable. For example, if a memory has a N-bit row address and M banks, $M \bullet 2^N$ two-bit counters are required. To reduce the area overhead while keeping prediction accuracy moderate, one two-bit counter is used for each bank (per-bank counter) instead of each row. As only M state machines are required in this case, significant area reduction is achieved at the loss of a little prediction accuracy. Among M state machines, one is selected by the bank address.

TABLE I
COMPARISON OF HIT-PREDICTION RATE FOR A HDTV DECODER

Benchmark	Previous scheme [14] (%)	Per-row counter (%)	Per-bank counter (%)
HDTV decoder	55.3	77.0	75.3

TABLE II
COMPARISON OF HIT-PREDICTION RATE FOR SPEC92 BENCHMARKS

Benchmarks	Previous scheme [14] (%)	Per-row counter (%)	Per-bank counter (%)
008.espresso	52.1	70.6	58.4
023.eqntott	73.9	86.9	80.5
056.ear	62.4	72.3	71.0
085.gcc	69.8	81.3	76.8
090.hydro2d	47.6	60.0	59.7
Average	61.2	74.2	69.3

IV. EXPERIMENTAL RESULTS

To evaluate the effectiveness of the proposed history-based mode prediction scheme, we measure memory latency and memory energy consumption by performing trace-driven simulation for a HDTV decoder system. Memory traces obtained by observing the shared bus are used as an input vector in the simulation. In addition, data memory traces of five SPEC92 benchmark programs are simulated to show the effectiveness in various applications.

A. Latency Estimation

As given in the following equation, the total memory latency is calculated by counting all the individual latencies.

$$\begin{aligned} \text{Latency} = {} & N_{idle} \times (t_{RCD} + t_{CL}) + \\ & N_{hit} \times t_{CL} + \\ & N_{miss} \times (t_{RP} + t_{RCD} + t_{CL}) \end{aligned} \quad (1)$$

where N_{idle} is the number of idle states, N_{hit} is the number of page hits in row-active state, and N_{miss} is the number of page misses in row-active state. In the simulation, a SDRAM that has a 13-bit row address, a 9-bit column address, and 4 banks is assumed. The precharge time (t_{RP}), row-activation time (t_{RCD}), and column select time (t_{CL}) are assumed to be three, three, and two (zero for write operations) cycles, respectively, which are quoted from a commercial SDRAM.

Table I and Table II show the ratio of the number of correct predictions to the number of total references for a HDTV decoder and SPEC92 benchmarks, respectively. The history-based scheme with per-row counters shows the highest hit-prediction rate, and the history-based scheme with per-bank counters predicts more accurately than the previous scheme. In the HDTV decoder system, the history-based scheme with per-bank counters predicts more accurately than the previous scheme by 20%. The large difference in hit-prediction ratio is caused by the irregularity of memory reference made by several functional blocks and processors.

TABLE III
COMPARISON OF TOTAL LATENCY CYCLES FOR A HDTV DECODER

Benchmark	Always idle	Previous scheme [14]	Per-row scheme	Per-bank scheme
HDTV decoder	1002375	934395	803714	813948
Normalized latency (%)	100.0	93.2	80.2	81.2

TABLE IV
COMPARISON OF TOTAL LATENCY CYCLES FOR SPEC92 BENCHMARKS

Benchmark	Always idle	Previous scheme [14]	Per-row scheme	Per-bank scheme
008.espresso	832900	839083	733312	803023
023.eqntott	992517	735501	645729	689577
056.ear	953780	834698	766691	775580
085.gcc	941097	726195	650187	679917
090.hydro2d	1087215	1035732	942354	944796
Normalized average (%)	100.0	86.8	77.8	81.0

As a result of more accurate prediction, the memory latency is significantly reduced even compared to the previous mode control scheme. The latency results are summarized in Table III and Table IV, where we can find that the proposed per-bank prediction scheme outperforms the scheme that always maintains the SDRAM in idle state by 18.8% and 19.0% on the average for the HDTV decoder system and SPEC92 benchmarks, respectively.

