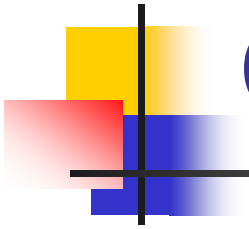
Sleepy Keeper : a New Approach to Low-Leakage Power VLSI Design

Se Hun Kim and Vincent J. Mooney III

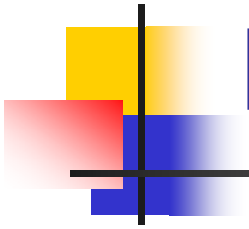
Center for Research on Embedded Systems and Technology
School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, Georgia, U.S.A

October 2006



Outline

- Introduction
- Related work
- Sleepy Keeper
- Experimental Methodology
- Experimental Results
- Conclusion



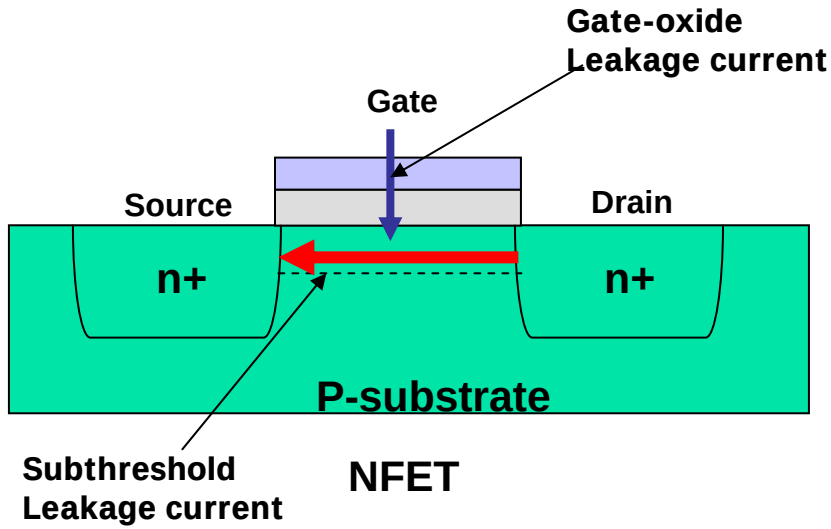
Background

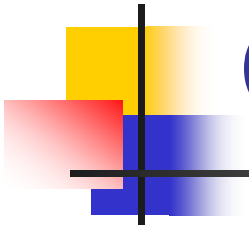
- Technology Trend
  - High density → Transistor size scaling down
  - High performance → V_{th} scaling down

    ➔ Increase of leakage power
- Increase of portable device (e.g., Cell phone, PDA)
- Power Consumption = Dynamic + Static
- Static Power Consumption (Leakage Power) became a significant issue

Leakage Power

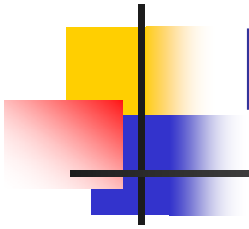
- Gate-oxide leakage
  - Gate tunneling due to thin oxide
- Subthreshold leakage
  - Scaling down of V_{th}
  - Short-channel effect
  - Our research focus





Outline

- Introduction
- Previous work
- Sleepy Keeper
- Experimental Methodology.
- Experimental Results
- Conclusion

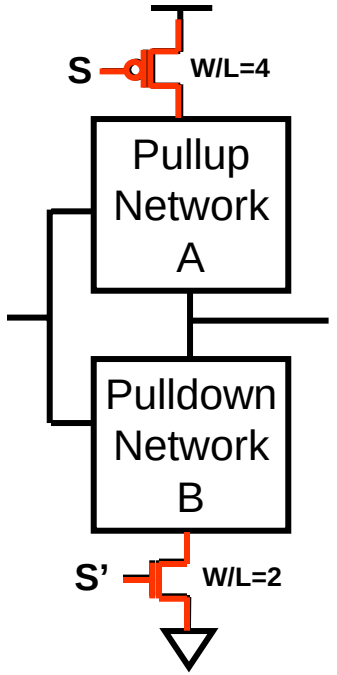


Previous Work

- Sleep
- ZigZag
- Stack
- Sleepy-Stack
- Leakage Feedback

Approaches : Sleep

- Source gating
- State destructive
  - Floating output
- Additional routing
- Dual V_{th} applicable

