PROJECT REPORT

32K-BIT SLEEPY SRAM

VLSI-I PROJECT DOCUMENT

JOON-SUNG YANG / GAHNGSOO MOON

DEC. 9, 2005

TABLE OF CONTENTS

I. ABSTRACT

The most research on the power consumption of circuits has been concentrated on the switching power and the power dissipated by the leakage current has been relatively minor area. However, in the current VLSI process, the sub-threshold current becomes the one of the major factors of the power consumption, especially in high-end memory. To reduce the leakage power in the SRAM, the power gating method can be applied and a major technique of the power gating is using sleep transistors to control the sub-threshold current. In this project, dual threshold voltages are adopted; normal SRAM cells have lower threshold voltages and THE higher threshold voltages control the sleep transistors. The size of sleep transistors can be chosen by the worst case current and are applied to every block.

For this project, we extend our discussion and present the result on the advantages of using sleep transistor in terms of delay, area and power reduction. The simulation of sleepy 32K-bit SRAM in tsmc 20 μm process, showed 47% of power saving without getting worst-case delay increased.

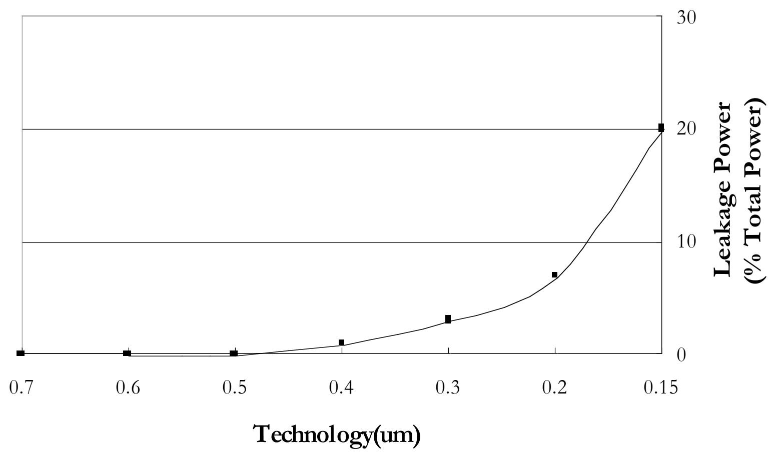
Index Terms: SRAM, sub-threshold current, leakage power, sleep transistor, delay, power saving

II. INTRODUCTION

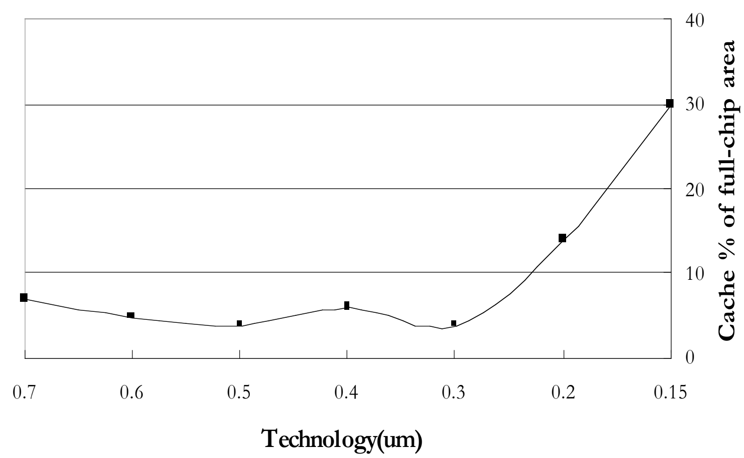
Complementary metal-oxide semiconductor (CMOS) technology development brings the performance enhancement and new challenges in VLSI circuit design such as process variation and increasing transistor leakage. The leakage current expressed as

$$I_{leakage} = I_0 e^{(V_{gs} - V_{th})/nV_T}$$
$$where\ I_0 = \mu_0 C_{ox}(W/L)V_T^2 e^{1.8},$$

takes more and more proportion in modern VLSI process as semiconductor devices are getting smaller and smaller. The following figures show the trend of the leakage power in terms of fabrication process. High-performance VLSI design is steadily required with the development of CMOS technology.



[Figure II-1] Trend of Leakage Power vs. Technology



[Figure II-2] Trend of Area Percentage vs. Technology

The demand for static random-access memory (SRAM) is increasing with large use of SRAM in mobile products, System On-Chip (SoC) and high-performance VLSI circuits. As the density of SRAM increases, the leakage power has become a significant component in chip design. A various methods have been adopted to reduce the leakage power. In this project, multi-threshold voltage is applied to construct sleep transistors that has higher threshold voltage. However, those multi-threshold voltages must reflect the characteristic of SRAM. That is, memory is generally a huge cluster of cells so the performance and cost may depend on clustering for higher threshold voltage or overall layout. Additionally, the analysis should include about the wire model and transistor sizing as well.

