A Three-Port nRERL Register File for Ultra-Low-Energy Applications

Jun-Ho Kwon
School of Electrical Engineering
Seoul National University
Kwanak P.O. Box 34,
Seoul 151-742, KOREA
+82-2-880-1815
jhkwon@sdgroup.snu.ac.kr

Joonho Lim
Global Communication Technology,
CA 94024
USA
klaatu@gct21.com

Soo-Ik Chae
School of Electrical Engineering
Seoul National University
Kwanak P.O. Box 34,
Seoul 151-742, KOREA
+82-2-880-1815
chae@sdgroup.snu.ac.kr

ABSTRACT

In this paper, we propose an adiabatic register file for ultra-low-energy applications, which uses a new reversible adiabatic logic, nRERL [1]. The nRERL register file discards *garbage* information with minimal energy dissipation. We designed a 16×8b three-port nRERL register file. From SPICE simulations, we found that the nRERL register file consumes less than 10% of the energy consumed in the conventional register file at the frequency of lower than 1MHz. We also describe how to design a RAM, a large array of the storage cells.

1. INTRODUCTION

The adiabatic circuits are useful for ultra-low-energy applications at low operating frequencies because they consume less energy as their operating speed decreases [2]. Recently we proposed nMOS reversible energy recovery logic (nRERL) [1]. Because it exploits the bootstrapped switches and nMOS transistors only, its circuit complexity and energy consumption are reduced substantially, compared to other fully adiabatic logic.

There have been several papers on the adiabatic memories for low-energy applications [3][4][5][6], which were based upon the conventional SRAM cell. However they have limitations in reducing energy consumption at lower-speed operation because they have large non-adiabatic loss, which does not depend on the operating frequency.

In contrast, a reversible memory uses a swap operation instead of erasing the bit stored during read or write operations [7][8] because erasing a bit accompanies non-adiabatic energy dissipation. Recently, a reversible memory was proposed in [9]. However, this fully reversible memory has limitation in implementation due to the *garbage* information. When a new data is written to a memory cell, the old data becomes *garbage*, which should be stored somewhere to avoid non-adiabatic loss due to erasing. Therefore, a huge garbage stack is required. Although a small array of reversible memory was implemented in [9], a large array of fully reversible memory is still not realizable because of the large overhead of garbage stack.

This paper describes an nRERL register file, which employs the reversible adiabatic logic, nRERL, and discards garbage information with minimal energy dissipation. A new nRERL storage cell and other blocks are designed. Also, the method to design a RAM, a large array of the storage cells is described.

ISLPED '00, Rapallo, Italy.

2. nRERL STORAGE CELL

2.1 nRERL

The nRERL [1], which uses a simpler 6-phase clocked power, is an improved version of previous RERL circuits [10][11][12][13]. It is simpler and its energy consumption and area overhead are reduced substantially because the nRERL uses nMOS transistors only by exploiting the bootstrapped nMOS switches. An nRERL buffer is shown in Fig. 1 as well as its 6-phase clock. The detailed description about nRERL and its clocked power generator can be found in [1].