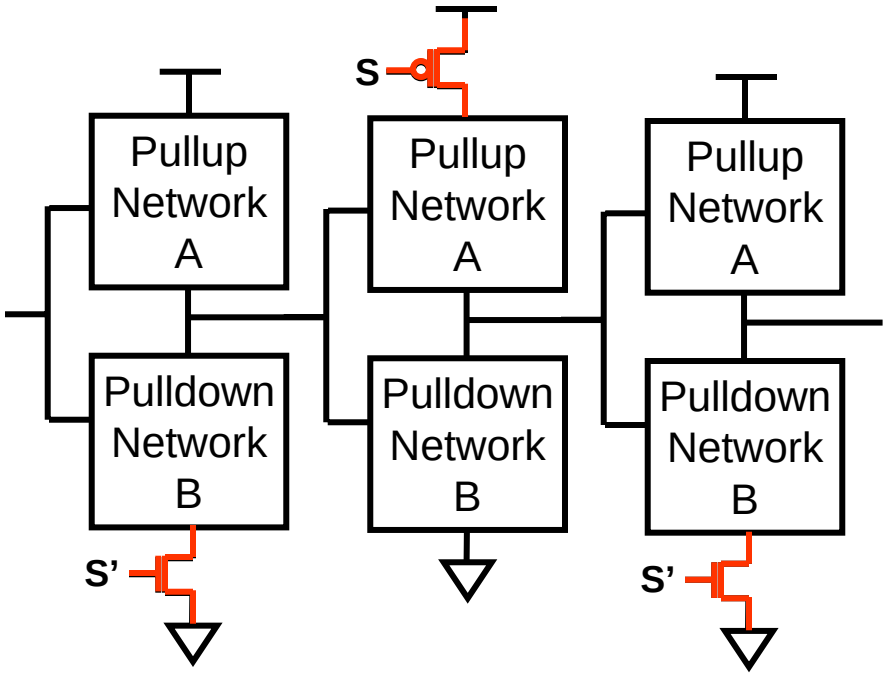


Approaches : Zigzag

- Favored input vector

  →Reduced wake-up overhead than sleep approach

- State destructive
- Dual V_{th} applicable



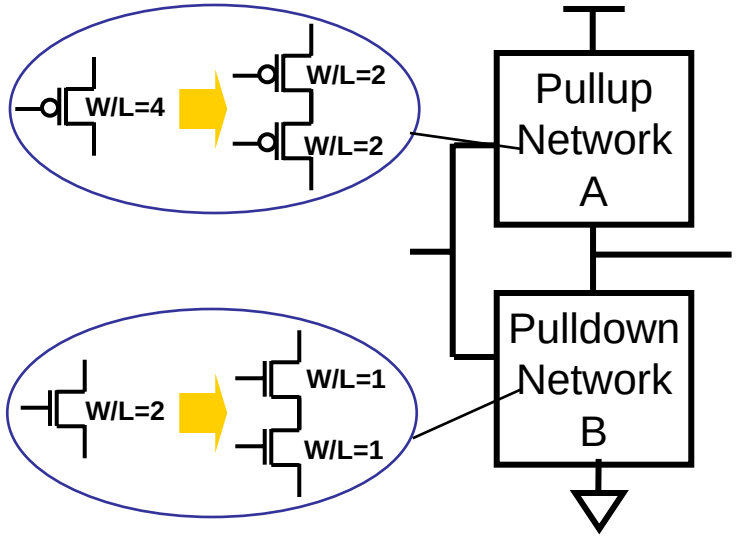
Approaches : Stack

- Duplicated transistors

  Induce reverse bias in cutoff
- State-Saving
- Delay penalty

  greater gate capacitance

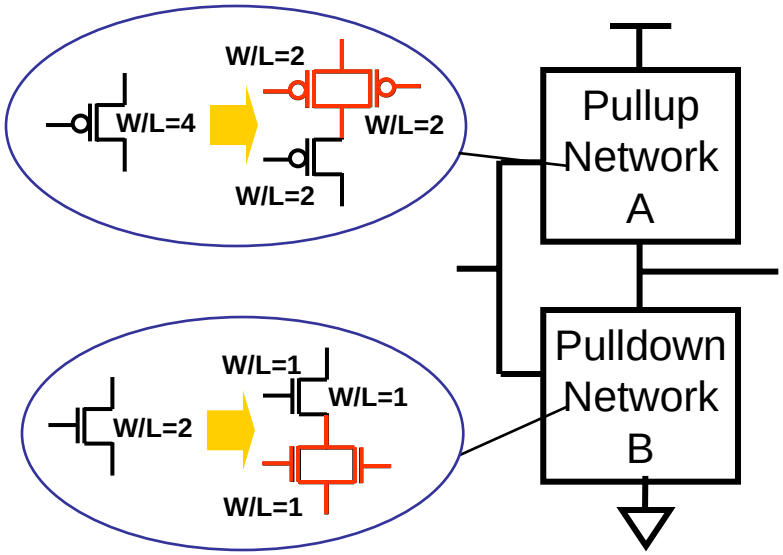
  greater resistance



Approaches : Sleepy-Stack

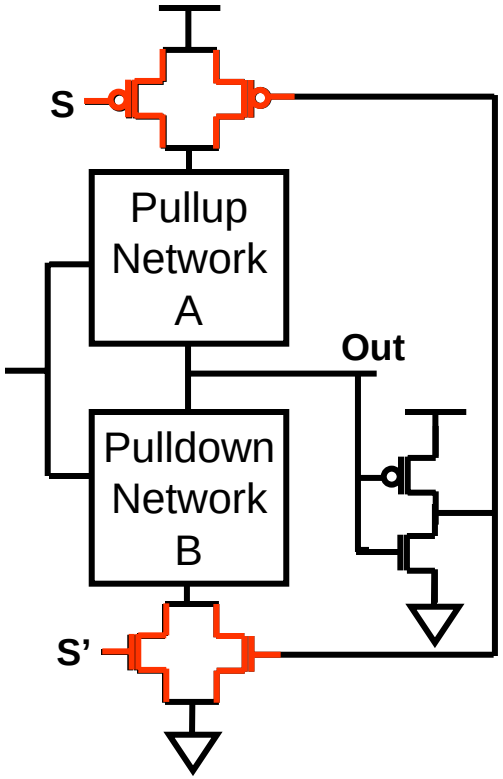
- Combination of Sleep and Stack

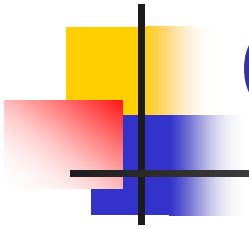