III. SPECIFICATION

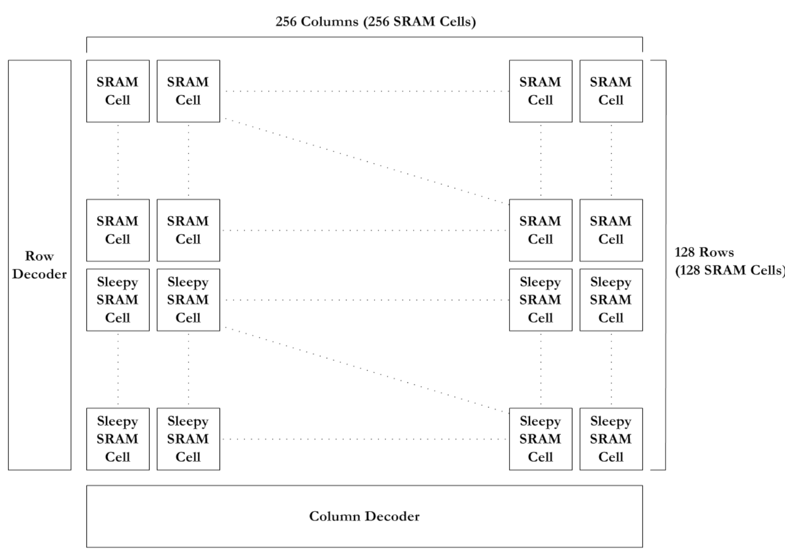
First, the sort of SRAM is to be determined. SRAM is roughly divided into two groups, sense amp SRAM and normal SRAM without sense amp. Sense Amp using SRAM is better for small signal handling and it is true that this kind SRAM has advantages over normal one. But a disadvantage is sense amp using SRAM takes difficulty in handling threshold voltages. So in this project, normal 6T SRAM is to be used as the main area we are interested in is the leakage power reduction using multi-threshold voltages.

There are many factors for 32K-bit SRAM, but this project will focus on the major parameters that can directly affect the indices we are interested in. Key parameters are listed as,

Parameters	Values
Supply Voltage	3.3V
nMOS Threshold Voltage	$V_{t,HI}$ = 0.5V, $V_{t,LO}$ = 0.38V
pMOS Threshold Voltage	$V_{t,HI}$ = -0.5V, $V_{t,LO}$ = -0.38V

[Table III-1] Major Controllable Parameters

Of course, transistor sizing is one of critical factors and accordingly carrier mobility must be taken into consideration. But we assume that parameter as uncontrollable, and accept it. The ratio between the mobility of n-type and p-type transistors is, in this project, 2.37; electron mobility, $\mu_n \approx 275 cm^2/(V \cdot s)$, hole mobility, $\mu_p \approx 116 cm^2/(V \cdot s)$.



[Figure III-1] Overview of 32K-bit Sleepy SRAM

Another fact that must be considered is that memory is quite slower compared with a processor unit, and because memory is a sort of size critical devices, the overall area should be limited at a proper level.

This is why sleep transistors are applied partially not to the whole system. The target values for 32K-bit SRAM are arranged below.

Gain/Overhead	Target
Power Reduction	40~50%
Area Overhead	Leakage Control Transistor 10~15%
Delay Overhead	Worst-case Delay 0% Increased
	Best-case Delay 20% Increased

[Table III-2] Target Values for 32K-bit Sleepy SRAM

Delay overhead might mislead that the overall delay is increased 20%. But the delay here separates the best and worst cases, so the maximum latency remains the same; the fastest latency before might not be kept. And considering the performance is generally determined by the worst-case delay, the targeted value can be interpreted as zero delay increased with large leakage power reduction.

And the transistor models and tools for the design, implementation and testing is,

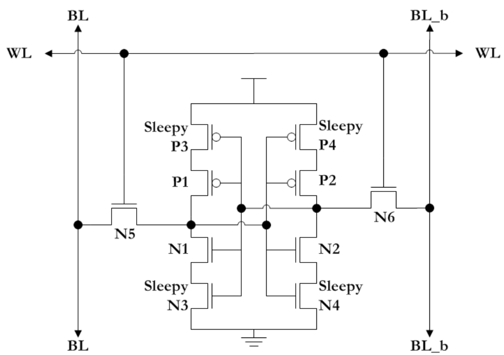
Tool/Simulator	Cadence/Hspice/Verilog-XL
Technology	0.20μm
Transistor Model	tsmc20N, tsmc20P

[Table III-3] Tools & Models for 32K-bit Sleepy SRAM

IV. DESIGN

A. TRANSISTOR SIZING

Transistor sizing for SRAM can be approached in two ways. One is the basic 6T transistor sizing. For the function of SRAM cell, read & write stability needs to be guaranteed. In read stability, N1 transistor is required to be much larger than N5 transistor to make sure that node between N1 and N5 transistors must not flip. When in write mode, bit lines (BL or BL_b) overpower cell with new value. However, high bit lines must not overpower inverters during read operation. That results in the determination of sizing P3 transistor weaker than N5 transistor.



