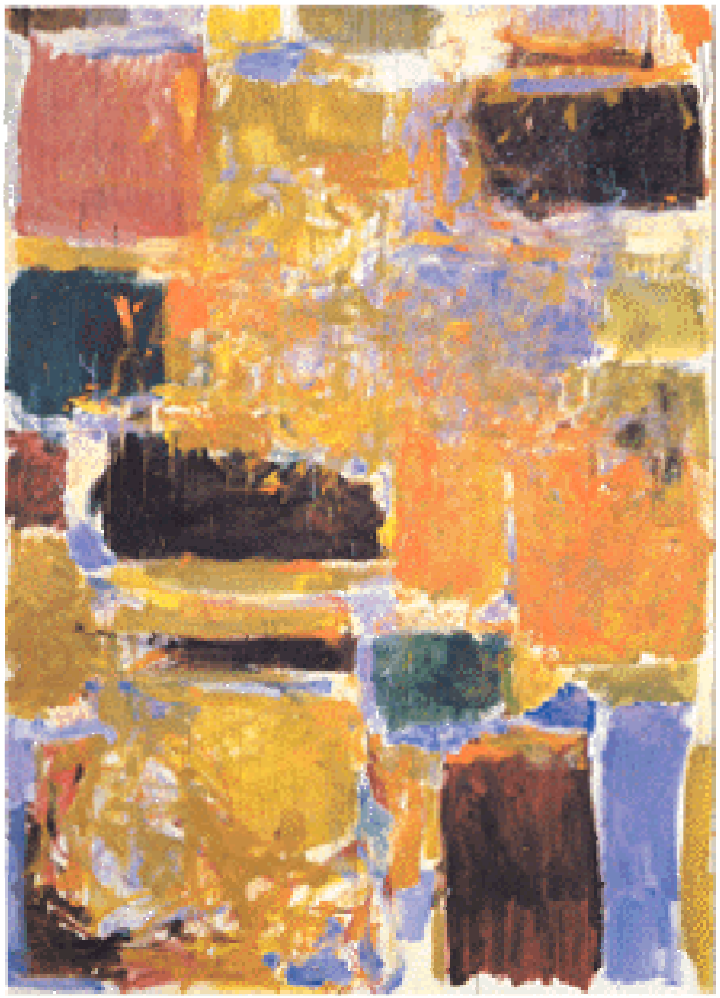
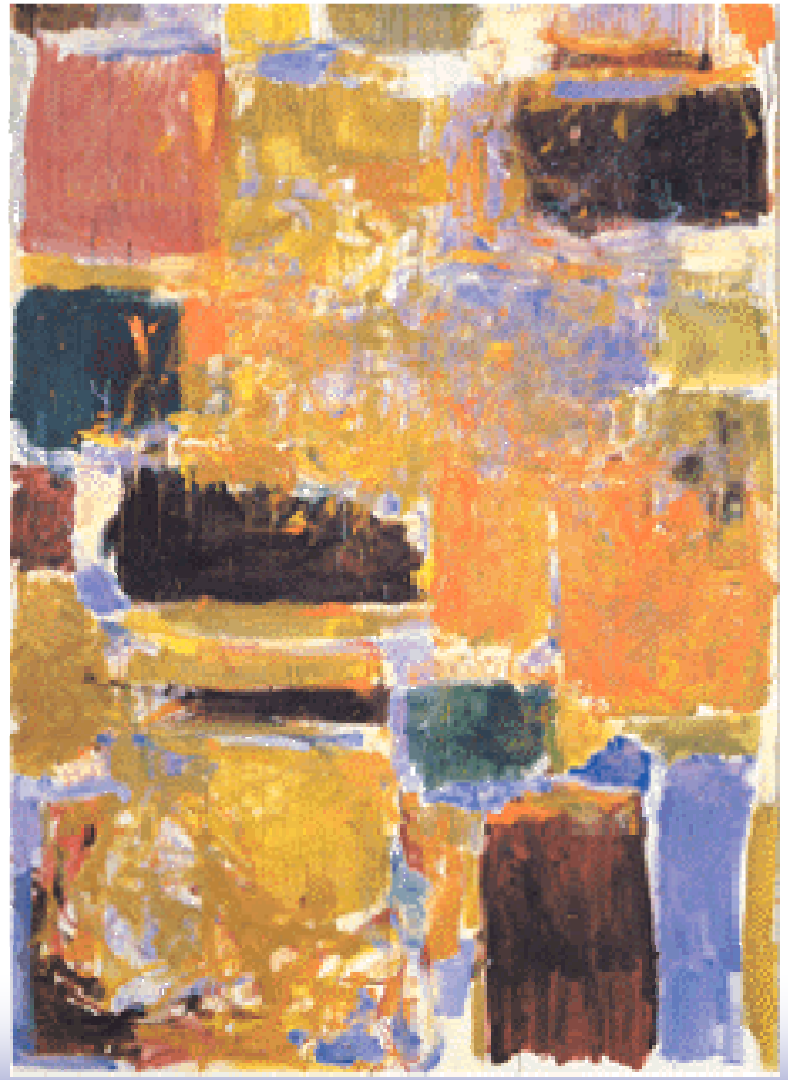


CS150 - EE141/241A
Fall 2014
Digital Design and Integrated Circuits

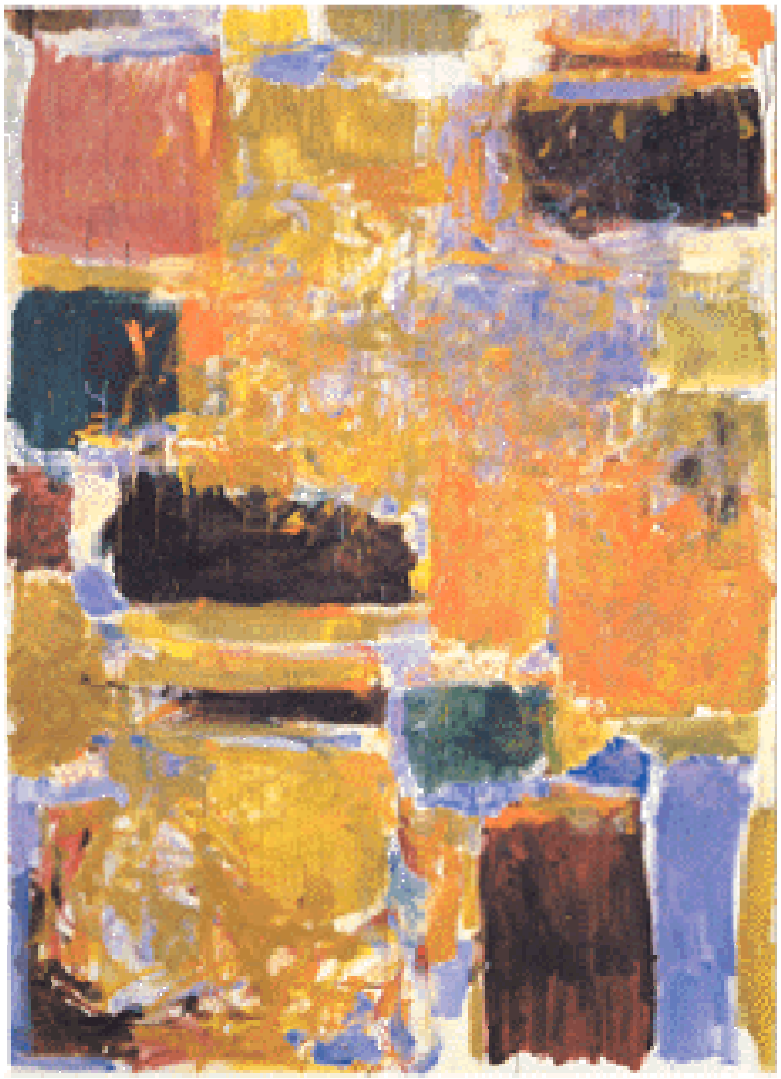
Instructors:
John Wawrzynek and Vladimir Stojanovic

Lecture 11

Outline

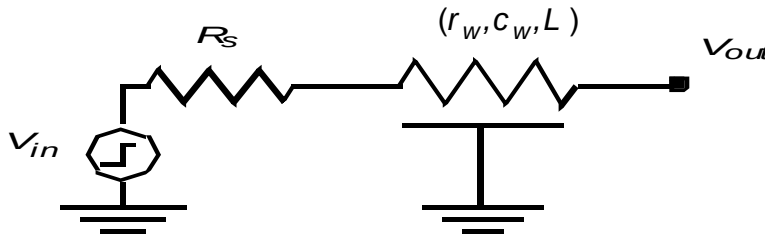


- Energy and Power metrics
- Energy and Power in logic
- Energy-delay trade-offs
- Voltage-scaling
- Managing Leakage