  Source gating, Stack effect
- State-saving
- Ultra-low leakage
- Area penalty



Approaches : Leakage Feedback

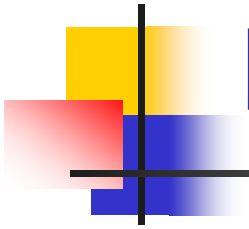
- Based on Sleep approach
- State-saving
- Leakage in inverter
- Area penalty





Outline

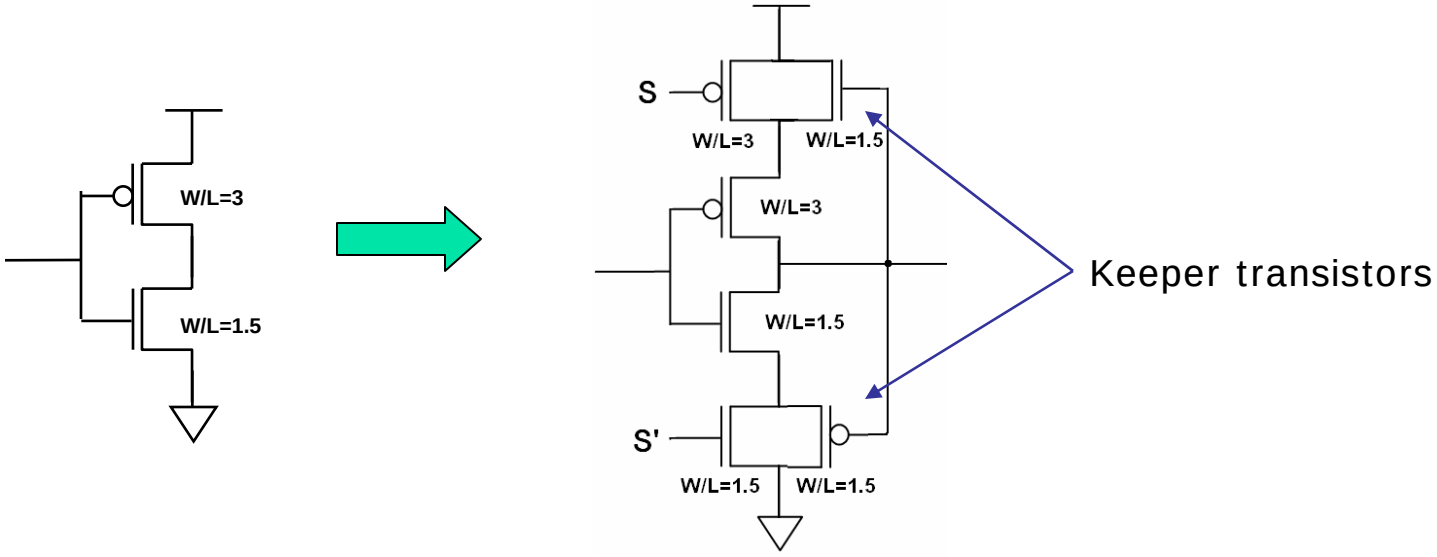
- Introduction
- Previous work
- Sleepy Keeper
- Experimental Methodology
- Experimental Results
- Conclusion



Motivation of Sleepy Keeper

- Ultra low leakage with dual V_{th}
- State-saving
- Less area penalty and faster than sleepy stack approach

