An Embedded 8-bit RISC Controller for Yield Enhancement of the 90-nm PRAM

Hyejung Kim[1], Kyomin Sohn[2], Jerald Yoo[1], and Hoi-Jun Yoo[1]
Semiconductor System Laboratory, Department of Electrical Engineering and Computer Science, KAIST, Daejeon, Korea[1]
Samsung Electronics, Hwasung, Korea[2]
seeseah@eeinfo.kaist.ac.kr

***Abstract*—An embedded 8b RISC for advanced memories is designed to control, analyze and optimize the memory timing and voltage parameters. The processor-based built-in-self-optimize (BISO) algorithm is proposed to enhance the memory yield. A test PRAM with the RISC is fabricated in 90nm, 3-metal diode-switch process. By applying BISO, the PRAM margin window increases by 221%. It operates at 100MHz and consumes 28.4mW at 1.0V supply voltage. The embedded RISC enables 100Mb/s/pin read/write throughputs to PRAM.**

I. Introduction

There have been many studies to integrate logics with memories such as IRAM (Intelligent RAM) [1] and PPRAM (Parallel Processing RAM) [2]. Most of the previous studies focused on the development high performance processors or specific memories. In this work, however, we would like to propose a new RISC architecture which is optimized to stabilize memory operation and to improve its yield [3]. And we applied the proposed RISC to enhance the yield of PRAM of the advanced technology.

The memories made by nanometer technology have difficulties in controlling internal device parameters because they are very sensitive to PVT variations. The proposed processor controls internal memory parameters individually to enhance its performance, stability and yield rather than implementation of other specific functions. Although there was a research using a BIST (Built-In Self-Test) block of simple FSM logic in order to test chip effectively [4], memory device in its wafer state can not be tested fully because of the I/O limits and operating frequency. In this paper, a new processor-based block-wise built-in-self-optimize (BISO) algorithm is presented to maximize memory performance and enhance its yield with an embedded RISC.

The PRAM (phase change random access memory) is a strong candidate for next-generation memory due to its many good features such as small cell size, fast read speed, and non-volatility [5]. However, the PRAM has many uncertain variations to be compensated, and it is a good test vehicle for application of the proposed RISC. In addition, cache-like PRAM operation can be implemented with the help of the proposed RISC to obtain 100Mb/s/pin read and write throughputs. We verify successful operation of the proposed RISC by fabrication of real silicon.

II. An Embedded 8-bit RISC Architecture

A. Architecutre of 8-Bit RISC

The figure 1 shows the block diagram of the proposed RISC. The 8-bit is adopted to implement the RISC for its small area overhead and offers general purpose functions.

The RISC has 3 pipeline stages, which are fetch, decode and execution. Pipelining is employed so that all parts of the processing and memory system can operate separately and continuously. The general instructions of RISC output the operation result in 3 stages, but some special instructions require multi cycle. The fetch block fetches the instructions from the code memory in the first stage. The control block decodes the fetched instructions to execute in the second stage. The last stage executes ALU operations, memory access, and write-back to the register file. Since the both operations of read and write the register file occurs in the same stage, the data hazard can be eliminated. The branch is performed with 2-cycle penalty. The RISC exchanges the data between SRAM data memory and PRAM.

