Lecture

14

ECE M216A

Energy-Delay Optimization

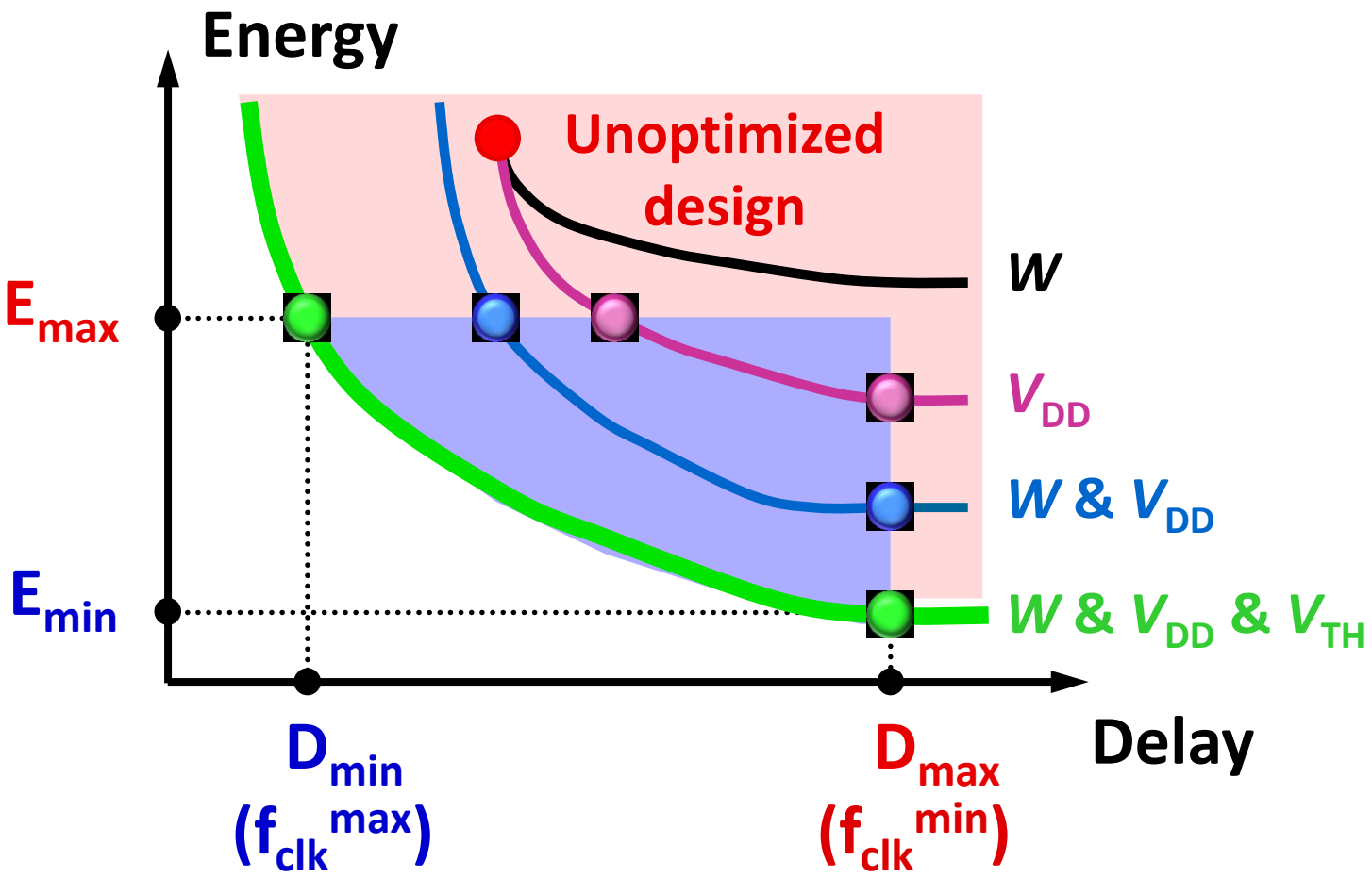
Prof. Dejan Marković

ee216a@gmail.com

Some Common Questions

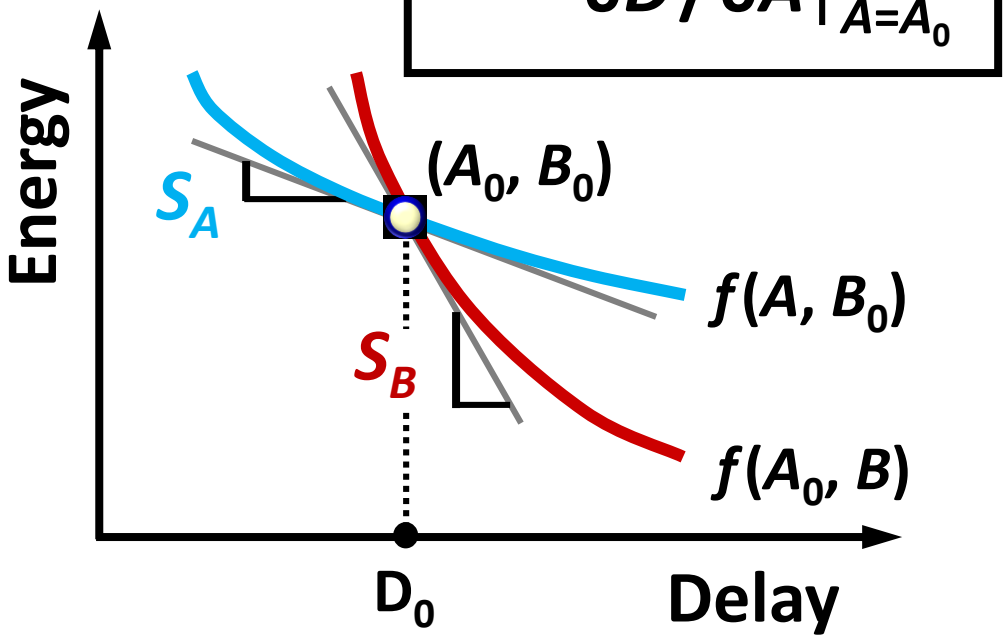
- Is sizing better than V_{DD} for energy reduction?
- Optimal values of gate size and V_{DD}?
- Increase or decrease V_{DD} for energy reduction?
- Optimal ratio of leakage / switching for min E?
- Optimal circuit topology?
- How many levels of parallelism?
- Etc.

Energy Minimization Problem



Energy-Delay Sensitivity

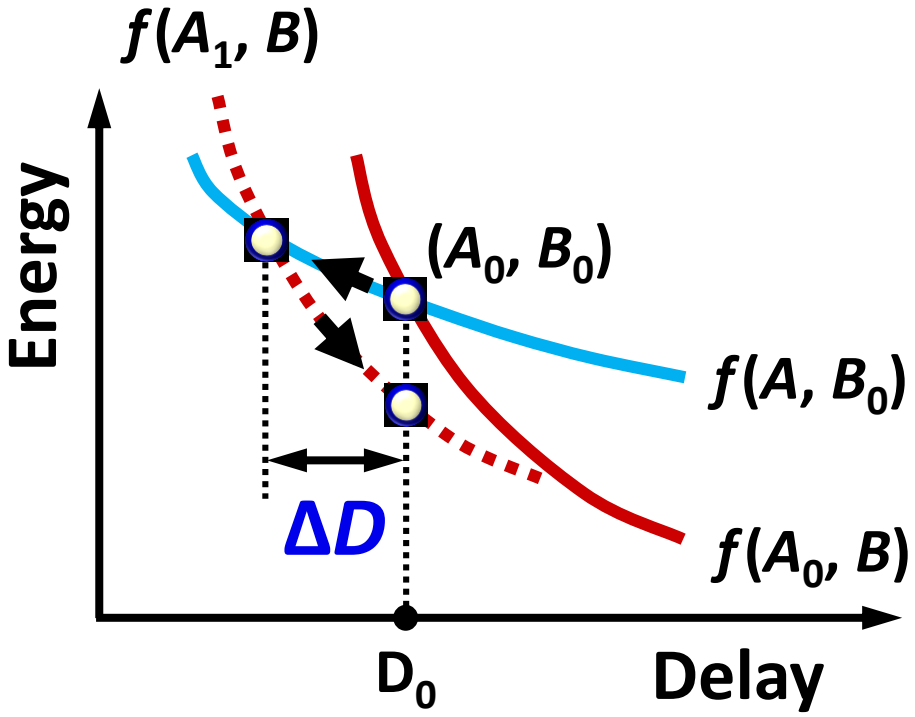
$$S_A = \left. \frac{\partial E / \partial A}{\partial D / \partial A} \right|_{A=A_0}$$



Slope of E-D curve around a design point (e.g. (A_0,B_0))

Solution: Equal Sensitivities

$$\Delta E = S_A \cdot (-\Delta D) + S_B \cdot \Delta D$$



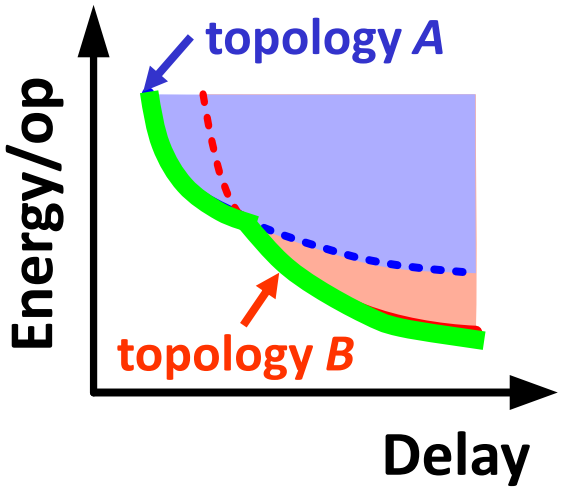
A fixed point is reached when all sensitivities are equal

Circuit Optimization

$$\begin{aligned} \text{minimize} \quad & Energy(V_{dd}, V_{th}, W) \\ \text{subject to} \quad & Delay(V_{dd}, V_{th}, W) \leq D_{con} \end{aligned}$$

Constraints

$$V_{dd}^{min} < V_{dd} < V_{dd}^{max}$$
$$V_{th}^{min} < V_{th} < V_{th}^{max}$$
$$W^{min} < W$$



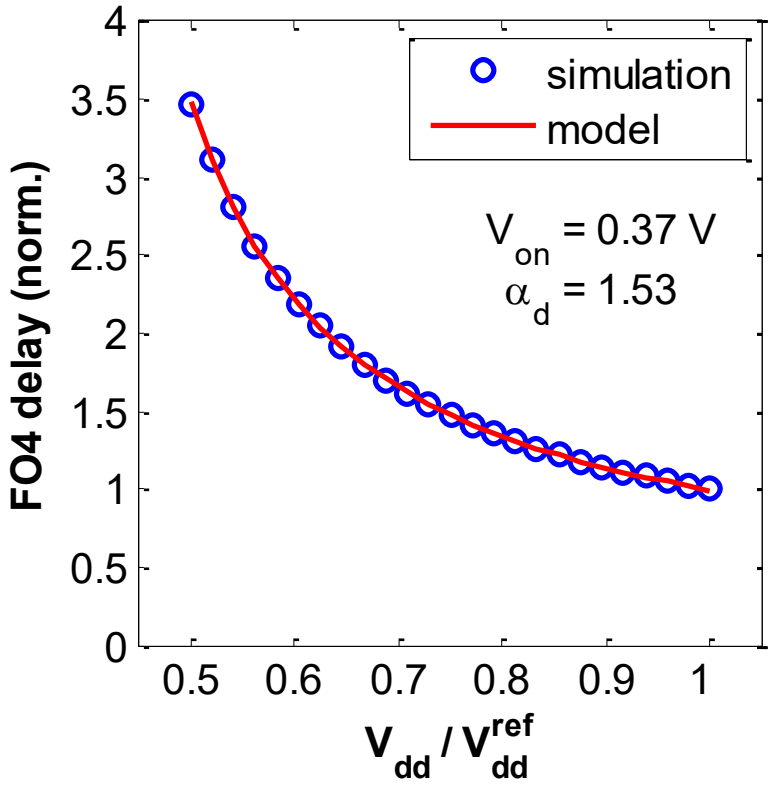
- **Reference design**
  - D_{min} sizing @ V_{DD}^{max}, V_{TH}^{ref}

Goal: find optimal *E-D* tradeoff for a logic function

Energy and Delay Models

Alpha-power based Delay Model

$$Delay = \frac{K_d \cdot V_{DD}}{(V_{DD} - V_{on} - \Delta V_T)^{\alpha_d}} \cdot \left(\frac{W_{out}}{W_{in}} + \frac{W_{par}}{W_{in}} \right)$$



Fitting parameters:

V_{on}, α_d, K_d

V_{DD}^{ref} = 1.2V

FO4 (V_{DD}^{ref}) = 25ps

(*) [Sutherland et al., Logical Effort, 1999]

Energy Model

- **Switching energy**

$$E_{sw} = \alpha_{0\to1} \cdot \left(C(W_{out}) + C\left(W_{par}\right) \right) \cdot {V_{DD}}^2$$

- **Leakage energy**

$$E_{lk} = \frac{W_{in}}{W_0} \cdot I_0(S_{in}) \cdot 10^{-\frac{V_T - \gamma_D V_{DD}}{S}} \cdot V_{DD} \cdot D$$

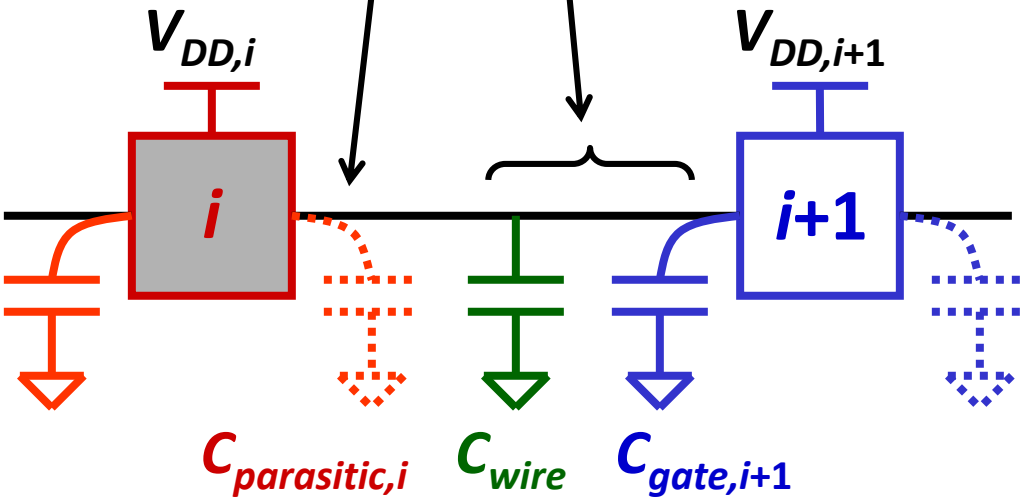
with:

D: the cycle time

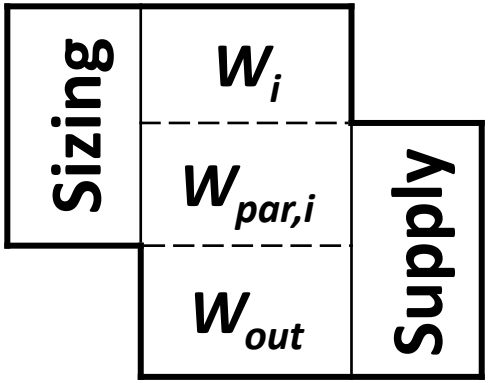
$I_0(S_{in})$: normalized leakage current with inputs in state S_{in}

Adjusting Switching Energy of a Gate

$$E = K_e \cdot (W_{par} + W_{out}) \cdot V_{DD}^2$$



Impact of opt variables



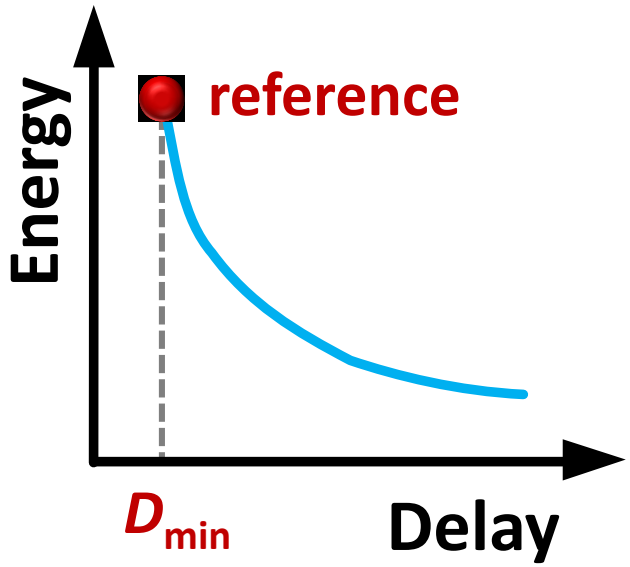
$$ec_i = K_e \cdot W_i \cdot (V_{DD,i-1}^2 + p_{i,ref} \cdot V_{DD,i}^2)$$

(energy stored on the logic gate *i*)

Optimization

Optimization Setup

- **Reference/nominal circuit**
  - Sized for D_{min} @ V_{DD}^{max}, V_T^{ref}
  - Known average activity
- **Define delay constraint**
  - $D_{con} = D_{min}\,(1 + d_{inc}/100)$
- **Minimize energy under delay constraint**
  - Gate sizing (W), optional buffering
  - V_{DD} and V_T scaling