Gates and Wires

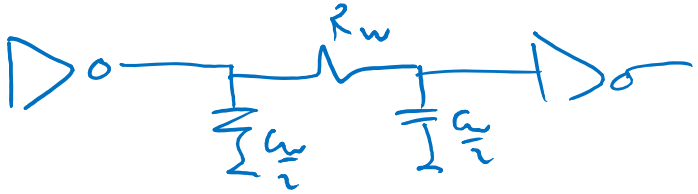
Driving an RC-line



$$\tau_D = R_s C_w + \frac{R_w C_w}{2} = R_s C_w + 0.5 r_w c_w L^2$$

$$t_p = 0.69 R_s C_w + 0.38 R_w C_w$$

The Global Wire Problem



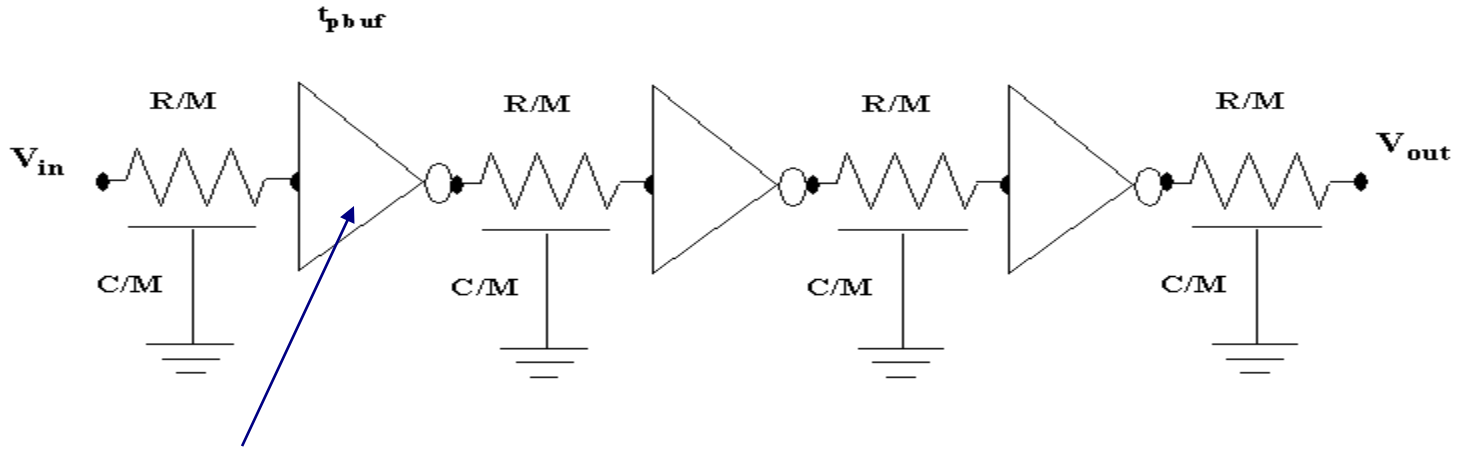
$R \cdot L \; C \cdot L \sim L^2$

$$Td = 0.38R_wC_w + 0.69(R_NC_{in} + R_NC_w + R_wC_{in})$$

Challenges

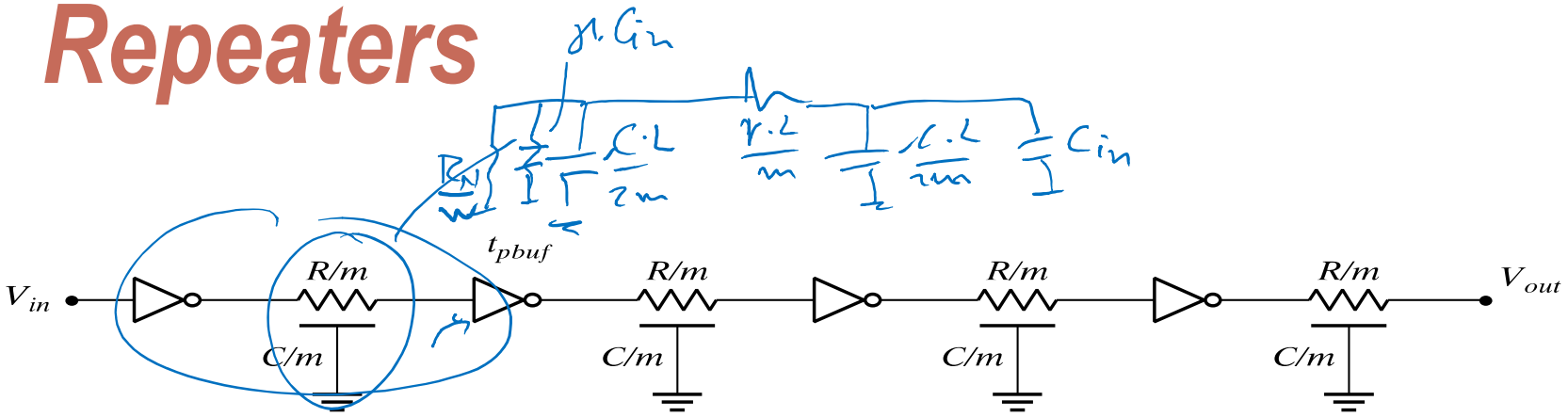
- No further improvements to be expected after the introduction of Copper (superconducting, optical?)
- Design solutions
  - Use of fat wires
  - Efficient chip floorplanning
  - Insert repeaters

Reducing RC-delay Using Repeaters



Repeater

Repeaters



$$t_p = 0.69m\left(\frac{R_N}{W}(W\gamma C_{in} + \frac{cL}{m} + WC_{in}) + \frac{rL}{m}(WC_{in} + 0.5\frac{cL}{m})\right)$$

$$m_{opt} = L\sqrt{\frac{0.38rc}{0.69R_N C_{in}(\gamma+1)}} = \sqrt{\frac{t_{pwire(unbuffered)}}{t_{p1}}}$$

$$W_{opt} = \sqrt{\frac{R_N c}{rC_{in}}}$$

Repeater Insertion

$$m_{opt} = L\sqrt{\frac{0.38rc}{0.69R_N C_{in}(\gamma+1)}} = \sqrt{\frac{t_{pwire(unbuffered)}}{t_{p1}}}$$

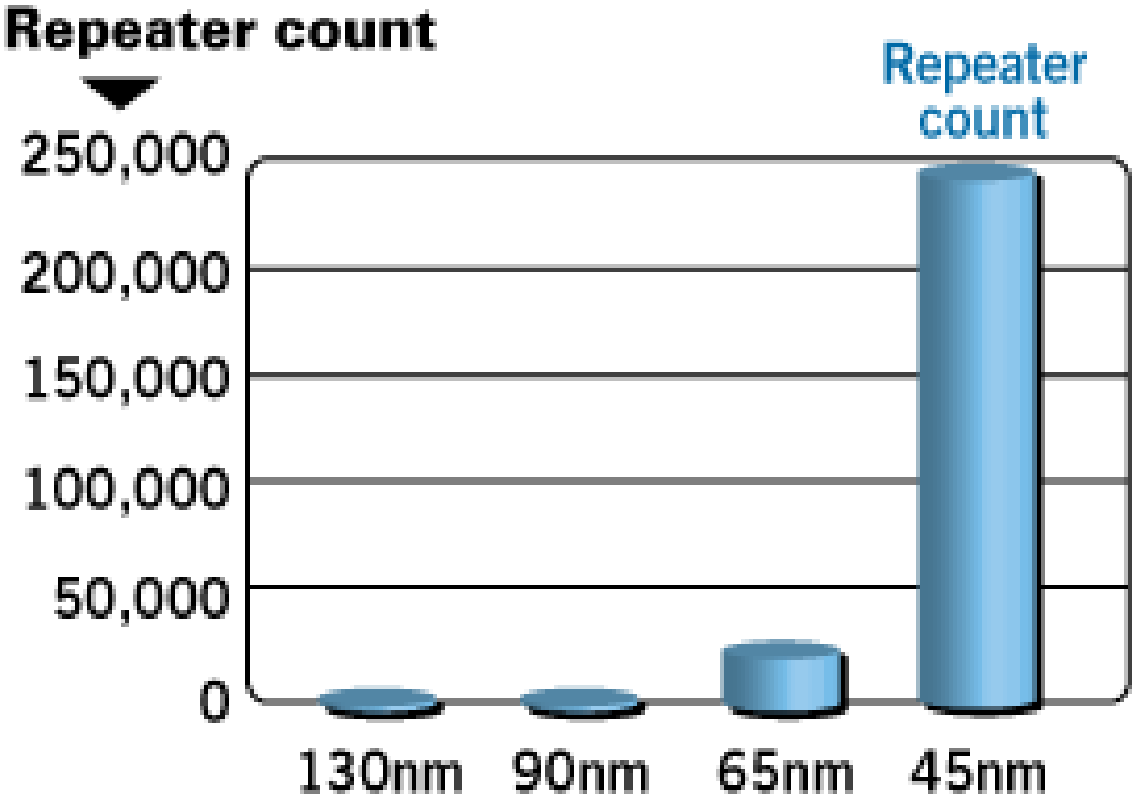
$$W_{opt} = \sqrt{\frac{R_N c}{r C_{in}}}$$

For a given technology and a given interconnect layer, there exists an optimal length of the wire segments between repeaters. The delay of these wire segments is independent of the routing layer!

$$L_{crit} = \frac{L}{m_{opt}} = \sqrt{\frac{t_{p1}}{0.38rc}} \qquad t_{p,crit} = \frac{t_{p,min}}{m_{opt}} = 2\left(1+\sqrt{\frac{0.69}{0.38(1+\gamma)}}\right)t_{p1}$$

From Elmore example: rc = 0.1fs/um², tp1 = 55ps, L_{crit} = 741um (rule of thumb ~ 0.5-1mm)