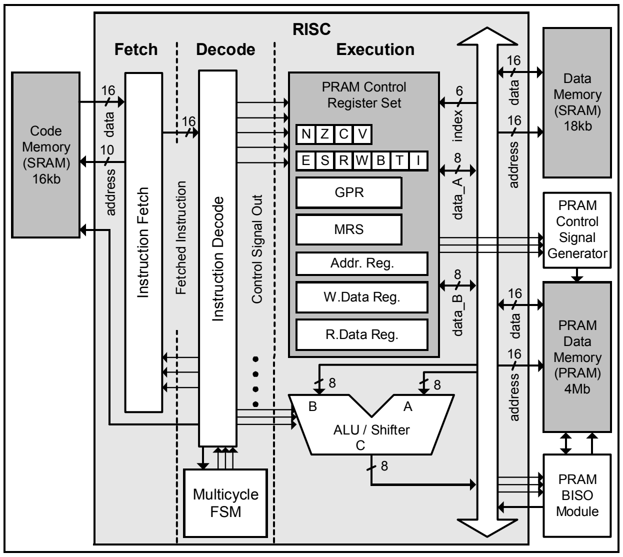


Figure 1. The Block Diagram of Embedded 8-bit RISC for PRAM

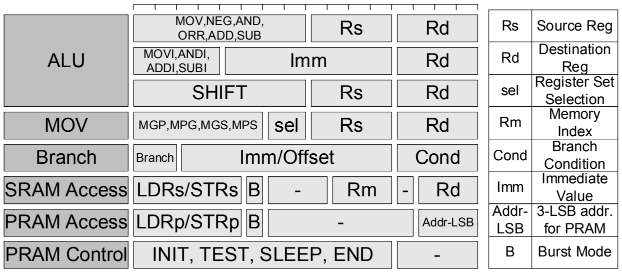
1-4244-1623-X/07/$25.00 ©2007 IEEE

Only small control logic is required for the proposed RISC through the well-defined compact 16-bit instruction encoding. The ISA encoding is summarized in figure 2. The ISA consists of 6 major categories. Special instructions are prepared for the burst memory access operation and BISO operation for PRAM. The MOV instructions transfer data efficiently between registers. The data load/store instructions are separated into two parts according to the destinations, internal SRAM or PRAM. The burst mode provides 128b data exchange at a time. The PRAM control instructions execute only the PRAM operations such as initialization, BISO, sleep and end operations. The 4-bit condition field (C/N/Z/V) determines the circumstances under which branch is to be executed.

16-kb and 18-kb internal SRAM are used for code memory and data memory, respectively. These SRAM blocks operate like a cache in order to enhance the throughput of the PRAM.

B. Memory Control Register Sets

The figure 3 shows the memory control register set. It consists of PRAM mode register set, address and data register for PRAM access operation, PRAM and RISC status registers. Only mode register (MRS0) are set by external input, the other registers are controlled by RISC or internal control block. Unlike the previous SRAM case, next generation RAM such as PRAM or MRAM core needs more various control signals and voltage sources. Additionally, basic information is not fixed yet because the cell type used in this work is wholly new. The 6 kinds of the signals are controlled, which are discharge, pre-charge, bias level, sense-amplifier enable, and read enable signals for read operation, reset, set, and write enable signals for write operation. They are stored in MRS part.



ALU	MOV,NEG,AND, ORR,ADD,SUB	Rs	Rd			
	MOVI,ANDI, ADDI,SUBI	Imm	Rd			
	SHIFT	Rs	Rd			
MOV	MGP,MPG,MGS,MPS	sel	Rs	Rd		
Branch	Branch	Imm/Offset	Cond			
SRAM Access	LDRs/STRs	B	-	Rm	-	Rd
PRAM Access	LDRp/STRp	B	-	Addr-LSB		
PRAM Control	INIT, TEST, SLEEP, END	-				

Rs	Source Reg
Rd	Destination Reg
sel	Register Set Selection
Rm	Memory Index
Cond	Branch Condition
Imm	Immediate Value
Addr-LSB	3-LSB addr. for PRAM
B	Burst Mode

Figure 2. The Proposed RISC ISA Organization

Name	MSB 8b	LSB 8b
GPR0 - GPR15	-	GPR
MRS0-MRS9	-	PRAM MRS
SR	-	RISC Status
PSR	-	PRAM Status
PWADR	PRAM Write Addr Reg.	
PRADR	PRAM Read Addr Reg.	
PWR0-PWR7	PRAM Write Data Reg.	
PRR0-PRR7	PRAM Read Data Reg.	