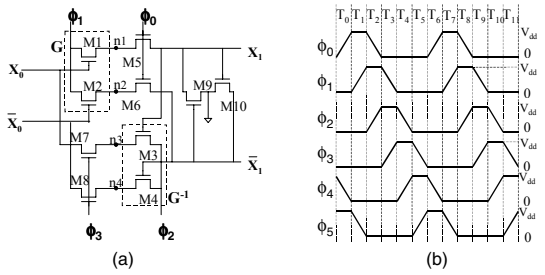

Fig. 1. nRERL: (a) a buffer (b) 6-phase clocked power

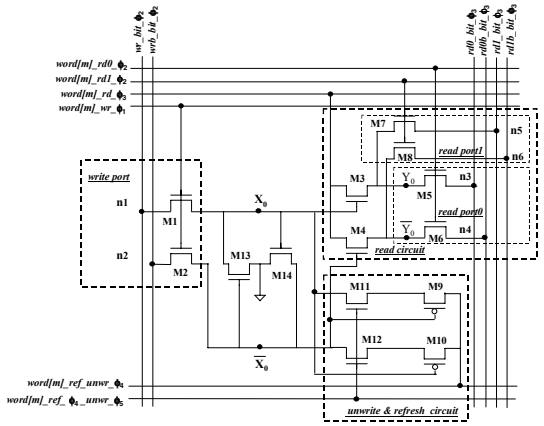

Fig. 2. Storage cell used in the 3-port (2R1W) nRERL register file

2.2 Storage Cell

The storage cell shown in Fig. 2 is used in the three-port (2-read 1-write) nRERL register file, which consists of 14 transistors. Basically it uses the same clocking scheme as the nRERL buffer in Fig. 1. The transistors, M1 and M2 control the write path from the write bit-lines to the storage nodes X_0 and $\overline{X_0}$, respectively, and M3, M4, M5 and M6 control the read path for read port0. Similarly, M3, M4, M7, and M8 control the read path for read port1. The transistors of M9, M10, M11 and M12 comprise an Self-Energy-Recovery Circuit (SERC) [10] which recovers the energy of X_0 and $\overline{X_0}$ in the unwrite operation and supplies the energy to in the refresh operation. Note that the SERC in [10] was designed with transmission gates. The clamp transistors, M13 and M14, make the undriven storage node stay at ground. The gate capacitances of M3, M4, M9, M10, M13 and M14 are the storage capacitance of the memory cell. All transistors in the storage cell are 0.36μm/0.24μm, which is minimum-sized, except for M3 and M4, which size are 0.92μm/0.24μm.

2.3 Clock Gating Method for a Single-Rail Signal

A clock gating signal should *nest* the clock signal not to have non-adiabatic loss. When a gating signal is enabled, it must stay high at least for three phases, as shown in Fig. 3, so that this signal can nest the clocked power. This clock gating method for a single-rail signal is used to activate only a selected part of a memory.

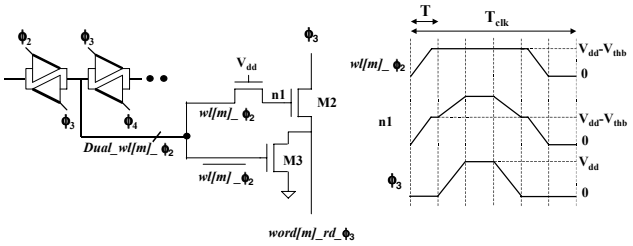

Fig. 3. Clock gating method for a single-rail signal

2.4 Control Signals and Operations

The control signals in the storage cell in Fig. 2 are clock gated signals with decoder output signals, as shown in Fig. 4. We use the naming conventions for control signal names like the followings: *rd* for read, *wr* for write, *ref* for refresh, and *unwr* for unwrite. The clock phase specified at the end of the name of each control signal stands for its evaluation phase. For example, *word[m]_wr_*ϕ_1 means that the signal is used for writing *m-th* word data, which is evaluated when $\phi_1$is rising.

Read Operation: *word[m]_rdi_*ϕ_2 and *word[m]_rd_*ϕ_3 are used in the read operation. Assume X_0 is high. When *word[m]_rd0_*ϕ_2 is rising, M5 and M6 are turned on to connect the cell to the read bit-lines without non-adiabatic loss because their both ends are in the *low*. Then, when *word[m]_rd_*ϕ_3 is rising, the storage node X_0 is boosted and X_0 and $\overline{X_0}$ are copied to n3 and n4, respectively. Later, when *word[m]_rd0_*ϕ_2 is falling, the cell is isolated from the read bit-line and when *word[m]_rd_*ϕ_3 is falling, the energy stored at the intermediate nodes Y_0 and $\overline{Y_0}$ is recovered, and the boosted X_0 is settled down.