Sleepy Keeper Structure



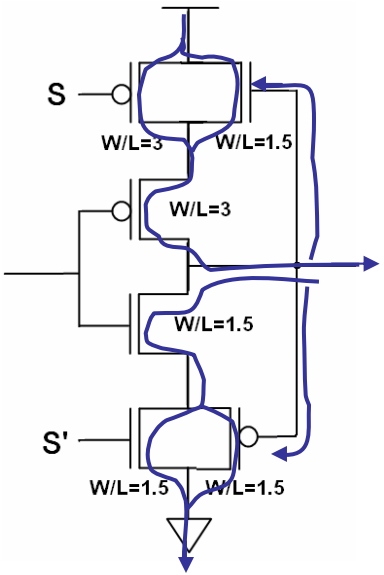
Conventional CMOS inverter

Sleepy keeper inverter

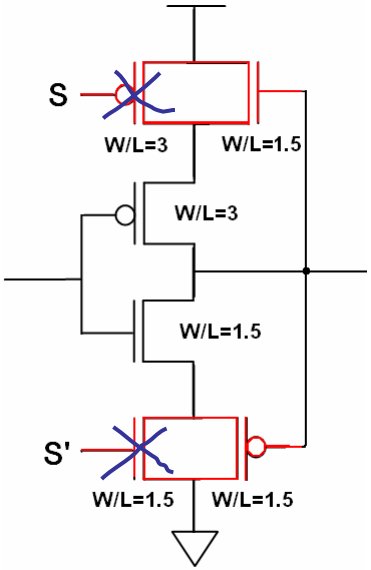
- Add sleep transistors
- Add two transistors driven by output

  NMOS to Pull-up Network, PMOS to Pull-down Network

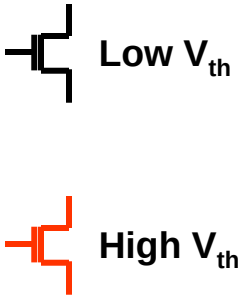
Sleepy Keeper Operation



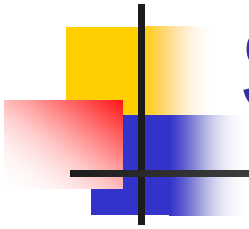
Active mode



Sleep mode



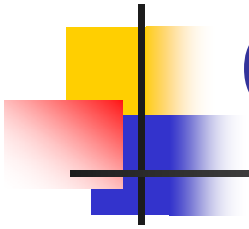
- During active mode, sleep transistors are on
  → reducing delay
- During sleep mode, sleep transistors are off
  → saving state
- Can apply dual V_{th}



Sleepy Keeper Operation

- Assumptions
  - Small delay between active mode and sleep mode
- Keeper transistors
  - keeper transistors are not for switching

    → lower voltage can be applied to maintain state



Outline

- Introduction
- Previous work
- Sleepy Keeper
- Experimental Methodology
- Experimental Results
- Conclusion

Experimental Methodology

- Seven techniques are compared

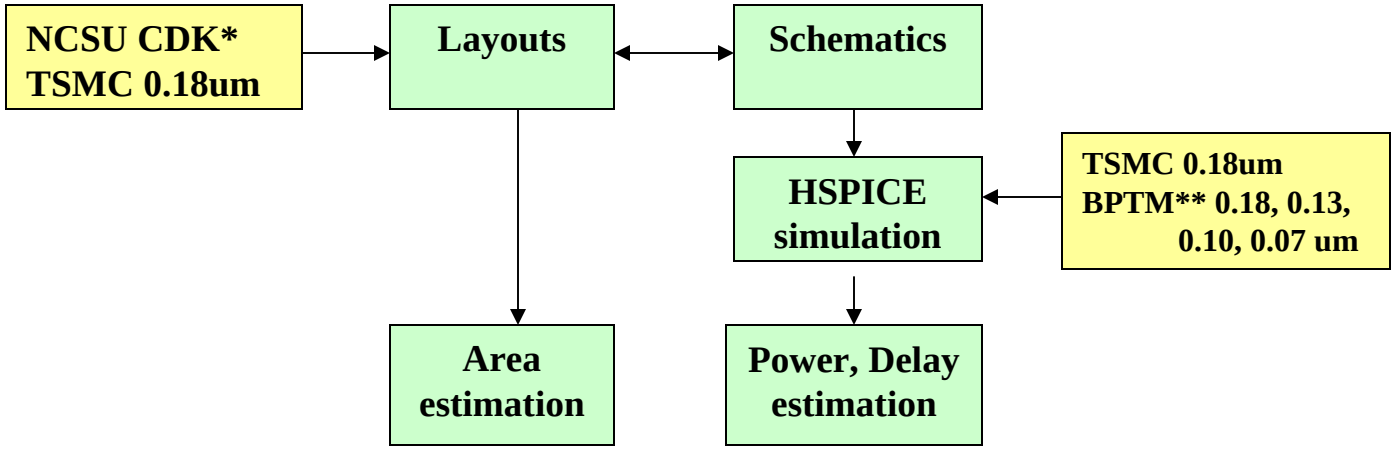
  base case, forced stack, sleep, zigzag, sleepy stack, leakage feedback, and sleepy keeper

- Dual-V_{th} applied for sleep, zigzag, sleepy stack, leakage feedback, and sleepy keeper

  Dual-V_{th} applied (0.2V and 0.4V)

Experimental Methodology

- Worst-case propagation delay, static power and dynamic power for each approach measured
- Area estimated by scaling down 0.18um layout