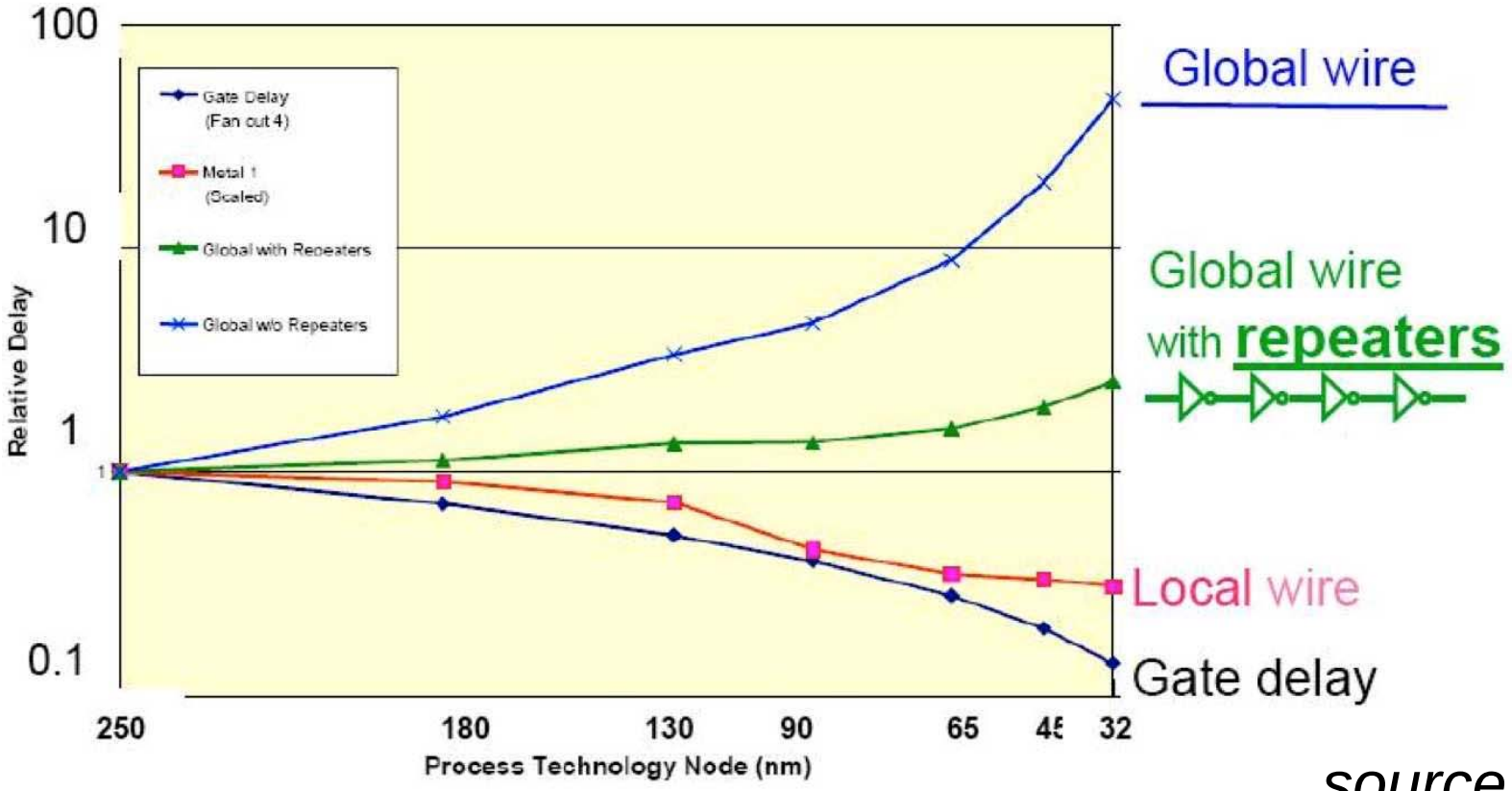
Importance of Repeaters



Source: IBM POWER processors,
R. Puri et al SRC Interconnect Forum 2006

- In modern designs the number of repeaters increases dramatically

Wire and Gate Delay Scaling



source: ITRS

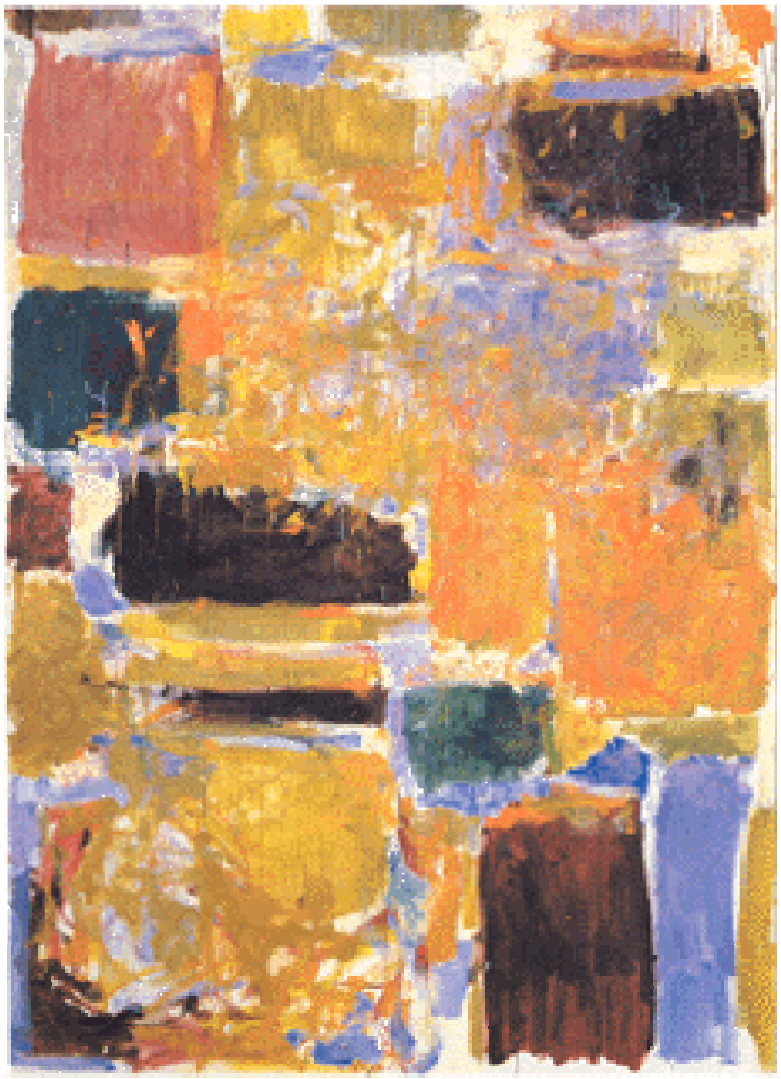
Delay for Metal 1 and Global Wiring versus Feature Size

- Gate delay gets better, wire delay gets worse

Visualization

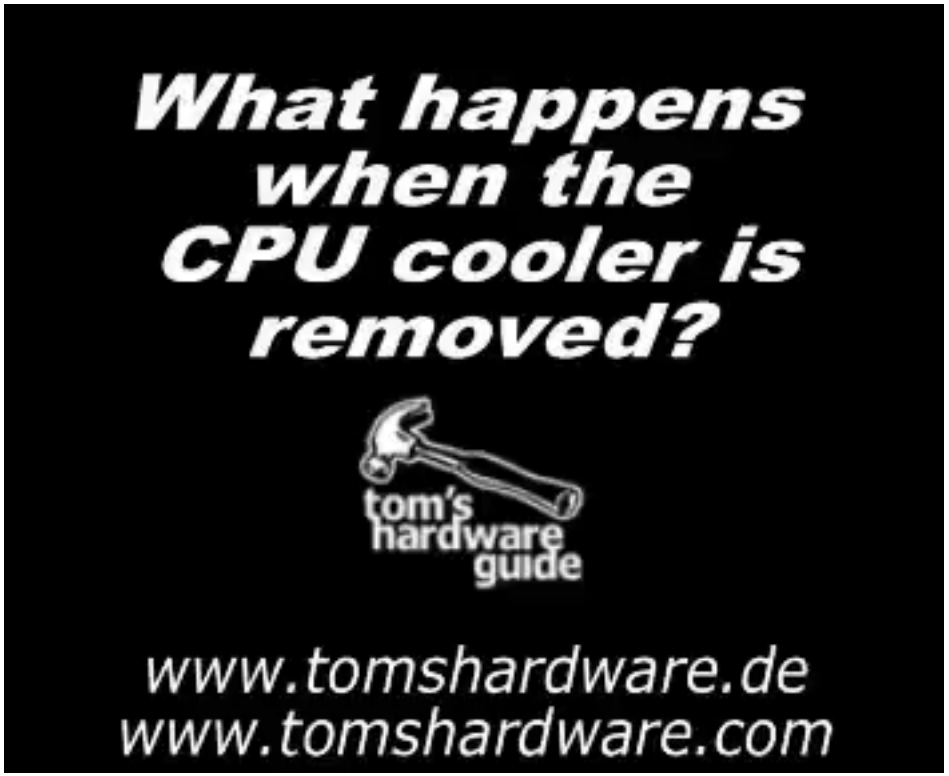
Play with the model and see how currents and voltages change

http://www.research.ibm.com/people/r/restle/Animations/DAC01top.html



Design Metric: Energy and Power

Why Power is Important



Where Does Power Go in CMOS?

- Switching power
  - Charging/discharging capacitors
- Short-circuit power
  - Both pull-up and pull-down on during transition
- Leakage power
  - Transistors are imperfect switches
- Static currents
  - Biasing currents, in e.g. analog, memory

Power Dissipation

Instantaneous power:

