



# Pareto Points in SRAM Design Using the Sleepy Stack Approach

**Jun Cheol Park<sup>^</sup> and Vincent J. Mooney III\***

\*Associate Director, <sup>^</sup>Center for Research on Embedded Systems and Technology (CREST), <http://www.crest.gatech.edu>

\*Associate Professor, <sup>^</sup>School of Electrical and Computer Engineering

\*Adjunct Associate Professor, College of Computing

\*Founder, Hardware/Software Codesign Lab, <http://codesign.ece.gatech.edu>  
Georgia Institute of Technology, Atlanta, GA, USA

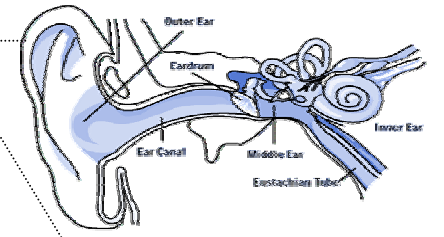
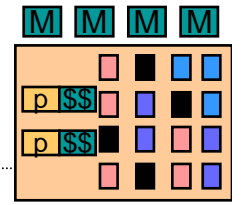
IFIP VLSI-SoC  
October 2005

# Outline

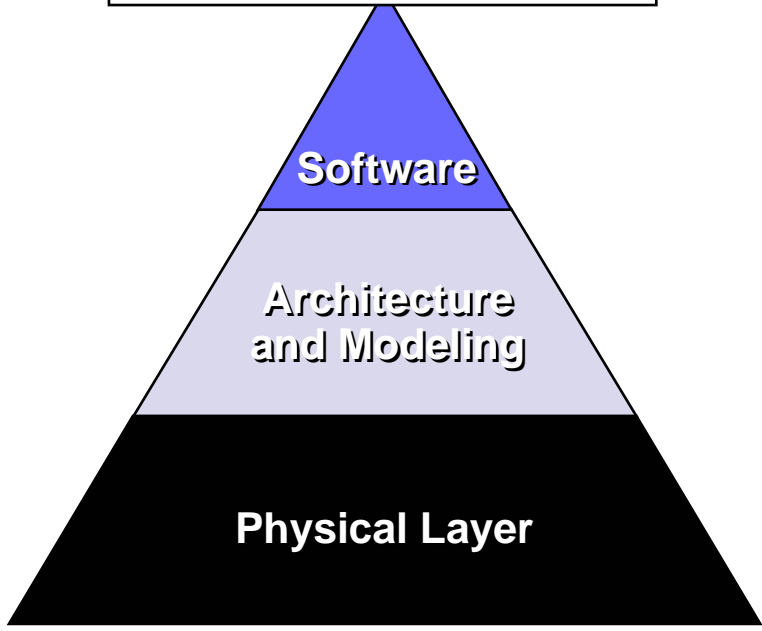
- Introduction
- Related work
- Sleepy stack structure
- Sleepy stack SRAM
- Conclusion



# CREST Faculty & Research

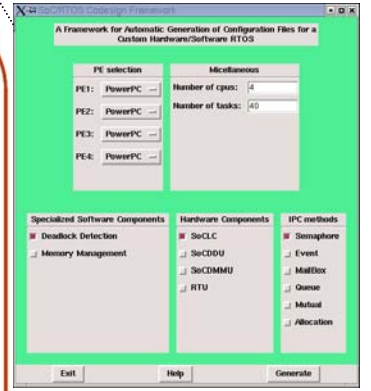
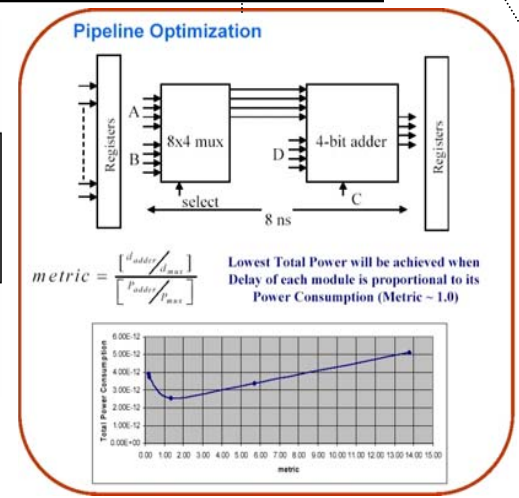
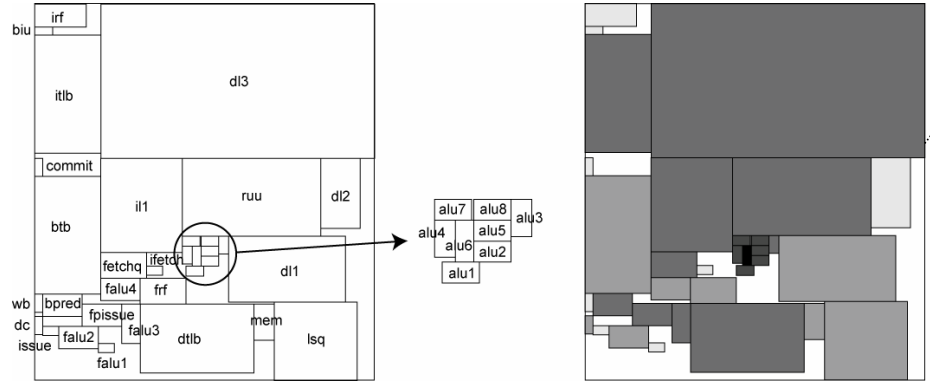