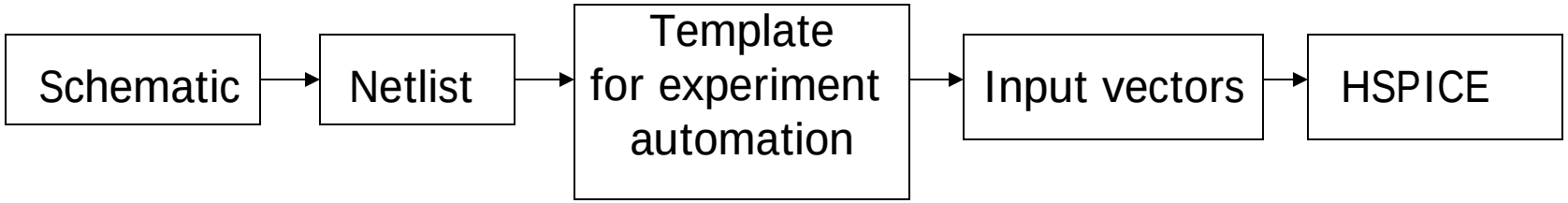


*NC State University Cadence Design Kit [Online]. Available http://www.cadence.ncsu.edu.

**Berkeley Predictive Technology Model (BPTM) [Online]. Available http://www.eas.asu.edu/~ptm.

Experimental Methodology

- Power, delay estimation



- Area estimation

  Scaled by ratio of squares & 10% overhead for nonlinear scaling layers

  Ex) 100um^2 in TSMC 0.18um

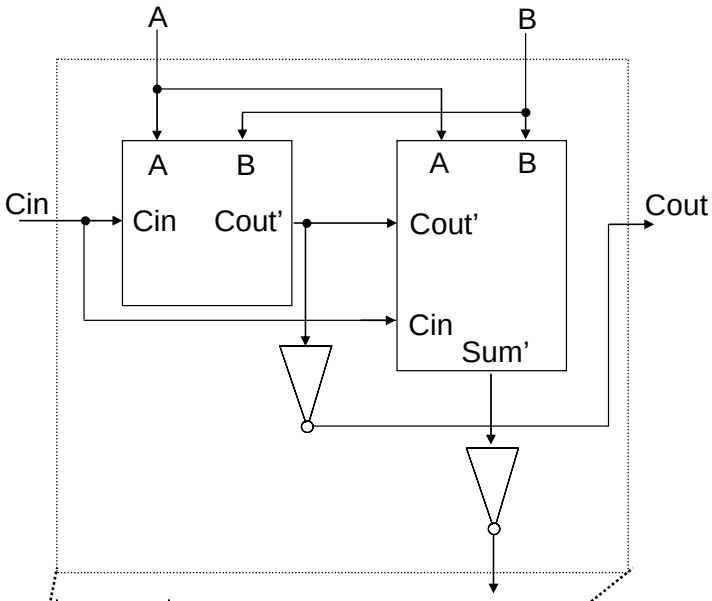
  For 0.10um => 100 * (0.10^2/0.18^2) * 1.1

Test circuit (4-bit adder chain)

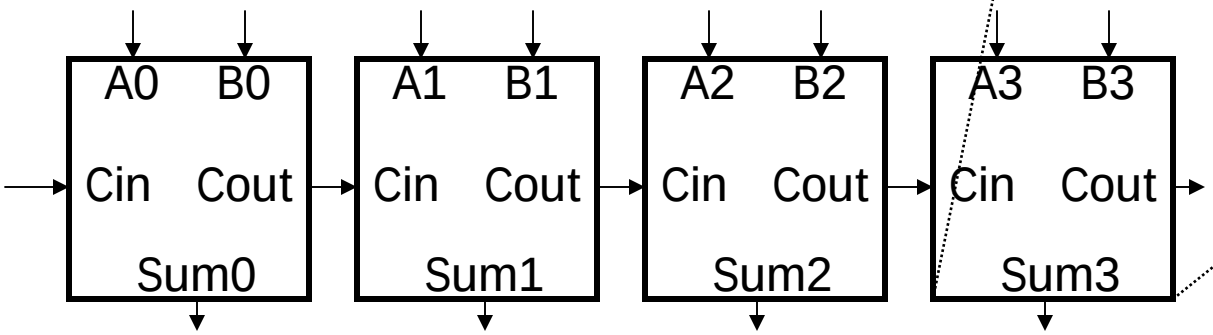
- 1-bit adder

  Cout = AB + Cin(A+B)