[Figure IV-1] Diagram of Sleepy SRAM Cell

Transistor	W/L
N1	600nm/200nm
N2	600nm/200nm
N3	200nm/200nm
N4	200nm/200nm
N5	300nm/200nm
N6	300nm/200nm
P1	300nm/200nm
P2	300nm/200nm
P3	200nm/200nm
P4	200nm/200nm

[Table IV-1] Transistor Sizing of Sleepy SRAM Cell

The sleep transistors for pull-up and pull-down network are used to 6T SRAM cell for the purpose of reducing the leakage current. Once the 6T SRAM sizing is determined, we are able to start to size the sleep transistors in heuristic way. In sizing sleep transistors, we need to approach with the following mathematical equations that state SRAM performance with existence of sleep transistors and leakage current. For n-type MOSFET, when the sleep transistor is used, delay is increased with V_X, the voltage at the node between N1 & N3.

For n-type MOSFET, N1 should be in saturation mode when conducting the maximum current.

$$\tau_d \propto \frac{C_L V_{DD}}{(V_{DD} - V_{tL,n})^{\alpha}}$$

$$\tau_d^{sleep} \propto \frac{C_L V_{DD}}{(V_{DD} - V_{X,n} - V_{tL,n})^{\alpha}}$$

Suppose Δp the rate of tolerance for the delay penalty, then

$$\frac{\tau_{d,n}}{\tau_{d,n}^{sleep}} = 1 - \Delta_{p,n}$$

And setting the scaling factor, α = 1 gives,

$$V_{X,n} = \Delta_{p,n}(V_{DD} - V_{tL,n})$$

So the amount of current flowing through the linearly operating sleep transistor calculated as,

$$I_{sleep,n} = \mu_n C_{ox}\left(\frac{W}{L}\right)_{sleep,n}\left((V_{DD} - V_{tH,n}) \cdot V_{X,n} - \frac{V_{X,n}^2}{2}\right)$$

By the similar fashion, the leakage current through p-type sleep transistor is found as,

$$\tau_{d,p} \propto \frac{C_L V_{DD}}{(-V_{DD} + |V_{tL,p}|)^{\alpha}}$$

$$\tau_d^{sleep} \propto \frac{C_L V_{DD}}{(-V_{X,p} + |V_{tL,p}|)^{\alpha}}$$

$$(V_{DD} - V_{X,p}) = \Delta_p (V_{DD} - |V_{tL,p}|)$$

$$I_{sleep,p} = \mu_p C_{ox} \left(\frac{W}{L}\right)_{sleep,p} \left((-V_{DD} + |V_{tH,p}|) \cdot (V_{X,p} - V_{DD}) - \frac{(V_{X,p} - V_{DD})^2}{2} \right)$$

The arranged sizing data for n-type sleep transistor follow as,

Type	$\Delta_{penalty}$ Rate	$(W/L)_{sleep}$	$I_{calculated}$	$I_{measured}$	V_X
nMOS	0.197	0.50	6.394E-05 A	6.403E-05 A	5.752E-01 V
	0.130	**1.00**	**9.181E-05 A**	**9.206E-05 A**	**3.796E-01 V**
	0.100	1.50	1.098E-04 A	1.100E-04 A	2.920E-01 V
	0.081	2.00	1.212E-04 A	1.231E-04 A	2.365E-01 V
	0.063	3.00	1.443E-04 A	1.420E-04 A	1.840E-01 V
	0.050	4.00	1.549E-04 A	1.530E-04 A	1.460E-01 V
	0.042	5.00	1.641E-04 A	1.612E-04 A	1.226E-01 V