Embedded System Developer



**Faculty**

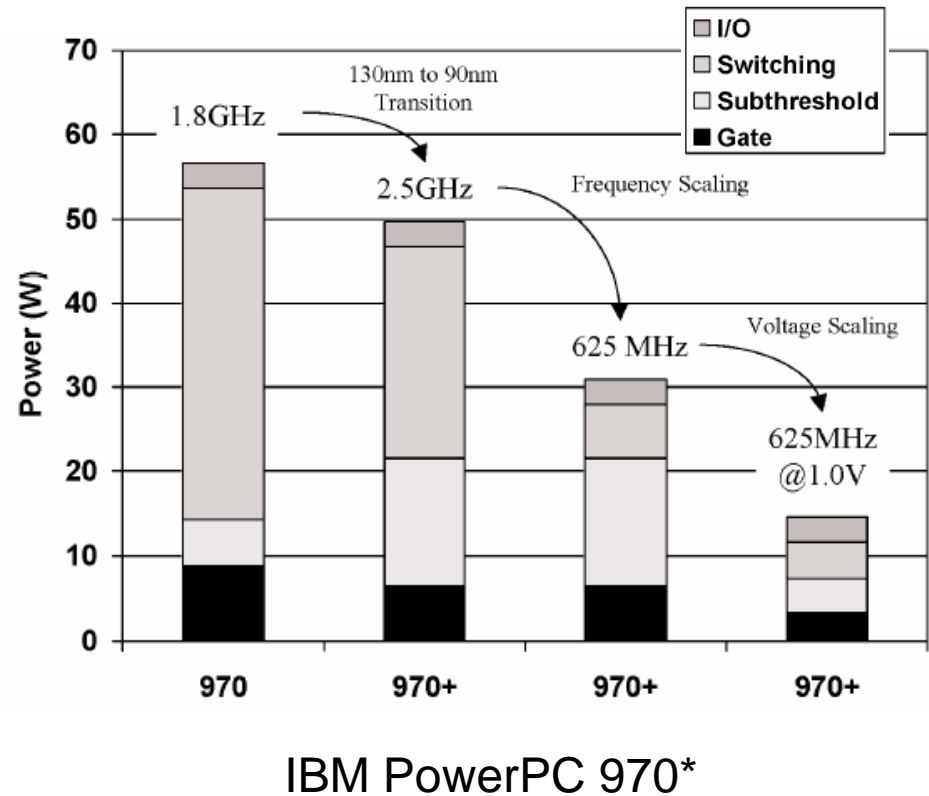
M. Egerstedt  
K. Palem  
S. Yalamanchili  
V. Mooney, D. Anderson  
S.-K. Lim, A. Chatterjee



# Power consumption



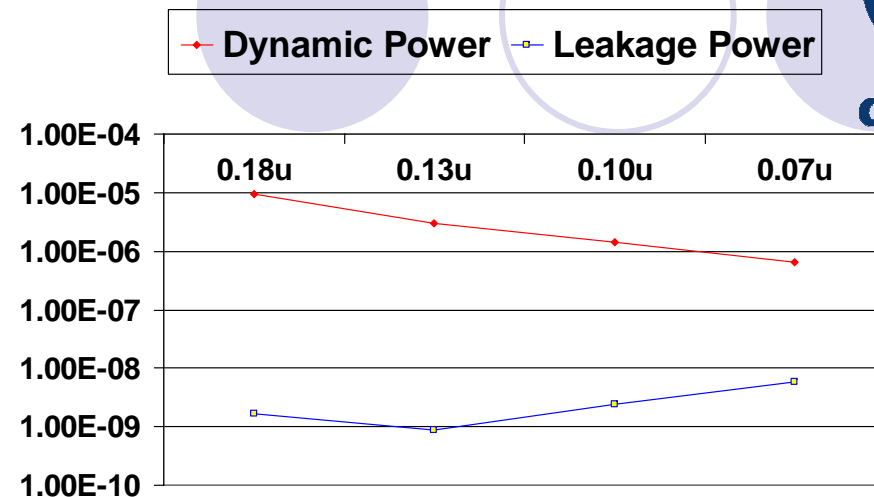
- Power consumption of VLSI is a fundamental problem of mobile devices as well high-performance computers
  - Limited operation (battery life)
  - Heat
  - Operation cost
- Power = dynamic + static
  - Dynamic power more than 90% of total power (0.18u tech. and above)
- Dynamic power reduction:
  - Technology scaling
  - Frequency scaling
  - Voltage scaling



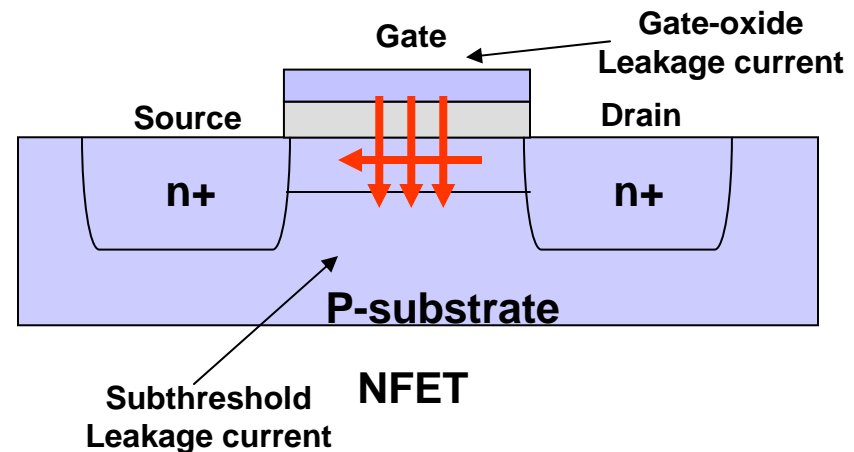
\*N. Rohrer et al., "PowerPC 970 in 130nm and 90nm Technologies," *IEEE International Solid-State Circuits Conference*, Vol. 1, pp. 68-69, February 2004.

# Leakage power

- Leakage power became important as the feature size shrinks
- Subthreshold leakage
  - Scaling down of  $V_{th}$ : Leakage increases exponentially as  $V_{th}$  decreases
  - Short-channel effect: channel controlled by drain
  - Our research focus
- Gate-oxide leakage
  - Gate tunneling due to thin oxide
  - High-k dielectric could be a solution



Experimental result 4-bit adder\*



\*Berkeley Predictive Technology Model (BPTM).  
 [Online]. Available <http://www-device.eecs.berkeley.edu/~ptm>.

# Outline

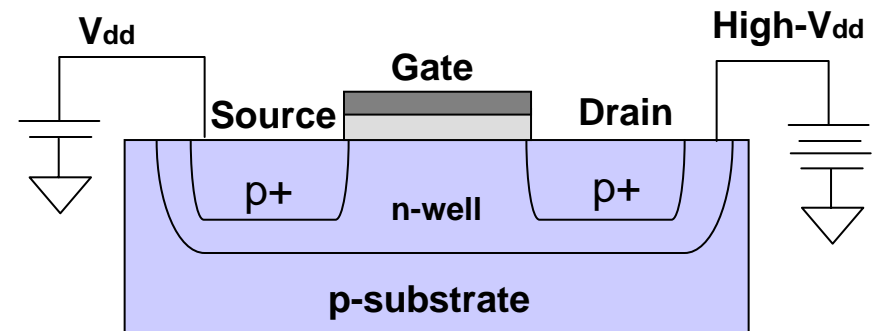
- Introduction
- Related work
- Sleepy stack structure
- Sleepy stack SRAM
- Conclusion