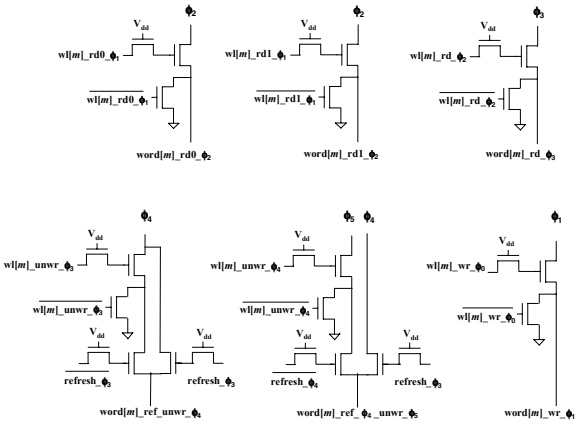

Fig. 4. Control signals to a memory cell

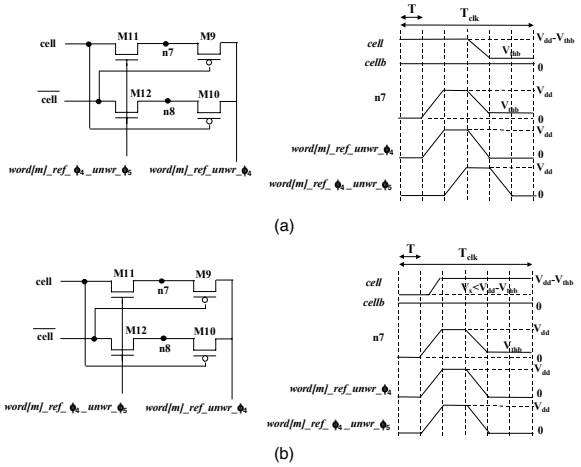

Fig. 5. Unwrite & refresh circuit and operation: (a) unwrite operation (b) refresh operation

Unwrite Operation: The *unwrite* operation must precede the write operation. The *unwrite* operation recovers the energy of the storage data with minimal energy dissipation and puts the memory cell in the *clear-state*. As shown in Fig. 5(a), First, *word[m]_ref_unwr_*ϕ_4 is rising so that n7 become V_{dd}. Then, *word[m]_ref_*ϕ_4*_unwr_*ϕ_5 turns on M11 and M12 without non-adiabatic loss. The energy of X_0 is recovered when *word[m]_ref_unwr_log_*ϕ_4 is falling. However, there exists remnant energy at one of the storage node, which will be dissipated with non-adiabatic loss during the next write operation.

Write Operation: The following write operation begins when *word[m]_wr_*ϕ_1 is rising, which turns on M1 and M2 to connect X_0 to the write bit-line. The write data is written to X_0 when ϕ_2 is rising. Then, when *word[m]_wr_*ϕ_1 is falling, X_0 is disconnected from the write bit-line. When ϕ_2 is falling, the energy of the write bit-line is recovered.

Refresh Operation: In nRERL storage array, the refresh operation is necessary because the storage cell in Fig. 2 is a dynamic one. Note that refresh operation is a *global* operation. All cells in a memory block are refreshed in a cycle simultaneously. *word[m]_ref_unwr_*ϕ_4 and *word[m]_ref_*ϕ_4*_unwr_*ϕ_5 are used in the refresh operation. As shown in Fig. 5(b), the control signals are rising to refresh degraded X_0^* and turns on M9, M11 and M12, and degraded X_0^* becomes 'V_{dd}-V_{thb}'. When the control signals are falling, M11 and M12 are off and refresh operation is ended.

3. Architecture

The block diagram of an nRERL register file is shown in Fig. 6. In a read operation, "read" signal is evaluated by ϕ_0, and its output data is evaluated by ϕ_4, and its latency of the read operation is 4 phases. In a write operation, "write" signal is evaluated by ϕ_2, its input data is written to the cell when ϕ_2 is rising in the next cycle, and its latency of the write operation is 6 phases because of an unwrite operation that is performed before a write operation.

