

# Fine-Grained Supply Gating Through Hypergraph Partitioning and Shannon Decomposition for Active Power Reduction

Lawrence Leinweber and Swarup Bhunia  
Electrical Engineering and Computer Science



CASE WESTERN RESERVE  
UNIVERSITY  
CASE SCHOOL OF ENGINEERING

# Outline

- **Introduction**
- **Background and Motivation**
- **Design Flow**
  - **Shannon Decomposition**
  - **Partitioning Algorithm**
  - **Cost Functions**
  - **Annealing Algorithm**
- **Results**
- **Conclusion**
- **Questions**

# Introduction

- **Power – The Major Challenge**
- **Causes**
  - **Smaller Transistors**
  - **Higher Clock Speeds**
- **Effects**
  - **Energy Consumption Limits Battery Apps**
  - **Power Density in High-Performance Apps**
    - **Higher Junction Temperatures**
    - **Reliability Issues**

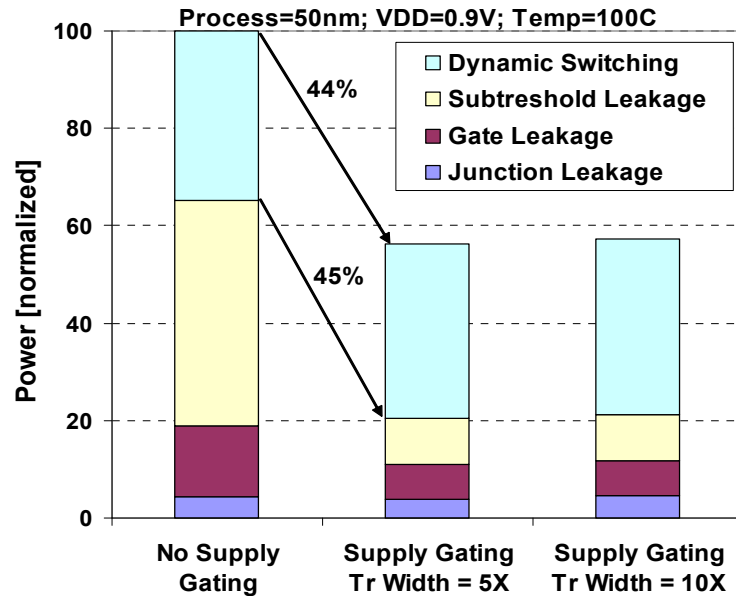
# Power Dissipation in Logic

- **Dynamic Power**
  - **Operating Voltage and Frequency**
  - **Switching Activity per Clock Cycle**
  - **Switching Capacitance**
- **Active Leakage Power**
  - **Gate Leakage**
  - **Band-to-Band Junction Tunneling Leakage**
  - **Subthreshold Leakage**
    - **Exponential Dependence on Temperature**

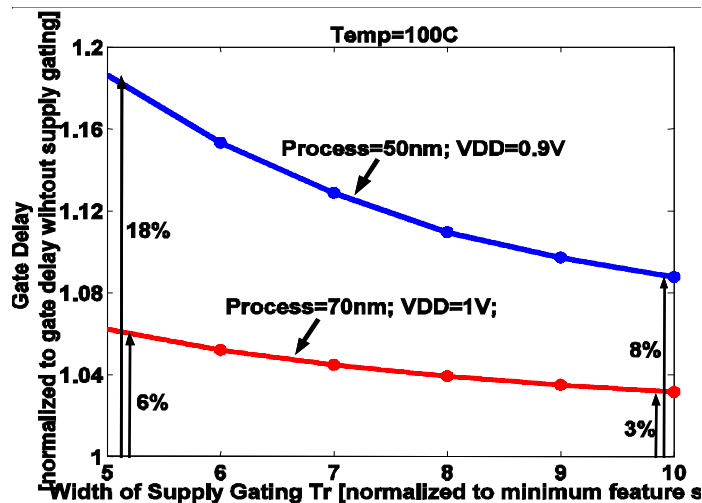
# Power Reduction Techniques

- **Dynamic Power Reduction**
  - **Static or Dynamic Voltage Scaling**
  - **Frequency Scaling**
  - **Clock Gating**
- **Active Leakage Reduction**
  - **Dual- $V_{th}$  Design**
    - **Requires Fabrication of High- and Low- $V_{th}$  Transistors on the Same Die**
    - **Increased Number of Timing-Critical Paths**
  - **Supply Gating**