Sensitivity to Sizing and Supply

- **Gate sizing (*W*)**

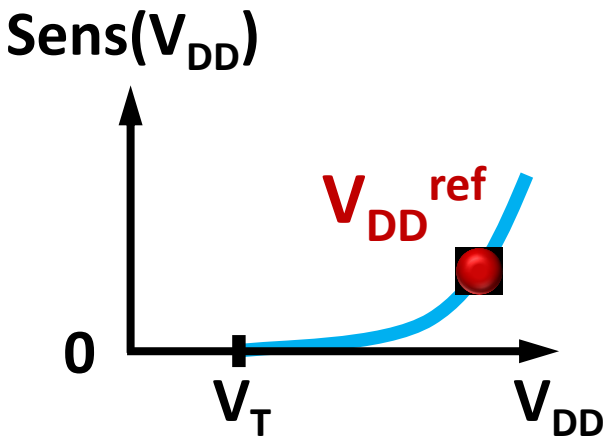
$$-\frac{\partial E_{Sw} / \partial W_i}{\partial D / \partial W_i} = \frac{ec_i}{\tau_{ref} \cdot \left(h_{eff,i} - h_{eff,i-1} \right)}$$

∞ for equal h_{eff} (D_{min})

- **Supply voltage (V_{DD})**

$$-\frac{\partial E_{Sw} / \partial V_{DD}}{\partial D / \partial V_{DD}} = 2 \cdot \frac{E_{Sw}}{D} \cdot \frac{1 - x_v}{\alpha_d - 1 + x_v}$$

$$x_v = \frac{V_{on} + \Delta V_{TH}}{V_{DD}}$$



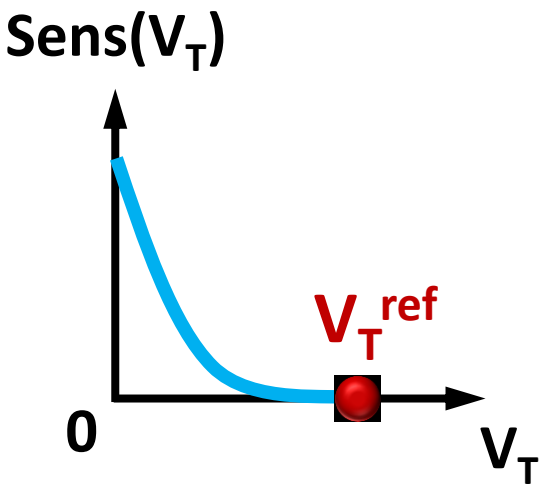
Sensitivity to Threshold Voltage

- **Threshold voltage (V_T)**

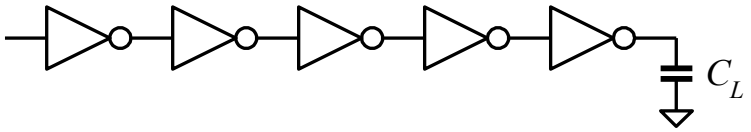
$$-\frac{\partial E / \partial(\Delta V_{TH})}{\partial D / \partial(\Delta V_{TH})} = P_{Lk} \cdot \left(\frac{V_{DD} - V_{on} - \Delta V_{TH}}{\alpha_d \cdot V_0} - 1 \right)$$

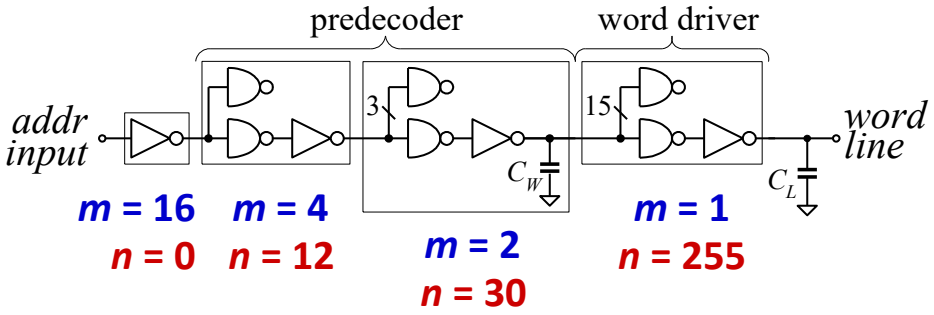
Low initial leakage

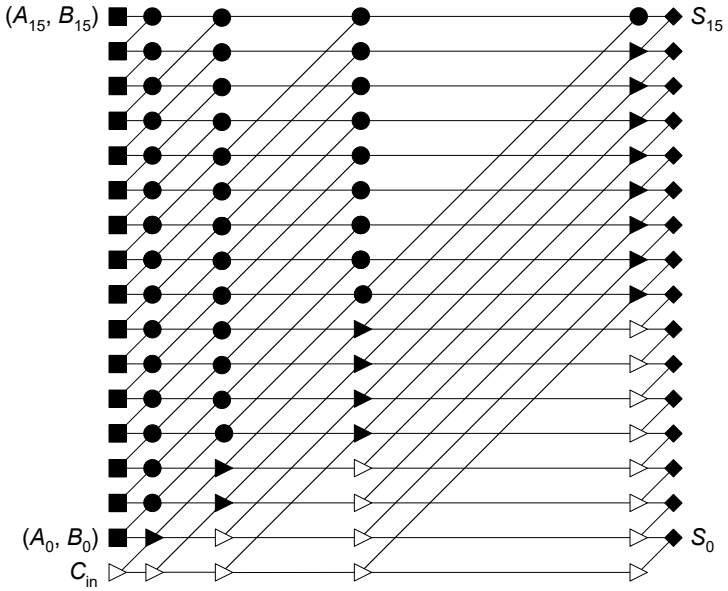
$\Rightarrow$ speedup comes for "free"



Circuit Optimization Examples



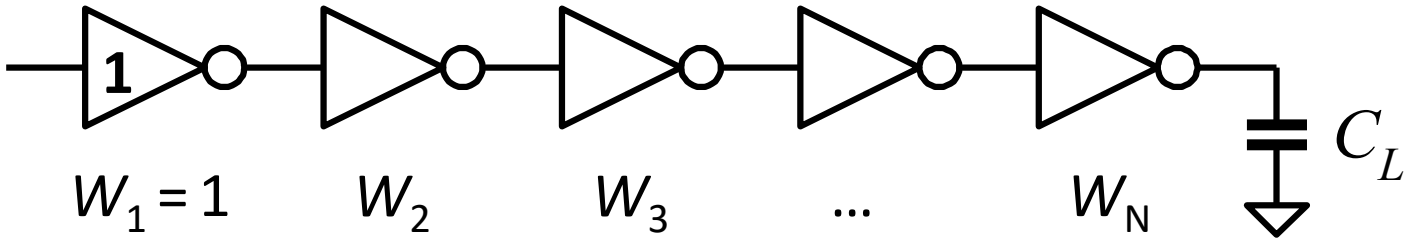




- **Inverter chain**
- **Memory decoder**
  - Branching
  - Inactive gates
- **Tree adder**
  - Long wires
  - Re-convergent paths
  - Multiple active outputs

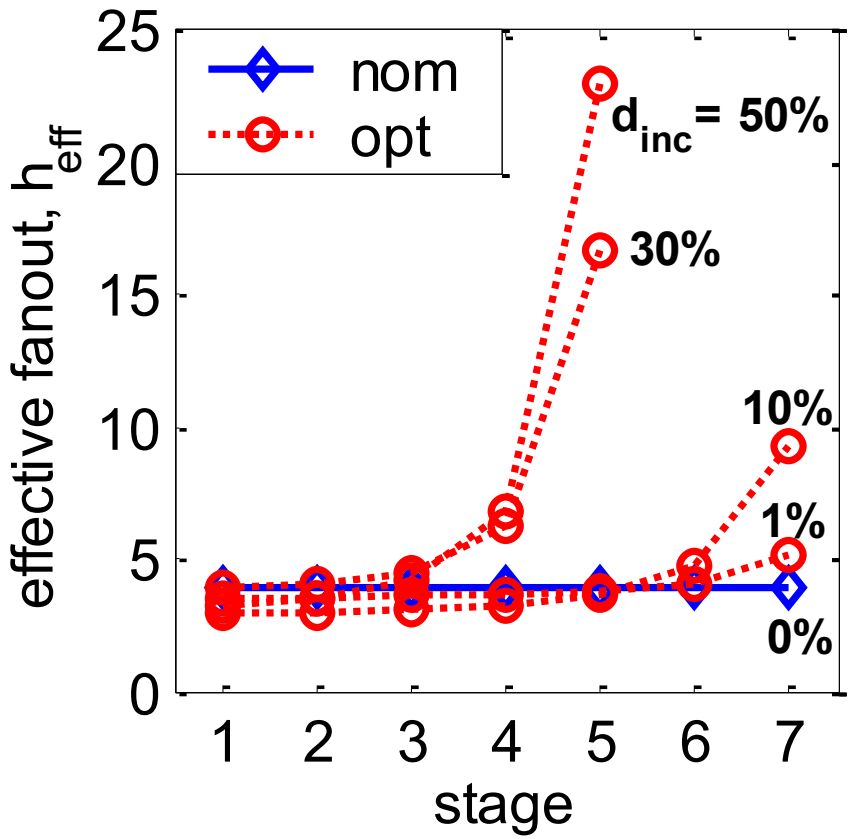
Example 14.1: Inverter Chain

- **Properties of inverter chain**
  - Single path topology
  - Energy increases geometrically from input to output



- **Goal**
  - Find optimal sizing $\boldsymbol{W} = [W_1, W_2, \ldots, W_N]$, supply voltage and buffering strategy to minimize energy

Inverter Chain: Gate Sizing & Buffering



$$W_i^2 = \frac{W_{i-1} \cdot W_{i+1}}{1 + \mu \cdot W_{i-1}}$$

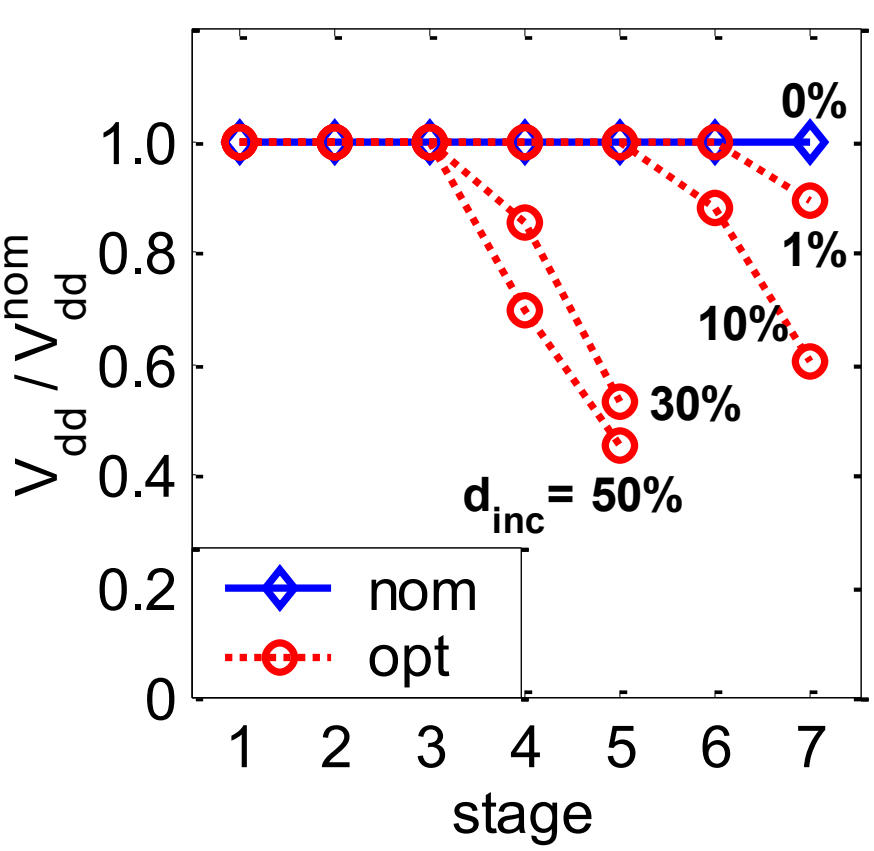
[Ma, Franzon, IEEE JSSC, 9/94]

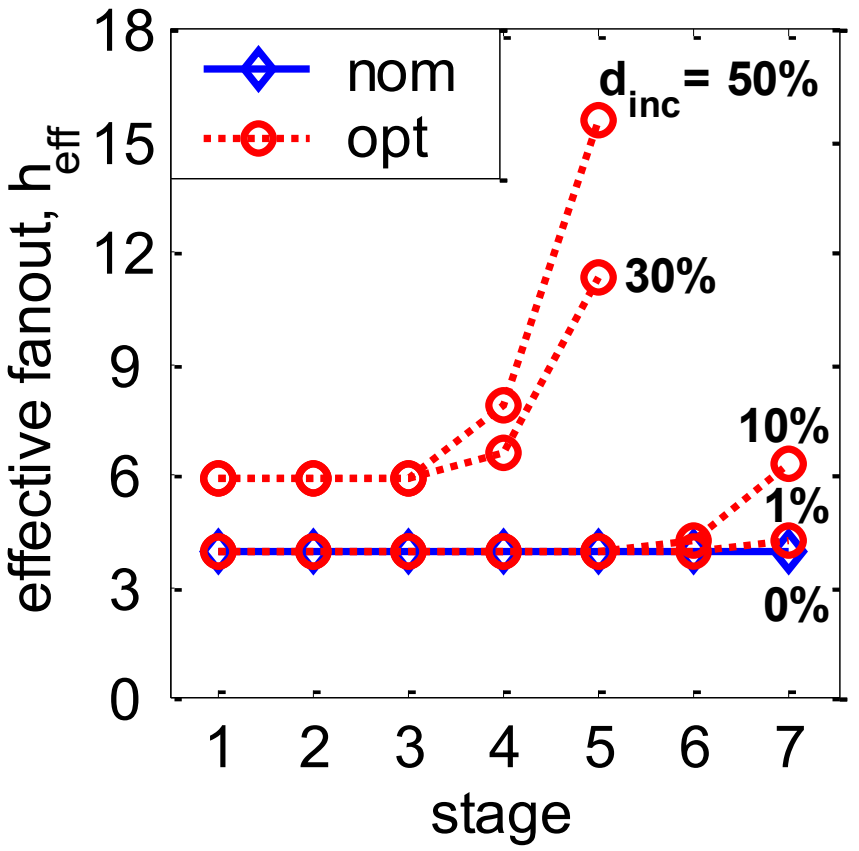
$$\mu = -\frac{2 \cdot K_e \cdot V_{DD}^2}{\tau_{ref} \cdot S_W}$$

$$S_W \propto \frac{ec_i}{h_{eff,i} - h_{eff,i-1}}$$

- **Variable taper achieves minimum energy**
- **Reduce number of stages at large d_{inc}**

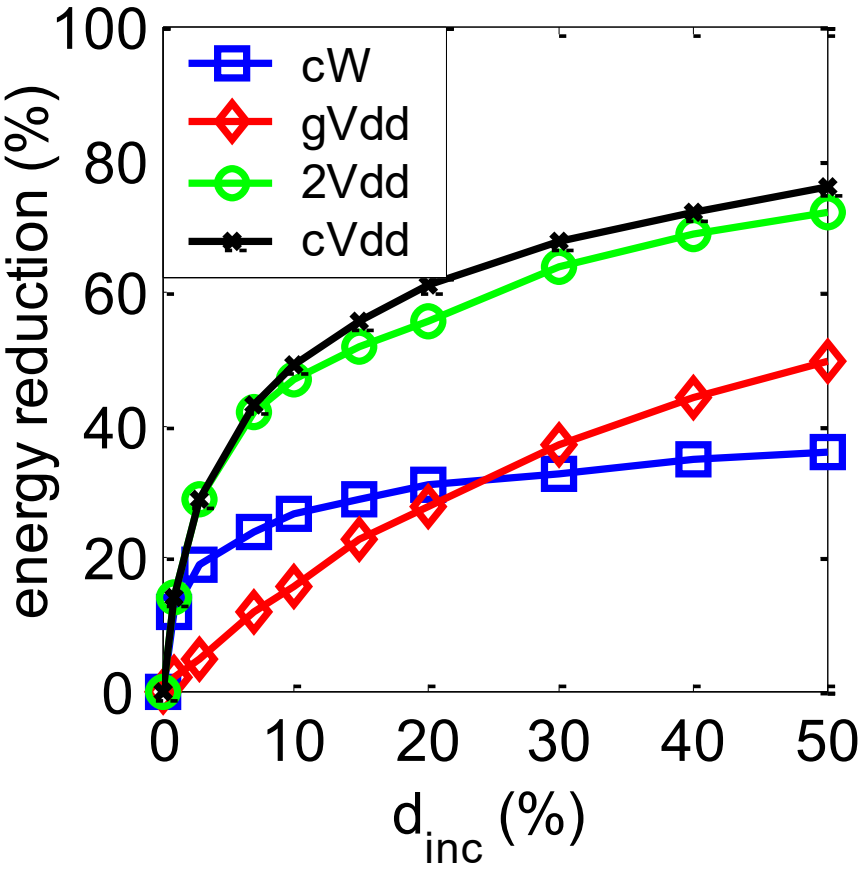
Inverter Chain: V_{DD} Optimization

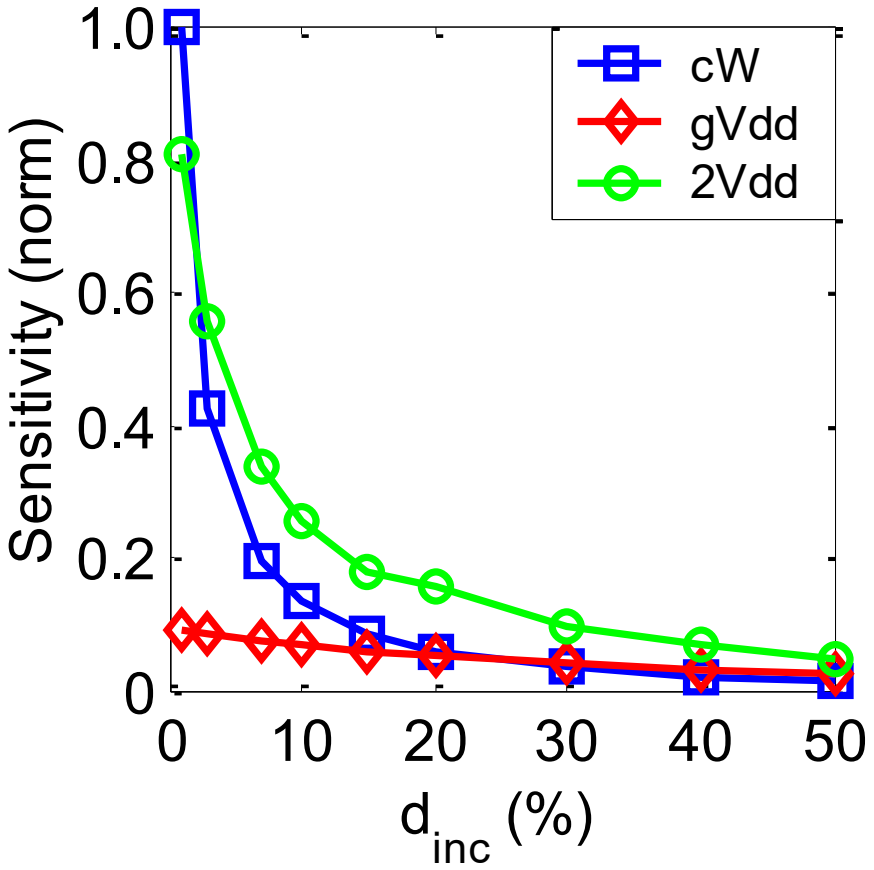




- **Variable taper achieved by voltage scaling**
- **V_{DD} reduces energy of the final load first**