Mode	**MRS0**	Mode_select : RD | WR | WR_back | Program (4b)
Test Mode	**MRS1**	Fail bit Count (16b)
	MRS2	
	MRS3	Incr_Flag | Data_Flip | Addr_Incr_Step (8b) | Data_Pattern (8b)
	MRS4	
	MRS5	
Memory Control Siganl	**MRS6**	Write IO (2b) | Delay_PBLDIS_PBIAS (3b) | Delay_PBIAS_PCHG (3b)
	MRS7	Delay_PCHG_PSA (5b)
	MRS8	Pulse_Width_PRESET (7b)
	MRS9	Pulse_Width_PSET (8b)

Status	**SR**	-	-	-	-	N	Z	C	V
Status	**PSR**	-	E	S	R	W	B	T	I

Figure 3. PRAM Control Register Set

In conventional memory, the control signals are generated by pre-defined delay paths and selected by mux. In this work, all memory control signals are saved in memory control register set digitally. The control signals are represented by clock periods for the flexibility of its application as shown in figure 4. A basic clock period is 10ns, and all timing margins and the pulse widths of control signals can be adjusted by programming MRS with new data. If there is severe clock difference, new values adjusted timing discrepancy can be programmed at MRS.

The 19-bit address and 128-bit data are stored in register set for PRAM read/write access operations. The PRAM status register represented for END, SLEEP, READ, WRITE, BURST, TEST, INIT. When a selected flag turns high, the operation indicated by the flag starts. After the reserved operation finishes, the END flag goes high so that all system can realize the ending. After all the arranged program codes are processed, the RISC sends a SLEEP flag. When receiving the SLEEP flag, the system gates off the RISC clock and the RISC returns to its sleep mode.

C. DataTtransferring Between GPR and SRAM

The GPR (General Purpose Register) and the SRAM data memory have 8-bit and 16-bit widths, respectively. Therefore, data width matching between GPR and SRAM is necessary to transfer data. In order to store/load data, four registers are necessary, 2 for SRAM access address and 2 for write/read data as shown in figure 5. The programmer specifies the only one register among register of R0-R7 for the address and data. To execute store/load instructions, the data should be read from both the specified registers of R0-R7 and the paired register of R8-R15. The SRAM access operations need 2-cycle execution, one cycle for the address data transfer and the other cycle for the write/read data and SRAM enable signal transfer.

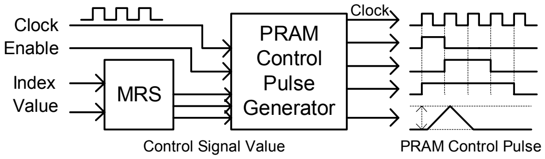


Figure 4. PRAM Control Signal Generation

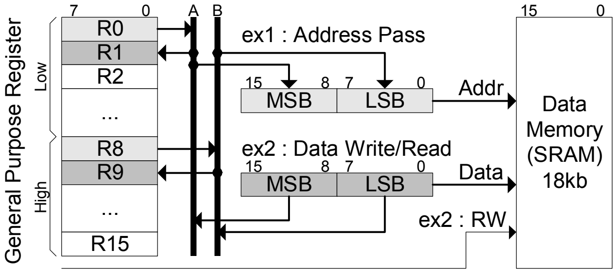


Figure 5. Data Transferring Between GPR and Memory

III. APPLICATION TO PRAM

The proposed RISC is applied to control PRAM by embedding it. The 4Mb PRAM core, 16kb code memory, 18kb data memory, PRAM BISO module, and peripheral control logics are connected to the RISC. The whole system is called uRAMP (Micro RAMP Processor). The uRAMP offers 4 operation modes: (1) PRAM single access mode, (2) PRAM burst access mode, (3) SRAM and register set access mode, and (4) RISC mode. Mode (2) and (4) require RISC operation and RISC is turned off by its clock gating in other modes. The mode (1) operation is like a synchronous SRAM with long latency. The typical latencies are 120ns and 3us for read and write operations, respectively. In mode (2), the PRAM operates in burst read/write mode by using RISC and cache-like SRAM blocks. Mode (3) is for setup and test by accessing the embedded SRAM and the MRS. Mode (4) is for parameter optimization (BISO) and other user program stored in the code memory. In this chapter, only mode (2) : PRAM burst access mode and mode (4) : BISO mode are explained in detail.