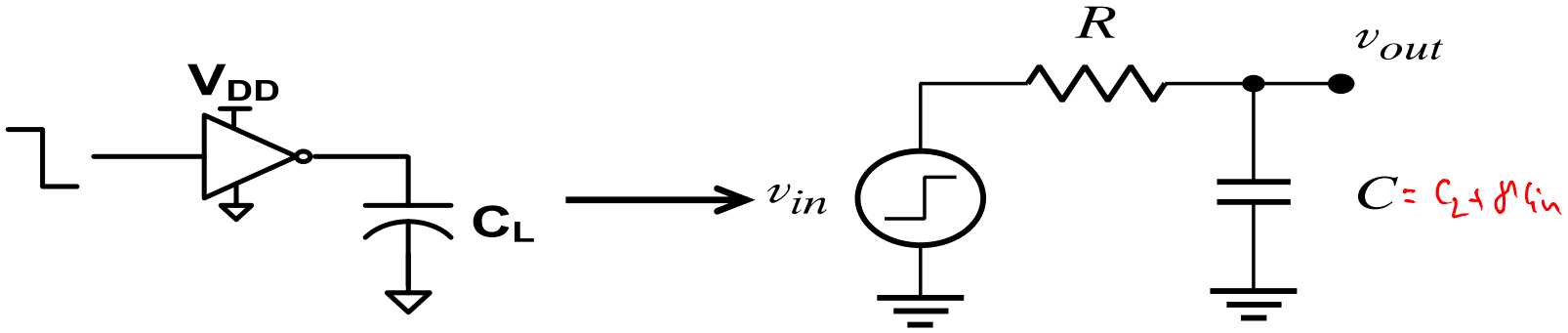
$$p(t) = v(t)i(t) = V_{supply}i(t)$$

Peak power:

$$P_{peak} = V_{supply}i_{peak}$$

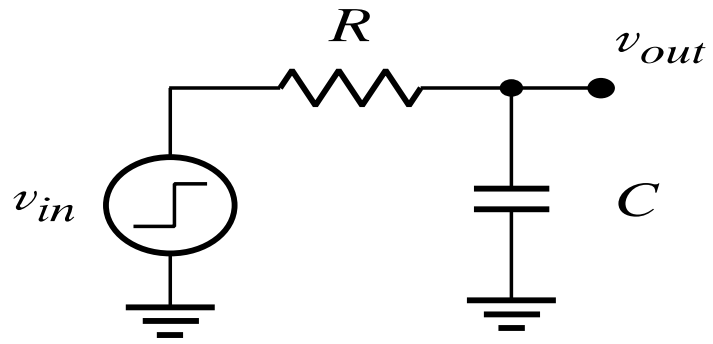
Average power: $P_{ave} = \frac{1}{T}\int_{t}^{t+T} p(t)\,dt = \frac{V_{supply}}{T}\int_{t}^{t+T} i_{supply}(t)\,dt$

Energy in Switch Logic



- The voltage on C_L eventually settles to V_{DD}
- Thus, charge stored on the capacitor is $C_L V_{DD}$
  - This charge has to flow out of the power supply
- So, energy is just $Q \cdot V_{DD} = (C_L V_{DD}) \cdot V_{DD}$

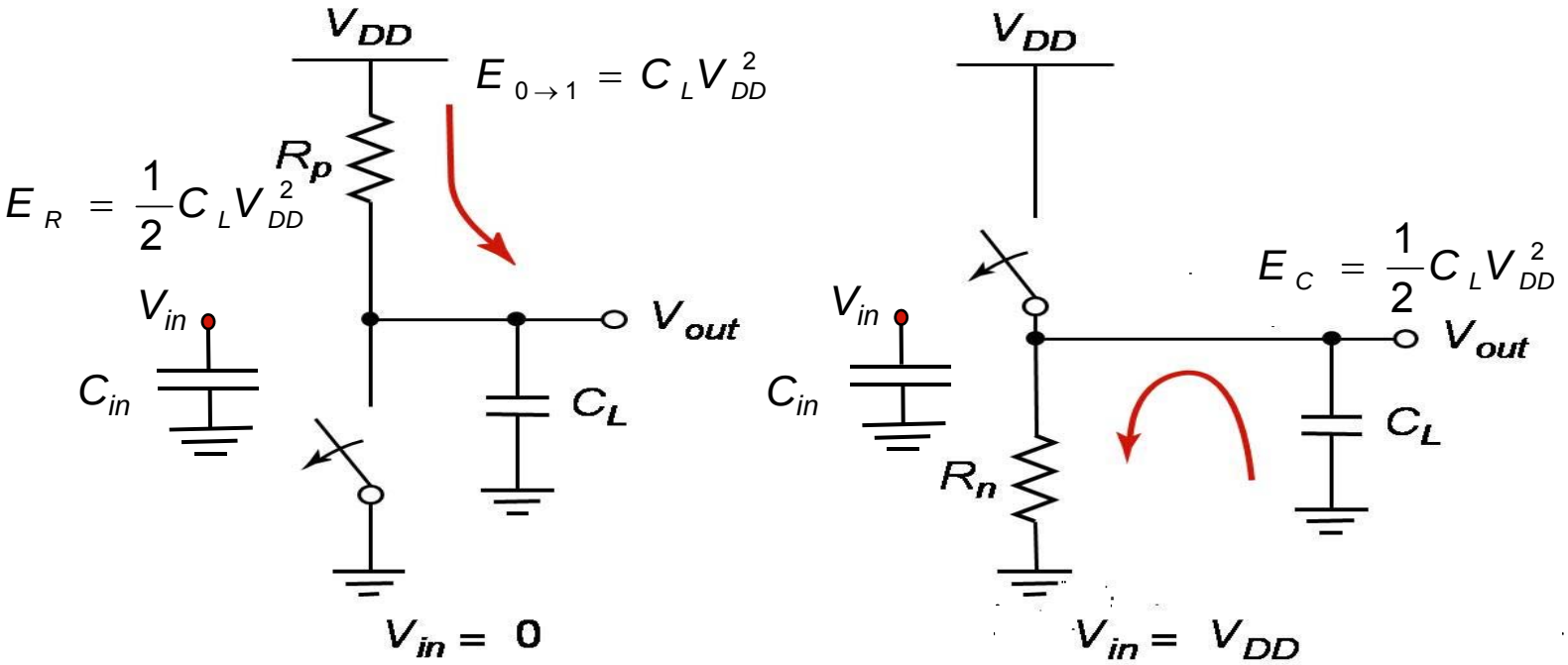
Energy (the harder way)



$$E_{0\to 1} = \int_0^T P_{DD}(t)dt = V_{DD}\int_0^T i_{DD}(t)dt = V_{DD}\int_0^{V_{DD}} C_L\, dv_{out} = C_L V_{DD}^2$$

$$E_C = \int_0^T P_C(t)dt = \int_0^T v_{out}\, i_L(t)dt = \int_0^{V_{DD}} C_L v_{out}\, dv_{out} = \frac{1}{2} C_L V_{DD}^2$$