# Supply Gating Power Reduction



- Power Reduction with Supply Gating for 50 nm



- Corresponding Impact on Delay

# Supply Gating Ideas

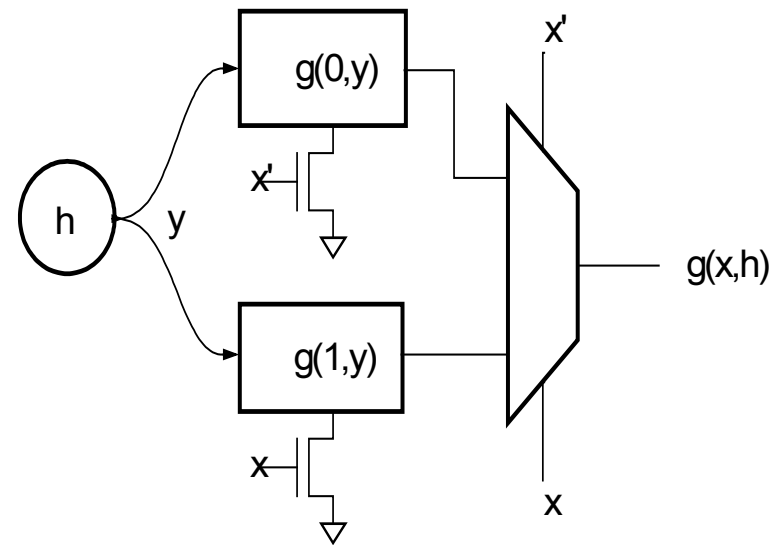
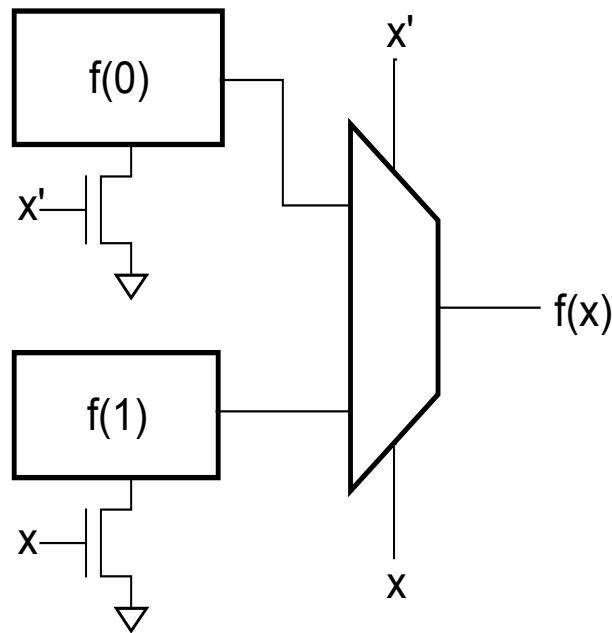
- **Stacking Transistors**
  - **Reverse Bias to Reduce Subthreshold Leakage**
  - **Prevent Switching to Save Dynamic Power**
- **Supply Gating Whole Modules, e.g., an ALU**
- **Fine-Grained Supply Gating – Bhunia, et al.**
  - **Cofactor Logic by Shannon Decomposition**
  - **Multi-Level Application of Technique**
- **Fine-Grained Supply Gating – Bhunia and Me**
  - **Shannon Decomposition Needs Small Circuits**
  - **Find Clusters by Hypergraph Partitioning**
  - **Subdivide Large Circuits**
  - **Design Methodology and Synthesis Tool**