A. PRAM Burst Read/Write Access Mode (Mode 2)

The conventional PRAM takes hundreds to thousands nanoseconds to write data to the cell because it requires large current of over 1mA and a slow quenching is necessary in SET write [3]. In this work, not only burst read operation, but also burst write operation is accomplished by using RISC and SRAM blocks. It adopts the direct-mapped cache-like architecture. Among 18kb data memory, 2kb space is used for tag area and 16kb space is used for data storage. The figure 6 shows the flow chart of PRAM burst access mode. This scheme enables 100Mb/s/pin read/write throughputs. Write back operation occurs in idling period by the external request or the detection of long idling period.

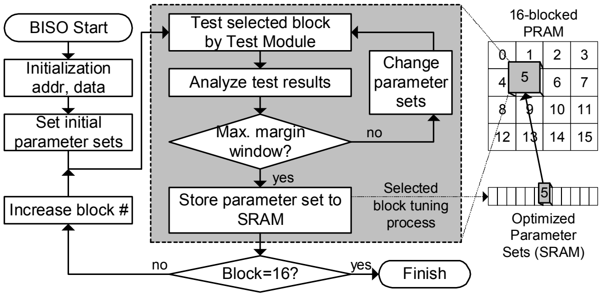
B. Built-In-Self-Optimize (BISO) Mode (Mode 4)

The BISO module provides block-wise parameter optimization. It writes data to PRAM core according to pre-defined conditions such as address sequence, a data flip flag, address increment/decrement flag, and data pattern. The conditions can be changed easily by simple RISC programming.

Figure 7 shows the flow chart of BISO operation. At first, the initial test condition sets such as various timings and voltage levels. The block-wise test is executed by the BISO test module. The cell resistance is tested by the current level. After the test of the each block, the fail bit counts is reported, and the RISC analyzes the test results to determine a specific set of tuning parameters. If the result is over the limit of specification, the test condition is adjusted and the test module is enabled again with the new test condition. After the optimal condition is determined for all blocks, they are stored in the data memory. The MRS is updated by the relevant data whenever new addresses are entered.

C. Code Density Analysis

The figure 8 shows the code density analysis of the proposed RISC. The code consists of initialization, burst read/write, write back and BISO operations. It is divided into four major categories of ALU/MOV/Branch, PRAM control, SRAM access, PRAM access. The total code size is 3.9kb, and there is enough room to store additional program code. An analysis of the total code percentage reveals that 58.5% ALU/MOV/branch, 14.9% PRAM control, 17.3% SRAM access, and 9.3% PRAM access instructions.



BISO Start
Initialization addr, data
Set initial parameter sets
Increase block #
Test selected block by Test Module
Analyze test results
Change parameter sets
Max. margin window?
no
yes
Store parameter set to SRAM
Selected block tuning process
Block=16?
no
yes
Finish
16-blocked PRAM
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
5
Optimized Parameter Sets (SRAM)