# Low-leakage SRAM



- Auto-Backgate-Controlled Multi Threshold CMOS (ABC-MTCMOS) [Nii98]
  - Reverse source-body bias during sleep mode
  - ➔ ○ Slow transition and large dynamic power to charge n-wells
- Gated-V<sub>dd</sub> [Powell00](Prof. K. Roy)
  - Isolate SRAM cells using sleep transistor
  - Loses state during sleep mode
- Drowsy cache [Flautner02]
  - Scaling V<sub>dd</sub> dynamically
  - Smaller leakage reduction (<86%) (we will show 3 orders magnitude reduction)

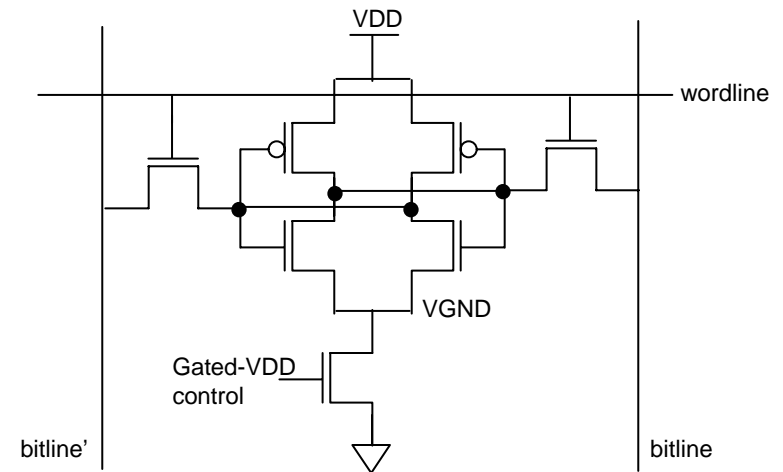


ABC-MTCMOS

# Low-leakage SRAM



- Auto-Backgate-Controlled Multi Threshold CMOS (ABC-MTCMOS) [Nii98]
  - Reverse source-body bias during sleep mode
  - Slow transition and large dynamic power to charge n-wells
- Gated-Vdd [Powell00](Prof. K. Roy)
  - Isolate SRAM cells using sleep transistor
  - ➔ ○ Loses state during sleep mode
- Drowsy cache [Flautner02]
  - Scaling Vdd dynamically
  - Smaller leakage reduction (<86%) (we will show 3 orders magnitude reduction)



Gated-V<sub>DD</sub>

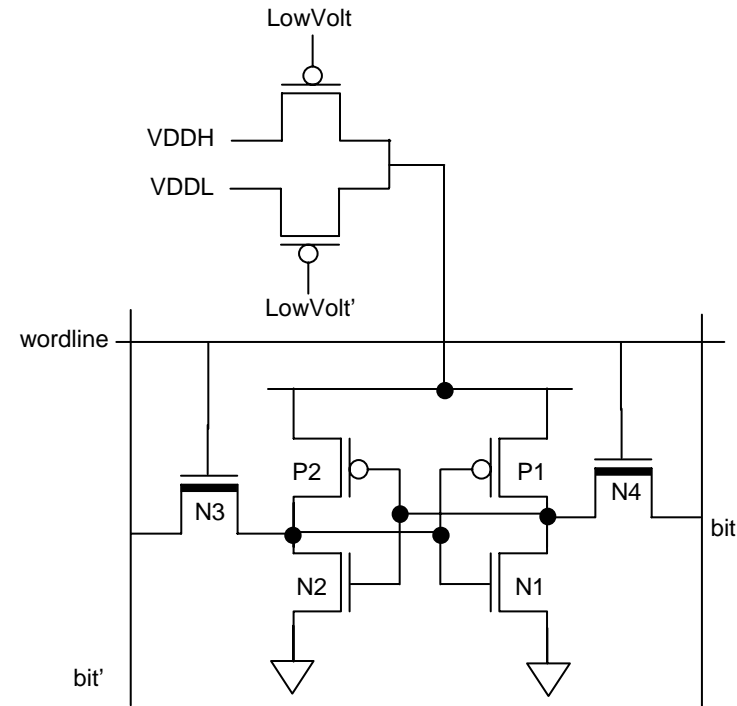
\*Intel introduces 65-nm sleep transistor SRAM from Intel.com , “65-nm process technology extends the benefit of Moore’s law”



# Low-leakage SRAM



- Auto-Backgate-Controlled Multi Threshold CMOS (ABC-MTCMOS) [Nii98]
  - Reverse source-body bias during sleep mode
  - Slow transition and large dynamic power to charge n-wells
- Gated-V<sub>dd</sub> [Powell00](Prof. K. Roy)
  - Isolate SRAM cells using sleep transistor
  - Loses state during sleep mode
- Drowsy cache [Flautner02]
  - Scaling V<sub>dd</sub> dynamically
  - Smaller leakage reduction (<86%) (we will show 3 orders magnitude reduction)



Drowsy cache

# Low-leakage SRAM comparison



- Sleepy stack SRAM cell
  - No need to charge n-well (ABC-MTCMOS)
  - State-saving (gated- $V_{dd}$ )
  - Larger leakage power savings (drowsy cache)

# Outline

- Introduction
- Related work
- Sleepy stack structure
- Sleepy stack SRAM
- Conclusion

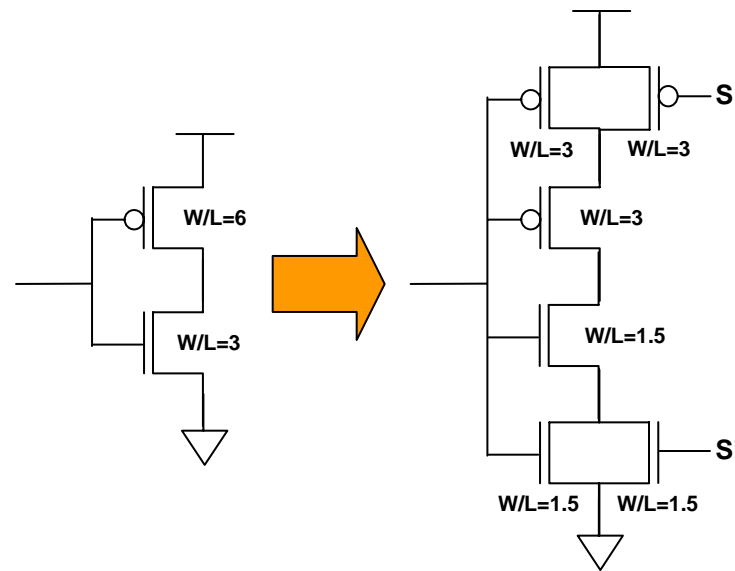


# Introduction of sleepy stack



- New state-saving ultra low-leakage technique
- Combination of the sleep transistor and forced stack technique
- Applicable to generic VLSI structures as well as SRAM
- Target application requires long standby with fast response, e.g., cell phone