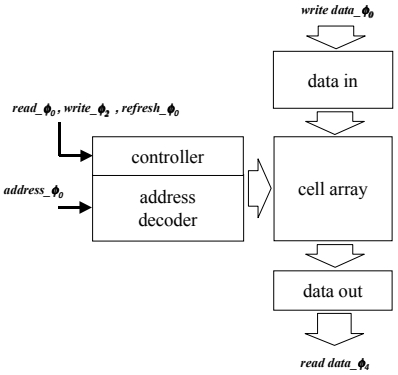

Fig. 6. Block diagram of an nRERL register file

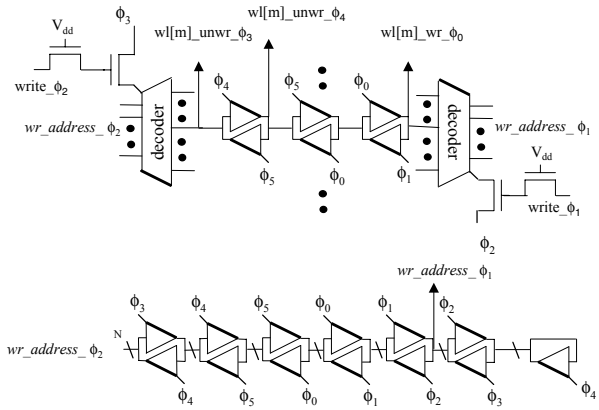

Fig. 7. Write address decoder in the nRERL register file, where N is the bit-width of the address. The buffer chain to make delayed copy *"wr_address_*ϕ_i*"* is needed to be reversible

For proper memory operation, the selection signals for each word-line in Fig. 4. must be generated separately. Fig. 7 shows the write address decoder, which is an N-to-2^N demultiplexer, where N is the bit-width of the address. To reduce the energy dissipation, the decoder is enabled only when its control is high. An additional buffer chain, which generates several delayed addresses, is necessary to recover the energy of the decoded signals. The energy of address is recovered at the end of its buffer chain with the SERC's after reducing the node capacitance of the delayed address. The read address decoder is similar to the write one.

The data in, data out and controller are simple, which mostly consists of buffers. Except for the nodes connected to the SERC's, the energy of any node is recycled without non-adiabatic loss in the nRERL register file. Especially the energy of the bit- and word-lines with large capacitance is recovered without any non-adiabatic loss.

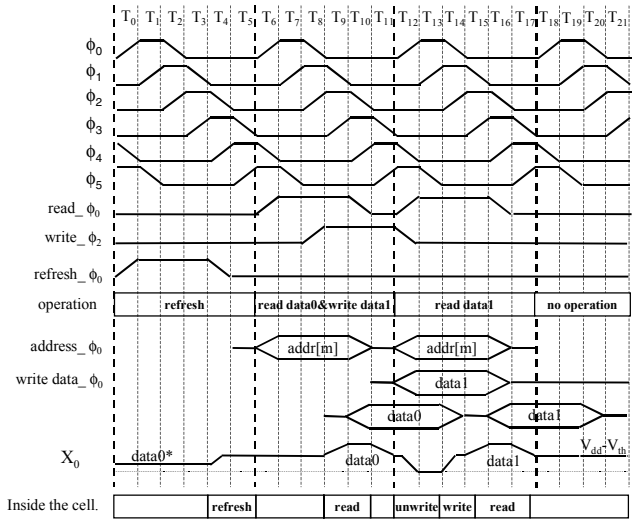

Fig. 8. Timings in the nRERL register file

The timings of write, read, and refresh operations are shown in Fig. 8. Each operation is pipelined and can be performed every cycle. Assume that a degraded data is initially in X_0 at the m-th

address. The operation sequence in the timing diagram is as follows: refresh the data, read it, write a new data, and read it again.

4. Hierarchical Design for a RAM Expansion

Generally, a RAM is a large array of the storage cells. In expanding an nRERL register file to a RAM, we have problems: the overhead of the large address decoder and the energy overhead due to the bit-lines with large capacitance. In this section, we propose to use a hierarchical design to solve these problems.

4.1 Two-Level Address Decoding

The area overhead of an address decoder is large when the bit-width of the address is large. This decoder overhead can be reduced substantially by using two-level decoding, because it can reduce the number of the buffers exponentially. In the two-level address decoding, the two decoded signals, *wl_high* and *wl_low*, from the two separated decoders can be combined, as shown in Fig. 9. If we divide the address decoder into more than two, the complexity of the decoder is further decreased with the additional cascaded gating circuit.