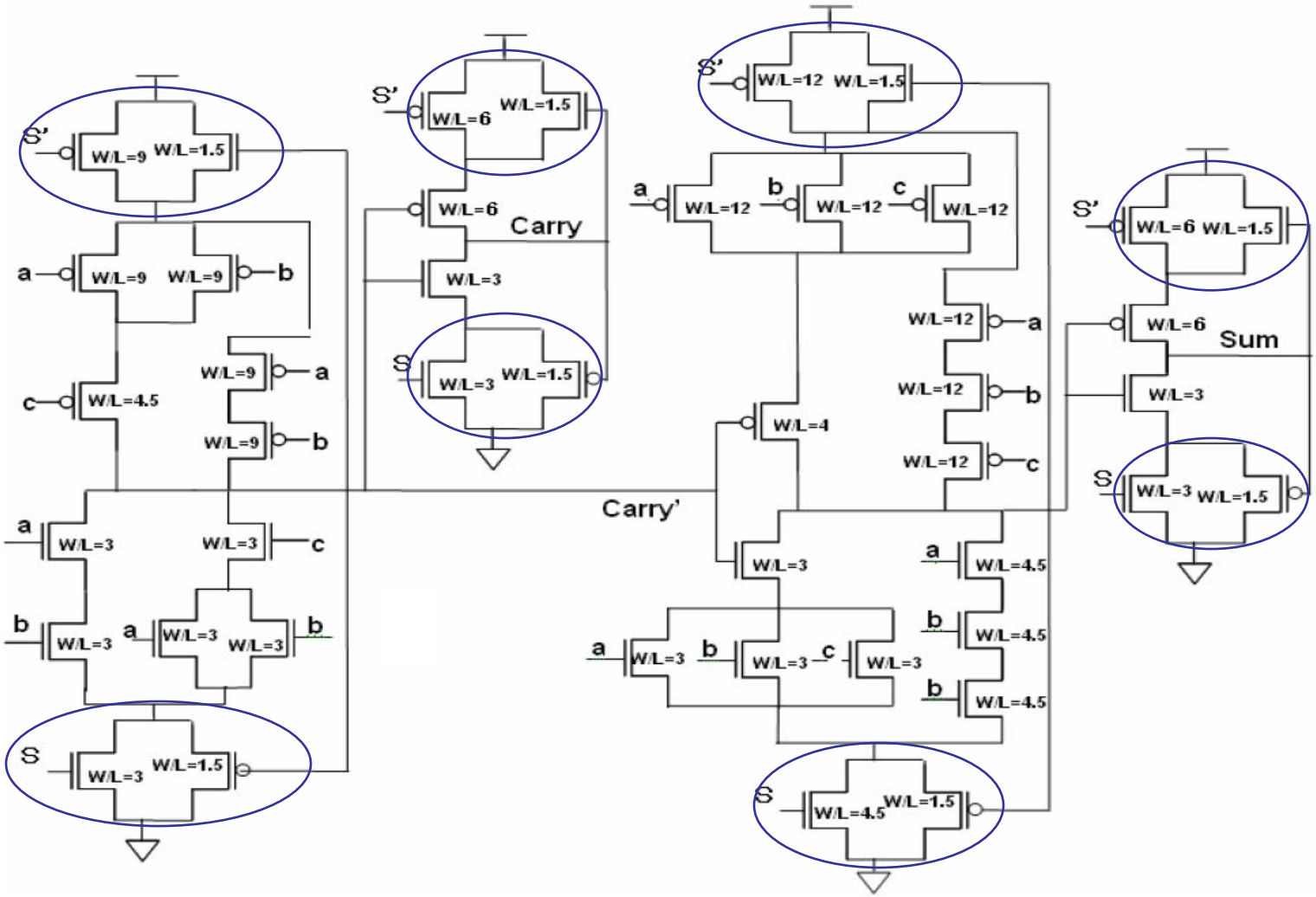
  Sum = ABCin+(A+B+Cin)Cout'