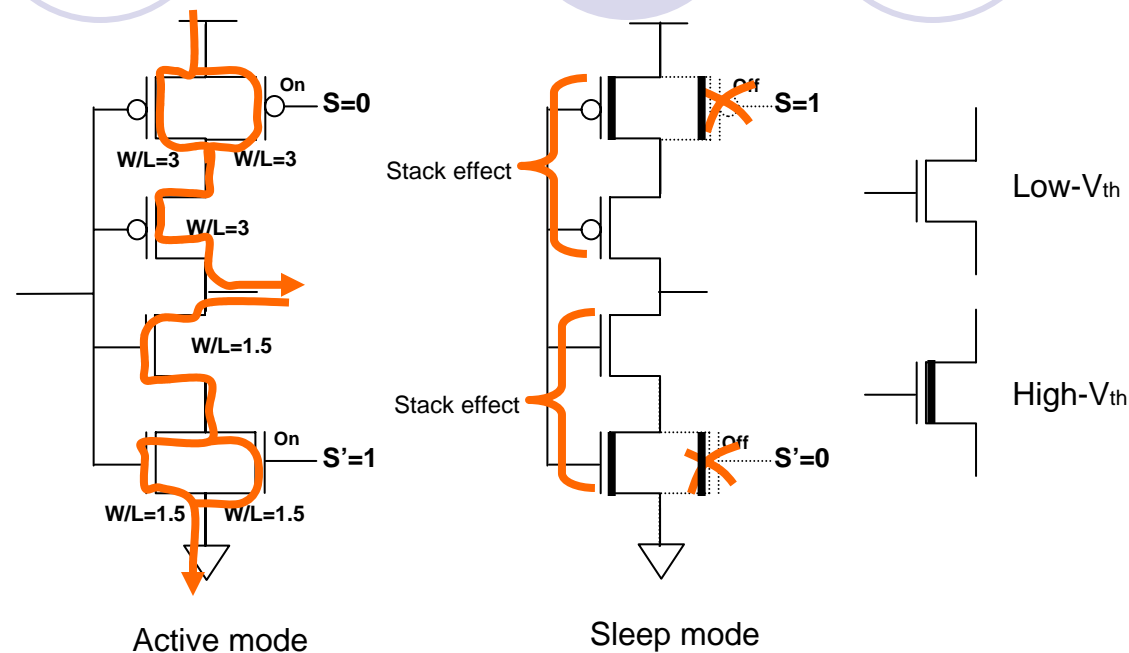
# Sleepy stack structure



Conventional CMOS inverter      Sleepy stack inverter

- First, break down a transistor similar to the forced stack technique
- Then add sleep transistors

# Sleepy stack operation



- During active mode, sleep transistors are on, then reduced resistance increases current while reducing delay
- During sleep mode, sleep transistors are off, stacked transistors suppress leakage current while saving state
- Can apply high- $V_{th}$ , which is not used in the forced stack technique due to the dramatic delay increase ( $>6.2X$ )

# Sleepy stack for logic



- Apply sleepy stack to a chain of 4 inverters
- Targeting 0.07u technology
- Compared to forced stack, the best prior state-saving low leakage technique, sleepy stack with dual- $V_{th}$  achieves 215X reduction in leakage power with 6% decrease in delay
- Sleepy stack is 51% larger than forced stack
- Published in PATMOS 2004

# Outline

- Introduction
- Related work
- Sleepy stack structure
- Sleepy stack SRAM
- Conclusion

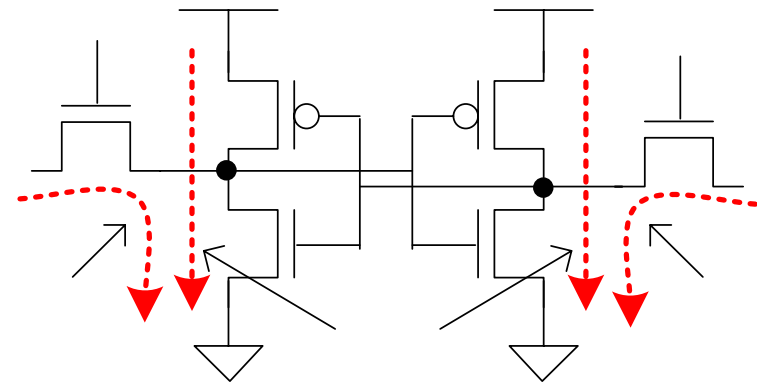




# Sleepy stack SRAM cell



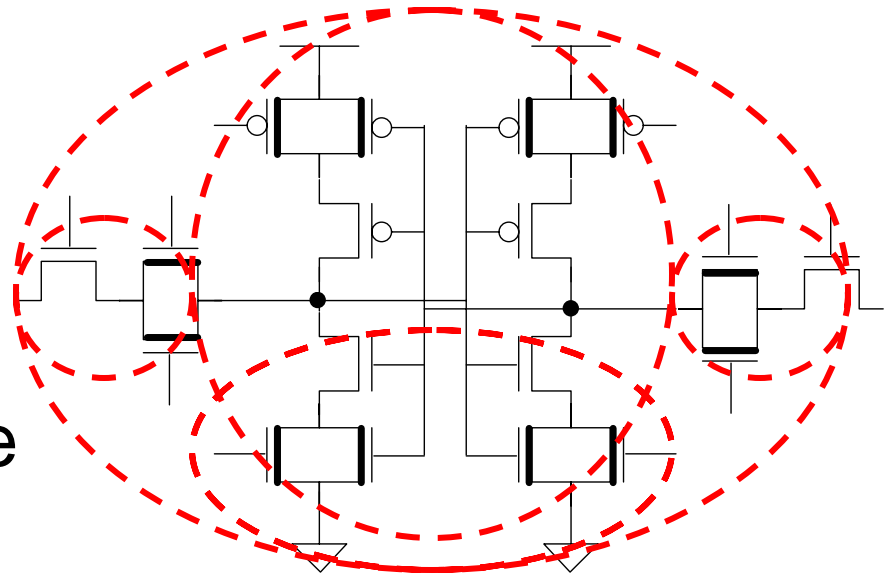
- Sleepy stack technique achieves ultra-low leakage power while saving state
- Apply the sleepy stack technique to SRAM cell design
  - Large leakage power saving expected in cache
  - State-saving
  - 6-T SRAM cell is based on coupled inverters
- SRAM cell leakage paths
  - Cell leakage
  - Bitline leakage