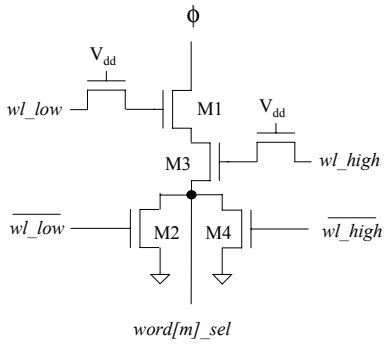

Fig. 9. Clock gating using 2-level address decoded signals

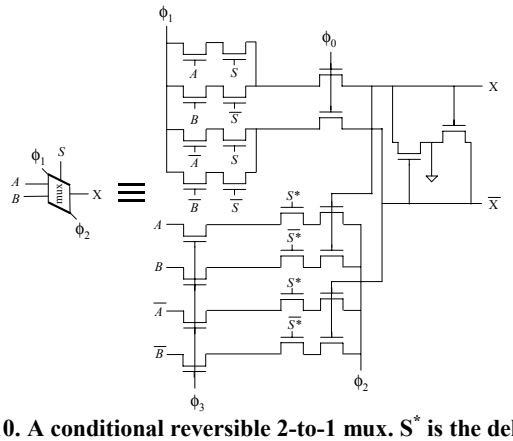

Fig. 10. A conditional reversible 2-to-1 mux. S^* is the delayed copy of the S

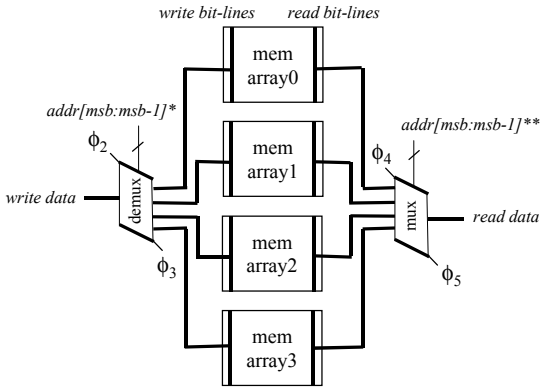

Fig. 11. Bit line separation using conditional reversible mux

4.2 Bit Line Separation Using a Conditional Reversible Mux

We also need to reduce the large capacitance of a bit line. We propose to divide a bit line into several lines and combine them with a *conditional reversible mux*. Generally, a mux is not reversible because it is not a one-to-one. However, it can be reversible if a condition is satisfied. For example, a conditional reversible 2-to-1 mux is shown in Fig. 10. This 2-to-1 mux requires a delayed copy S^* to make it reversible. If S is high, A is selected and its energy is recovered by ϕ_2, and B must be in the *clear state*. If S is low, B is selected and its energy is recovered by ϕ_2, and A must be in the *clear state*. The conditional reversible mux can be used to reduce the energy dissipation in a bit line as shown in Fig. 11. Each bit line is divided into 4, respectively. The 1-to-4 demux, which is the reverse of the 4-to-1 mux, selects only one out of four write bit lines and puts the others in the *clear state*. The 4-to-1 mux selects only one out of four read bit lines. This method saves the adiabatic loss substantially because the adiabatic loss of a node is proportional to the square of its capacitance.

5. Simulation Results

A 3-port 16×8b nRERL register file was designed, which used Anam 0.25-μm n-well 5-metal process. The area of a proposed register cell with 14 transistors was less than two times that of the conventional 3-port SRAM cell with 10 transistors. Each of the three address decoders was divided into two with two-level address decoding: one for the higher 2 bits and one for the lower 2 bits. In addition, each read and write bit line is divided into two.