# Background and Motivation

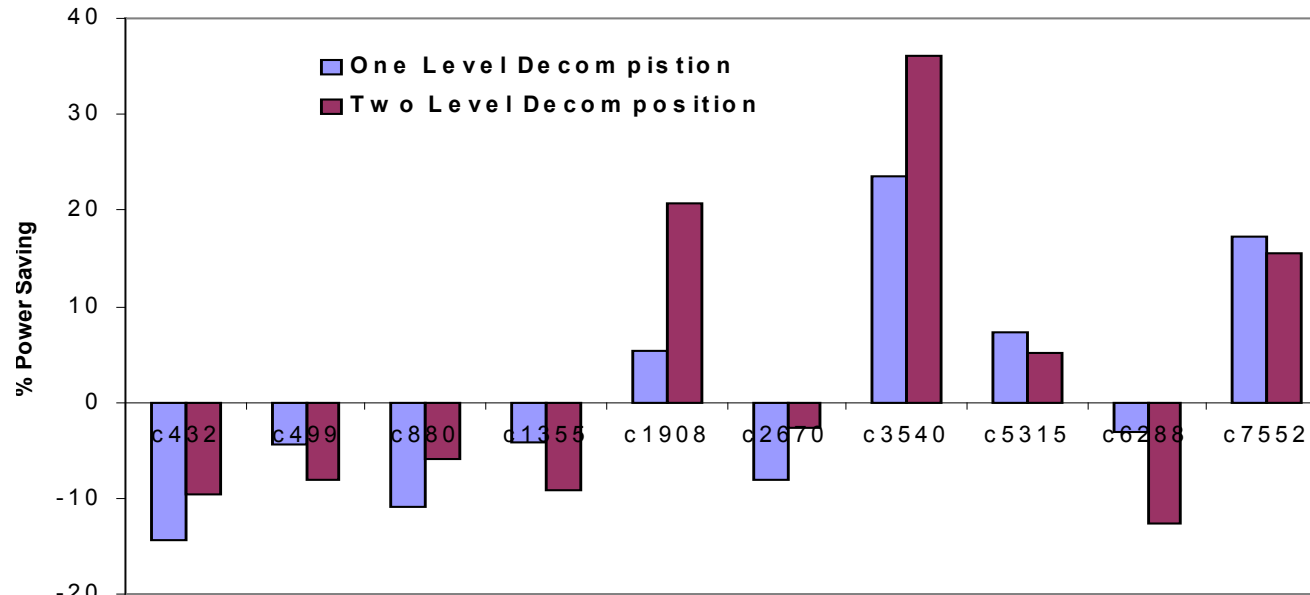
- **Shannon Decomposition; Factoring Shared Logic**

$$f(x) = f(0) \cdot \bar{x} + f(1) \cdot x$$

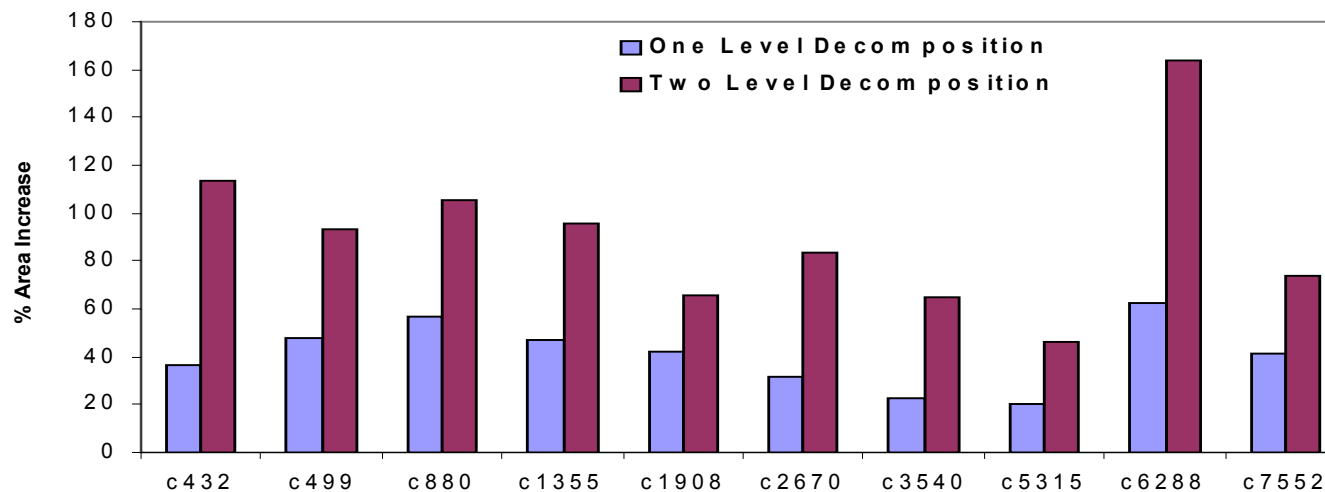
$$g(x, h) = g(0, h) \cdot \bar{x} + g(1, h) \cdot x$$