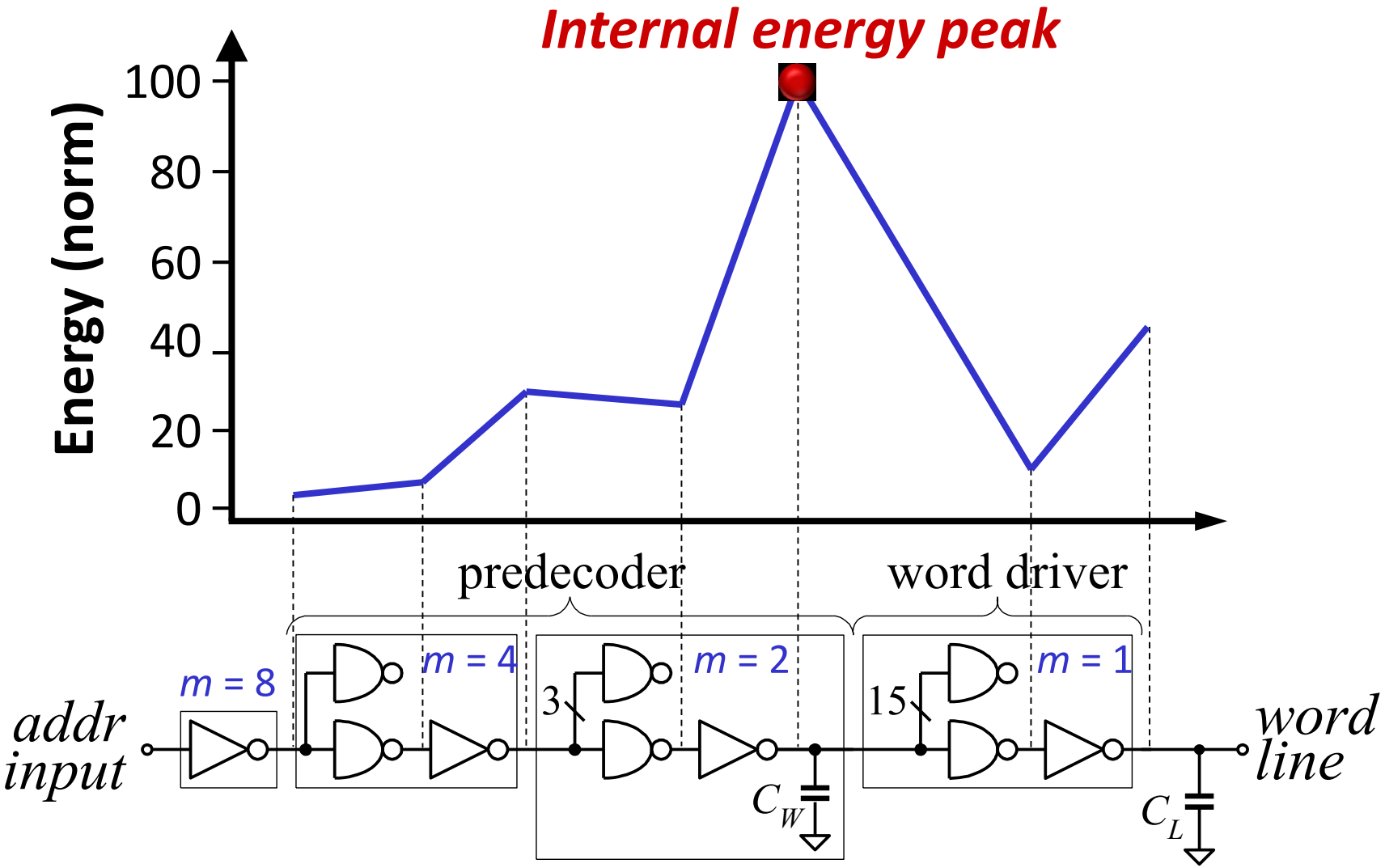
Inverter Chain: Optimization Results



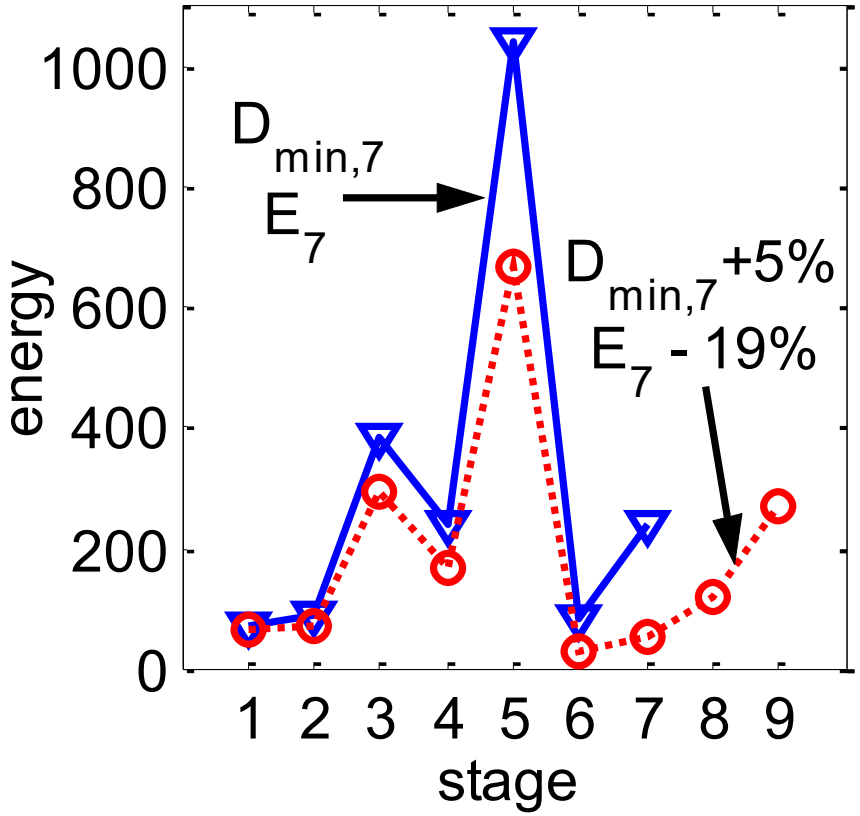


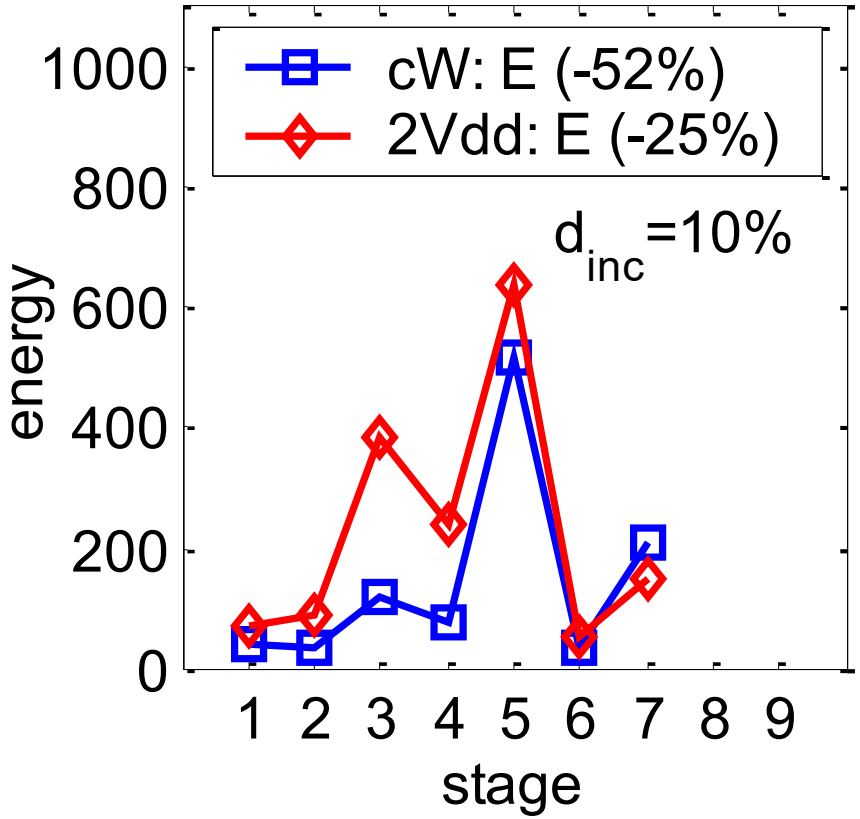
- **Parameter with the largest sensitivity has the largest potential for energy reduction**
- **Two discrete supplies mimic per-stage V_{DD}**

Example 14.2: SRAM Decoder



W vs. V_{DD} for Reducing Energy Peak

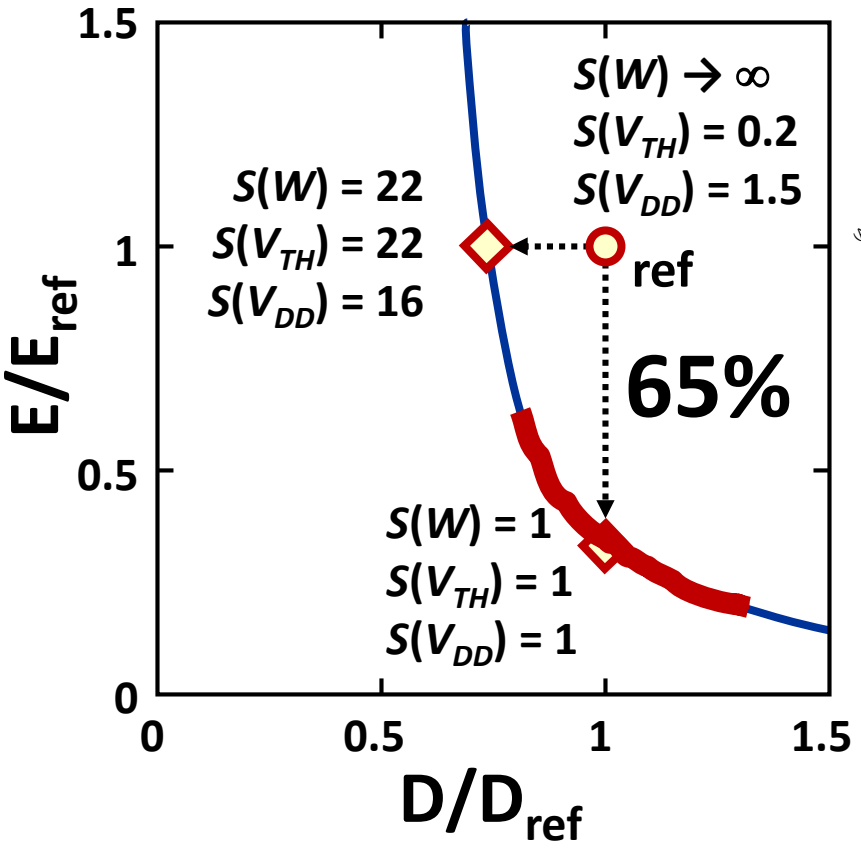


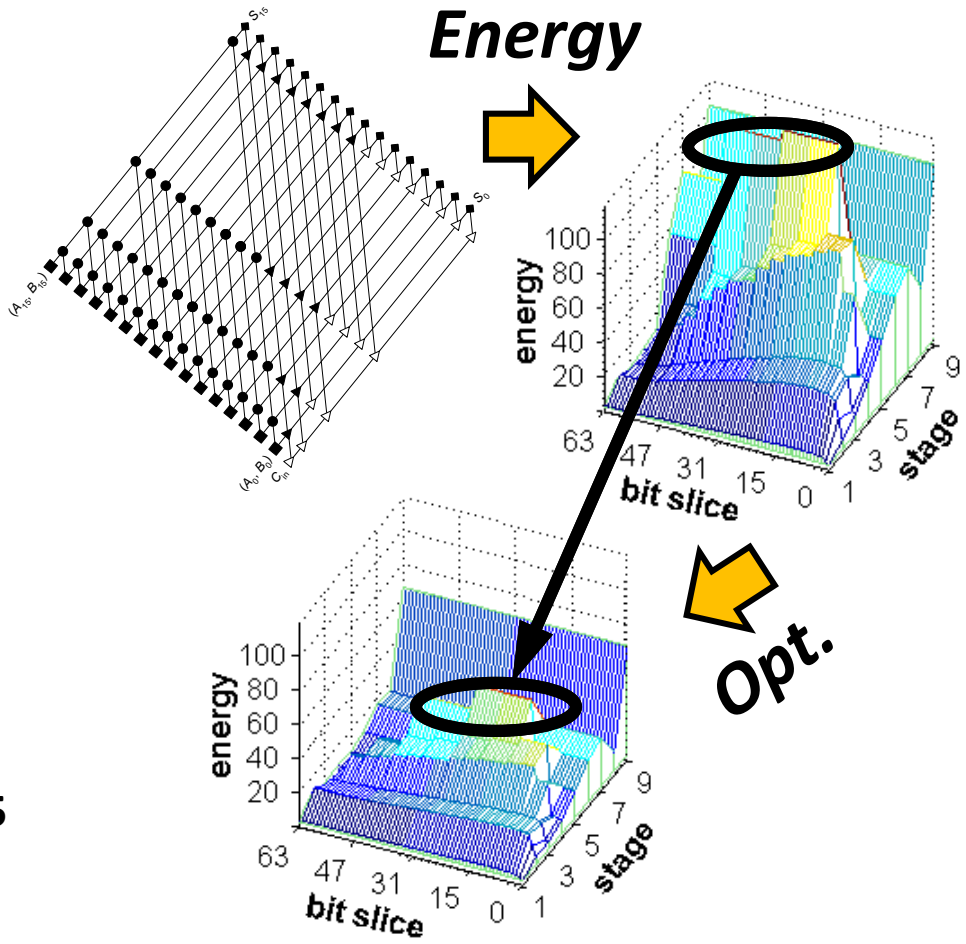


- V_{DD} less effective than W optimization
- Buffering also reduces energy peak

[B. Amrutur, Ph.D. Thesis, Stanford, 8/99]