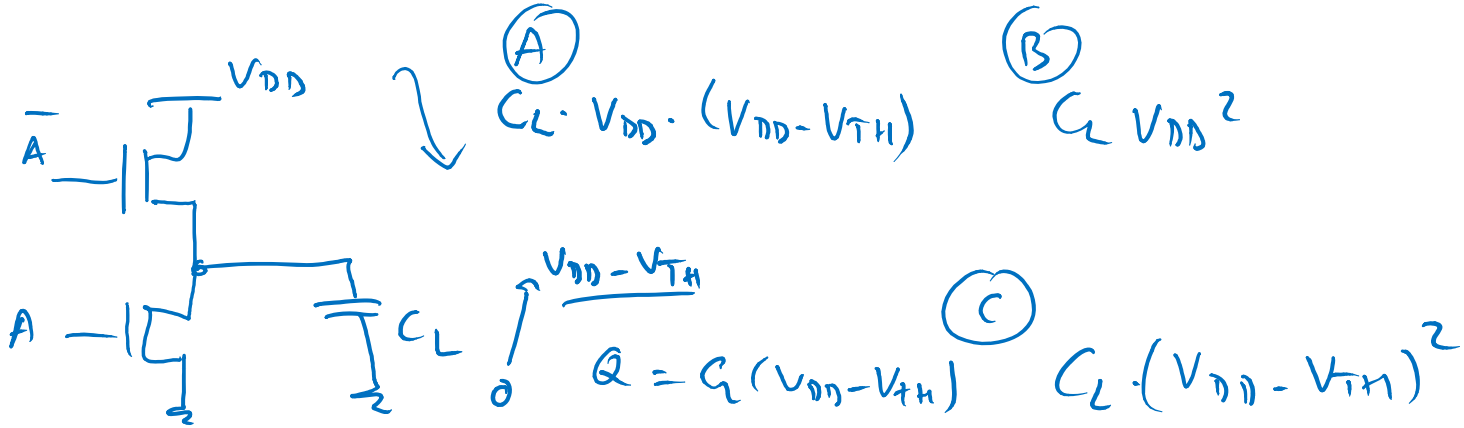
The Switch Inverter: Energy



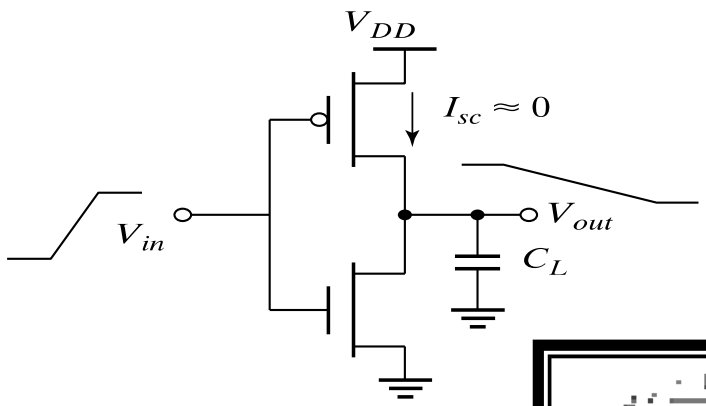
(a) Low-to-high

(b) High-to-low

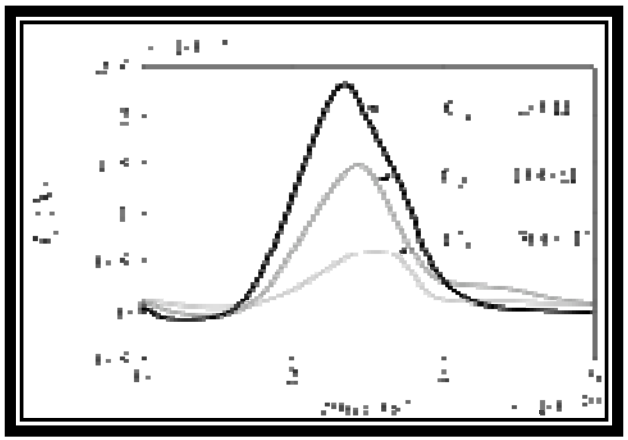
Imperfect Switch: Energy

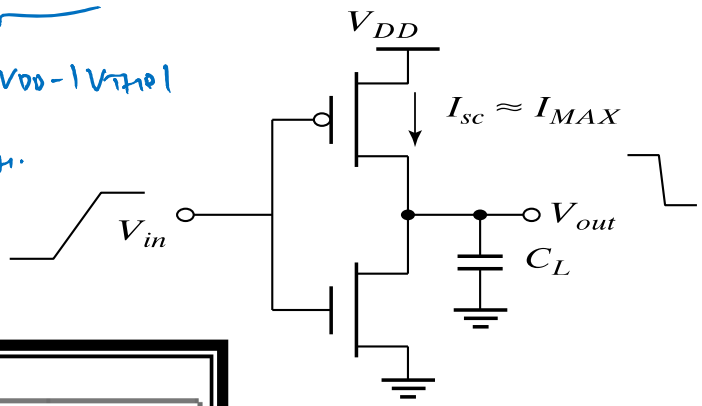


Short Circuit Current



Large load

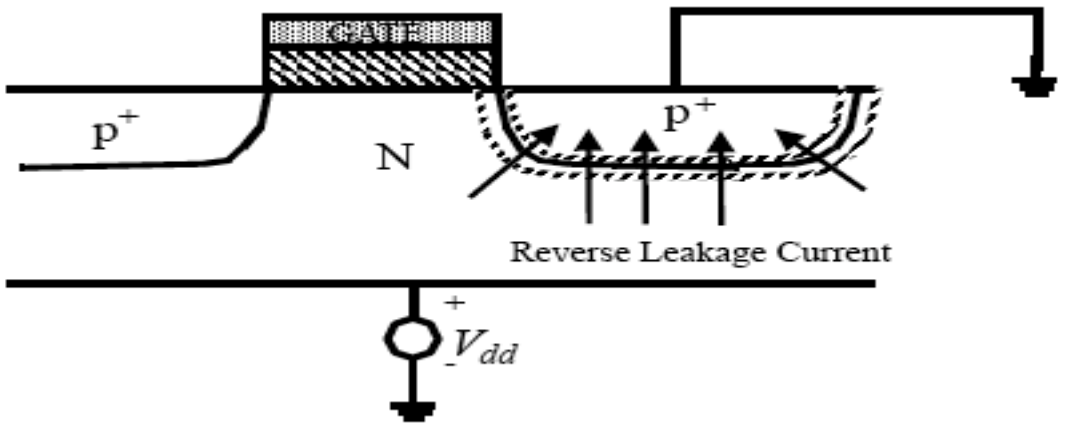




Small load

- Short circuit current usually well controlled

Diode Leakage



$$I_{DL} = J_S \acute{} A$$

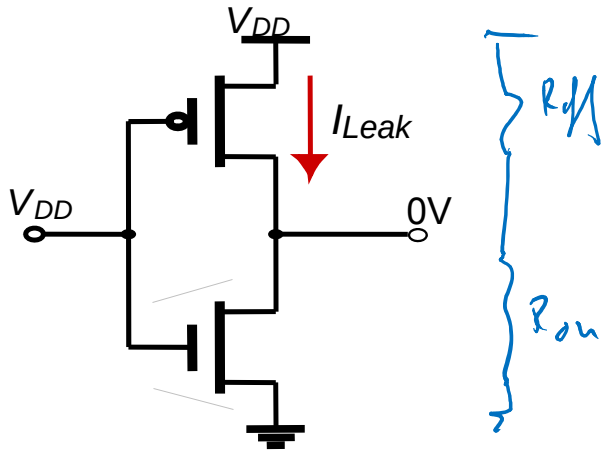
JS = 10-100 pA/μm2 at 25 deg C for 0.25μm CMOS

JS doubles for every 9 deg C!

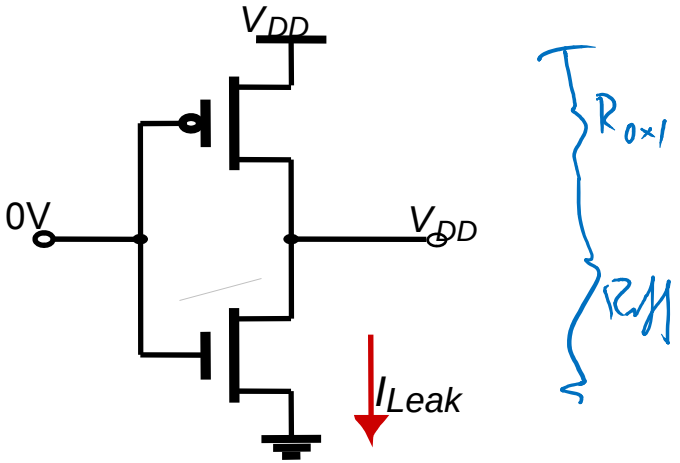
Much smaller than transistor leakage in deep submicron