# Shannonized ISCAS-85 Circuits



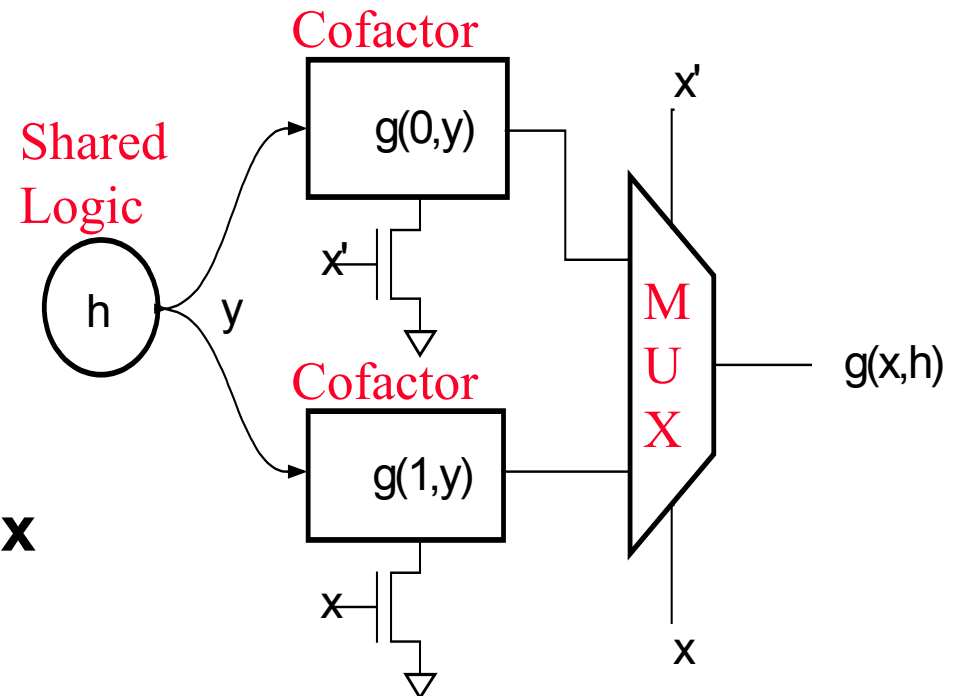
• % Power Saving



• % Area Increase

# Problems with Large Circuits

- **Not Cohesive w.r.t. a Single “Control” Variable ( $x$ )**
  - **Very Small Cofactors**
- **Collapsing Logic to Two Levels Becomes Exponentially Complex ( $2^n$ )**
  - **Much Logic Appears in Both  $f(0)$  and  $f(1)$**
- **If Many Outputs, Multiplexer Overhead Dominates**



# Hypergraph Partitioning

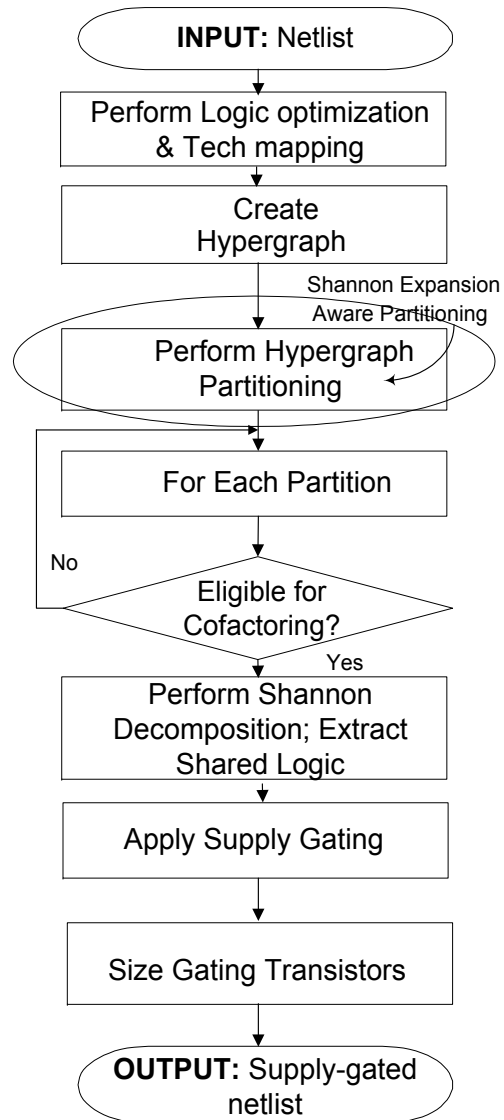
- **Apply Shannon Decomposition to Partitions**

$$o_1 = x \cdot y \cdot z + \bar{x} \cdot y \cdot \bar{z}$$

$$o_2 = p \cdot q \cdot \bar{r} + \bar{p} \cdot q \cdot r$$

- **Hypergraph  $H = (V, E^h)$** 
  - **Set of Vertices,  $V$ , (Gates)**
  - **Set of Hyperedges,  $E^h$ , (Wires)**
    - Each Hyperedge is a Subset of  $V$
  - **Represents the Topology of a Circuit**
- **Shannon Decomposition Aware Partitioning**
  - **Cluster Logic for the Benefit of Power Saving by Shannon Decomposition**