B. Energy Estimation

The energy consumption (E) is calculated based on the equations presented in [16], as follows.

$$E = \sum_{i=0}^{M-1} (E^i_{static} + E^i_{dynamic}) \quad (2)$$

$$E^i_{static} \approx P_{stby} \cdot t^i_{stby} \quad (3)$$

$$E^i_{dynamic} \approx N^i_{pa} \cdot E_{pa} + N^i_{rw} \cdot E_{rw} \quad (4)$$

where M is the number of banks, N^i_{pa} is the number of precharge/activation's in bank i, and N^i_{rw} is the number of read/write's in bank i. The energy parameters are shown in Table V, which are quoted from [16]. The operating frequency is assumed to be 133 MHz.

The total memory energy consumptions for a HDTV decoder and SPEC92 benchmarks are summarized in Table VI and Table VII, respectively. The proposed per-bank prediction scheme reduces the energy consumption by 23.3% and 40.8% for the HDTV decoder system and SPEC92 benchmarks, respectively, over the scheme that always maintains the SDRAM in idle state.

Compared to the previous mode control scheme, the proposed mode control scheme considerably improves memory performance without increasing the energy consumption. Therefore, it can be successfully applied to HDTV decoder systems.

TABLE V
ENERGY PARAMETERS

Symbol	Meaning	Value
P_{stby}	Standby power	50 mW
E_{pa}	Energy of a precharge/activation	14000 pJ/miss
E_{rw}	Energy of a read/write	2000 pJ/access

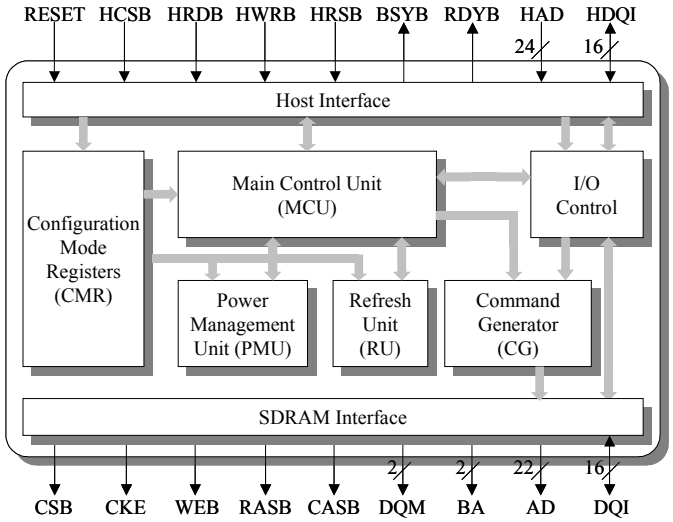

Fig. 7. Architecture of the memory controller.

V. VLSI ARCHITECTURE AND IMPLEMENTATION

Fig. 7 shows the overall architecture of the memory controller. The configuration and mode registers provide parameters needed for the initialization and the control of SDRAM. The refresh unit is responsible for the periodic generation of refresh cycle requests. The power management unit brings SDRAM into power-down mode after the memory request is not accepted for a predefined time interval and recovers the active state when a memory operation is requested. The main control unit performs memory operations by tracking the states of SDRAM.

TABLE VI
COMPARISON OF TOTAL ENERGY CONSUMPTION FOR A HDTV DECODER

Benchmark	Always idle (μJ)	Previous scheme [14] (μJ)	Per-row scheme (μJ)	Per-bank scheme (μJ)
HDTV decoder	5921	4630	4475	4541
Normalized energy (%)	100.0	78.2	75.6	76.7

TABLE VII
COMPARISON OF TOTAL ENERGY CONSUMPTION FOR SPEC92 BENCHMARKS

Benchmark	Always idle (μJ)	Previous scheme [14] (μJ)	Per-row scheme (μJ)	Per-bank scheme (μJ)
008.espresso	5449	3373	3260	3350
023.eqntott	6561	3757	3650	3837
056.ear	6485	3548	3343	3566
085.gcc	6269	3652	3530	3770
090.hydro2d	7168	4807	4341	4391
Normalized average (%)	100.0	59.9	56.8	59.2

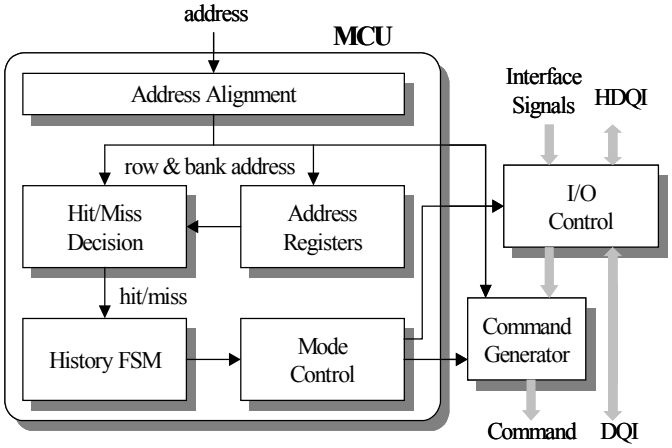

Fig. 8. Main control unit.