# Sleepy stack SRAM cell



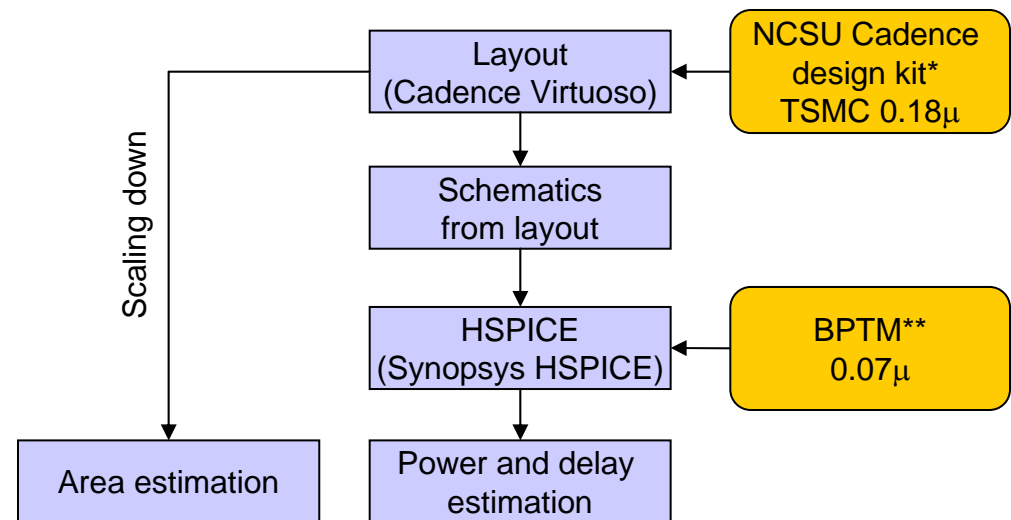
- Sleepy stack SRAM cell
  - PD sleepy stack
  - PD, WL sleepy stack
  - PU, PD sleepy stack
  - PU, PD, WL sleepy stack
- Area, delay and leakage power tradeoffs



# Experimental methodology



- Estimate area by scaling down 0.18 $\mu$  layout
- Estimate dynamic power, static power and cell read time using BPTM 0.07 $\mu$  technology



\*NC State University Cadence Tool Information.  
[Online]. Available <http://www.cadence.ncsu.edu>.

\*\*Berkeley Predictive Technology Model (BPTM).

[Online]. Available <http://www-device.eecs.berkeley.edu/~pt119>.

# Experimental methodology



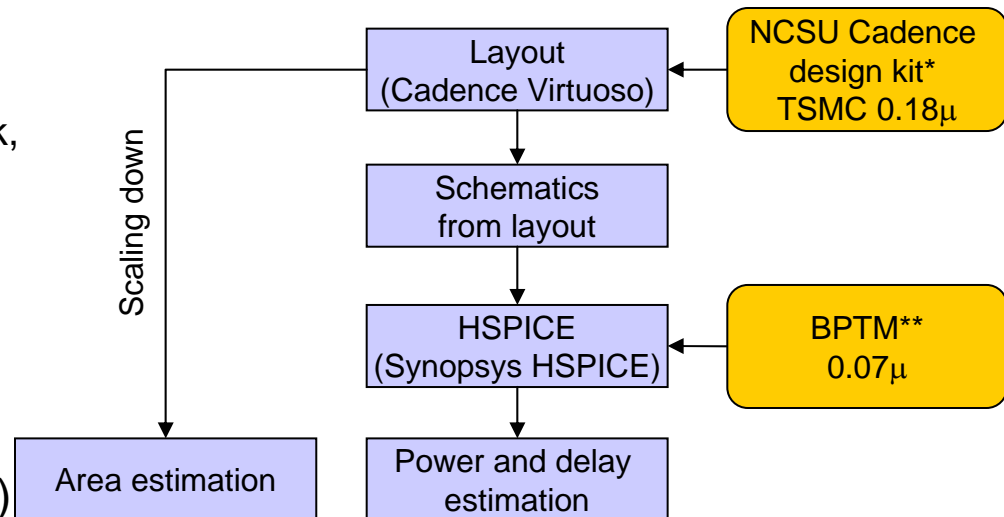
- Base case and three techniques are compared
  - High- $V_{th}$  technique, forced stack, and sleepy stack
- 64x64 bit SRAM array designed
- Area estimated by scaling down 0.18 $\mu$  layout
  - Area of 0.18 $\mu$  layout\*(0.07 $\mu$ /0.18 $\mu$ )
- Power and read time using HSPICE targeting 0.07 $\mu$
- 1.5 $\times V_{th}$  and 2.0 $\times V_{th}$
- 25°C and 110°C

	Technique	
<b>Case1</b>	Low- $V_{th}$ Std	Conventional 6T SRAM
<b>Case2</b>	PD high- $V_{th}$	High- $V_{th}$ applied to PD
<b>Case3</b>	PD, WL high- $V_{th}$	High- $V_{th}$ applied to PD, WL
<b>Case4</b>	PU, PD high- $V_{th}$	High- $V_{th}$ applied to PU, PD
<b>Case5</b>	PU, PD, WL high- $V_{th}$	High- $V_{th}$ applied to PU, PD, WL
<b>Case6</b>	PD stack	Stack applied to PD
<b>Case7</b>	PD, WL stack	Stack applied to PD, WL
<b>Case8</b>	PU, PD stack	Stack applied to PU, PD
<b>Case9</b>	PU, PD, WL stack	Stack applied to PU, PD, WL
<b>Case10</b>	PD sleepy stack	Sleepy stack applied to PD
<b>Case11</b>	PD, WL sleepy stack	Sleepy stack applied to PD, WL
<b>Case12</b>	PU, PD sleepy stack	Sleepy stack applied to PU, PD
<b>Case13</b>	PU, PD, WL sleepy stack	Sleepy stack applied to PU, PD, WL