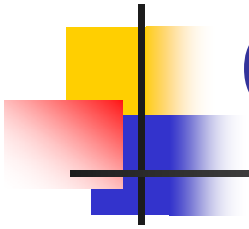


- 4-bit adder chain



1-bit adder schematic

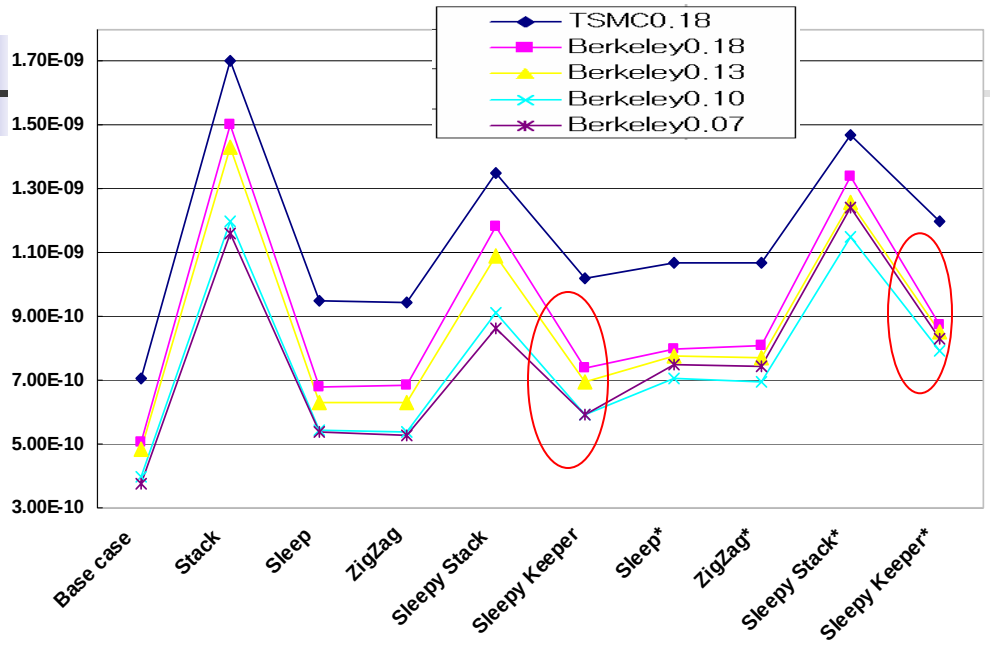




Outline

- Introduction
- Previous work
- Sleepy Keeper
- Experimental Methodology
- Experimental Results
- Conclusion

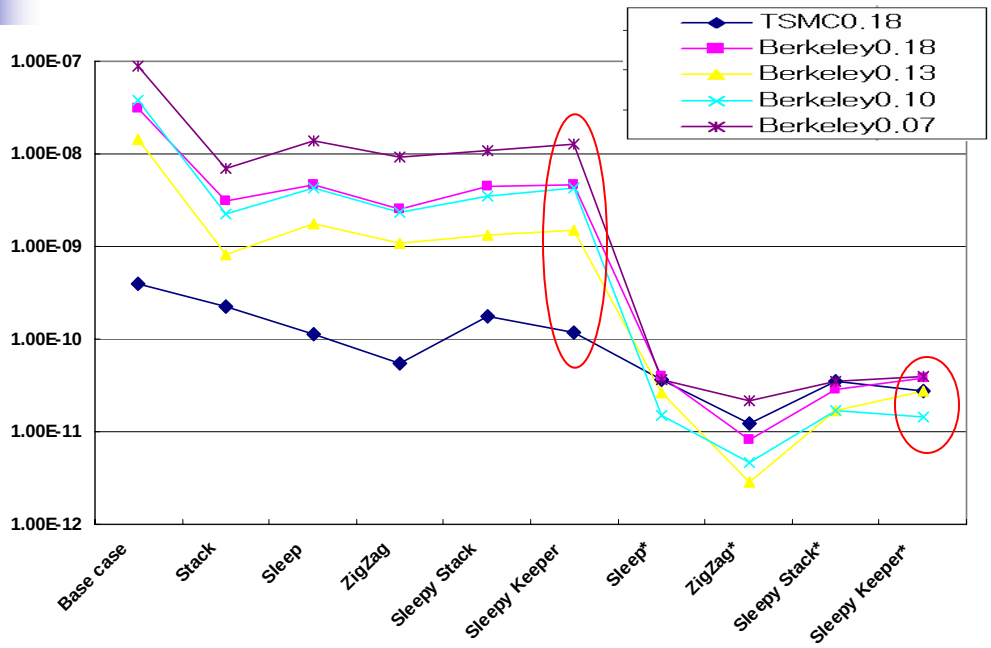
4-bit adder results – Propagation Delay



0.07µ	
Base case	3.76E-10
Stack	1.16E-09
Sleep	5.38E-10
ZigZag	5.25E-10
Sleepy Stack	8.64E-10
Sleepy Keeper	5.90E-10
Sleep*	7.52E-10
ZigZag*	7.43E-10
Sleepy Stack*	1.24E-09
Sleepy Keeper*	8.30E-10

- Compared mainly to sleepy stack (best prior leakage control technique while state saving)
- Sleepy keeper results 46% less delay than sleepy stack (49% less when dual Vth)
- Reason : Less number of transistors than sleepy stack