The main control unit performs read and write operations by regulating the command generator and the I/O control block as shown in **Fig. 8**. The address alignment block divides the address into a bank address, a row address, and a column address. The hit/miss decision block compares the row address with the previous row address of the same bank to decide whether the access results in a page hit or not. The hit/miss information is fed into the history FSM and updates the corresponding history counter. Considering the state of history counters and SDRAM, the mode control block controls the command generator and the I/O control block.

A fully synthesizable *Verilog* model was described in register-transfer level to implement the memory controller. The memory controller contains about 4700 gates. The operating frequency is 133 MHz and the core size is 0.4 mm × 0.4 mm. Fig. 9 shows a layout of the circuit that was implemented with 0.35 μm 3.3 V four-layer metal CMOS technology.

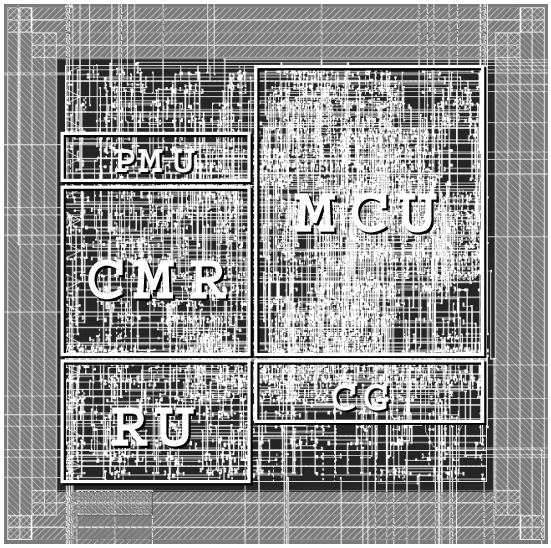

Fig. 9. Layout of the memory controller.

VI. CONCLUSION

To reduce memory latency of synchronous memories, we have proposed a memory control scheme that predicts whether the successive memory access leads to a page hit or not and changes the memory mode according to the prediction. Two-bit state machines are employed to predict the next memory mode based on the history of memory references. Two prediction schemes that use per-row and per-bank predictors are proposed to make a compromise between prediction accuracy and area overhead. Experimental results on benchmark programs and a HDTV decoder show that the proposed scheme is effective in reducing the number of row-activations and precharges, thereby improving memory performance and energy consumption.

REFERENCES

[1] T. Nishitani, "An approach to a multimedia system on a chip," in *Proc. IEEE Workshop on Signal Processing Systems*, Oct. 1999, pp. 13-22.

[2] E. Chou and S. Bing, "System-on-a-chip design for modern communications," *IEEE Circuits and Devices Magazine*, vol. 17, no. 6, pp. 12-17, Nov. 2001.

[3] D. Clark, "Mobile processors begin to grow up," *Computer*, vol. 35, no. 3, pp. 22-25, Mar. 2002.

[4] R. Sita, E.Brosz, R. Meyer, L. Phillips, and R. T. Ryan, "A single-chip HDTV video decoder design," *IEEE Trans. Consumer Electron.*, vol. 44, no. 3, pp. 519-526, Aug. 1998.

[5] O. Duardo, P. Graca, S. Hosotani, S. Sugawa, and H. Jiang, "A cost effective HDTV decoder IC with integrated system controller, down converter, graphics engine and display processor," *IEEE Trans. Consumer Electron.*, vol. 45, no. 3, pp. 879-883, Aug. 1999.

[6] S. Bae, S. Kim, S. Min, W. Kim, and C. Min, "A single-chip HDTV A/V decoder for low cost DTV receiver," *IEEE Trans. Consumer Electron.*, vol. 45, no. 3, pp. 887-893, Aug. 1999.