[Table IV-2] Transistor Sizing Data for n-type Sleep Transistor

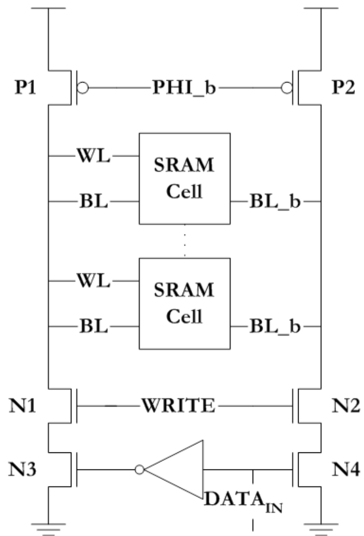
And for the p-type sleep transistor,

Type	$\Delta_{penalty}$ Rate	$(W/L)_{sleep}$	$I_{calculated}$	$I_{measured}$	V_X
pMOS	0.171	0.50	1.356E-04 A	1.337E-04 A	4.993E-01 V
	0.123	**1.00**	**2.070E-04 A**	**2.079E-04 A**	**3.592E-01 V**
	0.089	1.50	2.338E-04 A	2.316E-04 A	2.599E-01 V
	0.070	2.00	2.506E-04 A	2.513E-04 A	2.044E-01 V
	0.051	3.00	2.797E-04 A	2.766E-04 A	1.489E-01 V
	0.039	4.00	2.889E-04 A	2.919E-04 A	1.139E-01 V
	0.033	5.00	3.076E-04 A	3.002E-04 A	9.636E-02 V

[Table IV-3] Transistor Sizing Data for p-type Sleep Transistor

For both n-type & p-type, sizing was selected to be (W/L) = 1, because memory is a size critical devices and only SRAM cell capable of tolerating up to 50% delay penalty will have sleep transistors. In other words, all the sizing listed in the above tables do not increase the worst-case delay so once delay requirement is met, then transistor size should meet the other key requirement such as area load.

The last one is sizing for the peripheral transistors of SRAM. Basically the operation of SRAM is pre-charging and evaluating, and reminding each of bit line has large capacitances, discharging transistors should be large enough to evaluate the signal fast. And pre-charge transistors should be weak in order that writing function operates efficiently.



[Figure IV-2] Conceptual Diagram of SRAM Column

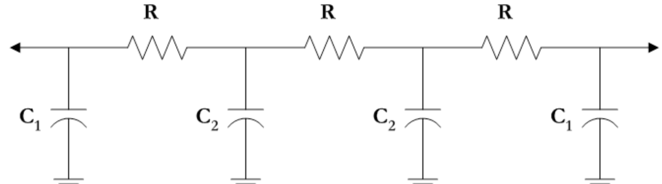
Transistor	W/L
N1	400nm/200nm
N2	400nm/200nm
N3	400nm/200nm
N4	400nm/200nm
P1	300nm/200nm
P2	300nm/200nm

[Table IV-4] Sizing of SRAM Peripheral Transistor

B. WIRE MODEL

Generally, memory is an array of huge number, which in turn means word line and bit line confront a large wire load. So it is necessary to include the proper wire model into simulation. Moreover, it is nearly impossible to simulate the whole 32K-bit SRAM Cell; hence the cells on the critical path are sampled and simulated. This limitation requires the wire model should include not only the resistance and capacitance of the wire itself but also the gate and junction capacitance connected to the wire. Starting with the area of the cell, the area of SRAM cell is 26×45λ.

The above area is based on the 6T transistor SRAM cell design – a sleepy SRAM cell consists of 10 transistors – but because the estimated value is a conservative and non-optimized, so there should be no significant size increase of SRAM cell array. Therefore we are able to apply this value to estimate the whole SRAM array. 3-segment Pi model was adopted as the 3-segment Pi model estimates the wire characteristics within 3% error and to get the accurate result, each capacitance includes the gate for word lines and junction capacitance for bit lines.



[Figure IV-3] Diagram of 3-Segment Pi Wire Model

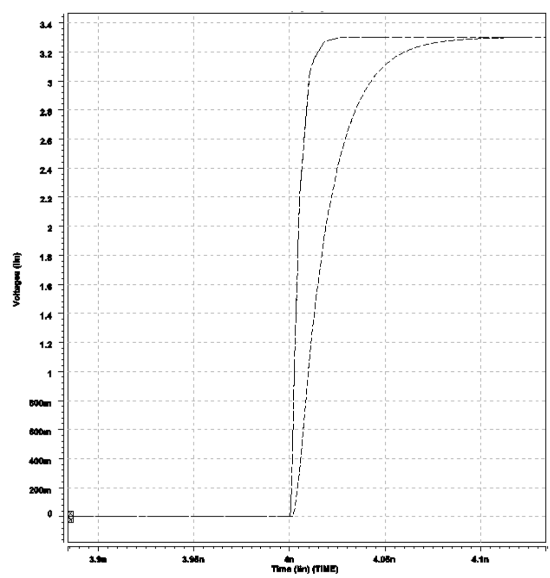
Type	R	C1	C2
Word Line	122 mΩ	0.278 fF	0.556 fF
Bit Line	211 mΩ	0.235 fF	0.470 fF

[Table IV-5] Numerical Values for 3-Segment Pi Wire Model

One of the important reasons for the wire models is that wire delay determines overall layout. In other words, sleep transistor can be placed only in the cell able to tolerate the load along sleep transistor. The result of wire simulation deserves to be recognized. Table below shows roughly double delay along bit lines, but there is not critical difference along word lines. This is sort of surprising but makes sense. As each transistor gating word line needs $V_{t,n}$ not $V_{DD}/2$, Word line delay take slight charge on the delay.