Example 14.3: Tree Adder

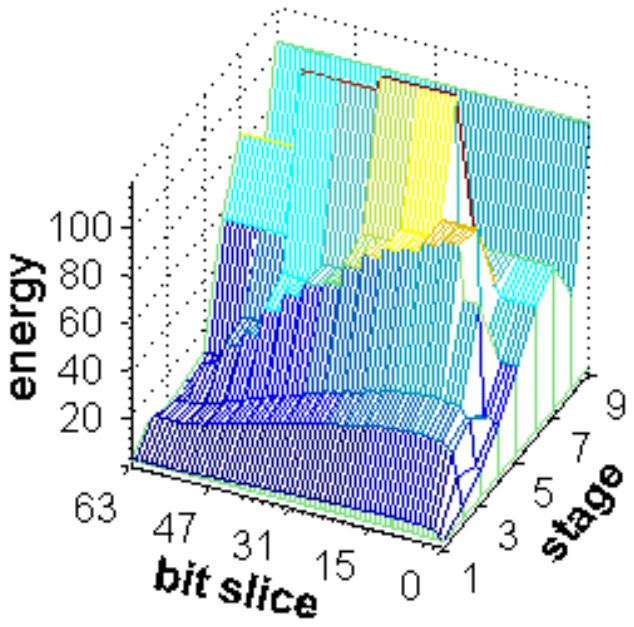




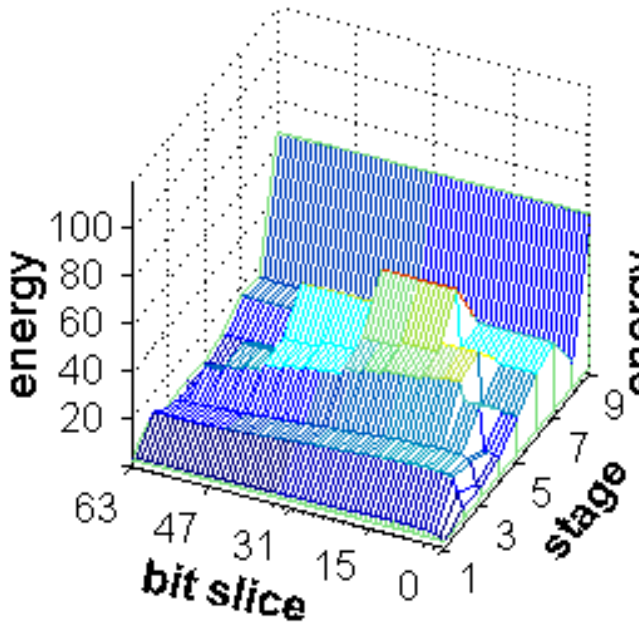
Tree Adder: Optimization Results

Reference: all paths are critical

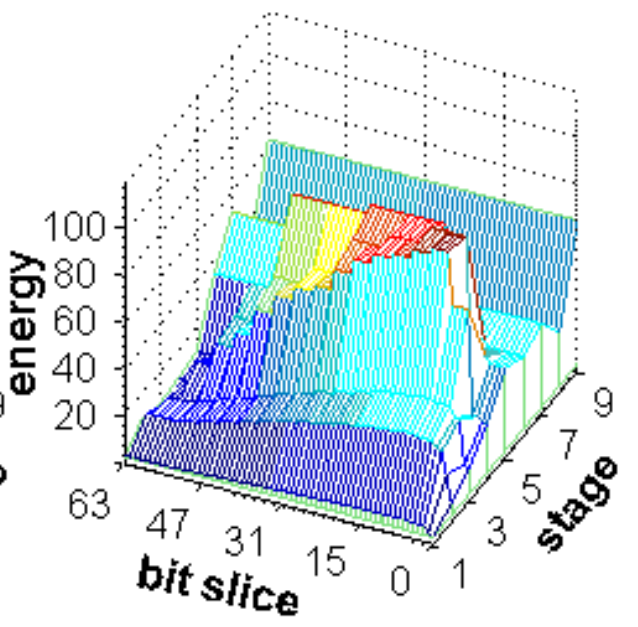
reference
D_{min}, E_{ref}



Sizing Opt.
$1.1D_{min}$, $0.45E_{ref}$



2-V_{DD} Opt.
$1.1D_{min}$, $0.73E_{ref}$

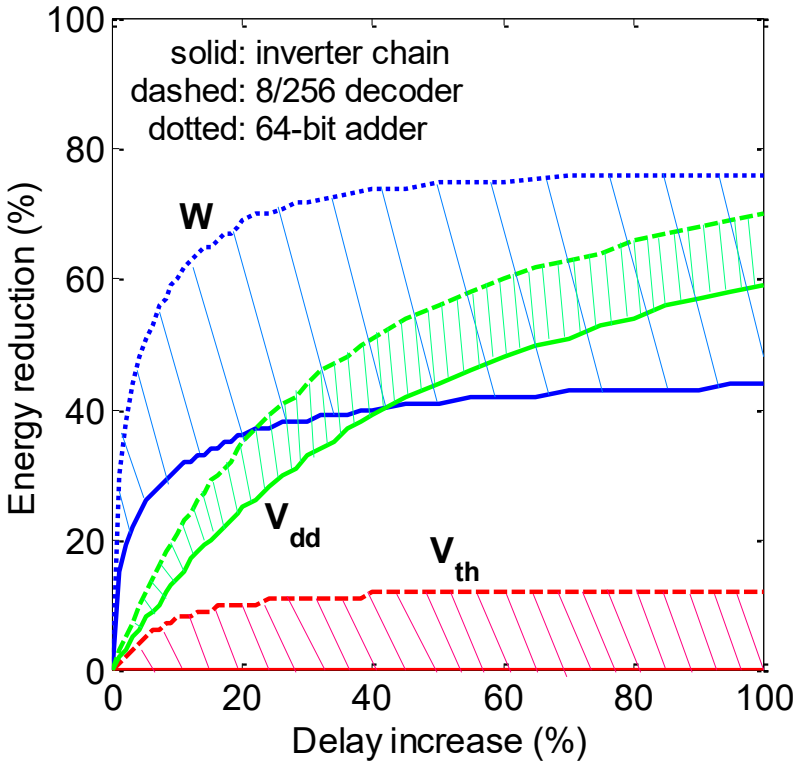


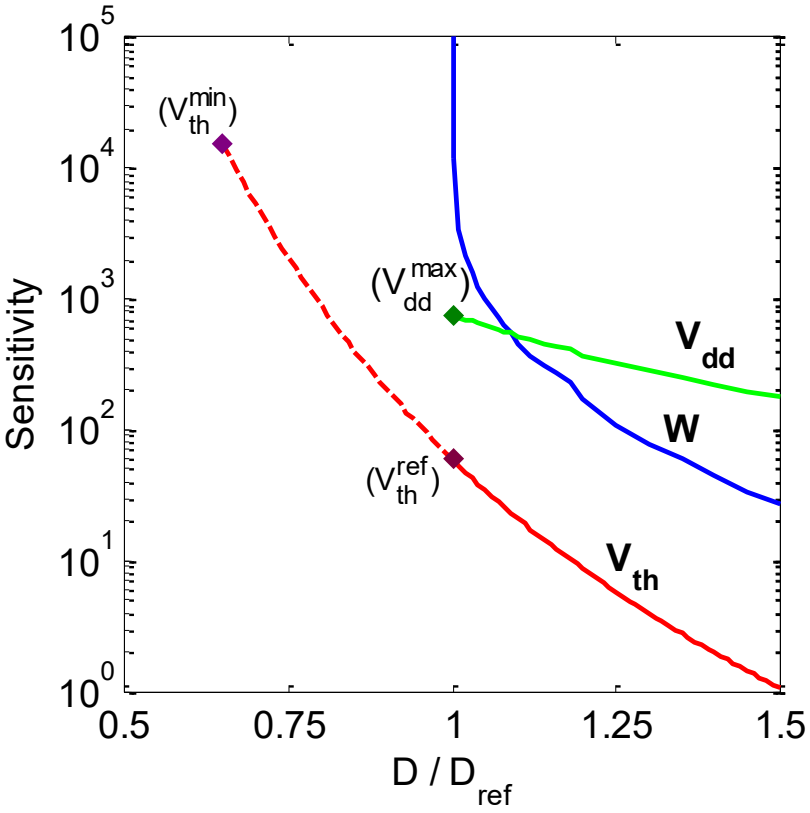
- **Internal energy: W more effective than V_{DD}**
  - For d_{inc} = 10%: ΔE_W = −55%, ΔE_{2Vdd} = −27%

A Few Insights

A Look at Tuning Variables…

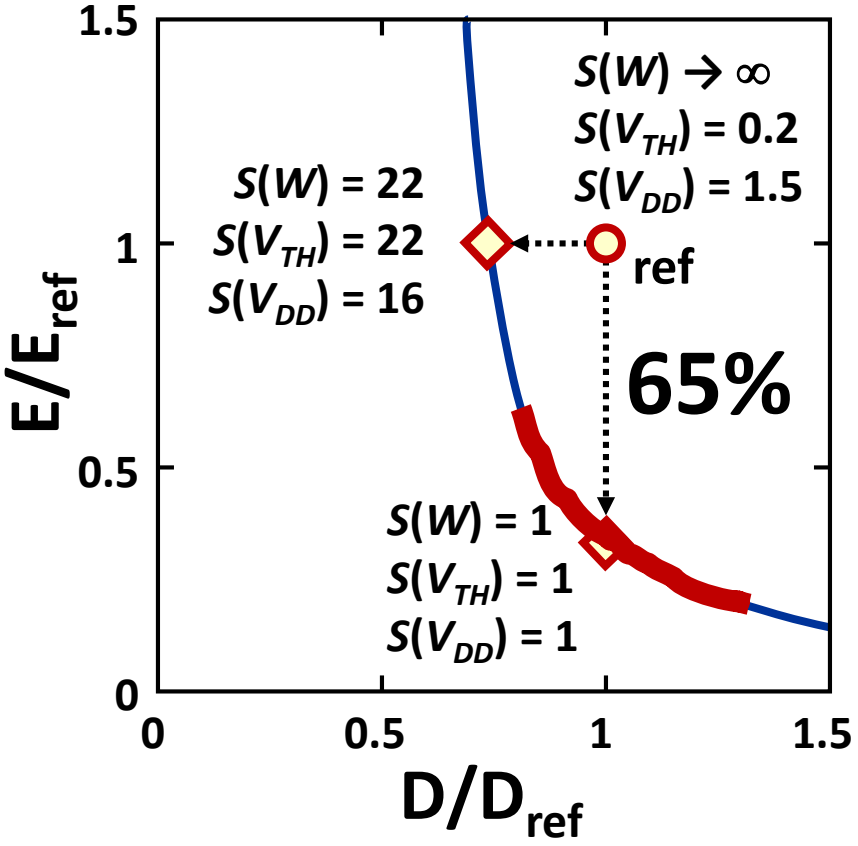
10% excess delay → 30-70% energy reduction





Peak performance is very power inefficient!

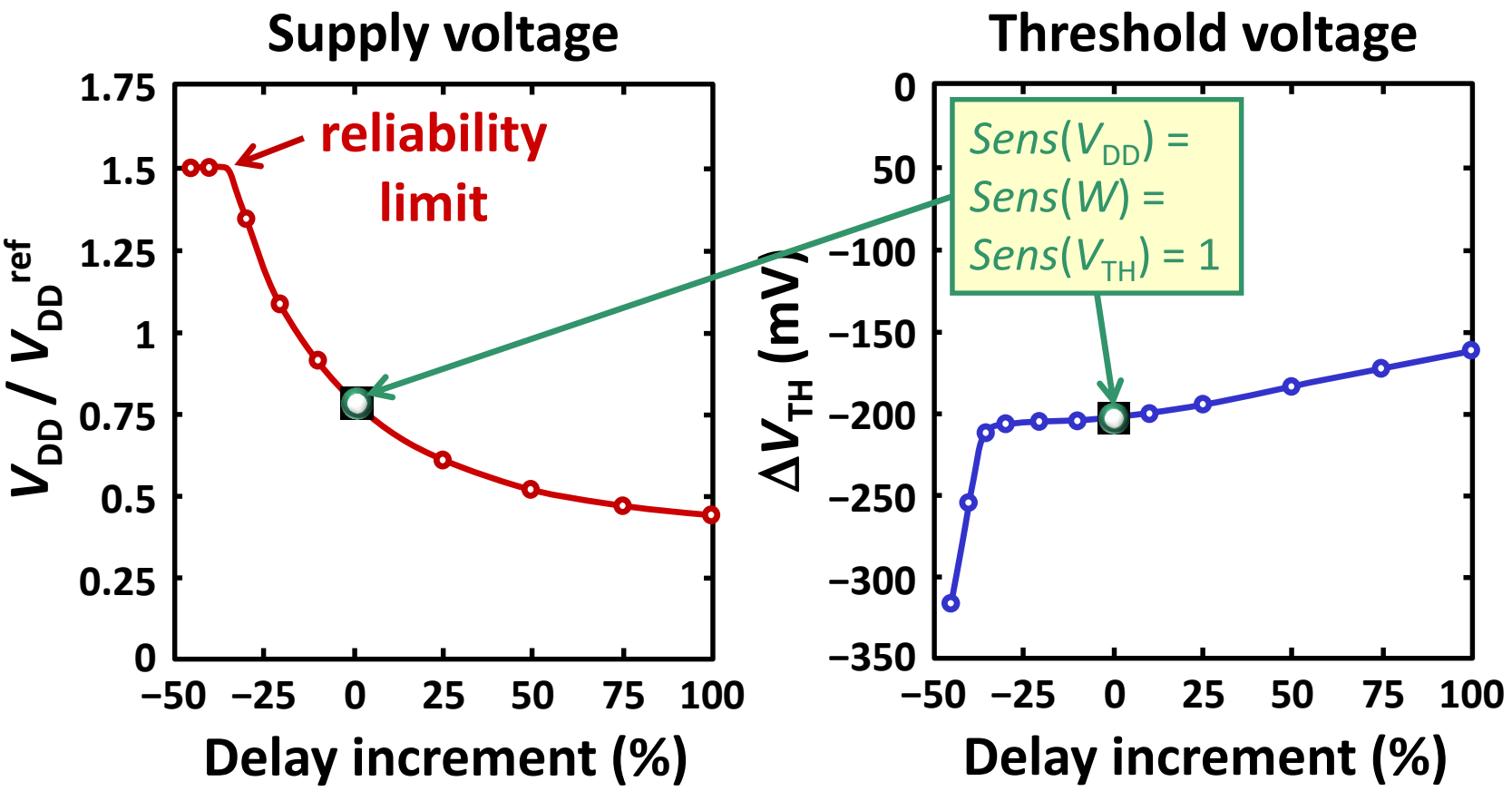
Limited Range of Circuit Optimization



- ±30% around D_{ref}
- Else, too much E or D
- Need for arch. opt.