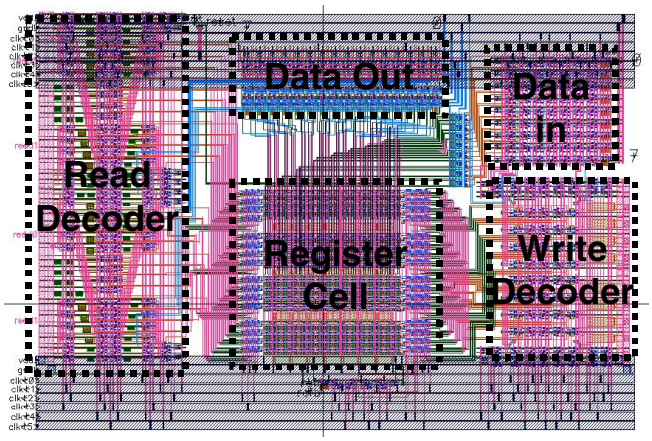

Fig. 12. Layout of the nRERL 3-port (2-read 1-write) 16×8-bit register file

The layout of the nRERL register file is shown in Fig. 12, which area is 500.1 μm × 331.8 μm. Its energy consumption per cycle is shown in Fig. 13, which was estimated with SPICE simulation in the condition, where two read and one write operations were performed simultaneously and the supply voltage is 2.5V. Its energy curve was similar to those of the nRERL logic circuits [1]. The energy consumption of the nRERL register file was separated into 7 components: two write address decoders, four read address decoders, controller, data-in, two write bit lines (include cell), four read bit lines, and data-out, which are shown in Fig. 13. The energy consumed in the decoders was the largest in the register file, even though it was designed by 2-level address decoding. Because the word line selection signals in each decoder are maintained for 4 phases and because the number of the decoders is six for 2R1W, the energy dissipation in the decoders is large.

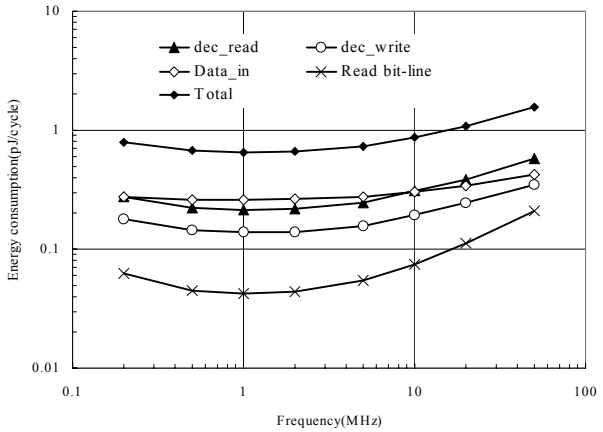

Fig. 13. Energy consumption of each part in the 3-port nRERL register file at V_{dd} = 2.5V

A three-port static CMOS register file was designed with a conventional 10-T SRAM cell and its power dissipation was compared with that of the nRERL register file as shown in Fig. 14. In the CMOS register file, we did not include the power consumption of the sense amplifiers and bit-line pull-up circuits because they are not required in the low-speed operation [6]. The results show that the nRERL register file consumes 6.6% of the power dissipated in the CMOS one at 1MHz, the energy-minimal frequency in Fig. 13.

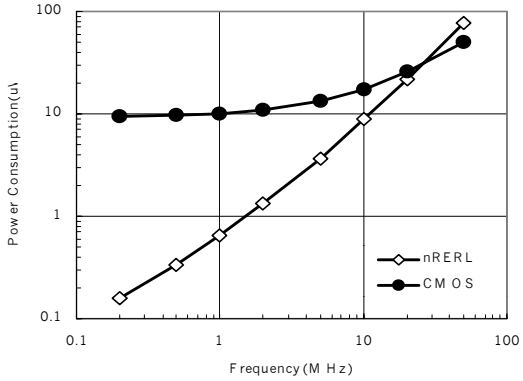

Fig. 14. Power consumption of the 3-port nRERL and static CMOS register files at V_{dd} = 2.5V. The power consumption in the sense amplifier and bit-line pull-up circuit is excluded in CMOS register file

Table I summarizes the data for energy consumption for several adiabatic memories, which data were collected from the published papers [3][4][5][6]. The adiabatic memories in [3], [4] and [5] are a hybrid design with the conventional CMOS SRAM cell. The adiabatic register file in [6] and our nRERL register file exploit fully adiabatic circuits.