[7] H. Kim, W. Yang, M. Shin, S. Min, S. Bae, and I. Park, "Multi-thread VLIW processor architecture for HDTV decoding," in *Proc. IEEE Custom Integrated Circuits Conf.*, May 2000, pp. 559-562.

[8] M. Winzker, P. Pirsch, and J. Reimers, "Architecture and memory requirements for stand-alone and hierarchical MPEG2 HDTV-decoders with synchronous DRAMs," in *Proc. IEEE Int'l. Symp.Circuits Syst.*, Apr. 1995, pp. 609-612.

[9] T. Takizawa, J. Tajime, and H. Harasaki, "High performance and cost effective memory architecture for an HDTV decoder LSI," in *Proc. Int'l. Conf. Acoustics, Speech, Signal Processing*, Mar. 1999, pp. 1981-1984.

[10] Y. Choi, M. Kim, H. Jang, T. Kim, S. Lee, H. Lee, C. Park, S. Lee, C. Kim, S. Cho, E. Haq, J. Karp, and D. Chin, "16-Mb synchronous DRAM with 125-Mbyte/s data rate," *IEEE J. Solid-State Circuits*, vol. 29, no. 4, pp. 529-533, Apr. 1994.

[11] Y. Takai, M. Nagase, M. Kitamura, Y. Koshikawa, N. Yoshida, Y. Kobayashi, T. Obara, Y. Fukuzo, and H. Watanabe, "250 Mbyte/s synchronous DRAM using a 2-stage-pipelined architecture," *IEEE J. Solid-State Circuits*, vol. 29, no. 4, pp. 426-431, Apr. 1994.

[12] K. Asheesh, P. R. Panda, N. D. Dutt, and A. Nicolau, "High-level synthesis with synchronous and RAMBUS DRAMs," in *Proc. Workshop on Synthesis and System Integration of Mixed Technologies*, Oct. 1998, pp.186-193.

[13] H. Kim and I. Park, "High-performance and low-power memory-interface architecture for video processing applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 11, pp. 1160-1170, Nov. 2001.

[14] S. Miura, K. Ayukawa, and T. Watanabe, "A dynamic-SDRAM-mode-control scheme for low-power systems with a 32-bit RISC CPU," in *Proc. Int'l. Symp. Low-Power Electronics and Design*, Aug. 2001, pp. 358-363.

[15] J. Hennessy and D. Patterson, "Fundamentals of computer design," in *Computer Architecture: A Quantitative Approach*, 2nd edition, pp. 1-67, Morgan Kaufmann, 1996.

[16] P. Marchal, D. Bruni, J.I. Gomez, L. Benini, L. Pinuel, F. Catthoor, and H. Corporaal, "SDRAM-energy-aware memory allocation for dynamic multi-media applications on multi-processor platforms," in *Proc. Design, Automation and Test in Europe Conference and Exhibition*, Mar. 2003, pp. 516-521.

Seong-Il Park (S'98) received the B.S. degree in electronics engineering from Korea University in 1996, and the M.S. degree in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST) in 1998. Currently, he is a Ph.D. student in the Department of Electrical Engineering and Computer Science at KAIST. His research interests include VLSI design for communication and multimedia applications. He is a student member of the IEEE.

Yongseok Yi (S'01) received the B.S. and M.S. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST) in 1999 and 2001, respectively. Currently, he is a Ph.D. student in the Department of Electrical Engineering and Computer Science at KAIST. His research interests include alternative placement methodologies for large VLSI circuits and high-performance logic simulation. He is a student member of the IEEE.

In-Cheol Park (S'88-M'92-SM'02) received the B.S. degree in electronics engineering from Seoul National University, in 1986, and the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST) in 1988, 1992, respectively. Since June 1996, he has been an Assistant Professor and now an Associate Professor in the Department of Electrical Engineering and Computer Science at KAIST. Prior to joining KAIST, he was with the IBM T.J.Watson Research Center, Yorktown, NY, from May 1995 to May 1996, where he researched high-speed circuit design. His current research interests include computer-aided design algorithm for high-level synthesis and very large scale integration architectures for general-purpose microprocessors. Dr. Park received the best paper award at the ICCD in 1999 and the best design award at the ASP-DAC in 1997.