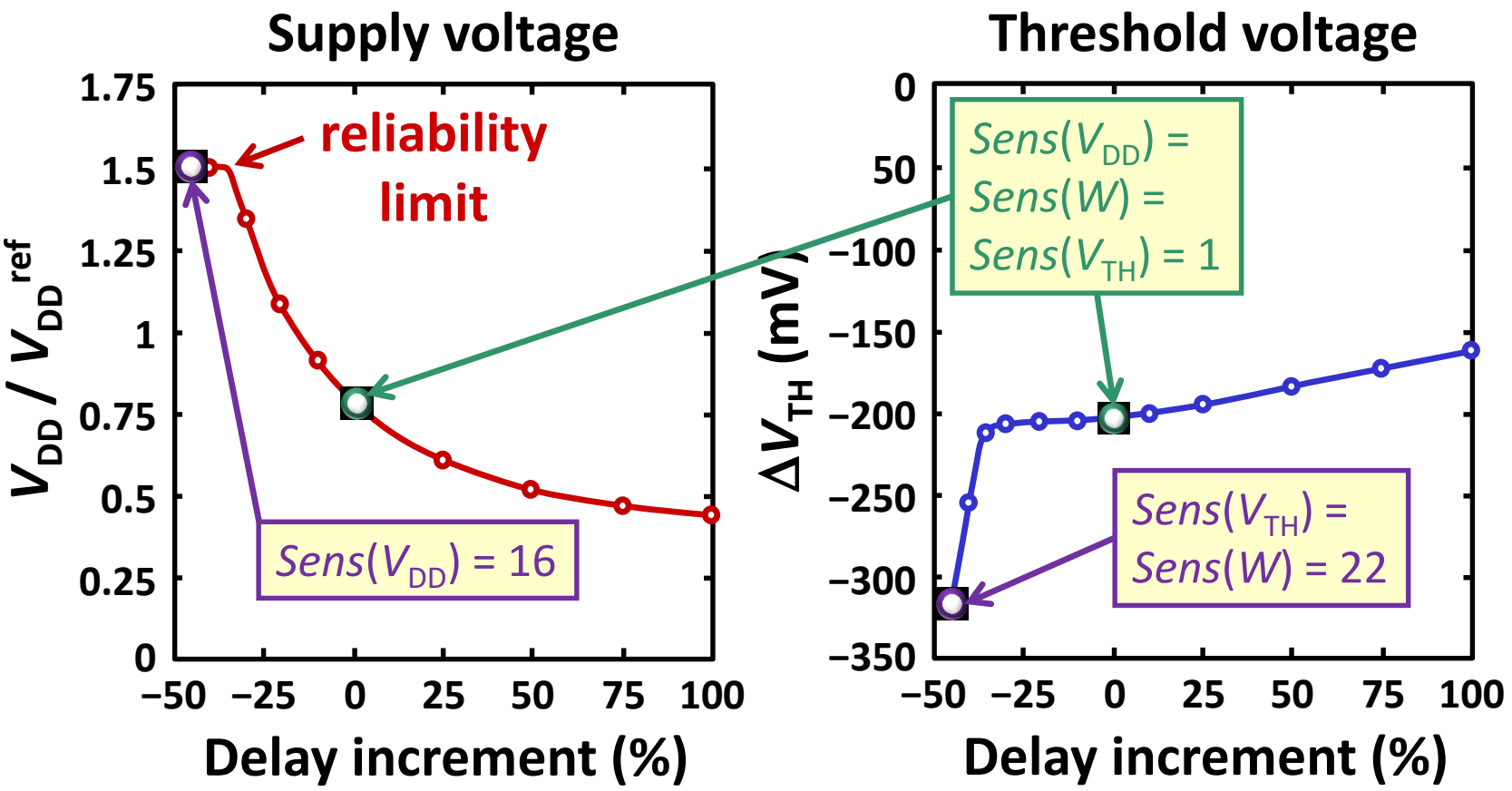
A Look at Tuning Variables

Equal sensitivity unless variables reach their bounds

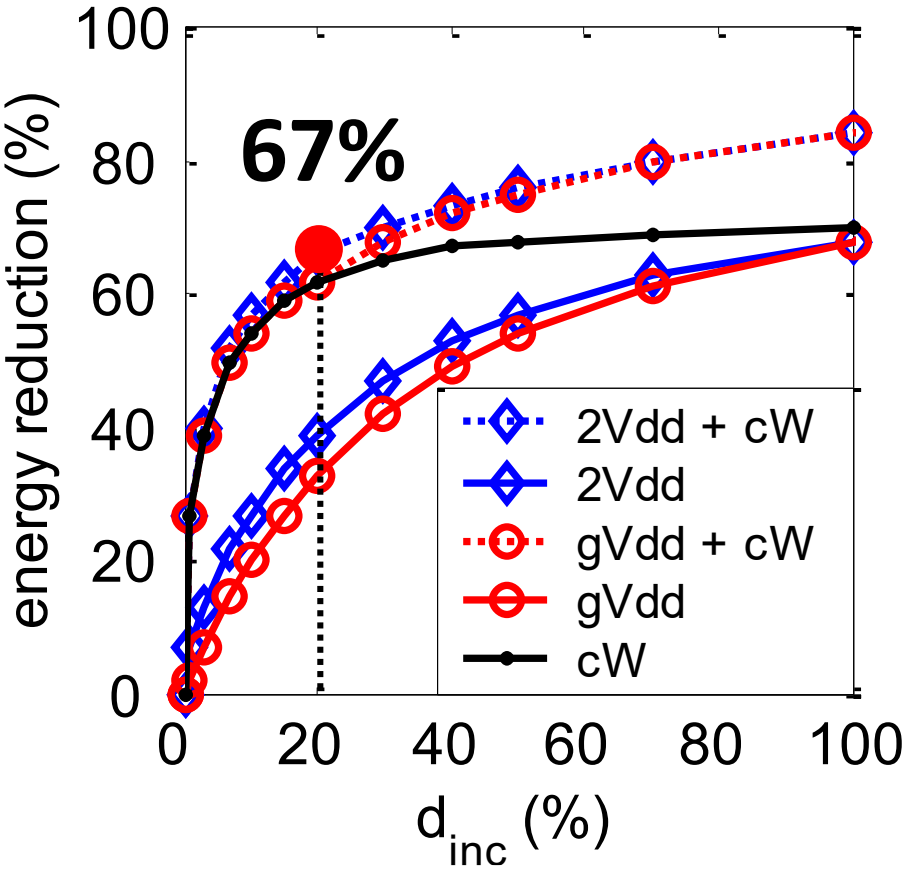


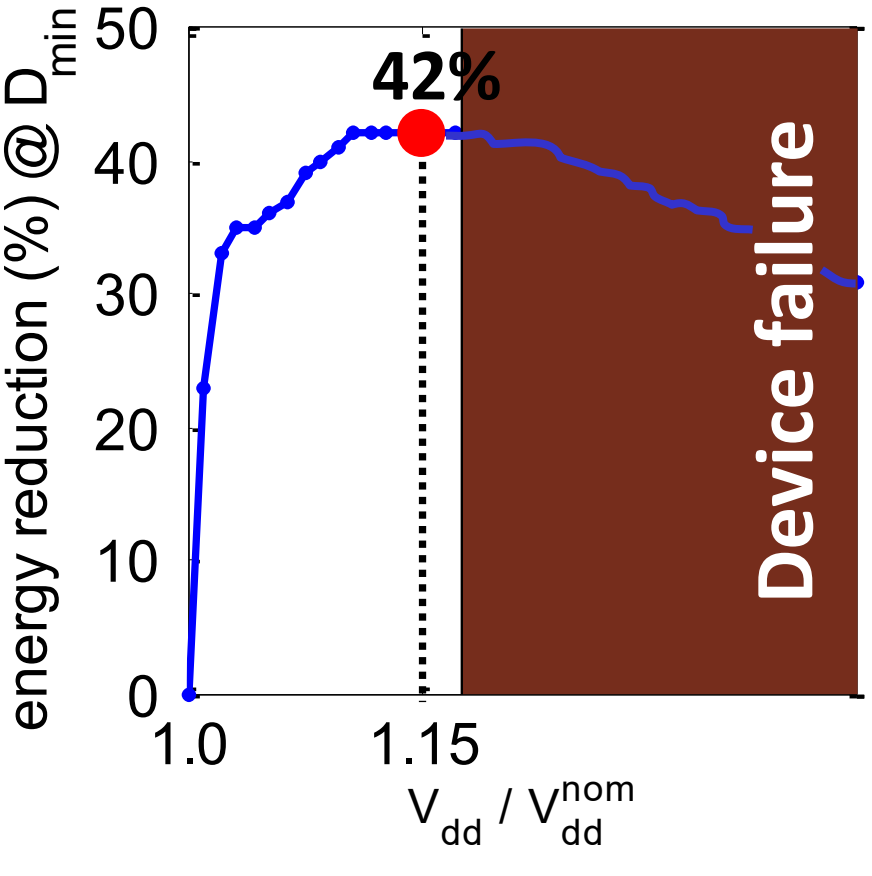
A Look at Tuning Variables

Bound reached = fewer variables to work with



Tree Adder: Joint Optimization

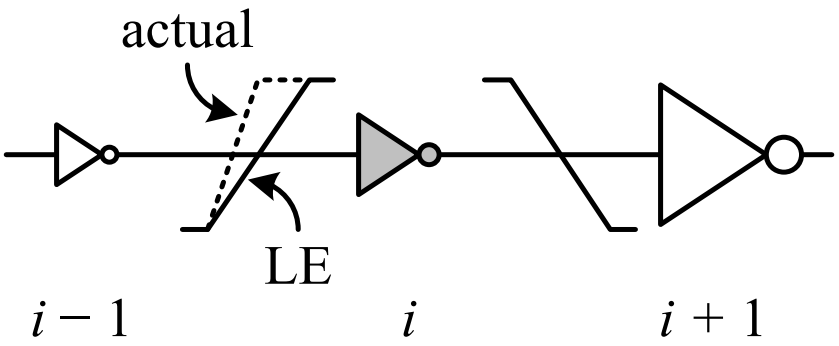




- Higher V_{DD} yields a lower energy solution
- Choose a more efficient variable

An Update: Slope-Aware Delay Model

- **Logical effort (equal slopes) overestimates delay**



- **Add slope correction**

$$D = \sum_{i=1}^{N} \left(g_i \cdot h_i + p_i - \frac{g_i \cdot h_i - g_{i-1} \cdot h_{i-1}}{K_i} \right)$$

Gate & V_{DD} dependent

Intuition: Slope Factor

$$t_p = t_{LE} - \frac{t_{slope,out} - t_{slope,in}}{K_{slope\ factor}}$$

Same RC ⇩

$$t_p = t_{LE} - \tau \cdot \frac{f_{out} - f_{in}}{K_{FO\ factor}}$$

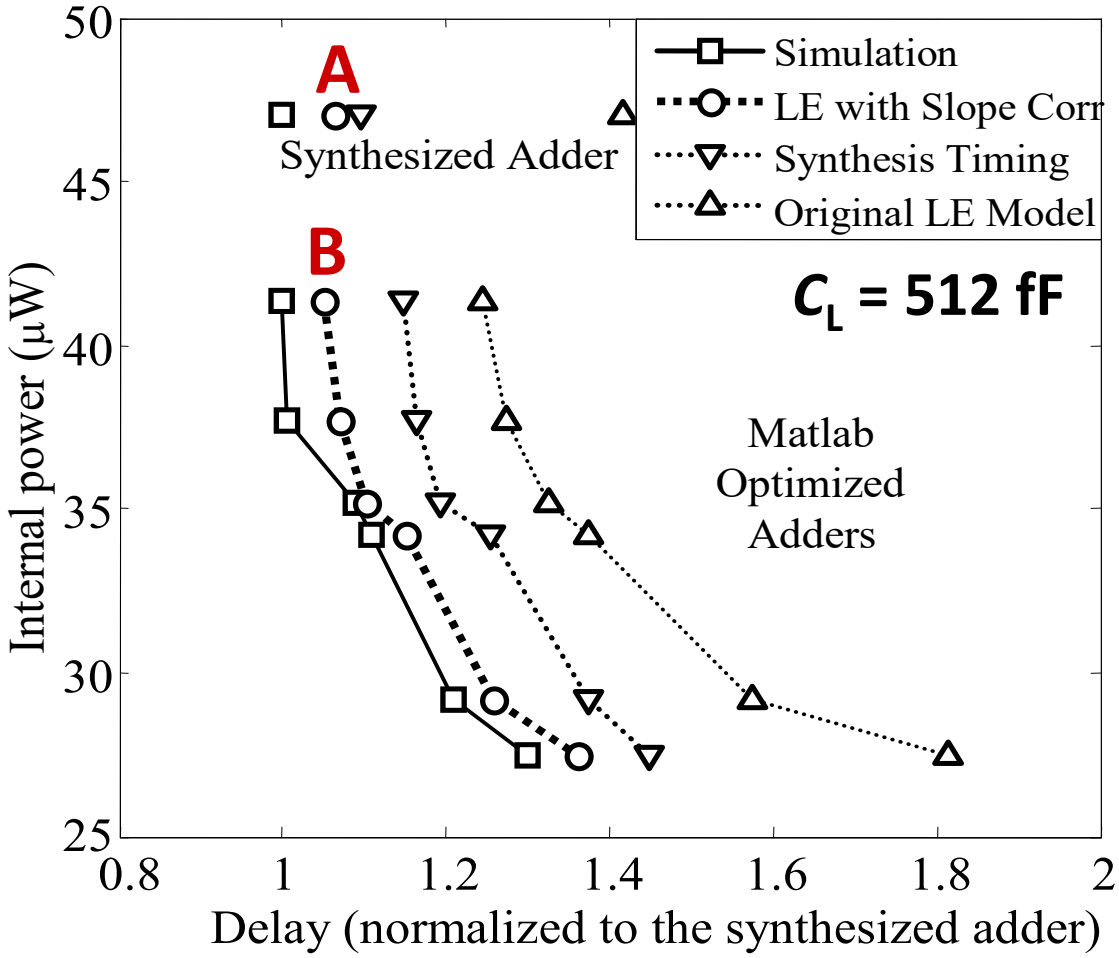
Divide by τ ⇩

$$d = d_{LE} - \frac{f_{out} - f_{in}}{K_{gate}}$$

- **K_{gate} converts slope to delay (fanout)**
  - Should be around 2-ish

Result: 16-b Adder Example (H = 256)

Logical effort largely overestimates non-critical paths



- **A: LE sizing**
- **B: SC LE sizing**

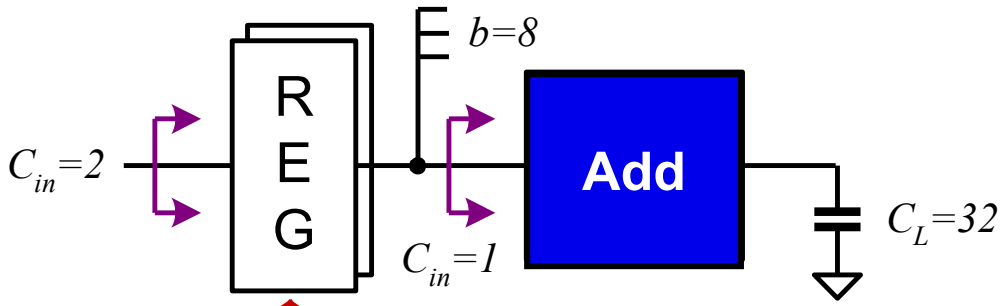
[C.C. Wang, D. Markovic, TCAS-II, Aug'09]

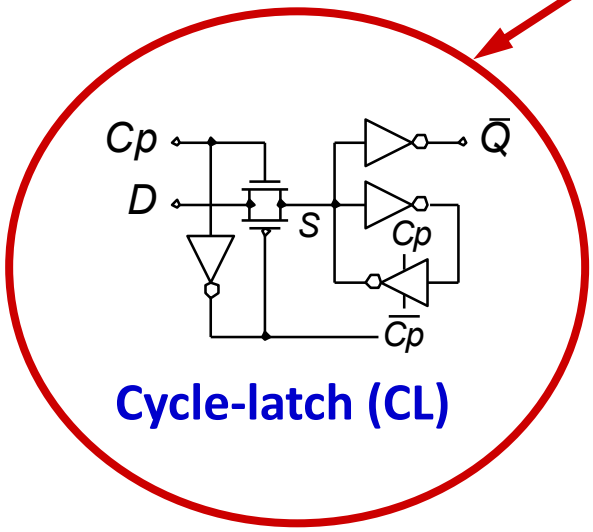
Lessons from Circuit Optimization

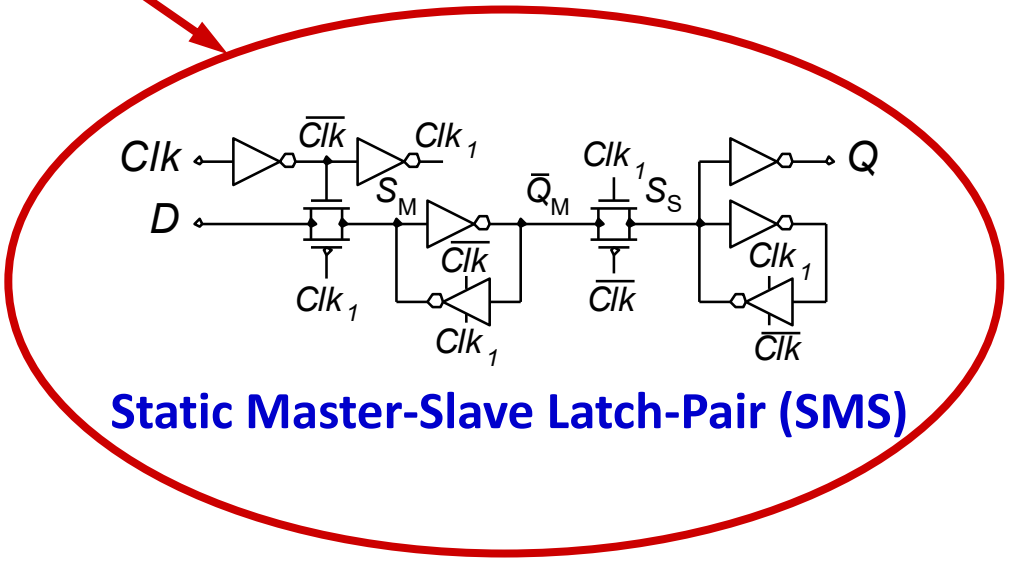
- **Sensitivity-based optimization framework**
  - Equal marginal costs ⇔ Energy-efficient design
- **Effectiveness of tuning variables**
  - Sizing is the most effective for small d_{inc}
  - V_{DD} is better for large delay increments
- **Peak performance is VERY power inefficient**
  - ~70% energy reduction for 20% delay penalty
- **Limited performance range of tuning variables**
  - Additional variables for higher energy-efficiency

Choosing Circuit Topology: Optimal Register?