# Design Flow

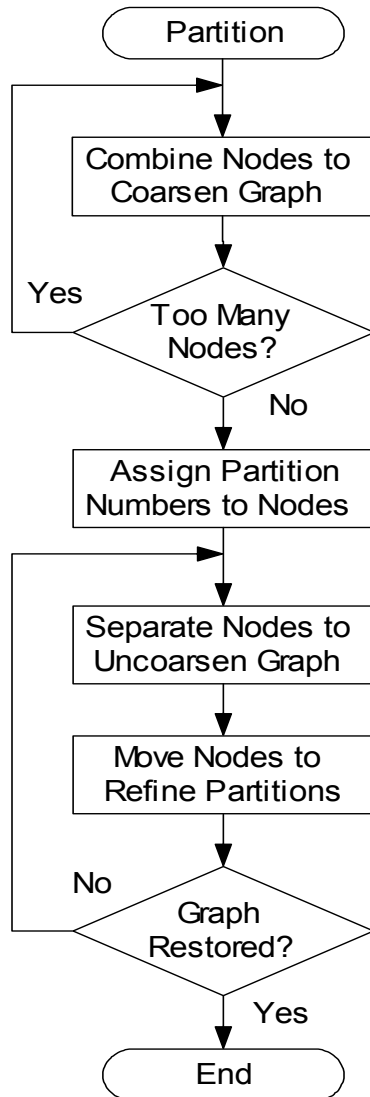


- **Tech Mapping and Partitioning**
- **Control Variable Selection**
  - **Test Cofactor Simplification**
- **Extract Shared Logic**
  - **Find Common Cubes**
  - **Trace Output Logic Cones**
- **Size Gating Transistors to  $N\lambda$** 
  - **Based on  $5\lambda$  Inverter Size**
- **Apply Recursively to Two Levels**
  - **Cofactors and Shared Logic**

# Shannon Decomposition

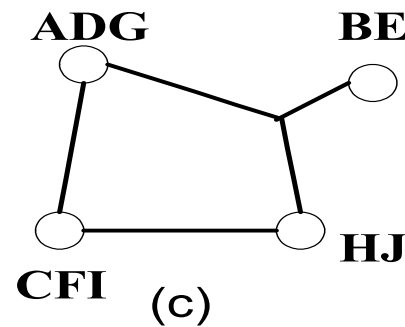
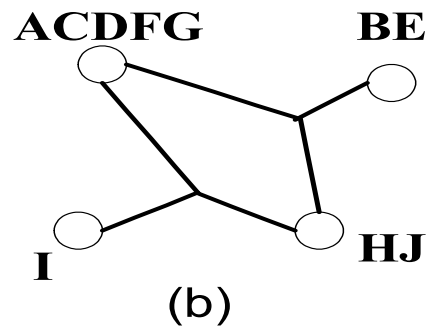
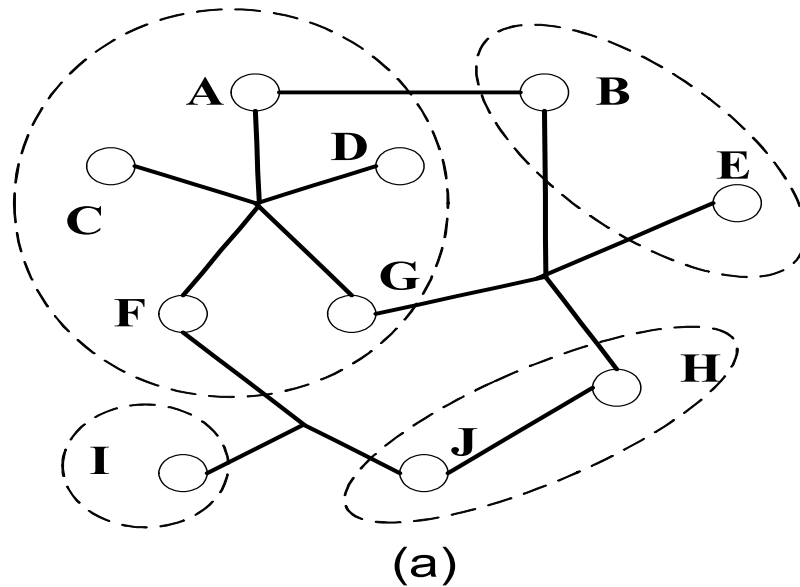
- **Use Multi-Level Logic and not Two-Level Logic**
- **Two Rounds of Control Variable Selection**
  - 1. For Each Input Variable**
    - **Separate Cofactors**
    - **Perform Constant Propagation**
    - **Count Gates**
  - 2. For 3 Best of 1<sup>st</sup> Round**
    - **Fully Synthesize**
    - **Least Gate Count Wins**
- **Perform Common Subexpression Elimination to Find Shared Logic**