Delay	127 Word Lines	255 Word Lines
128 Bit Lines	9.436E-10 sec	9.597E-10 sec
64 Bit Lines	5.061E-10 sec	5.222E-10 sec

[Table IV-6] Wire Delay simulation for Critical Positions

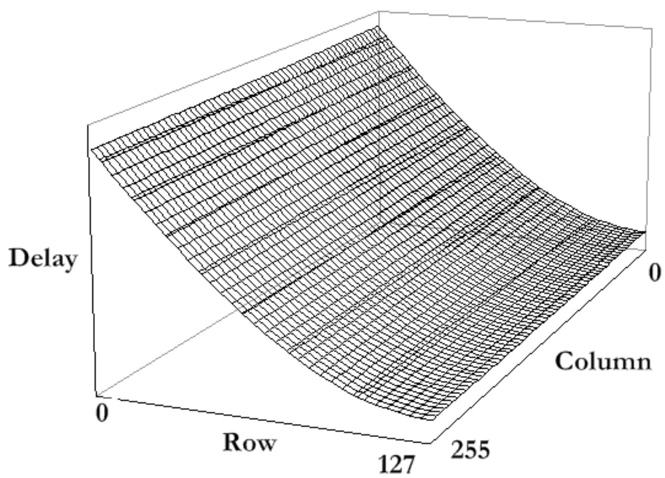


[Figure IV-4] Wire Model Simulation Waveform

C. Clustering & Delay Distribution

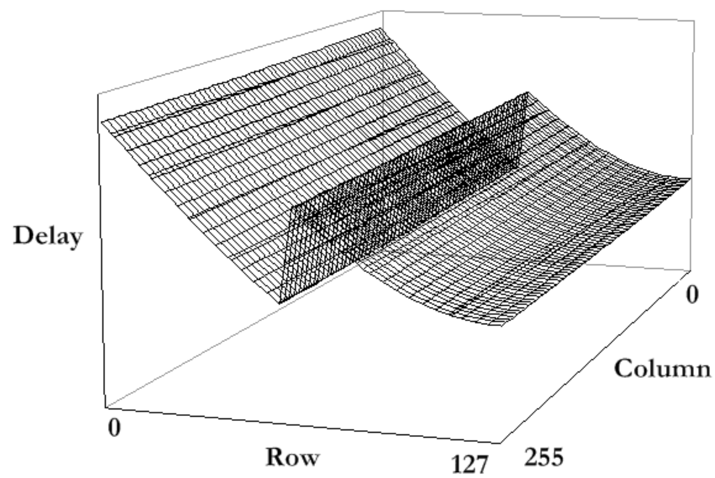
Leakage power reduction using multi-threshold voltages shows different spectrum depending on the clustering size. Generally known is, global block severely count on the input vector and but it has reduced area overhead. Meanwhile, local block has input-independent delay overhead but quite large area overload. So mostly hybrid technique is applied, which means installing sleep transistor by block. However that hybrid technique requires a logically homogeneous block and for this project, each SRAM cell is logically and perfectly independent from each other. Therefore, hybrid technique cannot be a candidate and only local sleep transistor can be applied, because each of SRAM cell may have logical “1” or logical “0” values without any rule. This constraints narrows choices and make layout more conspicuous. Instead of clustering, partial install of sleep transistor is chosen for the alternative.

Seeing the wire model simulation result, the whole SRAM cells can be grouped into two categories, cells near critical path and cells with more slack. The next figure shows this relation and if sleep transistors are used in the latter group then the leakage power will be reduced without increasing the worst-case delay.



[Figure IV-5] Word Line & Bit Line Delay Distribution without Sleep Transistors

Another figure below shows the delay distribution expected when sleepy transistors are partially used for the cells with more slack.



[Figure IV-6] Word Line & Bit Line Delay Distribution with Sleep Transistors

Theoretically, if multiple sleep transistors are placed depending on the amount of slack so that all the delays are equal, then leakage power reduction can be maximized without increasing the worst-case delay. But practically, the number of threshold voltage is limited to two, and heterogeneous cells require additional processing steps, so cost may cover the benefit of leakage power reduction. Therefore, dual threshold voltages and locally installed sleep transistors by group are the optimal strategy for this project as in [Figure III-1] Overview of 32K-bit Sleepy SRAM.