- **64-bit ALU**



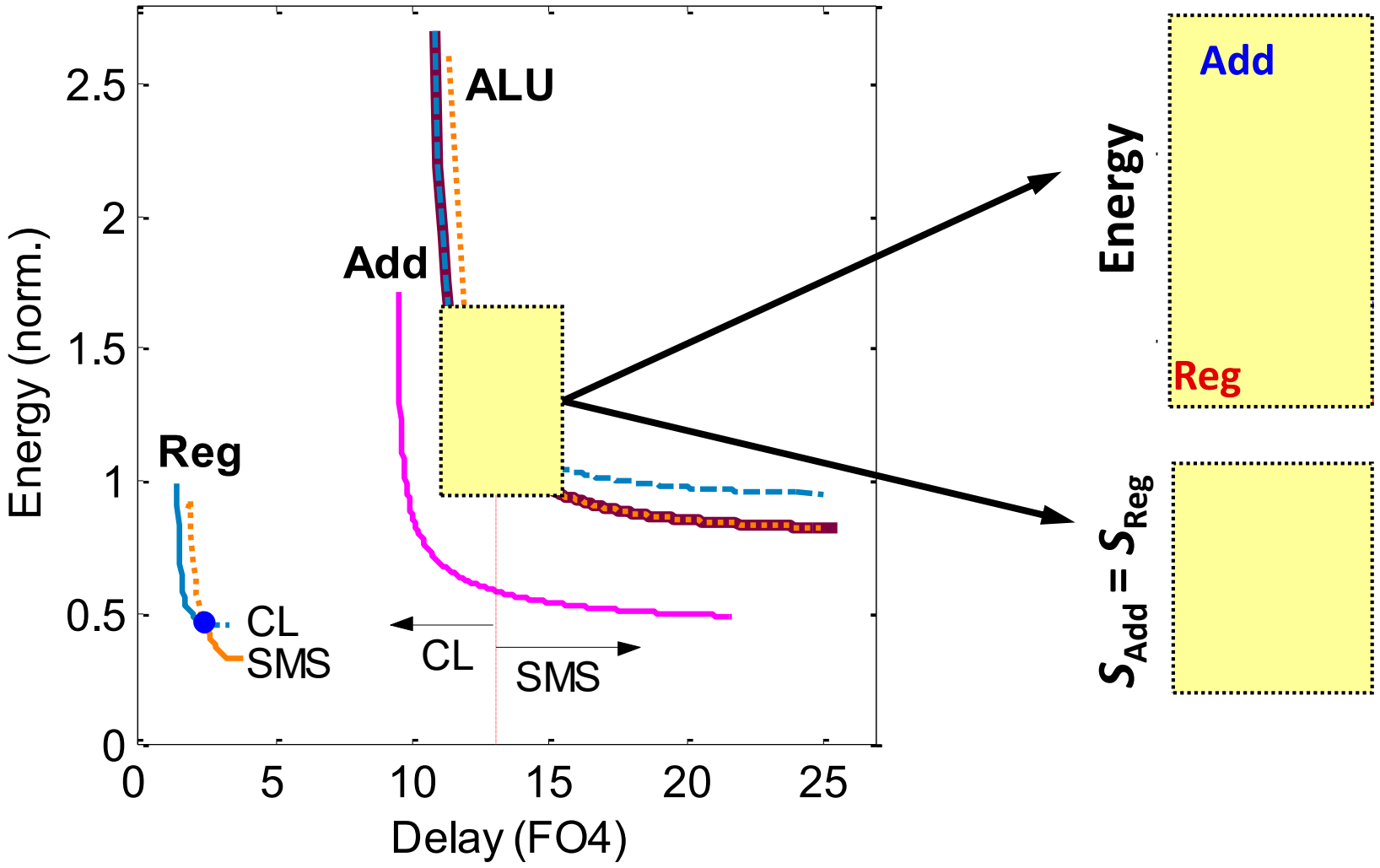




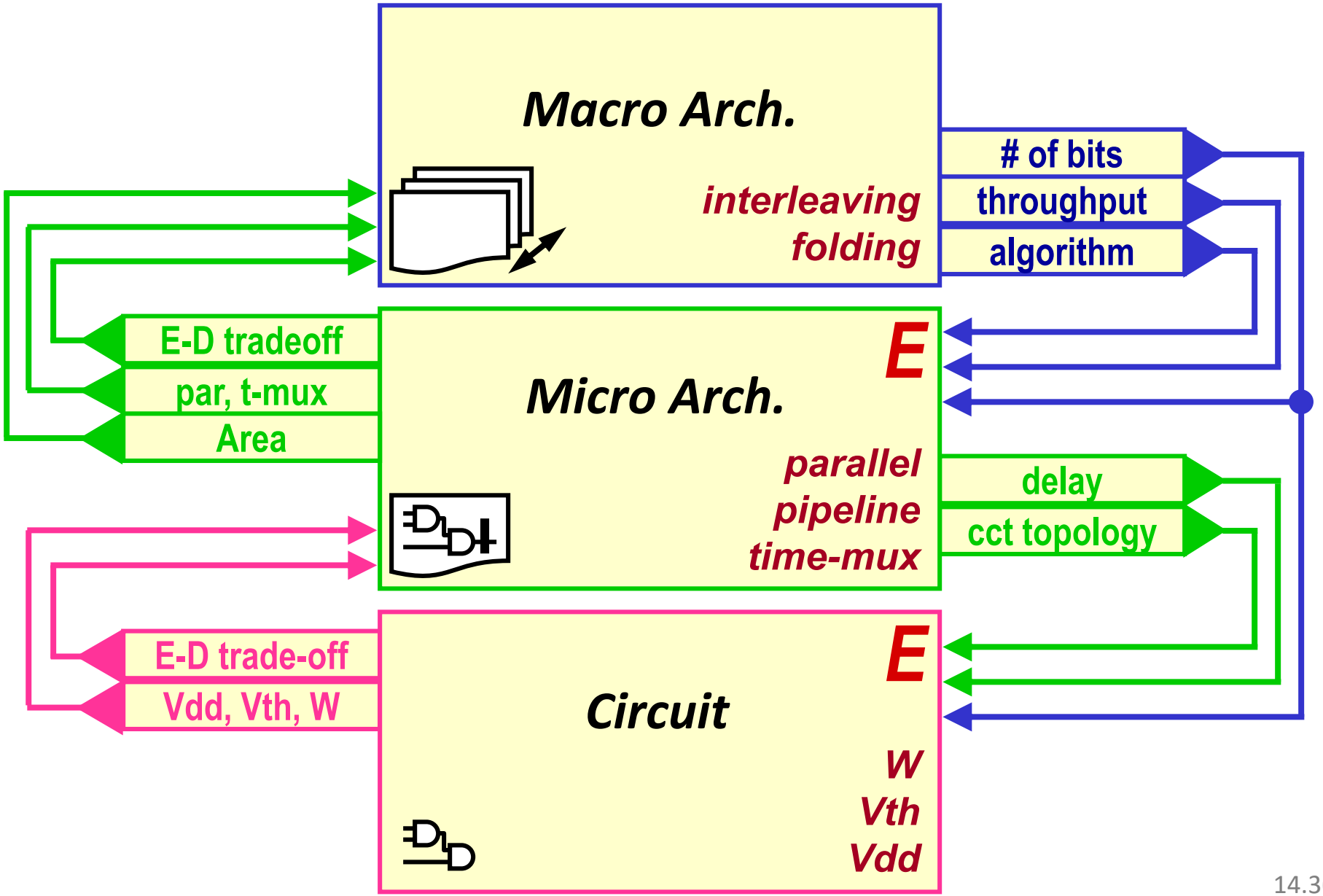
- **Given energy-delay tradeoff for adder and register (two register options), what is the best energy-delay tradeoff in the ALU?**

Balancing Sensitivity Across Circuit Blocks

Upsize lower activity blocks (Add) to save energy



Micro-Architectural Optimization

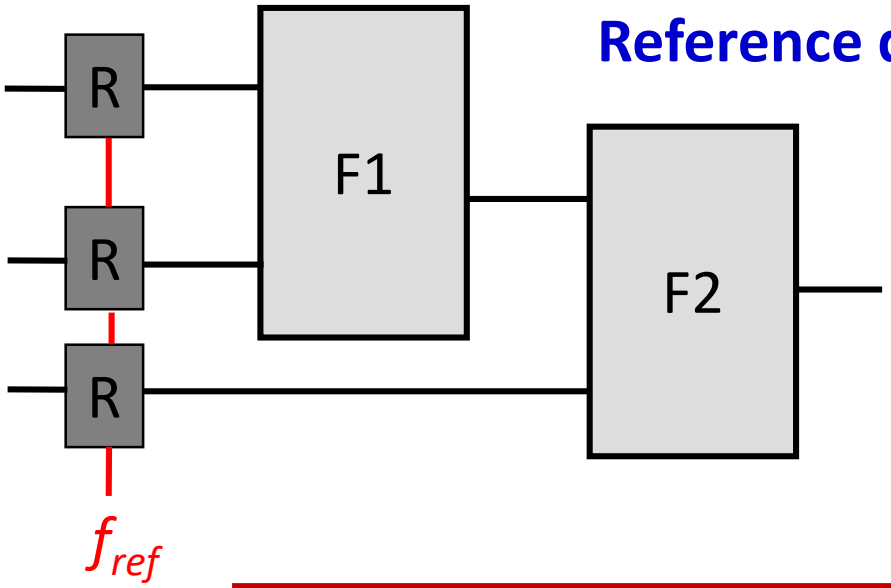


Reducing the Supply Voltage

(while maintaining performance)

Concurrency:

trading off clock frequency versus area to reduce power



Reference design

R: register,
F1,F2: combinational logic (adders, ALUs, etc)

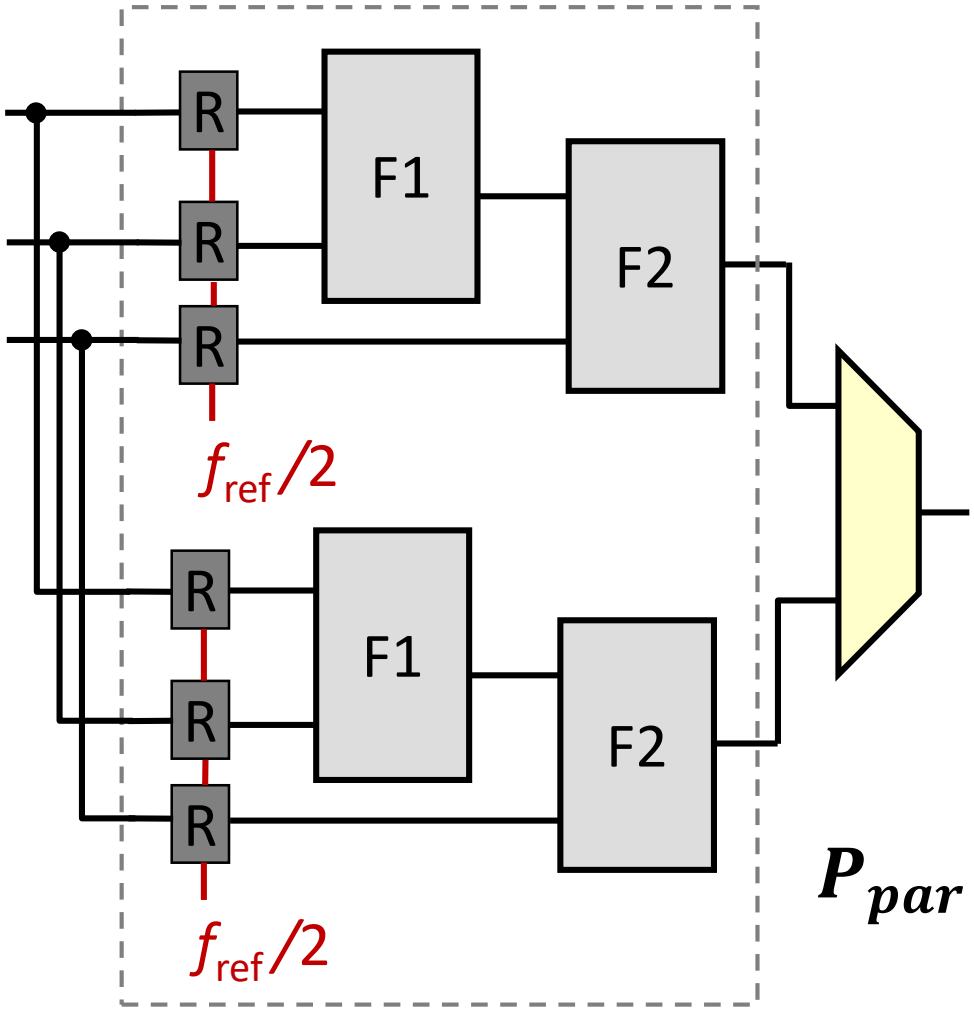
[A. Chandrakasan, et al., JSSC'92]

$$P_{ref} = C_{ref} \cdot V_{dd,ref}^2 \cdot f_{ref}$$

C_{ref}: average switching capacitance

A Parallel Implementation

- Slower logic = lower V_{DD} = lower power



$$f_{par} = \frac{f_{ref}}{2}$$

$$C_{par} = (2 + ov_{par}) \cdot C_{ref}$$

$$V_{DD,par} = \epsilon_{par} \cdot V_{DD,ref}$$

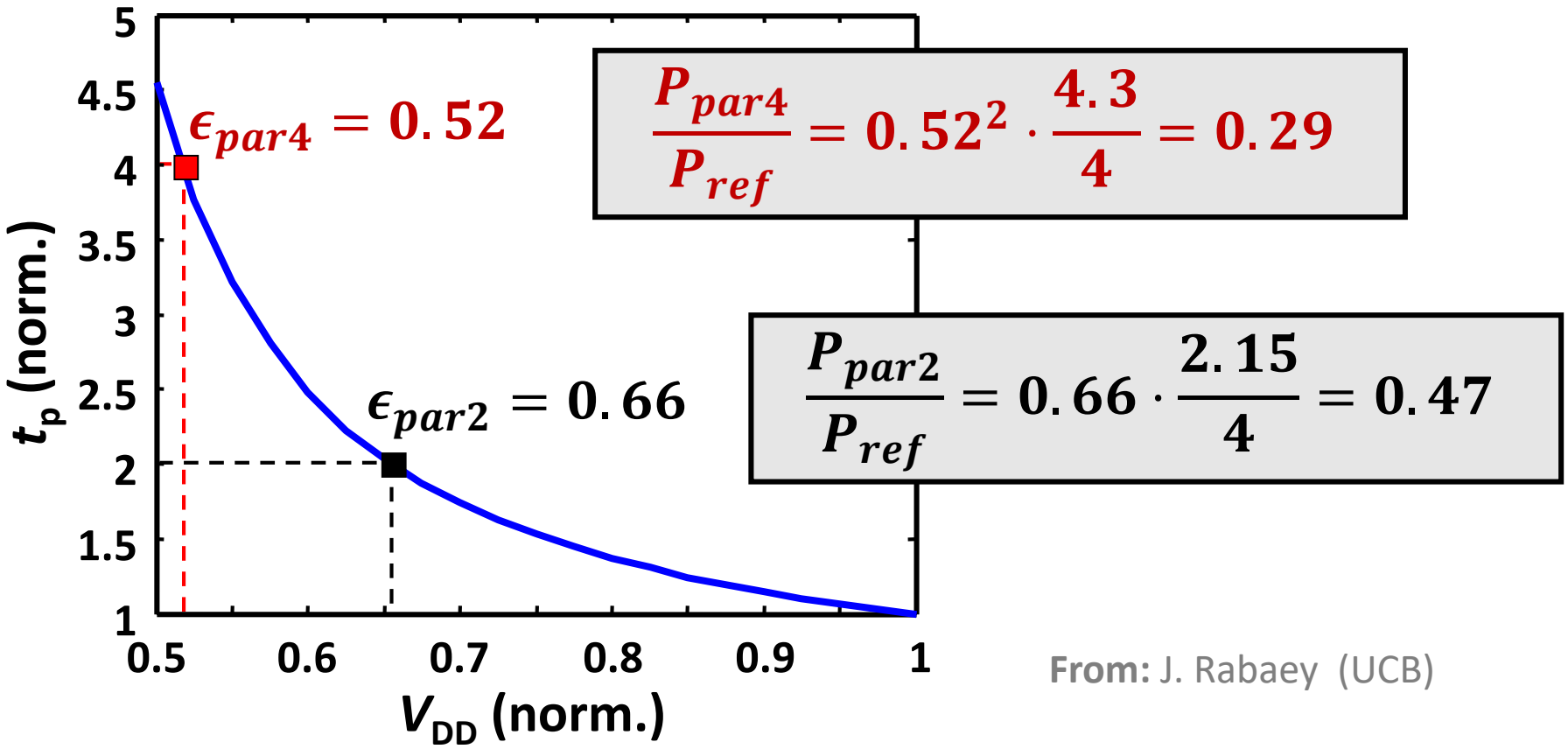
Power saving

Almost cancels

$$P_{par} = \epsilon_{par}^2 \cdot \frac{2 + ov_{par}}{2} \cdot P_{ref}$$

Parallelism Example (90nm CMOS)

Power (assuming ov_{par} = 7.5%)

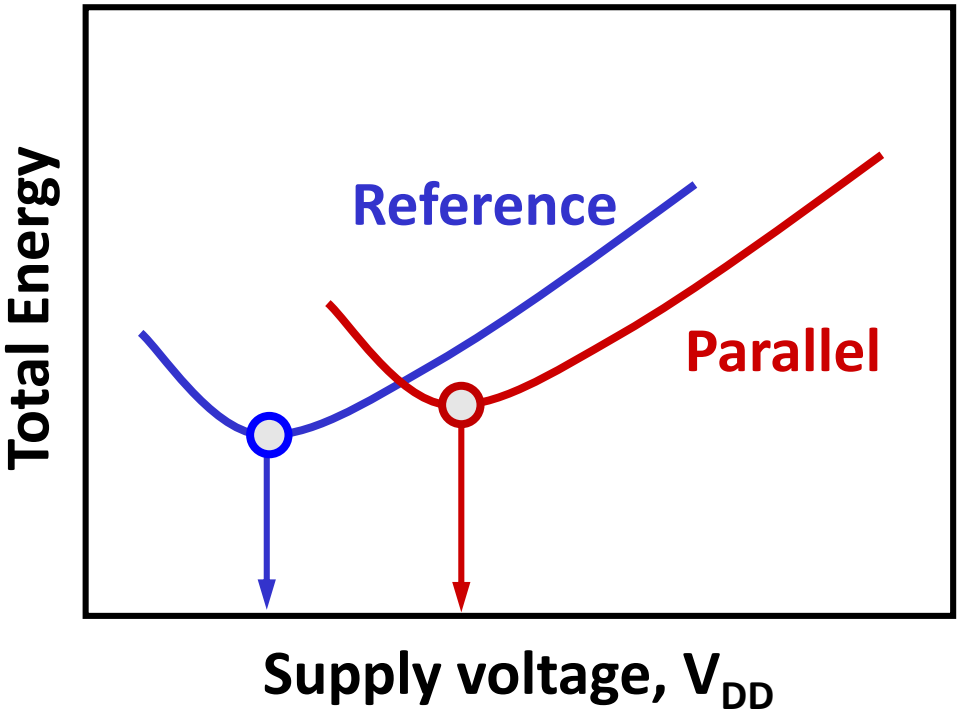


From: J. Rabaey (UCB)

How many levels of parallelism?