Transistor Leakage

- Transistors that are supposed to be off - leak

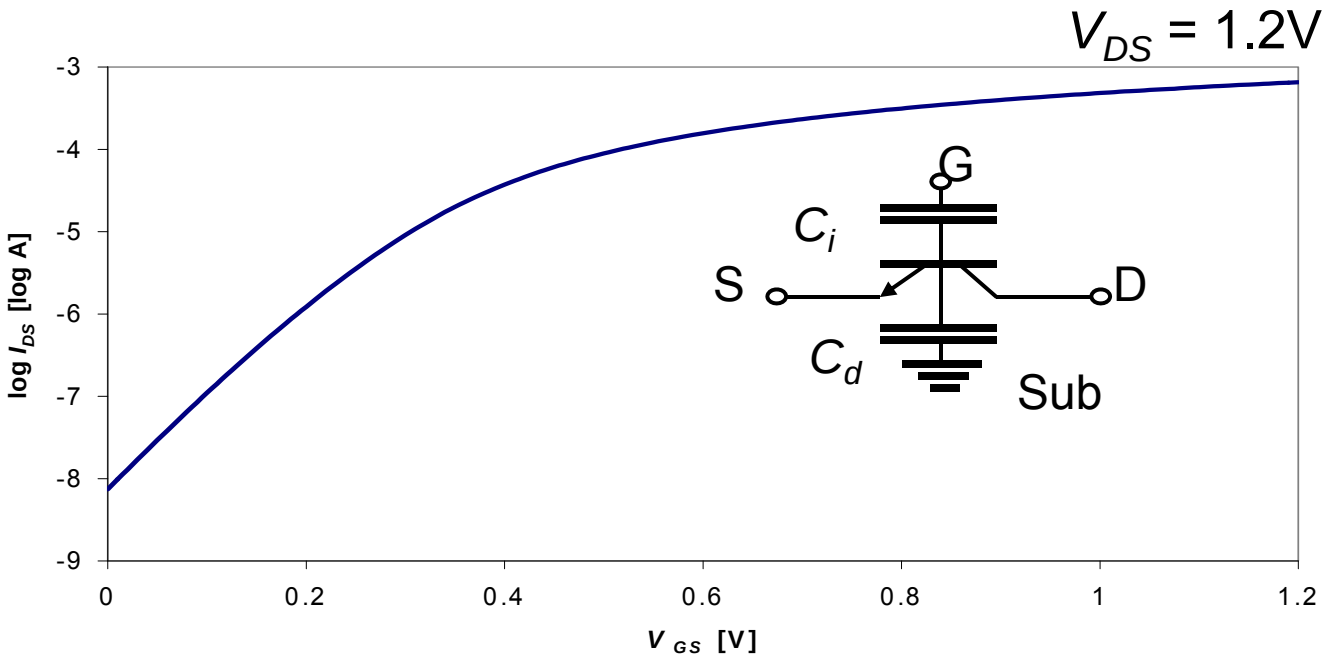


Input at V_{DD}



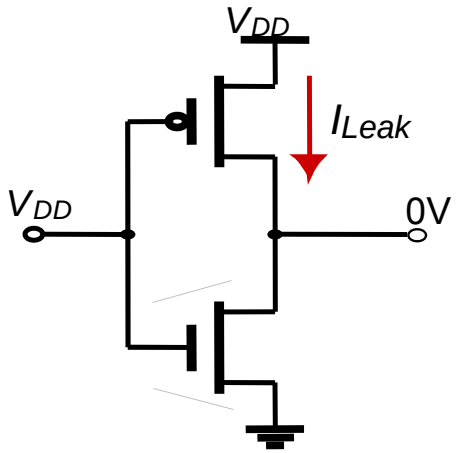
Input at 0

Transistor Leakage

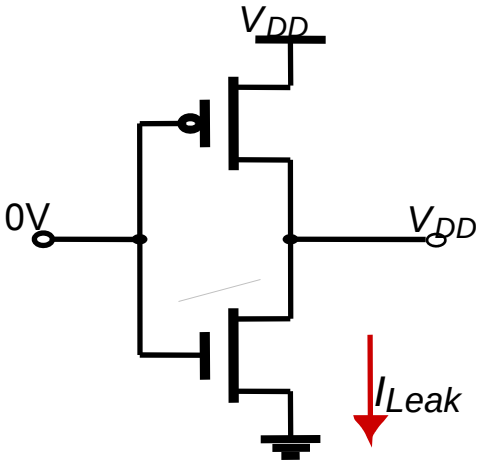


Drain leakage current is exponential with V_{GS}-V_T

Roff



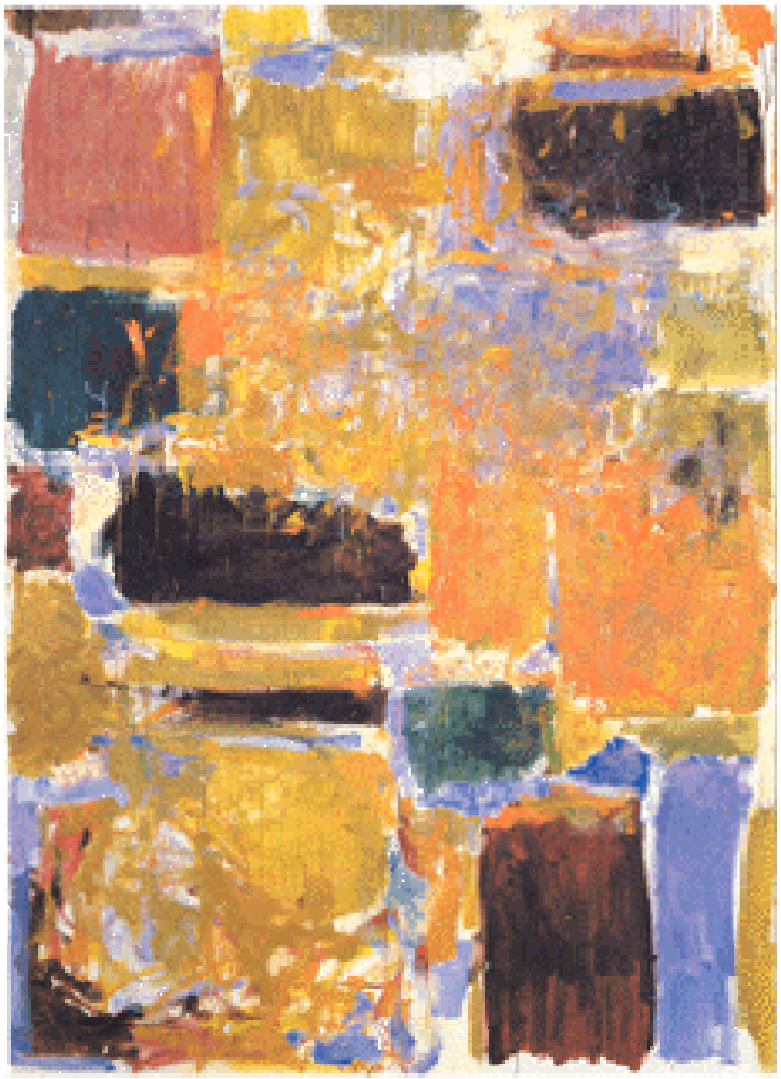
Input at V_{DD}



Input at 0

$$I_{Leak} \sim I_0 e^{\frac{-V_T}{nkT/q}}$$

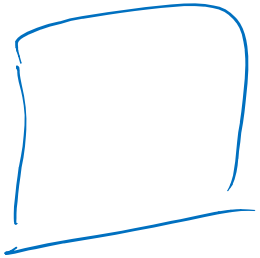
$$R_{off} = \left(V_{DD} / I_0\right) e^{\frac{V_T}{nkT/q}}$$



Energy and Power in Logic

Transition Activity and Power

- Energy consumed in N cycles, E_N:



$$E_N = C_L \bullet V_{DD}^2 \bullet n_{0\to1}$$

$n_{0\to1}$ – number of $0\to1$ transitions in N cycles

$$P_{avg} = \lim_{N\to\infty} \frac{E_N}{N} \cdot f = \left(\lim_{N\to\infty} \frac{n_{0\to1}}{N} \right) \cdot C_L \cdot V_{DD}^2 \cdot f$$

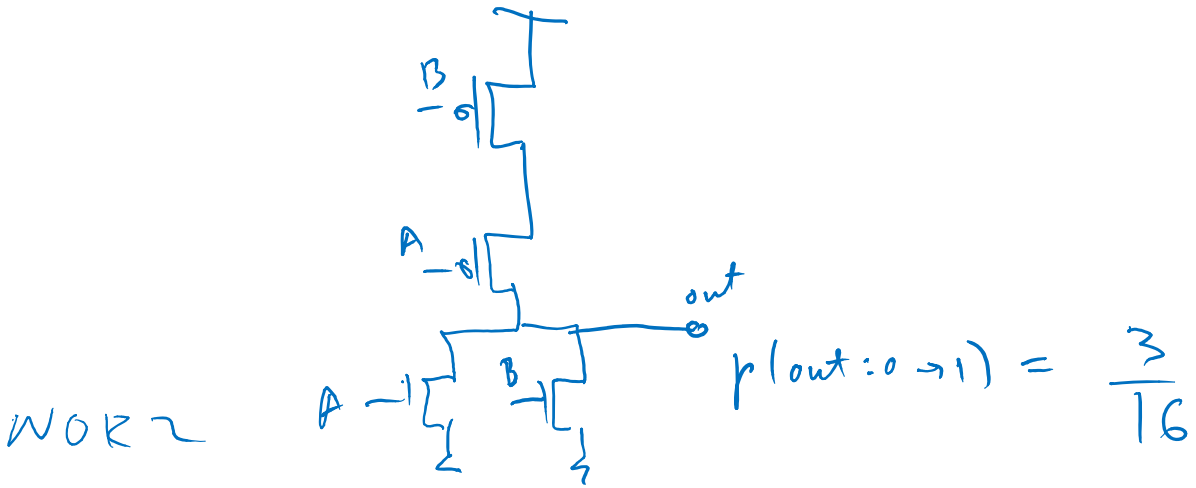
$$\alpha_{0\to1} = \lim_{N\to\infty} \frac{n_{0\to1}}{N} \cdot f$$

$$\boxed{P_{avg} = \alpha_{0\to1} \cdot C_L \cdot V_{DD}^2 \cdot f}$$

Factors Affecting Transition Activity

- "Static" component (does not account for timing)
  - Type of logic function (NOR vs. XOR)
  - Type of logic style (static, pass-gate, dynamic)
  - Signal statistics
  - Inter-signal correlations
- "Dynamic" or timing dependent component
  - Circuit topology
  - Signal statistics and correlations

Type of Logic Function or Family



$p(out: 0 \rightarrow 1) = \frac{3}{16}$

NOR2

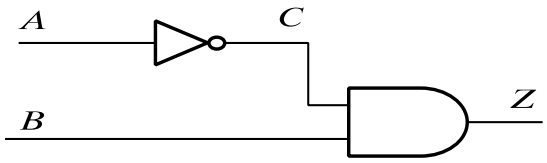
A	B	out
0	0	1
0	1	0
1	0	0
1	1	0

$p(out = 1) = \frac{1}{4}$

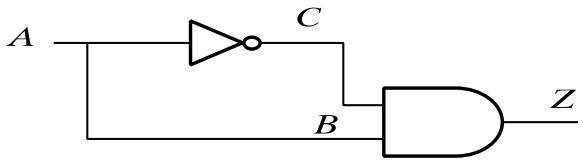
$p(out = 0) = \frac{3}{4}$

Inter-Signal Correlations

- Need to use conditional probabilities to model inter-signal correlations
- CAD tools best for performing such analysis



(a) Logic circuit without reconvergent fan-out



(b) Logic circuit with reconvergent fan-out

Logic without reconvergent fanout

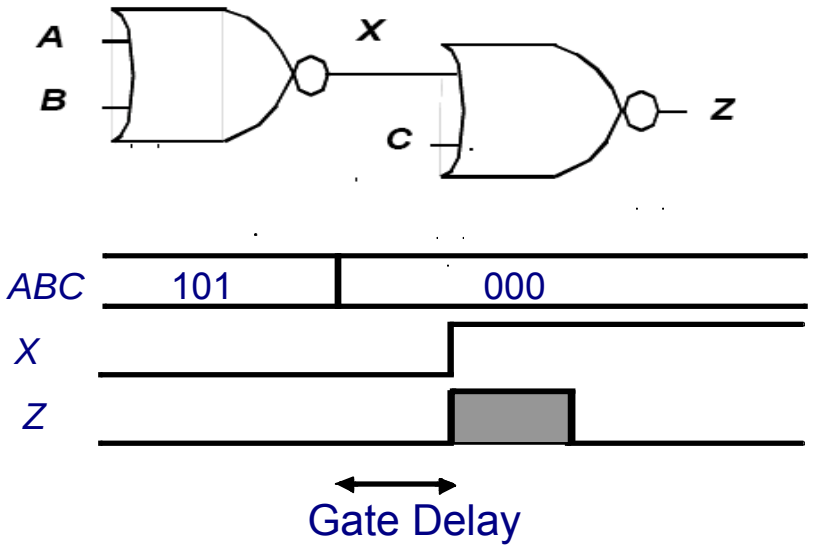
$p_{0\to1} = (1 - p_{\bar{A}} p_B)\, p_{\bar{A}} p_B$