V. USER DOCUMENT

A. TITLE

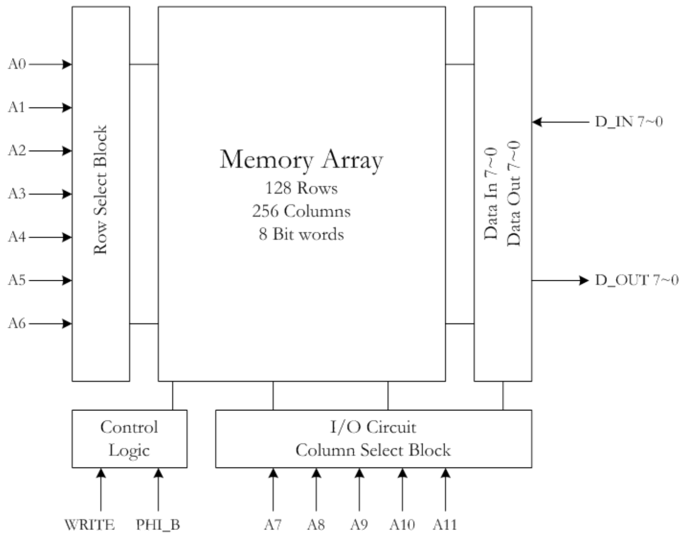
32K-bit SRAM: 128 rows, 256 columns, 8-bit words (3.3V operating voltage)

B. GENERAL DESCRIPTION

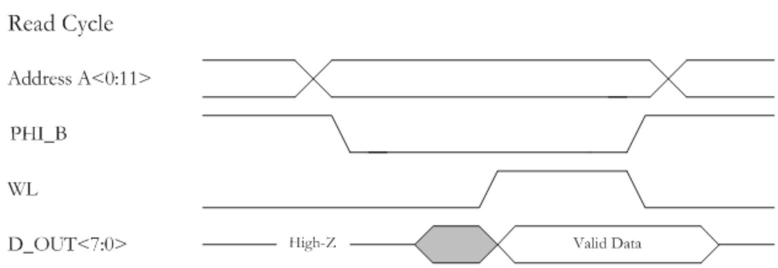
This is a 32,768 bit Static Random Access Memory (SRAM) organized by 4096 words by 8 bits. This memory has own input and output lines and has control signals, WRITE and PHI_b. SRAM fully operates in static mode. Therefore, no clock or refreshment is required.

A<11:0>	Address Input
D_IN<7:0>	Data Input
D_OUT<7:0>	Data Output
WRITE	Write Command Input
PHI_b	Bit line Pre-charge Command Input

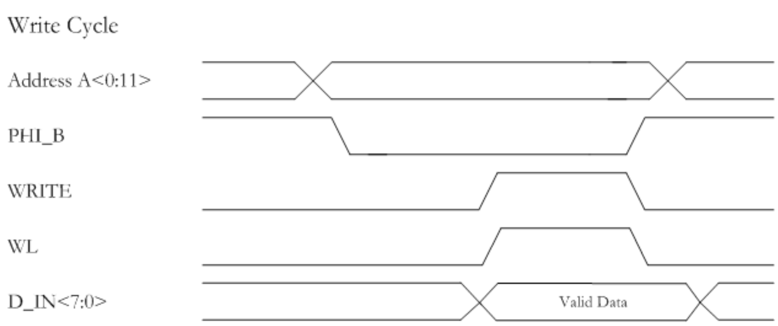
[Table V-1] Pin Description



[Figure V-1] 32K-bit SRAM Functional Block Diagram



[Figure V-2] Read Cycle Timing Diagram



[Figure V-3] Write Cycle Timing Diagram

Cell Standby Power Consumption	1.48E-3 mW
Chip Area	450907 □ m²
Maximum Latency	9.567E-10 sec