Figure 7. The Flow Chart of Block-wise BISO Operation

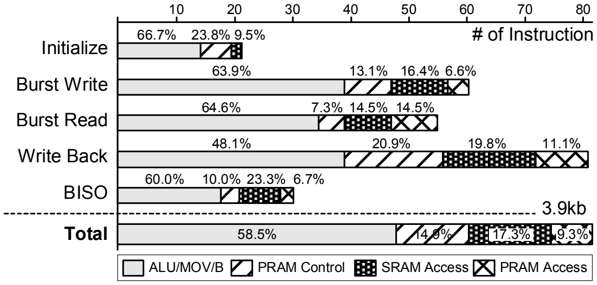


Figure 8. The Code Density Analysis

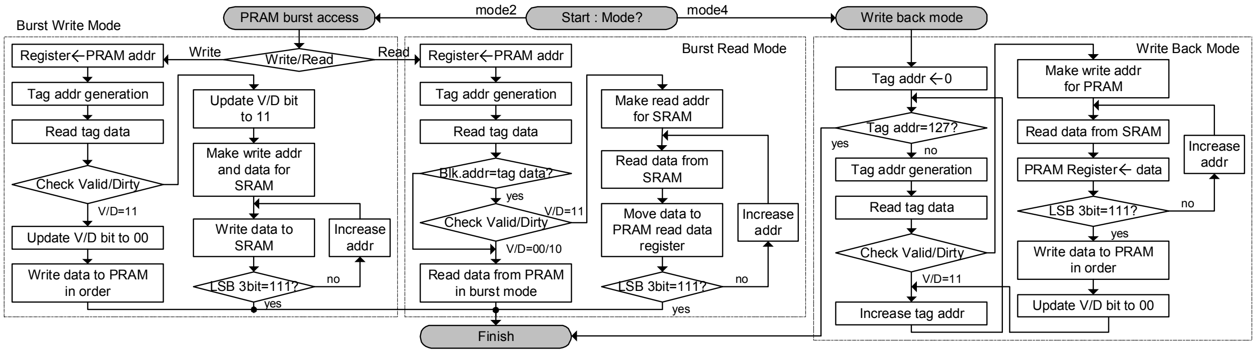


Figure 6. The Flow Chart of PRAM Burst Access Mode

IV. Implementation Results

The figure 9 shows the chip photograph and the block diagram of the uRAMP. It is fabricated in a 90nm CMOS process with 3 metals and diode-switch PRAM cell process. The PRAM core, SRAM blocks, register files, and I/O circuits are designed by full-custom design method. The RISC, BISO module and peripheral control logic are designed by synthesis flow. The chip size is about 6mm^2. The area of RISC and SRAM blocks are about 0.35 mm^2 and 0.70mm^2, respectively. Its area is under 1% of high density memory if 256Mb PRAM is integrated. The power consumption of the uRAMP is 21.6mW and 28.4mW in single access and burst access at 100MHz and 1.8V, respectively. Table-I summarizes the performance of the 8-bit RISC controller.

4Mb PRAM core is logically divided into 16 of 256kb cell blocks. Test block size is reconfigurable by RISC programming. Test chips with small margin window are selected to validate the effectiveness of the embedded RISC. The figure 10 shows the measurement results of set and reset resistance distributions. The graph (a) is the resistance distributions per 512kb without BISO optimization. Based on the analysis of the data, a software routine in the BISO is developed to generate the optimal parameters which improve the distributions of all blocks as shown in graph (b). The margin window increases by up to 221% by using BISO [3]. The resistance distribution of PRAM can be controlled and higher yield can be obtained without any process optimization or circuit tweaking.

The table-II shows the comparison results of the related micro-controller such as Atmel Mega 128L [7] and ARM7 Thumb [8]. These estimations are the results of running uRAMP specific program. The number of instruction of the initialization is not different since it mainly consists of ALU operations. On the other hand, if the burst access and write-back are executed with numerous memory accesses, the proposed RISC shows the least number of instructions. Moreover, the proposed RISC has small code size. Although ARM has also small number of instructions, the code size is larger since its long instruction and datapath length. The comparison result shows that the proposed RISC is optimized for memory control and optimal memory operation including yield enhancement of PRAM.