Logic with reconvergent fanout

$P(Z = 1) = p(C=1 \mid B=1)\, p(B=1)$

$p_{0\to1} = 0$

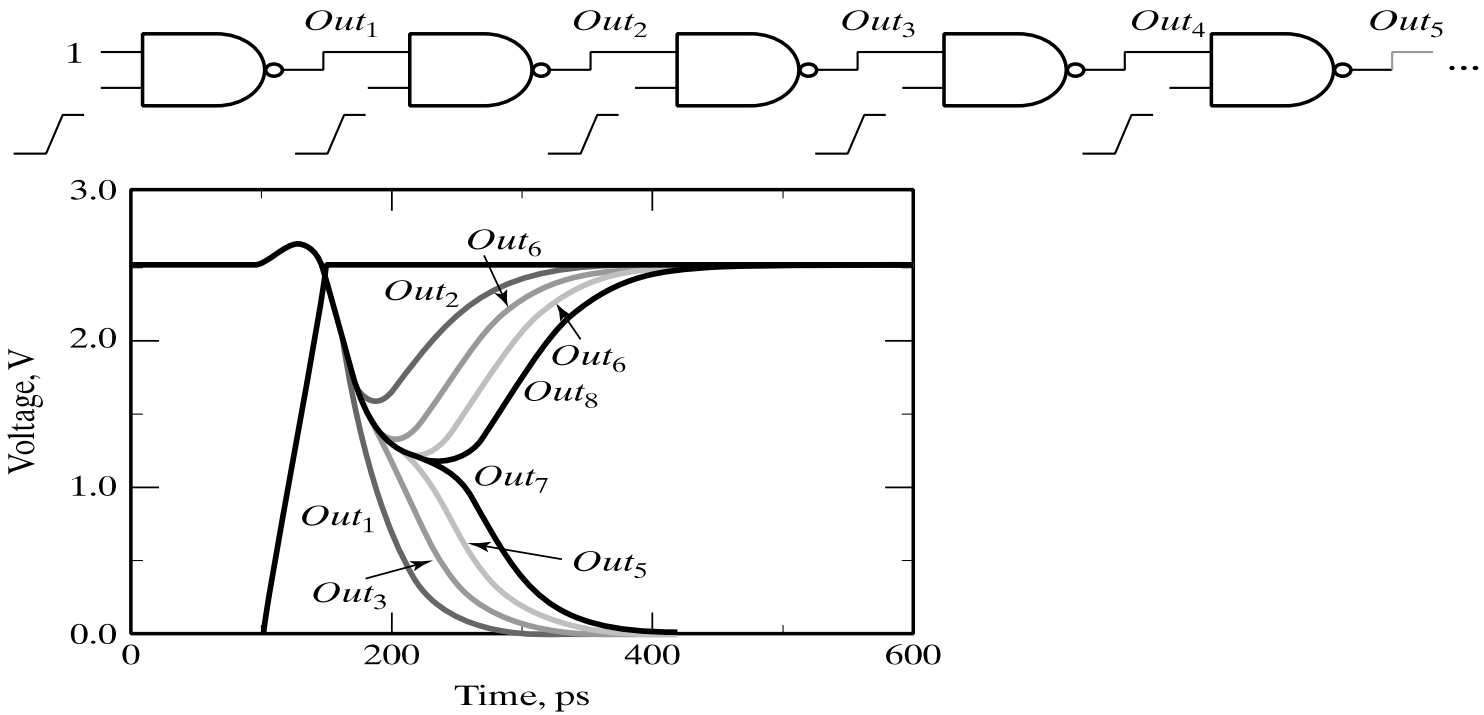
Glitching in Static CMOS

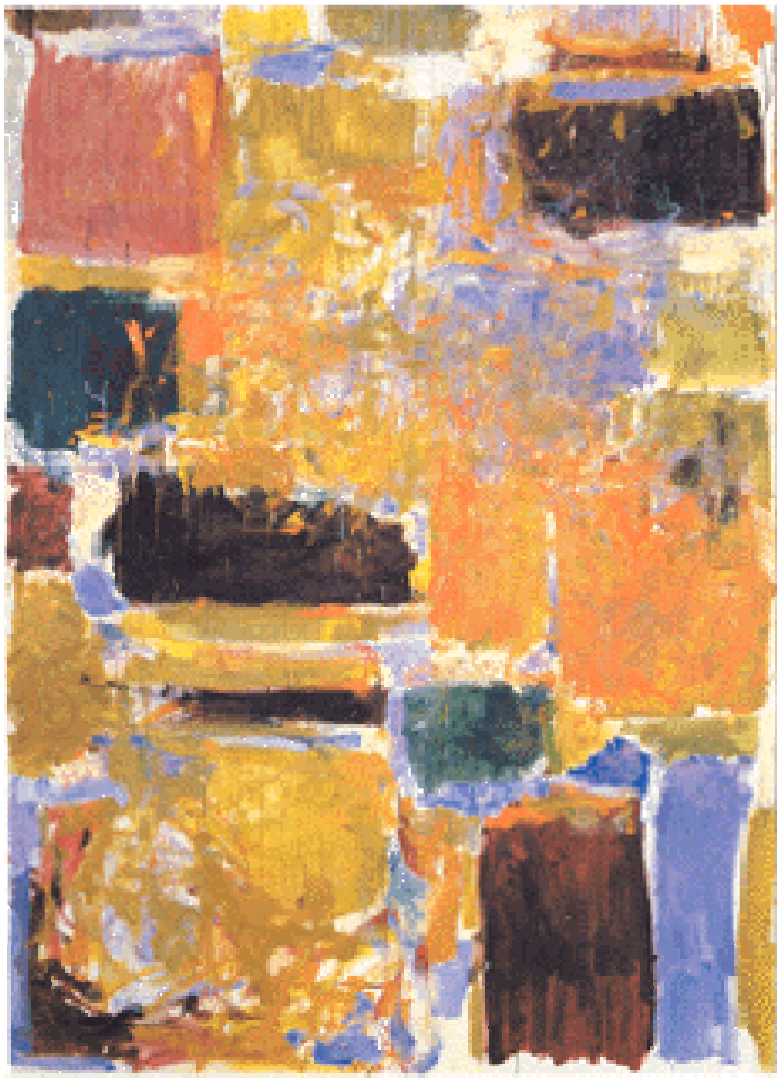


Also known as dynamic hazards

The result is correct,
but there is extra power dissipated

Example: Chain of NAND Gates





Energy-delay trade-offs

A simple current model

- If device always operates in velocity sat.:

$$I_D = k' \cdot \frac{W}{L} \cdot \left(V_{GS} - V_T - \frac{V_{D,VSAT}}{2} \right) V_{D,VSAT}$$

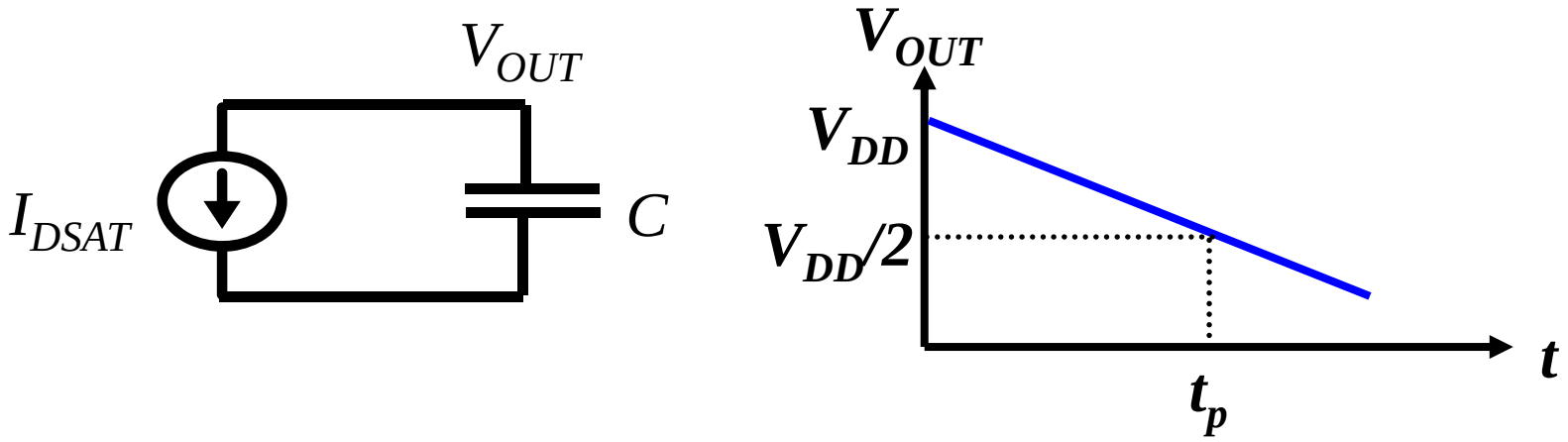
- "V_T*" model: $V_T^* \equiv V_T + \frac{V_{D,VSAT}}{2}$

$$I_D = k' \cdot \frac{W}{L} \cdot \left(V_{GS} - V_T^* \right) V_{D,VSAT}$$

- Good for first cut, simple analysis

Switching Delay

- In saturation, transistor basically acts like a current source:



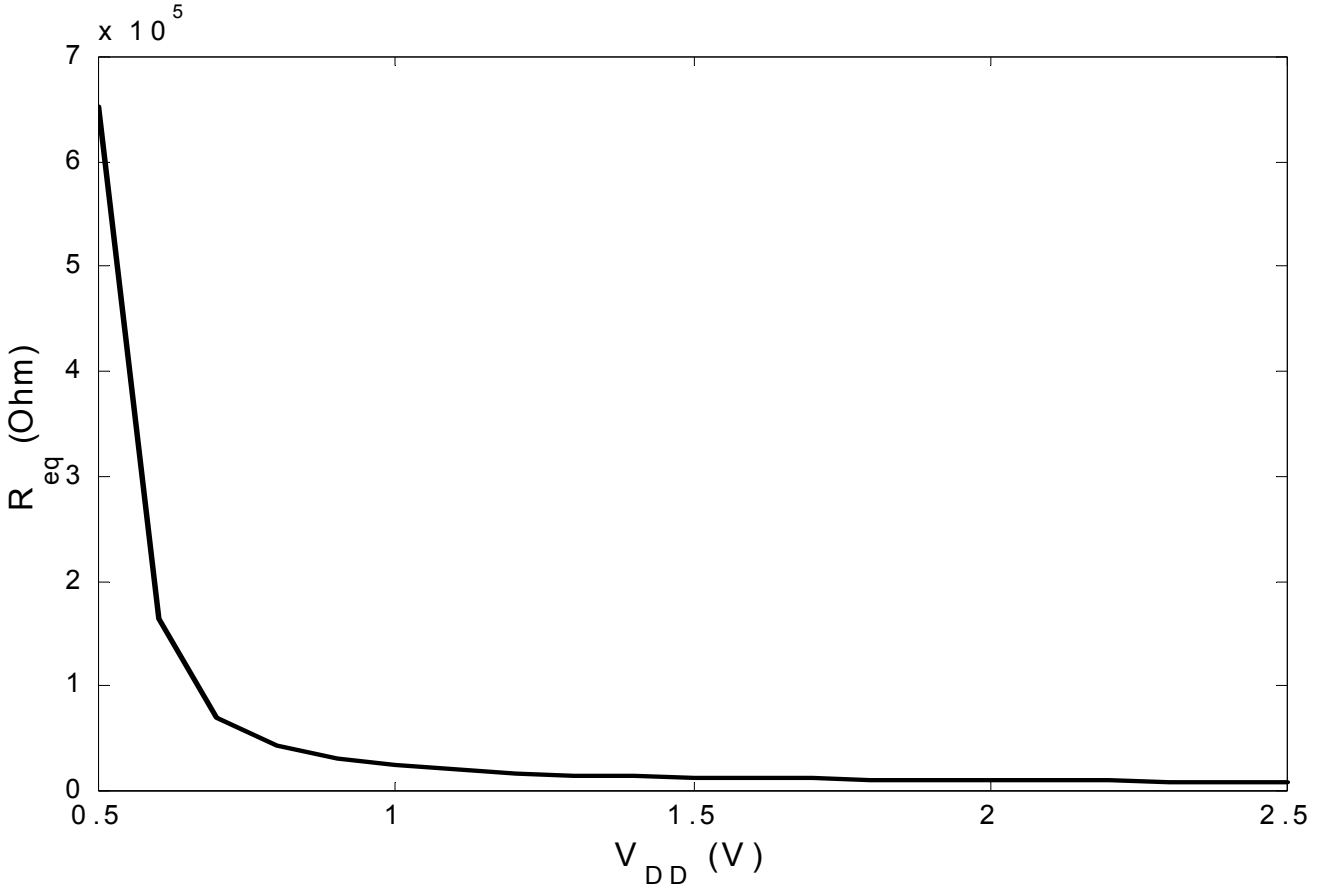
$$V_{OUT} = V_{DD} - (I_{DSAT}/C)t \longrightarrow \boxed{t_p = C(V_{DD}/2)/I_{DSAT}}$$