[Table V-2] Fundamental Parameters at Operational Points

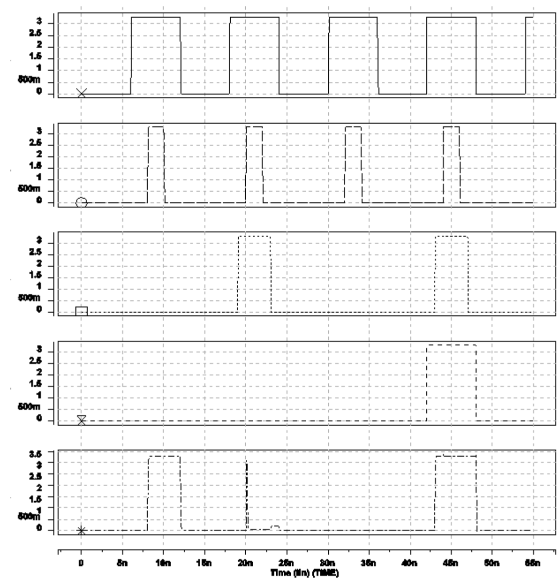
VI. TESTING

Testing for 32K-bit SRAM flows along the functional blocks; address decoders, SRAM cell and multiplexers. For the decoders and multiplexers, performance testing is not required to measure leakage power reduction of SRAM, so only functional test was performed. For the functional test of decoders & multiplexers, we made a program that generates Verilog test bench for all the case; this test bench includes a task that performs test. Following is the excerpt from the test bench.

```
reg [31:0] Calculated;

task test;
    input [31:0] Measured, Calculated;
    begin
        if ( Measured != Calculated ) begin
            $display( "ERROR: Measured = %h, Calculated = %h",
Measured, Calculated );
        end
    end
endtask

initial begin
        A[4:0]      = 5'b0;

#50;    A           = 7'h00;
        Calculated  = 128'h00000001;
#50;    test( WL, Calculated );
#50;    A           = 7'h01;
        Calculated  = 128'h00000002;
#50;    test( WL, Calculated );
#50;    A           = 7'h02;
        Calculated  = 128'h00000004;
#50;    test( WL, Calculated );
#50;    A           = 7'h03;
        Calculated  = 128'h00000008;
#50;    test( WL, Calculated );
#50;    A           = 7'h04;
        Calculated  = 128'h00000010;
#50;    test( WL, Calculated );
#50;    A           = 7'h05;
        Calculated  = 128'h00000020;
#50;    test( WL, Calculated );
#50;    A           = 7'h06;
        Calculated  = 128'h00000040;
```

[Excerpt of Verilog Test Bench for 5-to-32 Column Decoder]

SRAM cell has four cases for its operation; read "1" or "0", write "1" or "0" and all of these cases were tested thru Hspice simulator as following figure. Because SRAM cell requires a sort of tuned timing in input signals, each of PHI_b, WL, WRITE and $DATA_{IN}$ was set up to meet this requirement. And as transistor sizing critical in SRAM especially in 6T SRAM about noise issue, transistors were sized as discussed in the design documents.



[Figure VI-1] SRAM Cell Hspice Simulation Waveforms

Module	Coverage	Method/Tool
7-to-128 Row Decoder	100% (128 cases)	Verilog-XL
5-to-32 Column Decoder	100% (32 cases)	Verilog-XL
SRAM Cell	100% (4 cases)	Hspice
32-to-1 Multiplexer	100% (32 cases)	Verilog-XL

[Table VI-1] Testing Coverage Metrics

VII. RESULT & OPTIMIZATION

Leakage power in this project was measured at the steady state when each SRAM cell holds logical "1" or "0", which removes dynamic power and direct path power. And the result is,

> *Leakage Power of Sleepy SRAM Cell = 5.810E-12 W*
>
> *Leakage Power of Non-Sleepy SRAM Cell = 8.452E-11 W*

The leakage power of non-sleepy SRAM is 1454.73% larger than sleepy SRAM. To extend the analysis further, we assumed four cases as following. One noticeable is sleepy partition is nearer to the output and non-sleepy is farther from the output. As stated before, this is for holding the same worst-case delay of 32K-bit SRAM. Additional area increase was estimated as 40% per sleepy SRAM cell; this is estimated by width of transistors as,

$$\left(\frac{Dimension\ of\ \ Sleepy\ SRAM\ \ Cell}{Dimension\ of\ \ Nonsleepy\ SRAM\ \ Cell}\right) = \frac{1170 \times (1400/1000)\lambda}{1170\lambda} = 1.40$$

Mode	# Sleepy Cell	# Non-Sleepy Cell
100% Sleepy	32768	0
75% Sleepy	24576	8192
50% Sleepy	16384	16384
25% Sleepy	8192	24576
Non-Sleepy	0	32768

[Table VII-1] Sleepy SRAM Partition Mode

Mode	Leakage Power	Reduction Rate	Area Overhead
100% Sleepy	1.90E-07 W	93.13%	40%
75% Sleepy	8.35E-07 W	69.84%	30%
50% Sleepy	1.48E-06 W	46.56%	20%
25% Sleepy	2.12E-06 W	23.28%	10%
Non-Sleepy	2.77E-06 W	0	0