# Experimental methodology



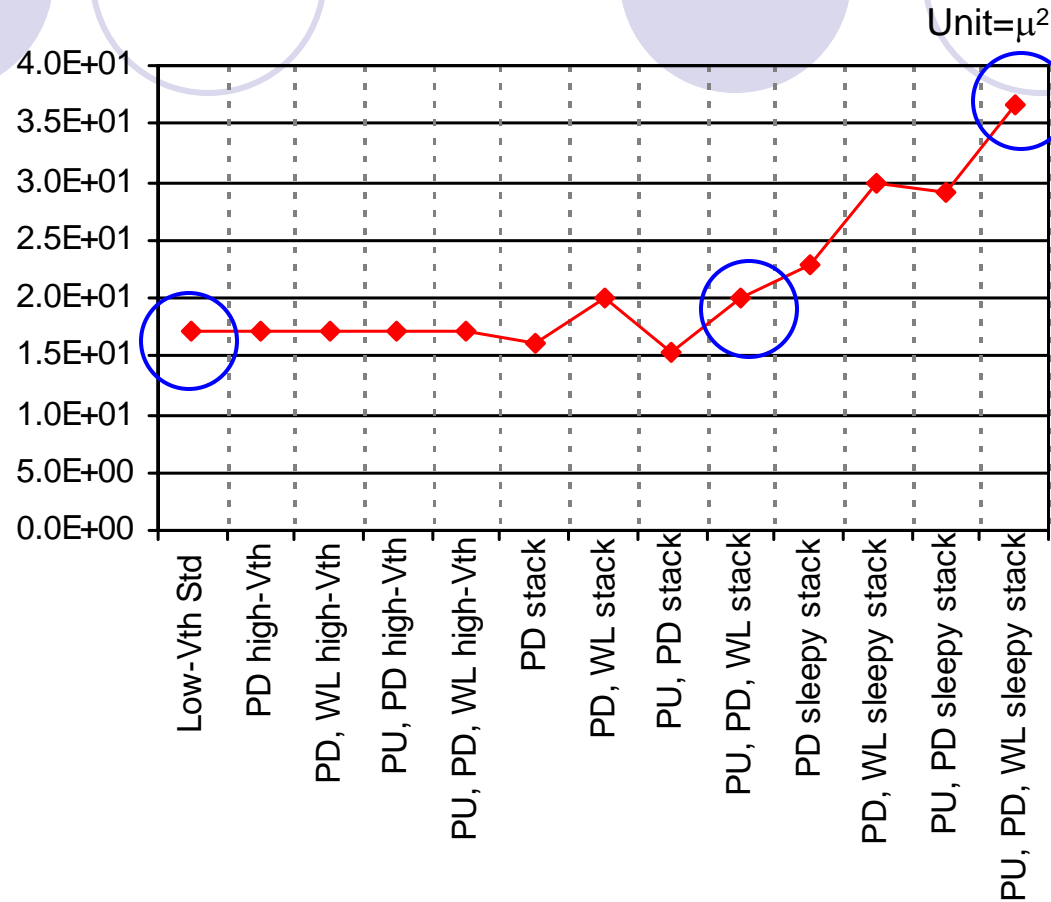
- Base case and three techniques are compared
  - High- $V_{th}$  technique, forced stack, and sleepy stack
- 64x64 bit SRAM array designed
- Area estimated by scaling down 0.18 $\mu$  layout
  - Area of 0.18 $\mu$  layout \*  $((0.07\mu/0.18\mu)^2 + 10\%)$
- Power and read time using HSPICE targeting 0.07 $\mu$
- 1.5 $\times V_{th}$  and 2.0 $\times V_{th}$
- 25 $^{\circ}C$  and 110 $^{\circ}C$



\*NC State University Cadence Tool Information. [Online]. Available <http://www.cadence.ncsu.edu>.

\*\*Berkeley Predictive Technology Model (BPTM). [Online]. Available <http://www-device.eecs.berkeley.edu/~ptm>.

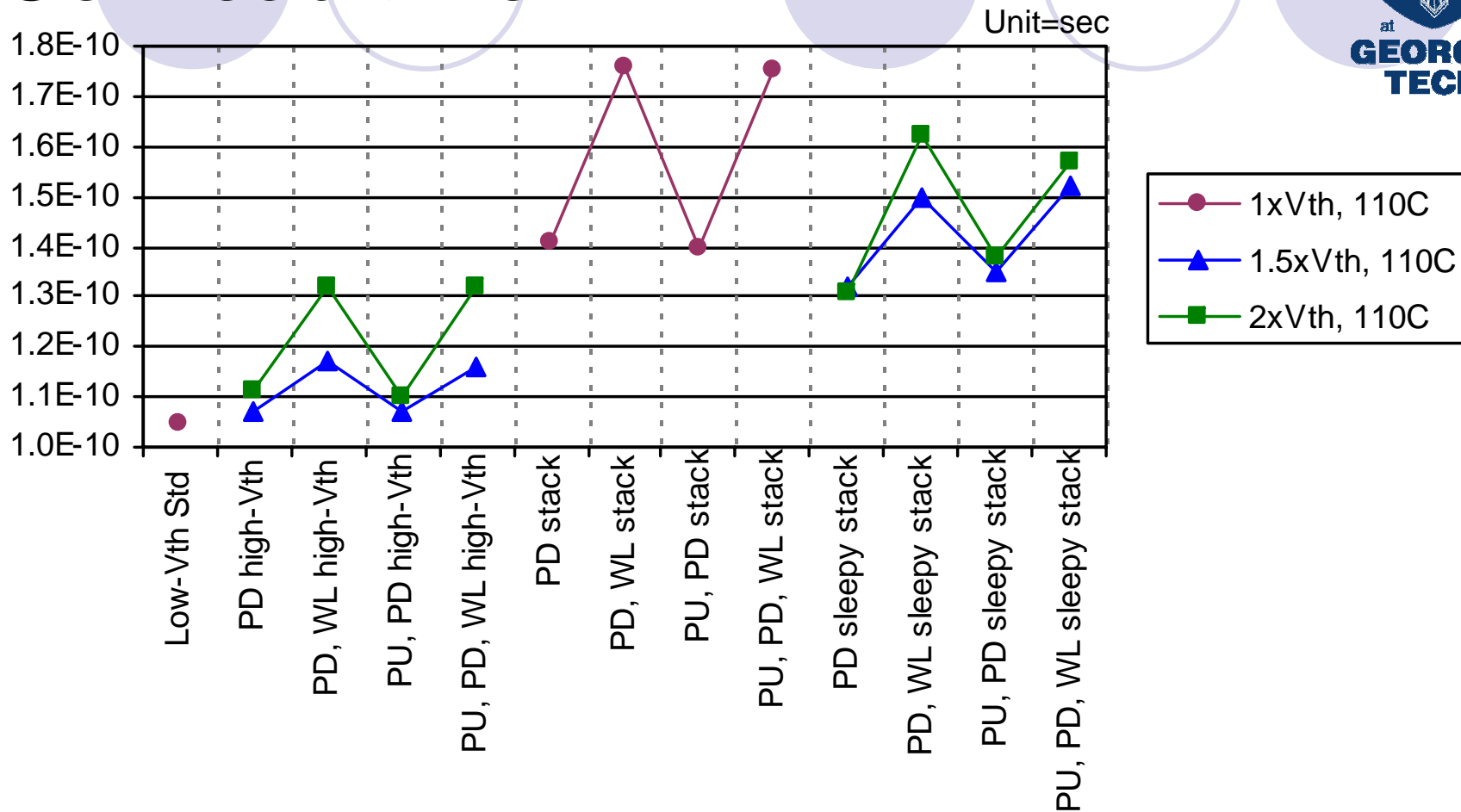
# Area



- PU, PD, WL sleepy stack is 113% and 83% larger than base case and PU, PD, WL forced stack, respectively