# Partitioning Algorithm



- **Coarsening - Simplification**
  - **Draw Nodes Together**
- **Recursively Until Optimal Number of Vertices per Partition**
- **Uncoarsening – Restore Hypergraph**
  - **Refine Partitions**
  - **Try Moving Nodes to Lower Cost**
  - **At Higher Levels, Each Node Represents Many Gates**

# Partitioning Example



- **Coarsening**
  1. ACDFG
  2. HJ
  3. BE
  4. I

- **Uncoarsening**
  1. F
  2. C

# Cost Functions

- **Cost of All Vertices in a Partition**

$$cost(nvert) = (vopt / nvert)^2 + (nvert / vopt)^2$$

- ***nvert*** – Number of Vertices in Partition

- ***vopt*** – Optimal Number of Vertices in a Partition

- **Cost of Each Edge:**

$$cost = kvpe \cdot (K - 1) / vopt$$

- ***K*** – Number of Partitions Edge Straddles

- ***kvpe*** – Cost of Vertex Relative to Edge = 10

- **Total Cost of Hypergraph = Vertex Cost for Each Partition + Sum of Edge Costs**

# Annealing Strategy

$$x = (newcost - currcost) / currcost$$

$$delta = 1 / \sqrt{ntries}$$

$$var = delta \cdot (x^2 - oldvar) + oldvar$$

$$zscore = x / \sqrt{var}$$

$$prob = 0.5 \cdot (ntries - n) / ntries$$

$$probit = 1.25 \cdot prob$$

$$r < prob \frac{zscore}{probit}$$

- **x** – normalized cost point
- **ntries** – algorithm run-time
- **delta** – time constant
- **var** – variance of x points
- **zscore** – Gaussian z-score
- **n** – iterations of algorithm
- **prob** – probability of accepting new partitioning
- **probit** – inverse Gaussian Cumulative Density Fn
- **r** – random number  $0 \leq r < 1$

# Test Setup – Baseline Conditions

- **Reference Netlist Pre-Synthesized with SIS**
- **Predictive 32 nm Technology**
- **0.9 V power supply**
- **Fan-Out-of-Four Invert Loads on Each Output**
- **5 ns Cycle Times (200 MHz Clock Frequency)**
- **1000 Cycles Per Test**
- **20% Input Activity Level**
- **20 ps Rise and Fall Times**
- **110° C**
- **Two Levels of Shannon Recursion**
- **Reject 50% Increase in Area and Est. Power**