The More Parallel the Better?

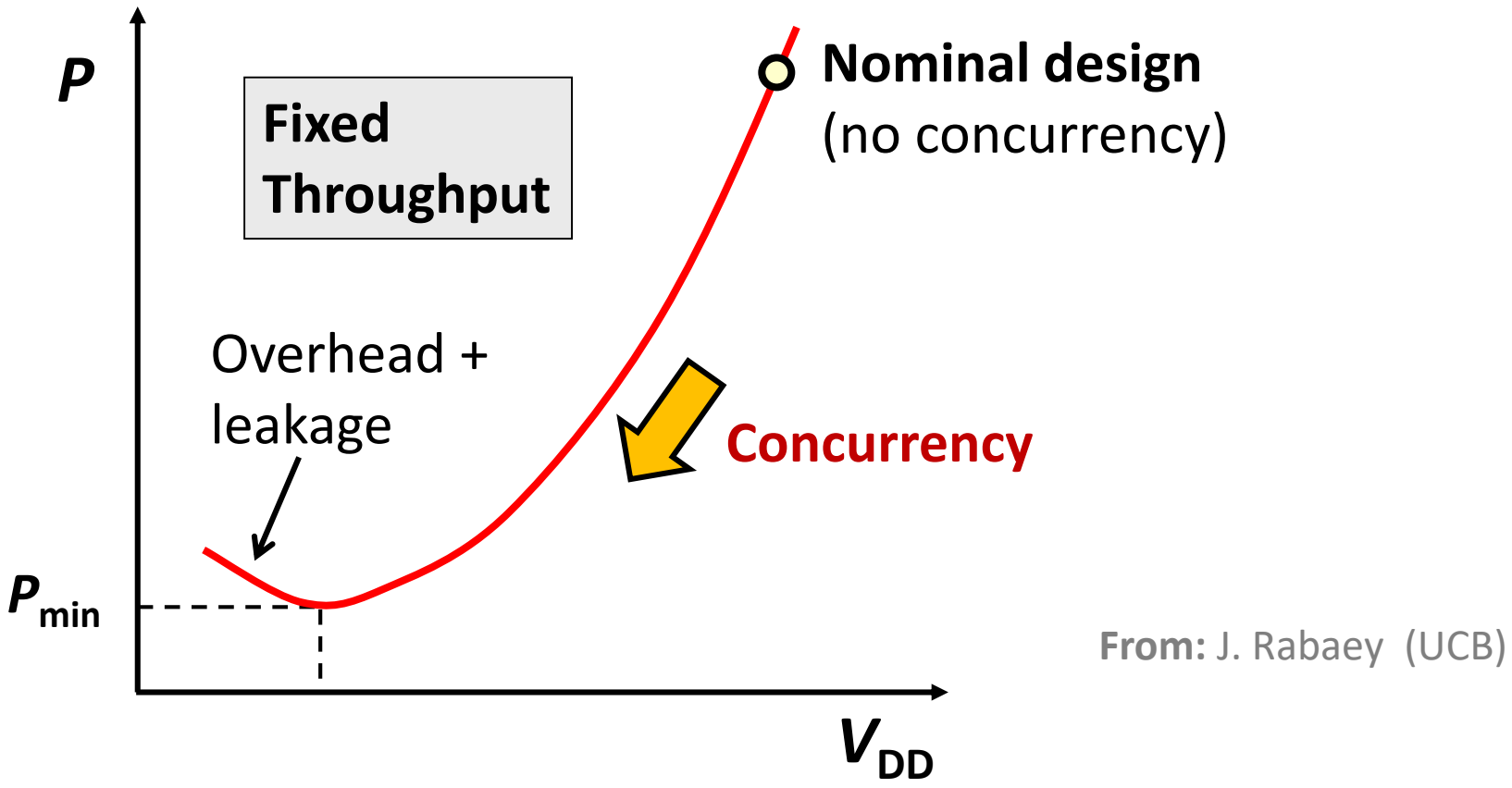
$$E_{tot} = E_{sw} + N \cdot E_{lk} + E_{overhad}$$



From:
J. Rabaey
(UCB)

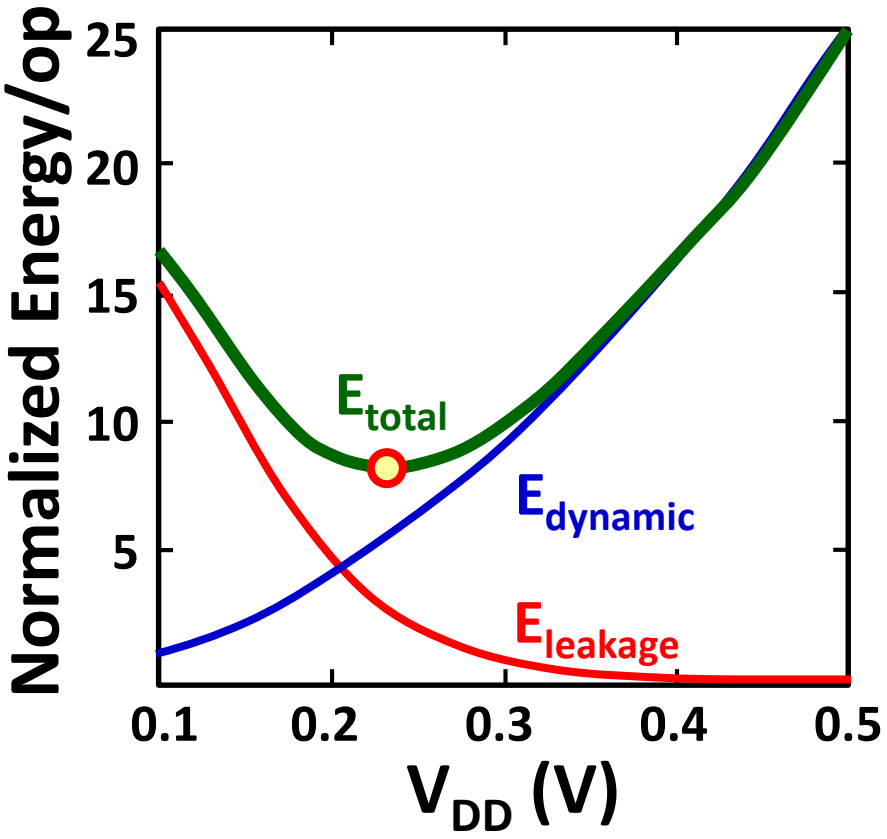
- Leakage and overhead start to dominate at high P
- Optimal V_{DD} and min E increase with parallelism

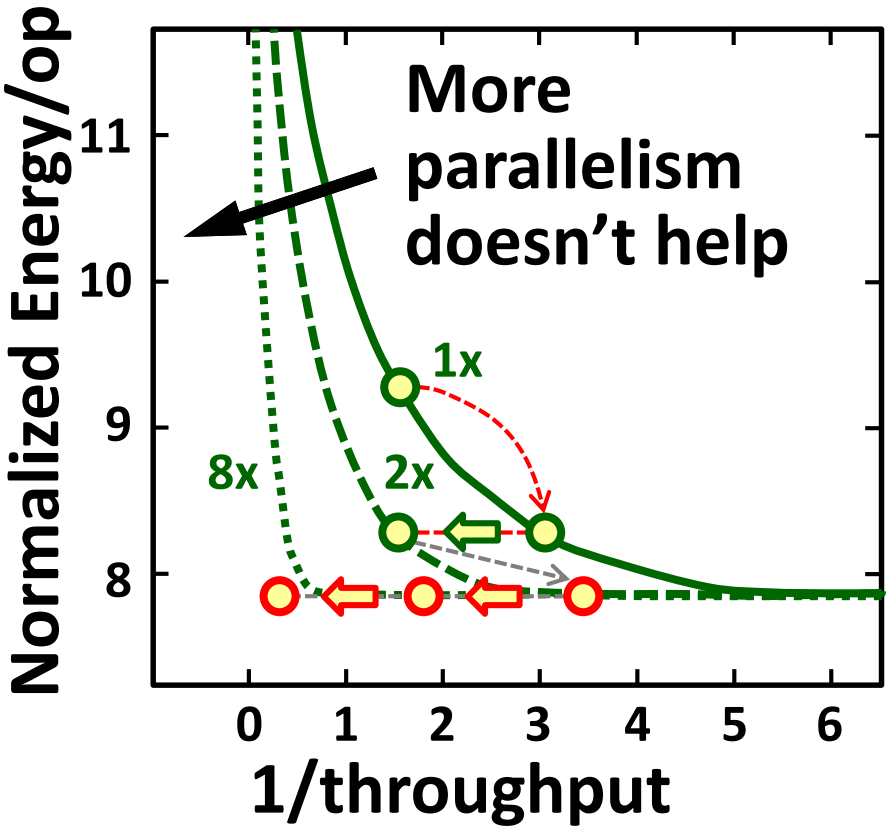
Increasing use of Concurrency Saturates



From: J. Rabaey (UCB)

Subthreshold Leakage: Game Over for CMOS

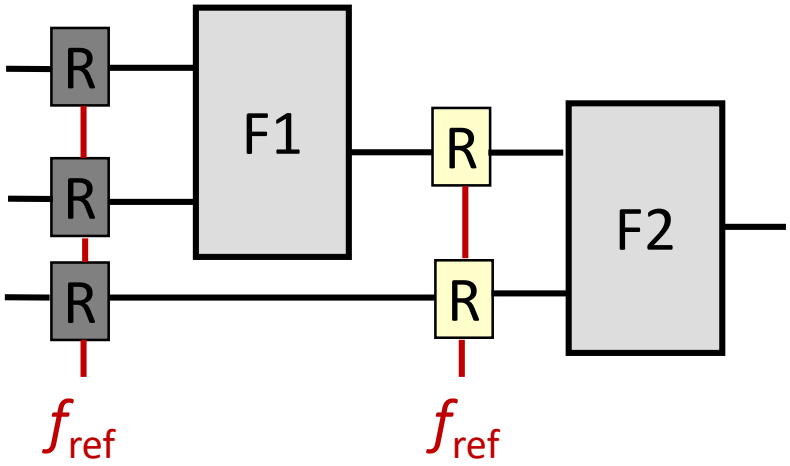




- Leakage and sub-threshold slope define minimum energy/op for CMOS
- Parallelism cannot reduce energy/op if operating at minimum energy/op

A Pipelined Implementation

Shallower logic reduces required supply voltage



$$f_{pipe} = f_{ref}$$

$$C_{pipe} = (1 + ov_{pipe}) \cdot C_{ref}$$

$$V_{DD,pipe} = \epsilon_{pipe} \cdot V_{DD,ref}$$

Power saving

Typically small

$$P_{pipe} = \square \cdot \square \cdot P_{ref}$$

Comparison of Par/Pipe @ Same V_{DD}

Parallel

(ov_{par} = 7.5%)

$$\frac{P_{par4}}{P_{ref}} = 0.52^2 \cdot \frac{4.3}{4} =$$

$$\frac{P_{par2}}{P_{ref}} = 0.66 \cdot \frac{2.15}{4} =$$

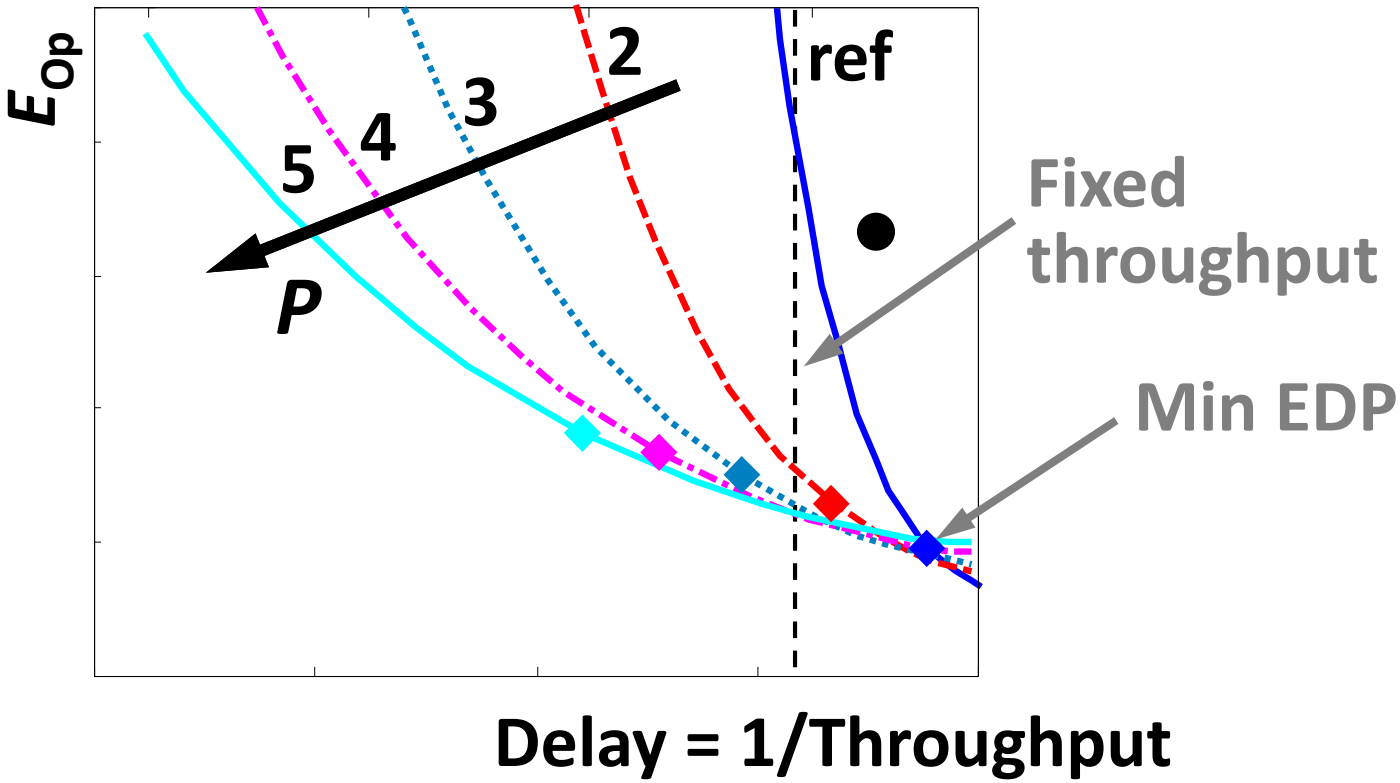
Pipeline

(ovpar = 10%)

$$\frac{P_{pipe4}}{P_{ref}} = 0.52^2 \cdot 1.1 =$$

$$\frac{P_{pipe2}}{P_{ref}} = 0.662 \cdot 1.1 =$$

Energy-Delay Space

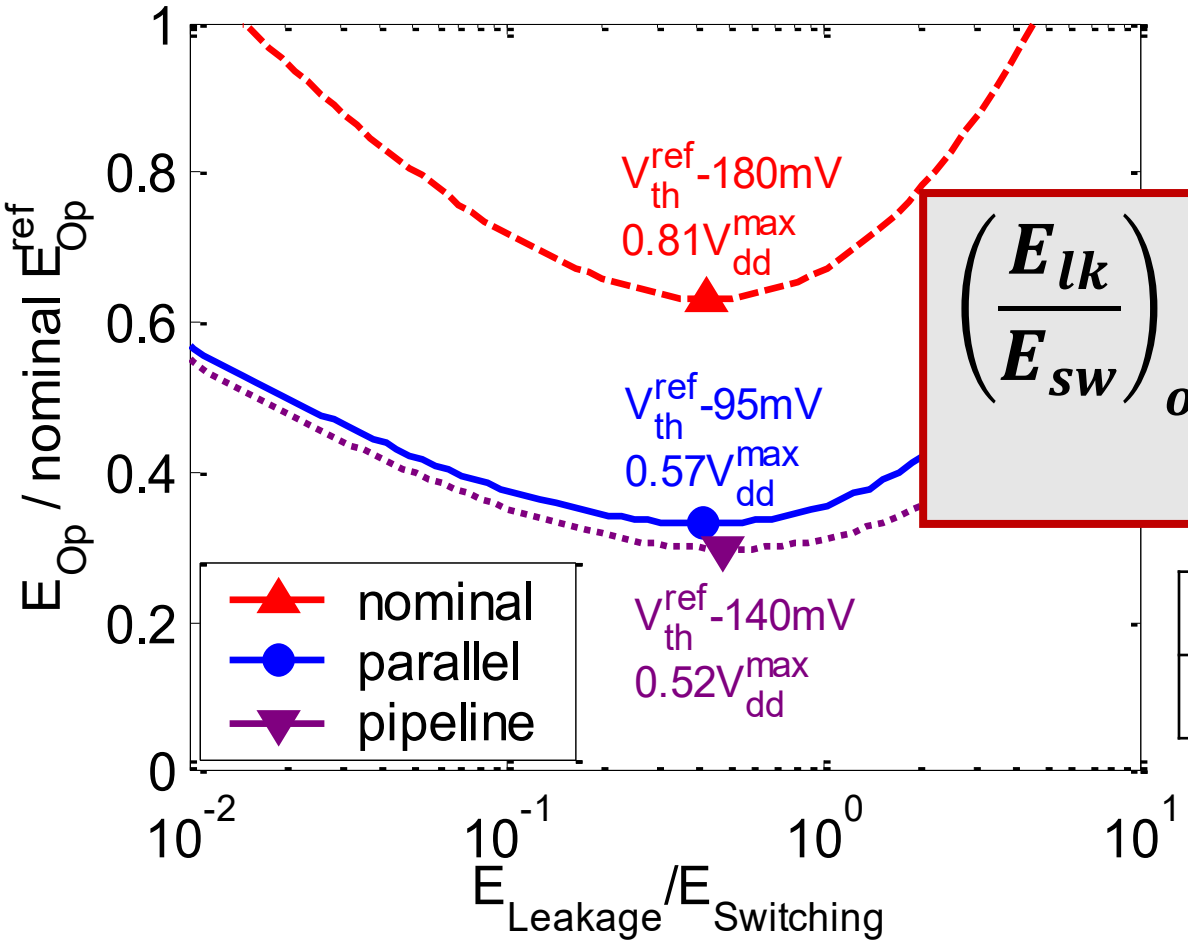


- Level of concurrency depends on target performance
- **Rule of thumb:** if speed exceeds MEP, add parallelism