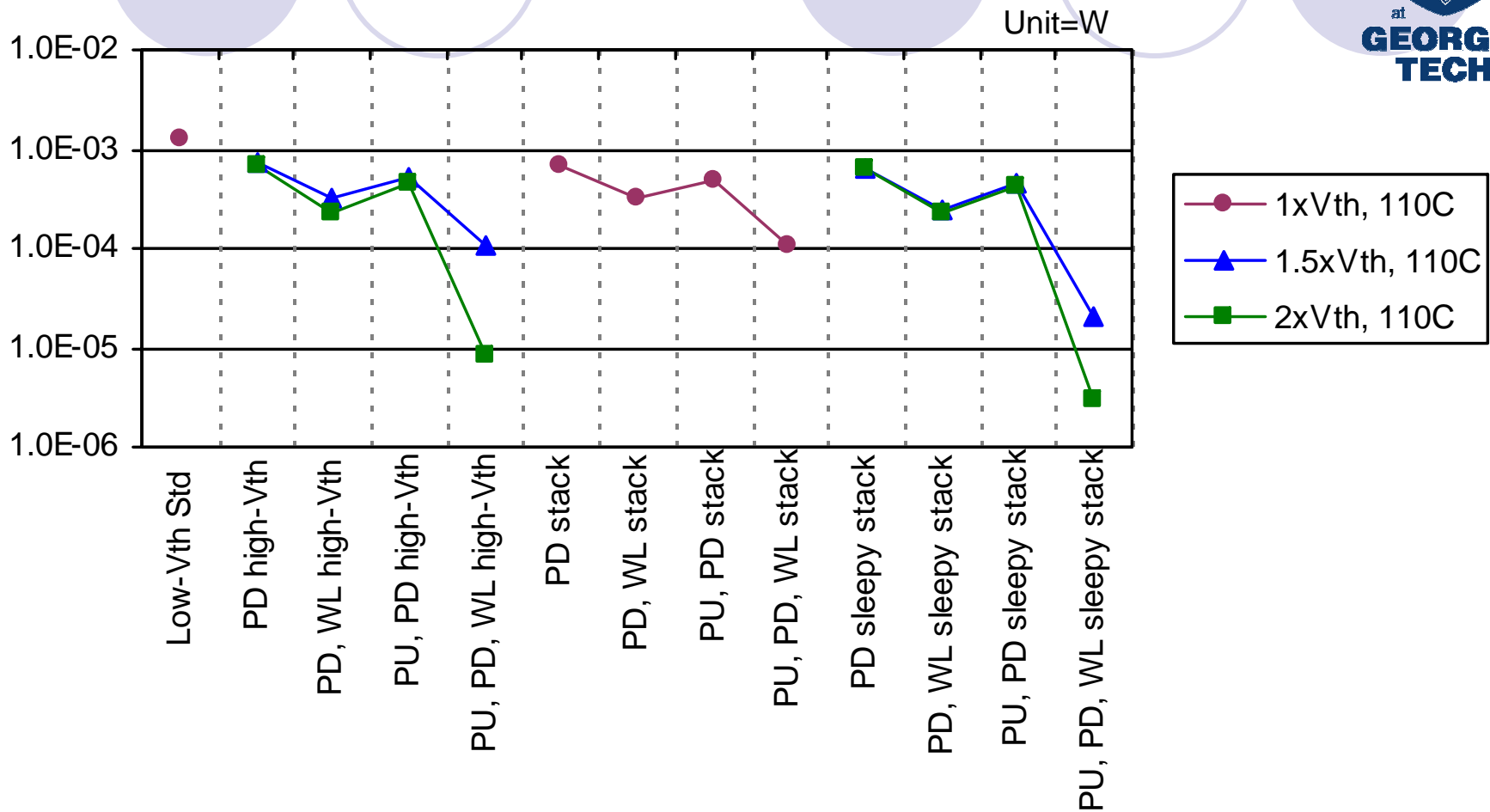


# Cell read time



● Delay: High- $V_{th}$  < sleepy stack < forced stack

# Leakage power



- At 110°C, the worst case, leakage power: forced stack > high-Vth 2xVth > sleepy stack 2xVth



# Tradeoffs

1.5xV<sub>th</sub> at 110°C



	Technique	Leakage power (W)	Delay (sec)	Area (u <sup>2</sup> )	Normalized leakage power	Normalized delay	Normalized area
Case1	Low-V <sub>th</sub> Std	1.254E-03	1.05E-10	17.21	1.000	1.000	1.000
Case2	PD high-V <sub>th</sub>	7.159E-04	1.07E-10	17.21	0.571	1.020	1.000
Case6	PD stack	7.071E-04	1.41E-10	16.22	0.564	1.345	0.942
Case10*	PD sleepy stack*	6.744E-04	1.15E-10	25.17	0.538	1.102	1.463
Case10	PD sleepy stack	6.621E-04	1.32E-10	22.91	0.528	1.263	1.331
Case4	PU, PD high-V <sub>th</sub>	5.042E-04	1.07E-10	17.21	0.402	1.020	1.000
Case8	PU, PD stack	4.952E-04	1.40E-10	15.37	0.395	1.341	0.893
Case12*	PU, PD sleepy stack*	4.532E-04	1.15E-10	31.30	0.362	1.103	1.818
Case12	PU, PD sleepy stack	4.430E-04	1.35E-10	29.03	0.353	1.287	1.687
Case3	PD, WL high-V <sub>th</sub>	3.203E-04	1.17E-10	17.21	0.256	1.117	1.000
Case7	PD, WL stack	3.202E-04	1.76E-10	19.96	0.255	1.682	1.159
Case11*	PD, WL sleepy stack*	2.721E-04	1.16E-10	34.40	0.217	1.111	1.998
Case11	PD, WL sleepy stack	2.451E-04	1.50E-10	29.87	0.196	1.435	1.735
Case5	PU, PD, WL high-V <sub>th</sub>	1.074E-04	1.16E-10	17.21	0.086	1.110	1.000
Case9	PU, PD, WL stack	1.043E-04	1.75E-10	19.96	0.083	1.678	1.159
Case13*	PU, PD, WL sleepy stack*	4.308E-05	1.16E-10	41.12	0.034	1.112	2.389
Case13	PU, PD, WL sleepy stack	2.093E-05	1.52E-10	36.61	0.017	1.450	2.127