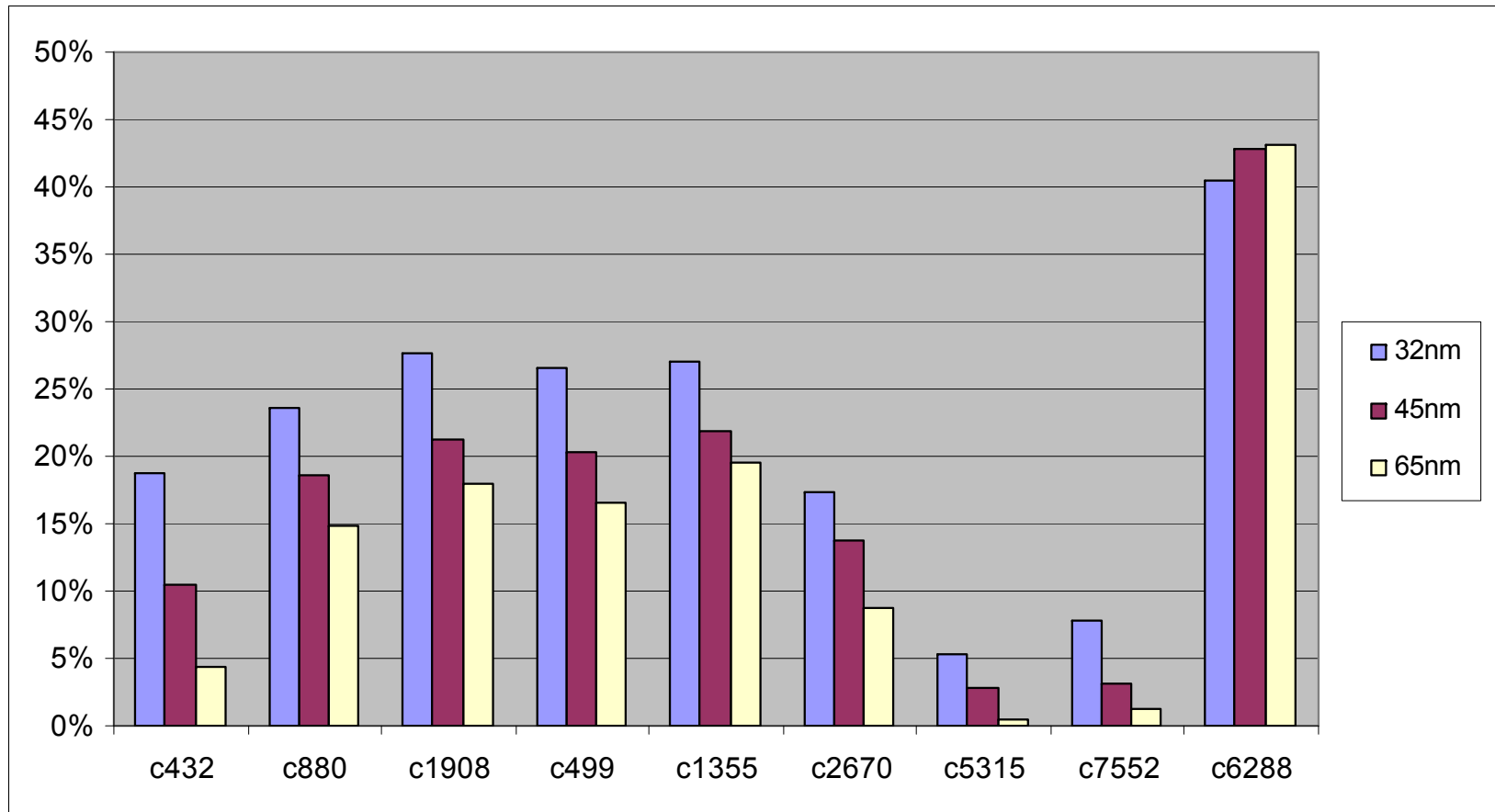
# One-Level Decomposition

<b>Circuit</b>	<b>Power (%)</b>	<b>Area (%)</b>	<b>Delay (%)</b>	<b><i>V<sub>opt</sub></i></b>
c432	<b>19.8</b>	-5.4	-1.5	11
c880	<b>22.2</b>	1.9	9.6	15
c1908	<b>25.6</b>	16.5	2.8	10
c499	<b>20.5</b>	23.7	-9.8	6
c1355	<b>20.5</b>	22.8	-7.6	7
c2670	<b>15.2</b>	5.3	11.8	30
c5315	<b>4.8</b>	4.8	5.0	16
c7552	<b>5.4</b>	-4.7	-10.3	22
c6288	<b>42.0</b>	16.0	10.7	5
Avg.	<b>19.6</b>	9.0	1.2	