Table I. Comparison of energy consumption for several adiabatic memories. The percentage in "Energy consumption" column is the energy consumption when it is compared with the corresponding conventional CMOS one.

	Energy consumption	Comment
RAM cell in [3]	16%	RAM core simulation
SRAM in [4]	24%	256×256b SRAM simulation
ECRL register file in[5]	28%	32×32b 2-port chip
Register file in [6]	10-15%	8×16b 3-port chip
nRERL register file	6.6%	16×8b 3-port simulation

6. Conclusions

We proposed an nRERL register file, which employs the reversible adiabatic logic, nRERL, and discards garbage information with minimal energy dissipation. A clock gating method for a single-rail signal was used to reduce the energy dissipation. We also recycle the energy of bit-line. From SPICE simulation, we found that an nRERL register file had substantial advantages in energy consumption at low operating frequencies. We also proposed to use a hierarchical design for an nRERL

RAM by using two-level decoding and the conditional reversible mux.

7. References

[1] J. Lim, D. -G. Kim, and S. -I. Chae, "nMOS reversible energy recovery logic for ultra-low-energy applications," *will be published in IEEE Journal of Solid-State Circuits, Jun. 2000.*

[2] W. C. Athas, L. "J". Svensson, J. G. Koller, N. Tzartzanis and E. Y. Chou, "Low-power digital systems based on adiabatic-switching principles," *IEEE Trans. VLSI Systems*, vol. 2, no. 4, pp. 398-407, Dec. 1994.

[3] D. Somasekhar, Y. Ye, and K. Roy, "An energy recovery static RAM core, " in *Proceeding of IEEE Symposium on Low Power Electronics*, pp. 62-63, 1995.

[4] N. Tzartzanis and W. C. Athas, "Energy recovery for the design of high-speed, low-power static RAMs," in *Proceeding of Int. Symposium on Low Power Electronics and Design*, pp. 55-60, 1996.

[5] Y. Moon and D. -K. Jeong, "A 32×32-bit adiabatic register file with supply clock generator, " *in Symposium on VLSI Circuits Dig. Tech. Papers*, pp. 27-28, 1997.

[6] S. Avery and M. Jabri, "A three-port adiabatic register file suitable for embedded applications, " in *Proceeding of Int. Symposium on Low Power Electronics and Design*, Monterey, CA, USA, pp. 288-292, 1998.

[7] R. Merkle, "Reversible electronic logic using switches, " *Nanotechnology*, vol 4, pp. 21-40, 1993.

[8] C. J. Vieri, "Pendulum: a reversible computer architecture, " M.S. thesis, Massachusetts Institute of Technology, June 1995.

[9] C. Vieri, M. J. Ammer, A. Wakefield, L. "J. " Svensson, W. C. Athas, and T. Knight, "Designing reversible memory," in *Proceeding of Unconventional Model of Computation*, pp. 386-405, 1998.

[10] J. Lim, D. -G. Kim and S. -I. Chae, "A 16-b carry-lookahead adder using reversible energy recovery logic for ultra-low-energy systems," *IEEE Journal of Solid-State Circuits*, vol. 34, no. 6, pp. 898-903, June 1999.

[11] J. Lim, K. Kwon and S. -I. Chae, "Reversible energy recovery logic circuit without non-adiabatic energy loss," *IEE Electronics Letters*, vol. 34, no. 4, pp. 344-346, Feb. 1998.

[12] J. Lim, D. -G. Kim and S. -I Chae, "Reversible energy recovery logic circuit and its 8-phase clocked power generator for ultra-low-power applications," *IEICE Trans. on Electronics*, vol. E82-C, no. 4, pp. 646-653, April 1999.

[13] J. Lim, D. -G. Kim and S. -I. Chae, "Reduction in energy consumption by bootstrapped nMOS switches in reversible adiabatic CMOS circuits," *IEE Proceedings: Circuits, Devices, and Systems,* vol.146, No. 6, Dec. 1999.