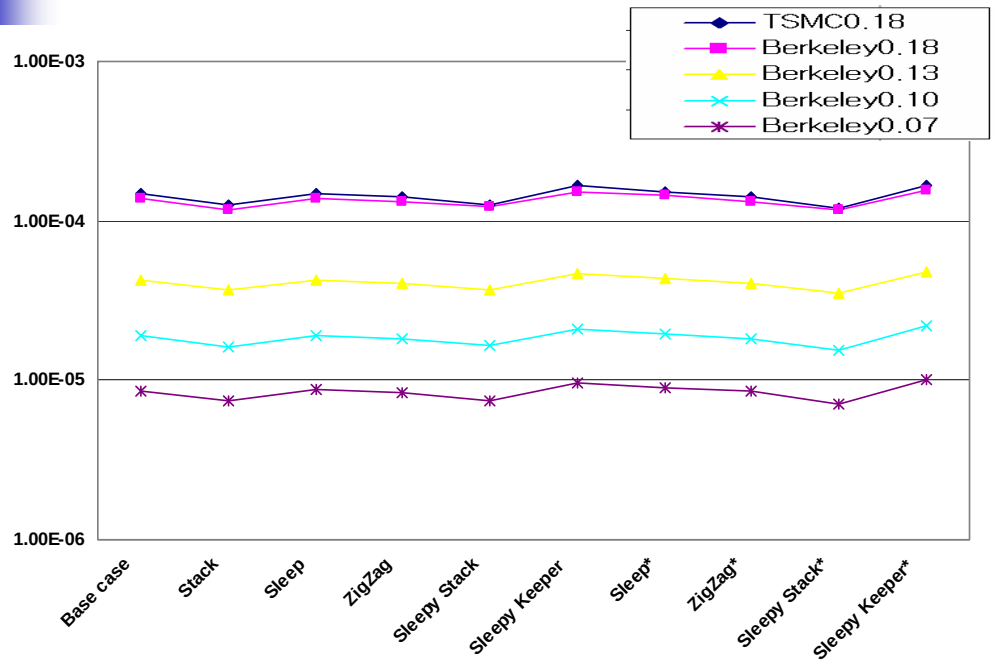
4-bit adder results – Static Power



0.07µ	
Base case	8.90E-08
Stack	6.83E-09
Sleep	1.36E-08
ZigZag	9.09E-09
Sleepy Stack	1.08E-08
Sleepy Keeper	1.30E-08
Sleep*	3.65E-11
ZigZag*	2.19E-11
Sleepy Stack*	3.50E-11
Sleepy Keeper*	3.89E-11

- Compared to stack, sleepy keeper reduce leakage power 175X
- Sleepy keeper results 20% more static power than sleepy stack (11% more when dual V_{th})

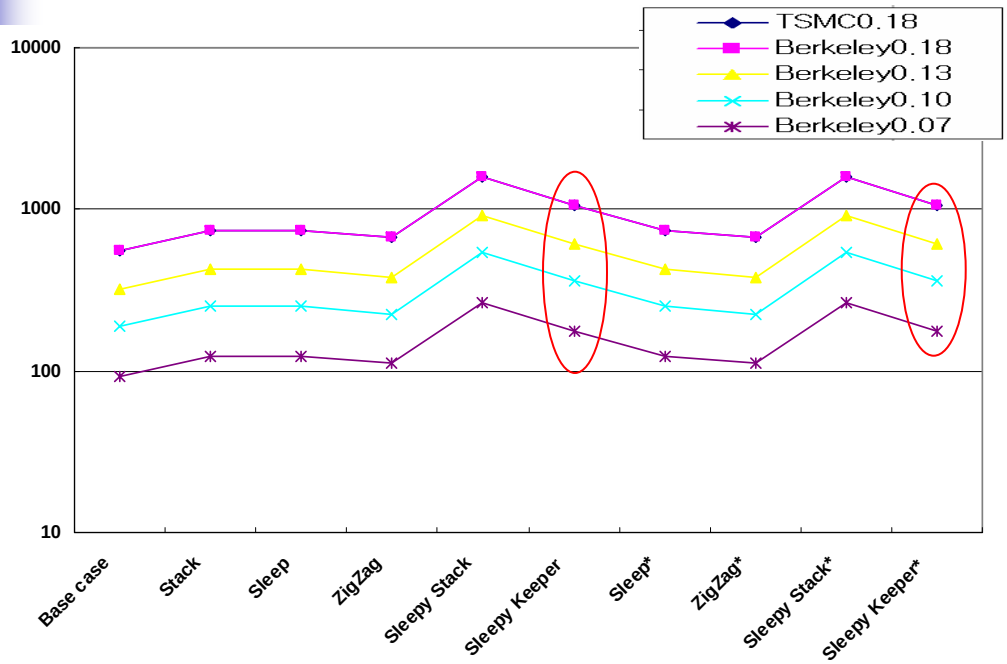
4-bit adder results – Dynamic Power



0.07µ	
Base case	8.63E-06
Stack	7.41E-06
Sleep	8.77E-06
ZigZag	8.37E-06
Sleepy Stack	7.39E-06
Sleepy Keeper	9.76E-06
Sleep*	9.03E-06
ZigZag*	8.46E-06
Sleepy Stack*	7.06E-06
Sleepy Keeper*	1.00E-05

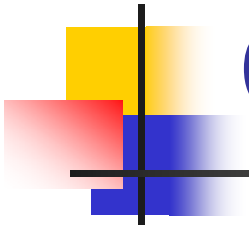
- Compared to sleepy stack, sleepy keeper results 31% more dynamic power (41% more when dual Vth)
- Conjecture : more short circuit current

4-bit adder results – Area



0.07μ	
Base case	91.84
Stack	123.76
Sleep	123.76
ZigZag	110.48
Sleepy Stack	263.52
Sleepy Keeper	177.11

- 93% larger than base case, 49% smaller than sleepy stack
- Reason : Sleep transistors, additional 2 transistors, and unusual placement of keeper transistor



Conclusion

- Ultra low static power with dual V_{th}
- State saving
- Less area, less delay than sleepy stack
- Dynamic power increased