Optimal Designs Have High Leakage

Set V_T, find V_{DD} to min E_{Op} for a fixed performance

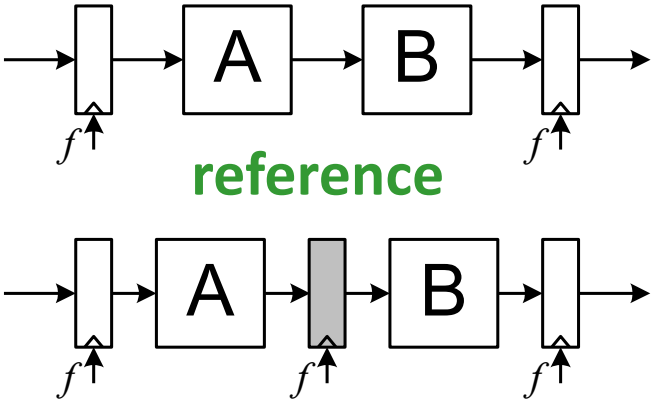
from 13.6

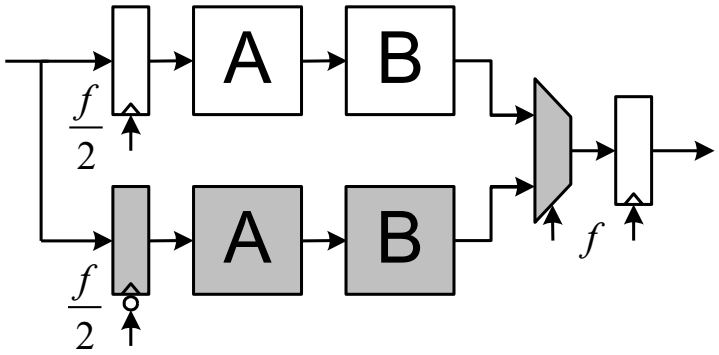


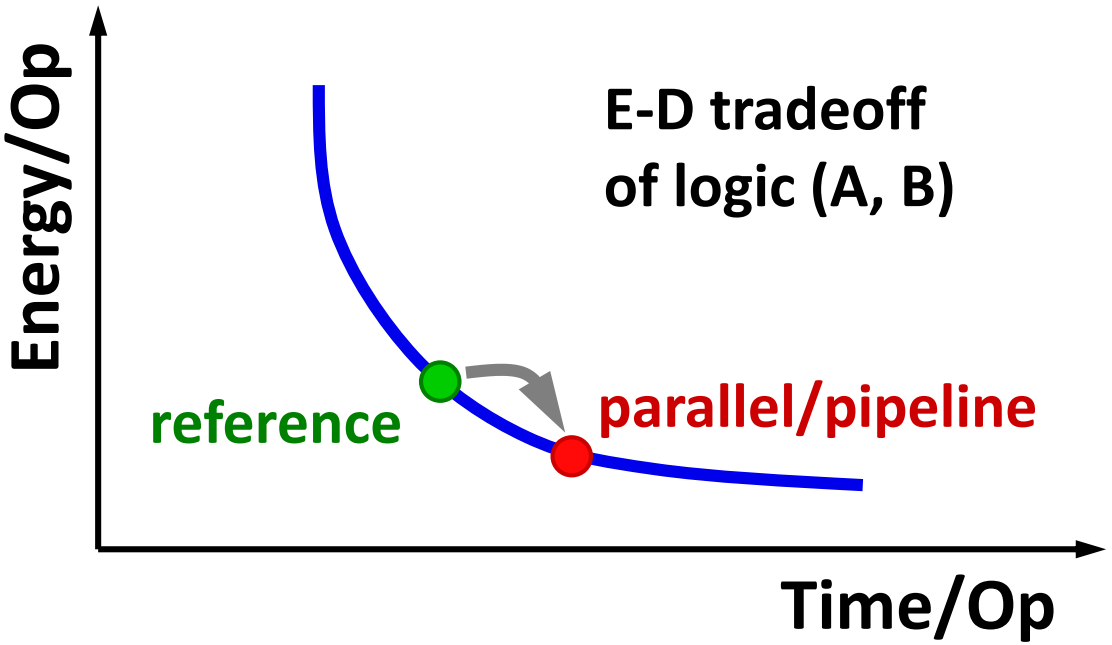
$$\left(\frac{E_{lk}}{E_{sw}}\right)_{opt} = \frac{2}{ln\left(\frac{L_d}{\alpha_{avg}}\right) - K}$$

Topology	Inv	Add	Dec
$(E_{Lk}/E_{Sw})^{opt}$	0.8	0.5	0.2

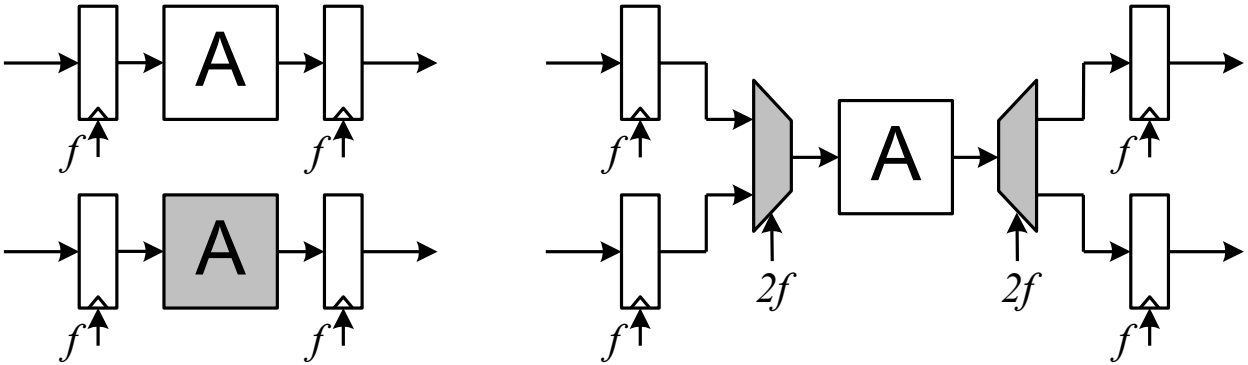
Parallelism and Pipelining in E-D Space





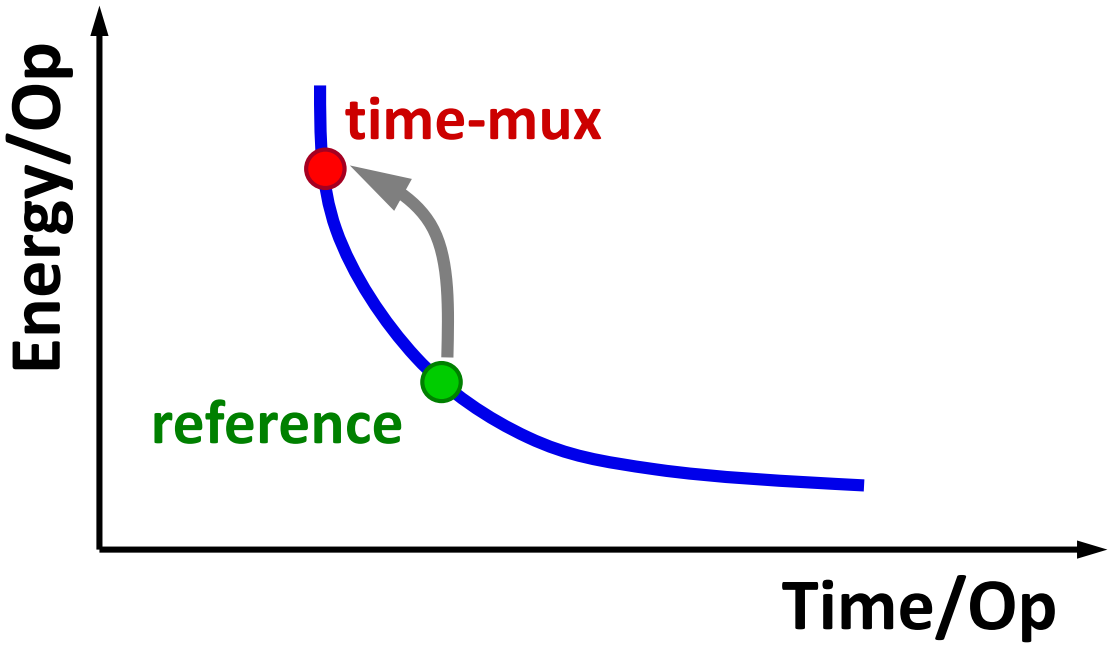


Time Multiplexing



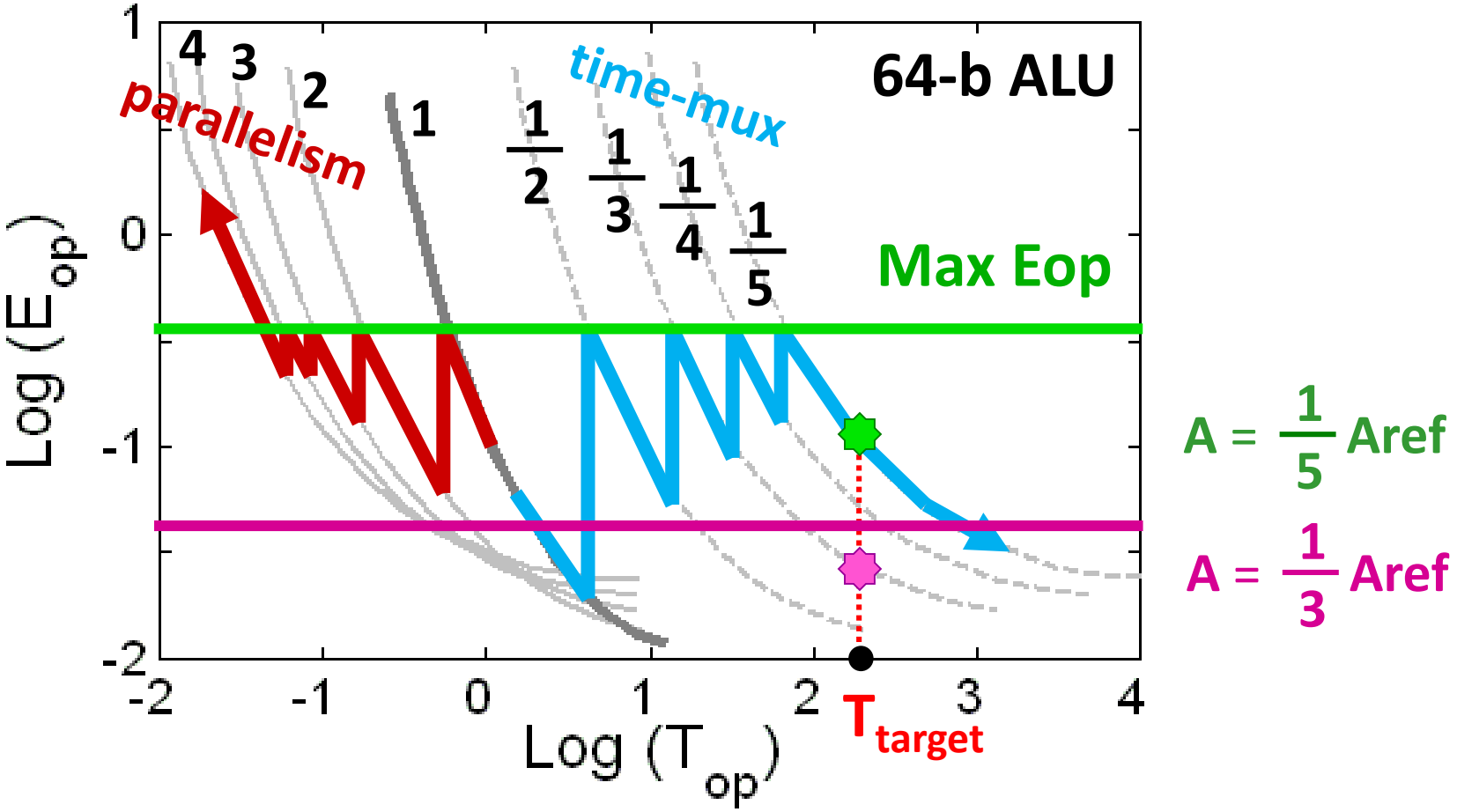
reference

time-mux



Energy-Area Tradeoff

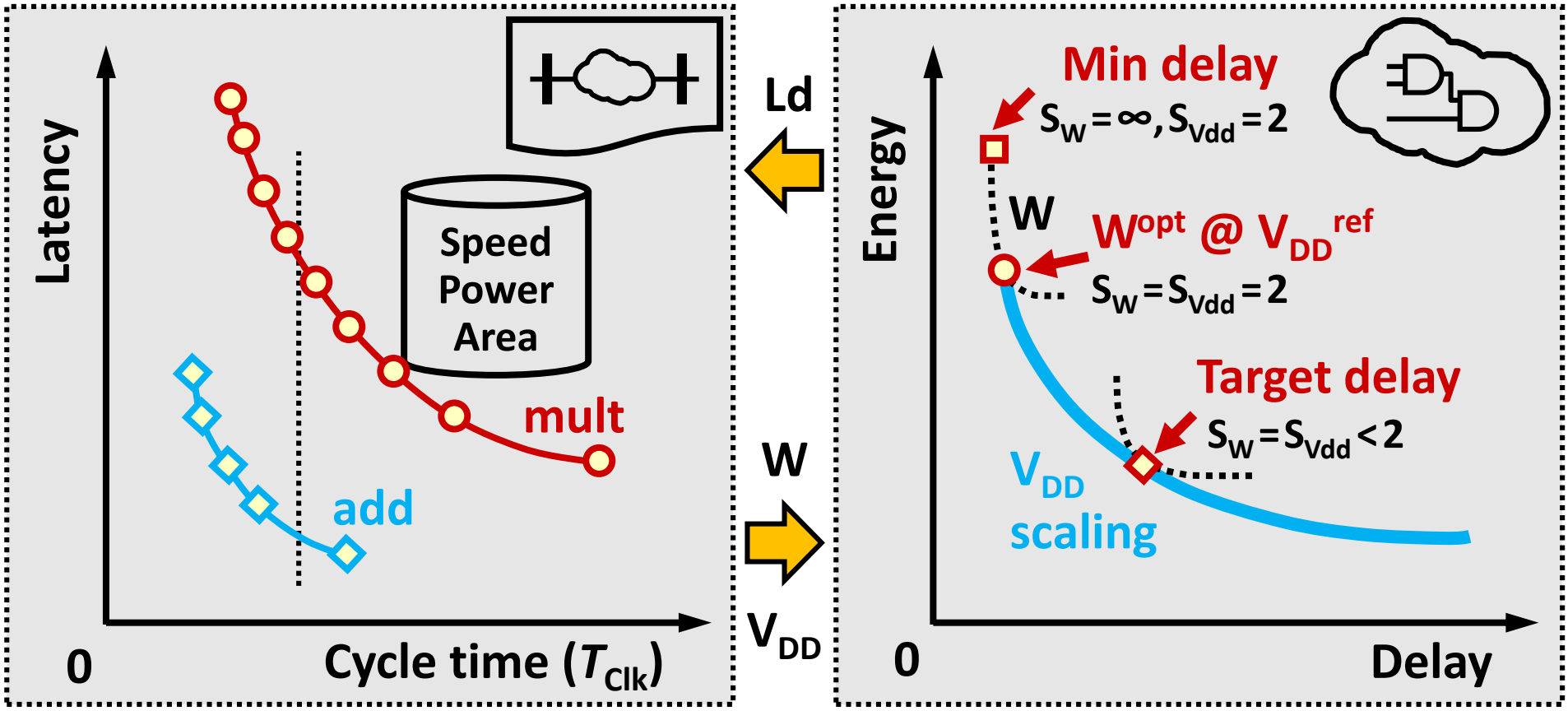
High throughput: Parallelism = Large Area



Low throughput: Time-Mux = Small Area

Putting it All Together

- Balance logic depth (L_d), adjust latency to reach T_{Clk}
- Optimize V_{DD} and W of the underlying pipelines



Micro Arch. (block) **Circuit (datapath)**

Summary: Design Guidelines

- **For maximum performance**
  - Maximize use of concurrency at the cost of area
- **For given performance**
  - Optimal amount of concurrency for minimum energy
- **For given energy**
  - Least amount of concurrency that meets performance
- **For minimum energy**
  - Solution with minimum overhead
    (direct mapping between function and architecture)

System Perspective

- **Optimizations at higher abstraction levels have greater potential impact**
- **While circuit techniques may yield improvements in the 10-50% range**
- **Architecture and algorithm optimizations can reach orders of magnitude power reduction**