# Two-Level Decomposition

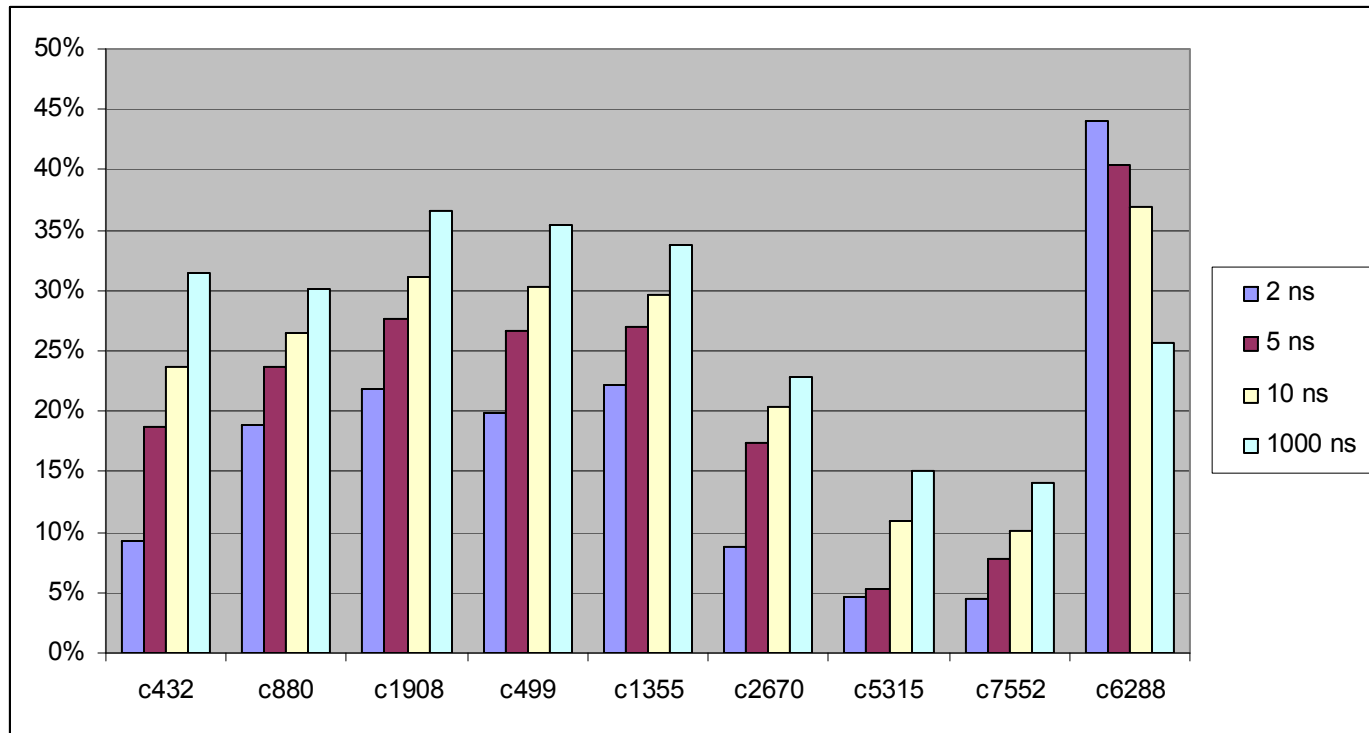
Circuit	Power (%)	Area (%)	Delay (%)	Total partitions	# of Shannon decomposed partitions		Execution Time (sec.)
					L = 1	L = 2	
c432	<b>18.7</b>	-6.8	3.4	9	9	13	220.77
c880	<b>23.6</b>	2.3	13.7	22	22	29	38.78
c1908	<b>27.7</b>	24.3	9.0	52	51	38	50.11
c499	<b>26.6</b>	34.6	4.0	57	57	37	44.95
c1355	<b>27.0</b>	35.7	0.6	61	58	34	46.88
c2670	<b>17.4</b>	12.0	19.1	81	73	62	129.52
c5315	<b>5.3</b>	6.0	13.8	108	105	140	335.37
c7552	<b>7.8</b>	-3.2	-2.8	97	89	139	434.62
c6288	<b>40.4</b>	14.7	9.1	469	427	87	613.46
Avg.	<b>21.6</b>	13.3	7.8				

# Effect of Technology Node



- **By Going to a Scaled Technology We Can Save More Power Since Leakage Power Increases With Scale**

# Effect of Cycle Time



- **Increasing Cycle Time Means Less Dynamic Power and More Leakage Power**
- **For 1000 ns Cycle Time, Power Saving Approximates Stand-By Power Saving**
- **Average Saving in Static Power: 27.2%**

# Conclusion

- **A Fine-Grained Supply Gating Methodology Is Presented, Leverages on**
  - **Efficient Partitioning**
  - **Extraction of Shared Logic**
  - **Efficient Shannon Decomposition**
- **We Observe Considerable Saving in Both Active and Stand-By Power**
  - **Little Impact on Area and Performance**
- **The Technique Can Be Automated and Integrated with Other Power Saving Methods**
- **The Proposed Technique is Scalable**
  - **Power Saving Improves with Technology Scaling**
- **Questions**