- Sleepy stack delay is matched to Case5 (“\*” means delay matched to Case5=best prior work)
- Sleepy stack SRAM provides new pareto points (blue rows)
- Case13 achieves 5.13X leakage reduction (with 32% delay increase), alternatively Case13\* achieves 2.49X leakage reduction compared to Case5 (while matching delay to Case5)

# Tradeoffs

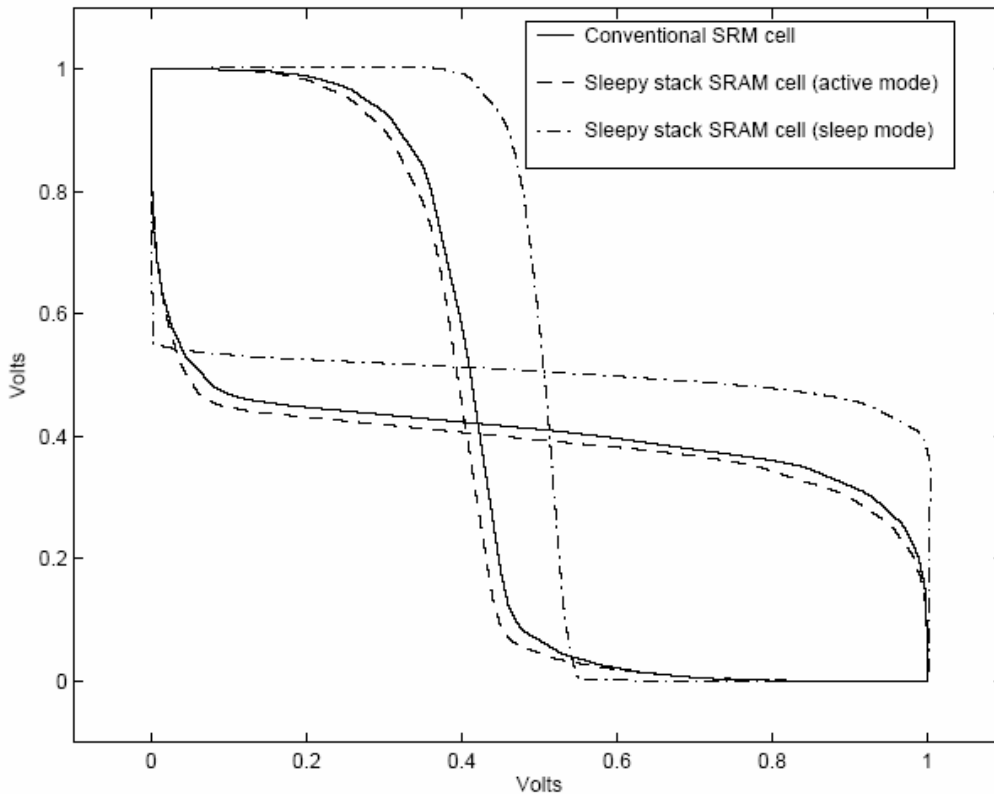
2.0xV<sub>th</sub> at 110°C



	Technique	Static (W)	Delay (sec)	Area (u <sup>2</sup> )	Normalized leakage	Normalized delay	Normalized area
Case1	Low-V <sub>th</sub> Std	1.25E-03	1.05E-10	17.21	1.000	1.000	1.000
Case6	PD stack	7.07E-04	1.41E-10	16.22	0.564	1.345	0.942
Case2	PD high-V <sub>th</sub>	6.65E-04	1.11E-10	17.21	0.530	1.061	1.000
Case10	PD sleepy stack	6.51E-04	1.31E-10	22.91	0.519	1.254	1.331
Case10*	PD sleepy stack*	6.51E-04	1.31E-10	22.91	0.519	1.254	1.331
Case8	PU, PD stack	4.95E-04	1.40E-10	15.37	0.395	1.341	0.893
Case4	PU, PD high-V <sub>th</sub>	4.42E-04	1.10E-10	17.21	0.352	1.048	1.000
Case12*	PU, PD sleepy stack*	4.31E-04	1.33E-10	29.48	0.344	1.270	1.713
Case12	PU, PD sleepy stack	4.31E-04	1.38E-10	29.03	0.344	1.319	1.687
Case7	PD, WL stack	3.20E-04	1.76E-10	19.96	0.255	1.682	1.159
Case3	PD, WL high-V <sub>th</sub>	2.33E-04	1.32E-10	17.21	0.186	1.262	1.000
Case11*	PD, WL sleepy stack*	2.29E-04	1.30E-10	32.28	0.183	1.239	1.876
Case11	PD, WL sleepy stack	2.28E-04	1.62E-10	29.87	0.182	1.546	1.735
Case9	PU, PD, WL stack	1.04E-04	1.75E-10	19.96	0.083	1.678	1.159
Case5	PU, PD, WL high-V <sub>th</sub>	8.19E-06	1.32E-10	17.21	0.007	1.259	1.000
Case13*	PU, PD, WL sleepy stack*	3.62E-06	1.32E-10	38.78	0.003	1.265	2.253
Case13	PU, PD, WL sleepy stack	2.95E-06	1.57E-10	36.61	0.002	1.504	2.127

- Sleepy stack delay is matched to Case5 (“\*” means delay matched to Case5=best prior work)
- Sleepy stack SRAM provides new pareto points (blue rows)
- Case13 achieves 2.77X leakage reduction (with 19% delay increase over Case5), alternatively Case13\* achieves 2.26X leakage reduction compared to Case5 (while matching delay to Case5)

# Static noise margin



	Technique	Static noise margin (V)	
		Active mode	Sleep mode
Case1	Low-Vth Std	0.299	N/A
Case10	PD sleepy stack	3.167	0.362
Case11	PD, WL sleepy stack	0.324	0.363
Case12	PU, PD sleepy stack	0.299	0.384
Case13	PU, PD, WL sleepy stack	0.299	0.384

- Measure noise immunity using static noise margin (SNM)
- SNM of the sleepy stack is similar or better than the base case

# Conclusion



- Sleepy stack SRAM cell provides new pareto points in ultra-low leakage power consumption
- 2.77X leakage reduction over high- $V_{th}$  with 19% delay increase or 2.26X without delay increase
- Sleepy stack SRAM cell shows the same or better SNM than the base case