[Table VII-2] Leakage Power, Rate of Reduction & Area Overhead

The above table shows the leakage power, the rate of reduction and area overhead. If power reduction is the only factor then 100% sleepy mode seems to be the best choice, however delay and area constraints make different decision. For this purpose, delay of the SRAM cell at the critical positions should be simulated and the result is,

Delay	127 Word Lines	255 Word Lines
128 Bit Lines	9.436E-10 sec	9.567E-10 sec
64 Bit Lines	6.797E-10 sec	6.957E-10 sec

[Table VII-3] Delays of SRAM Cell at Critical Positions

The simulation result exceeds the expected delay increase calculated data in sleep transistor sizing; however delays thru 64 bit lines are still shorter than 128 bit lines. Practically, memory latency for reading data is determined at the conditions where maximum delay occurs, so we can accept this delay penalty. The measured rate of delay penalty is,

$$\begin{aligned} \textit{rate of } \Delta_{penalty} &= \max\left(\frac{(6.797\times10^{-10} - 5.061\times10^{-10})}{5.061\times10^{-10}}, \frac{(6.957\times10^{-10} - 5.222\times10^{-10})}{5.222\times10^{-10}} \right) \\ &= 34.3\% \end{aligned}$$

and we can estimate overall delay penalty. Average delay penalty assumes the cell accesses are uniformly distributed, the worst-case and the best-case delay each indicate delay thru the farthest cell and the nearest partition from the output. And RC delay along bit line is not a perfect linear but wire delay simulation shows the rate of curve is very small that it is assumed that RC delay tends to be linear.

Mode	Worst-Case $\Delta_{penalty}$	Average $\Delta_{penalty}$	Best-Case $\Delta_{penalty}$
100% Sleepy	34.3%	34.3%	34.3%
75% Sleepy	9.3%	25.7%	34.3%
50% Sleepy	0	17.1%	34.3%
25% Sleepy	0	8.57%	34.3%
Non-Sleepy	0	0	0.00%

[Table VII-4] Sleepy Modes & Delay Penalty

Now, there is a decision change, 100% & 75% sleepy mode has large area overhead and increased worst-case delay meanwhile memory is a dimension critical device and slow compared with processing unit. Delay penalty rate also tells the maximum tolerance rate is 100-34.3=65.7% and if RC delay is distributed in linear fashion, maximum rate of sleepy SRAM partition becomes 65.7%. And 25% sleepy mode has not enough leakage power reduction as the whole system can tolerate ≈65.7% penalty. Therefore, taking all these factors into account gives 50~65.7% sleepy mode is the optimized for the project. If more than two threshold voltages are available, then the optimal partition is expected to change.

VIII. REFERENCES

N. Weste et al., “Principles of CMOS VLSI Design (3rd Ed. 2005)”, Addison-Wesley.

K. Zhang, “SRAM Design on 65-nm CMOS Technology With Dynamic Sleep Transistor for Leakage Reduction (Apr., 2005)”, IEEE Journal of Solid-State Circuits Vol. 40, No. 4.

V. Rayapati, “Interconnect Propagation Delay Modeling and Validation for the 16-MB CMOS SRAM Chip (Aug., 1996)”, IEEE Transactions on Components, Packaging, and Manufacturing Technology Vol. 19, No. 3.

R. Castagnetti et al., “A High-Performance SRAM Technology With Reduced Chip-Level Routing Congestion for SOC (Mar., 2005)”, Proceedings of the Sixth International Symposium on Quality Electronic Design (ISQED’05).

M. Anis, “Design and Optimization of Multithreshold CMOS (MTCMOS) Circuits (Oct., 2003)”, IEEE Transaction on Computer Aided Design of Integrated Circuits and Systems, Vol. 22, No. 10.

H. Zhou et al., “Adaptive Mode-Control: A Low-Leakage, Power-Efficient Cache Design”, Department of Electrical & Computer Engineering, North Carolina State University.

M. Johnson, “Leakage Control With Efficient Use of Transistor Stacks in Single Threshold CMOS (Feb., 2002)”, IEEE Transactions on Very Large Scale Integrated (VLSI) Systems, Vol. 10, No. 1.

B. Calhoun, et al., “A Leakage Reduction Methodology for Distributed MTCMOS (May, 2004)”, IEEE Journal of Solid-State Circuits, Vol. 39, No. 5.

A. Ramalingam et al., “Sleep Transistor Sizing Using Timing Criticality and Temporal Currents (Jan., 2005)”, Proc. Asia South Pacific Design Automation Conference (ASPDAC).

H. Qin et al., “SRAM Leakage Suppression by Minimizing Standby Supply Voltage”, Department of EECS, University of California at Berkeley.

IX. APPENDIX

A. SCHEMATIC DIAGRAMS