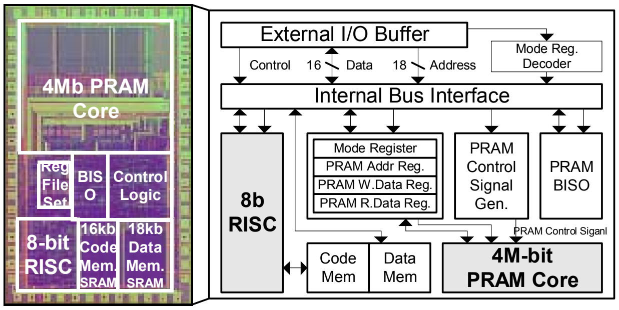


Figure 9. Block Diagram and Chip Phtograph of the uRAMP

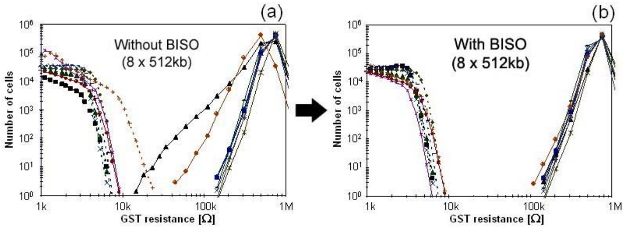


Figure 10. The Resistance Distributions without and with BISO scheme

TABLE I. A Summary table of the RISC

Process technology	Samsun 90nm PRAM technology
Supply voltage	1.8V
Clock frequency	100MHz
Gate counts	5k
Area	6mm^2(Chip) / 0.35mm^2(RISC) /0.7mm^2(SRAM)
Power consumption	21.6mW (single access) / 28.4mW (burst access)

TABLE II. Comparison Result with Related Microcontroller

	Atmel Mega 128L [7]	**ARM7 Thumb [8]**	**This Work**
Inst./data length (bit)	8 / 8	16 / 32	16 / 16
Norm (# of Inst.)	1.49 (379)	1.13 (279)	**1.0 (248)**
Initialization	1.19 (25)	1.04 (22)	1.0 (21)
Burst Write	1.72 (91)	1.15 (69)	1.0 (61)
Write Back	1.47 (139)	1.17 (93)	1.0 (81)
Total Code Size	3.0k	4.5k	3.9k

V. Conclusion

An 8-bit RISC architecture is proposed for stable and efficient operation of the advanced memory. Two major schemes of the cache-like burst access mode and BISO (Built-In-Self-Optimize) mode are proposed. The proposed 8-bit RISC is integrated with a test 4Mb PRAM to verify effectiveness of the proposed RISC. A 90-nm PRAM of this work requires more careful control and optimization due to process uncertainties. By adopting the proposed RISC, the distributions of GST resistances of PRAM core are improved remarkably. Moreover, the burst PRAM access mode provides performance of 100 Mbps/s/pin throughput.

References

[1] D.Patterson, et al., “Interlligent RAM (IRAM): chips that remember and compute,” ISSCC Dig. Tech. Papers, pp.224-225. Feb. 1997

[2] K.Murakami, et al. “Parallel processing RAM chip with 256Mb DRAM and Quad processors,” ISSCC Dig. Tech. Papers, pp.228-229, Feb.1997

[3] Kyomin Sohn, Hyejung Kim, Jerald Yoo, Jeong-Ho Woo, Seung-Jin Lee, Hoi-Jun Yoo et al., “Processor-Based Built-in Self-Optimizer for 90nm Diode-Switch PRAM,” Symp. VLSI Circuit, June 2007 (Accepted)

[4] J.Dreibelbis, et al., “Processor-based built-in self-test embedded DRAM,” IEEE Journal of Solid-State Circuits, vol.33, no.11, pp.1731-1740, Nov.1998

[5] Y.Shin, “Non-volatile memory techinologies for beond 2010,” Symp. VLSI Tech, pp.156-159, June, 2006

[6] C.Tseng, et al., “A 72Mb separate-I/O synchronous SRAM chip with 504Gb/s data bandwidth,” ISSCC Dig.Tech. Papers, pp.626-627, Feb.2006

[7] Atmel ATMega128L AVR Microcontroller Datasheet. Available: http://www.atmel.com

[8] ARM 7TDMI Data Sheet. Available:http://www.arm.com