$$R_{eq} \approx \frac{1}{2 \cdot \ln(2)} \frac{V_{DD}}{I_{DSAT}}$$

$t_p = \ln 2 \cdot R_{eq} \cdot C$

$\sim (V_{DD} - V_{th})$

The Transistor as a Switch

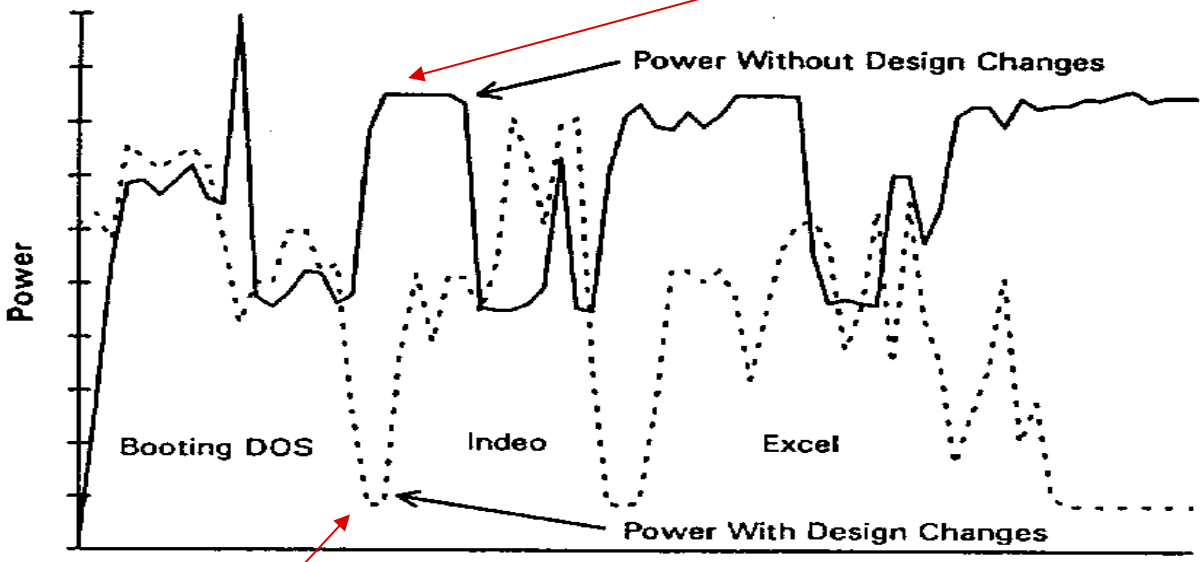


Principles for Power Reduction

- Most important idea: reduce waste
- Examples:
  - Don't switch capacitors you don't need to
    - Clock gating, glitch elimination, logic re-structuring
  - Don't run circuits faster than needed
    - Power α V_{DD}^2 – can save a lot by reducing supply for circuits that don't need to be as fast
    - Parallelism falls into this category
- Let's say we do a good job of that – then what?

Standby Power - Was Not A Concern In Earlier Days

Processor in idle mode!



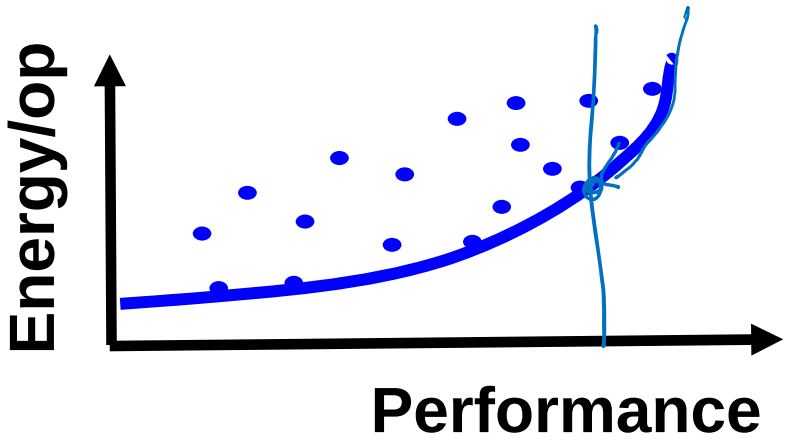
Floating Point Unit and Cache powered down when not in use

[Source: Intel]

"Power-Delay" and Energy-Delay

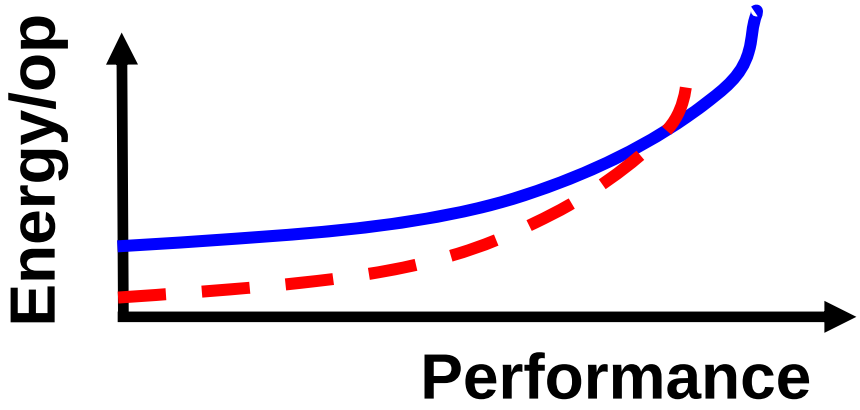
- Want low power and low delay, so how about optimizing the product of the two?
  - So-called "Power-Delay Product"
- Power·Delay is by definition Energy
  - Optimizing this pushes you to go as slow as possible
- Alternative gate metric: Energy-Delay Product
  - EDP = $(P_{av} \cdot t_p) \cdot t_p = E \cdot t_p$

Energy – Performance Space



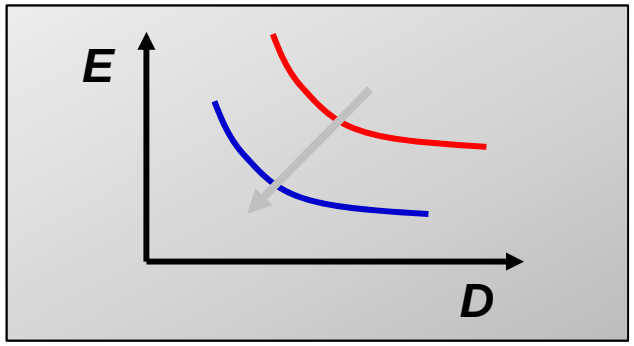
- Plot all possible designs on a 2-D plane
  - No matter what you do, can never get below/to the right of the solid line
- This line is called "Pareto Optimal Curve"
  - Usually (always) follows law of diminishing returns

Optimization Perspective

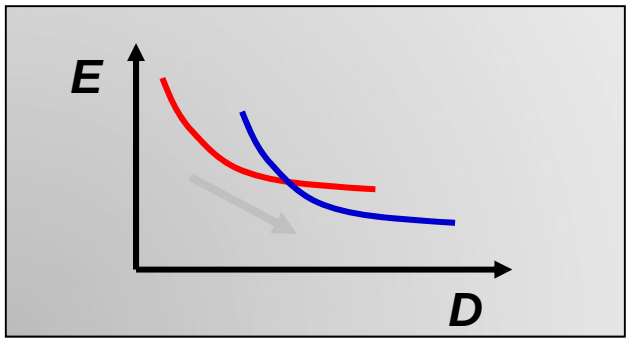


- Instead of metrics like EDP, this curve often provides information more directly
  - Ex1: What is minimum energy for XX performance?
  - Ex2: Over what range of performance is a new technique (dotted line) actually beneficial?

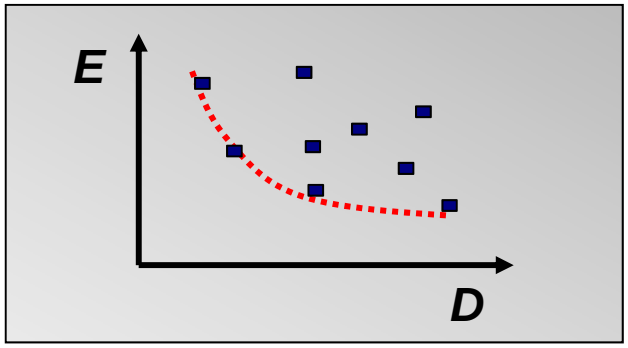
Expanding the Playing Field



Removing inefficiencies (1)

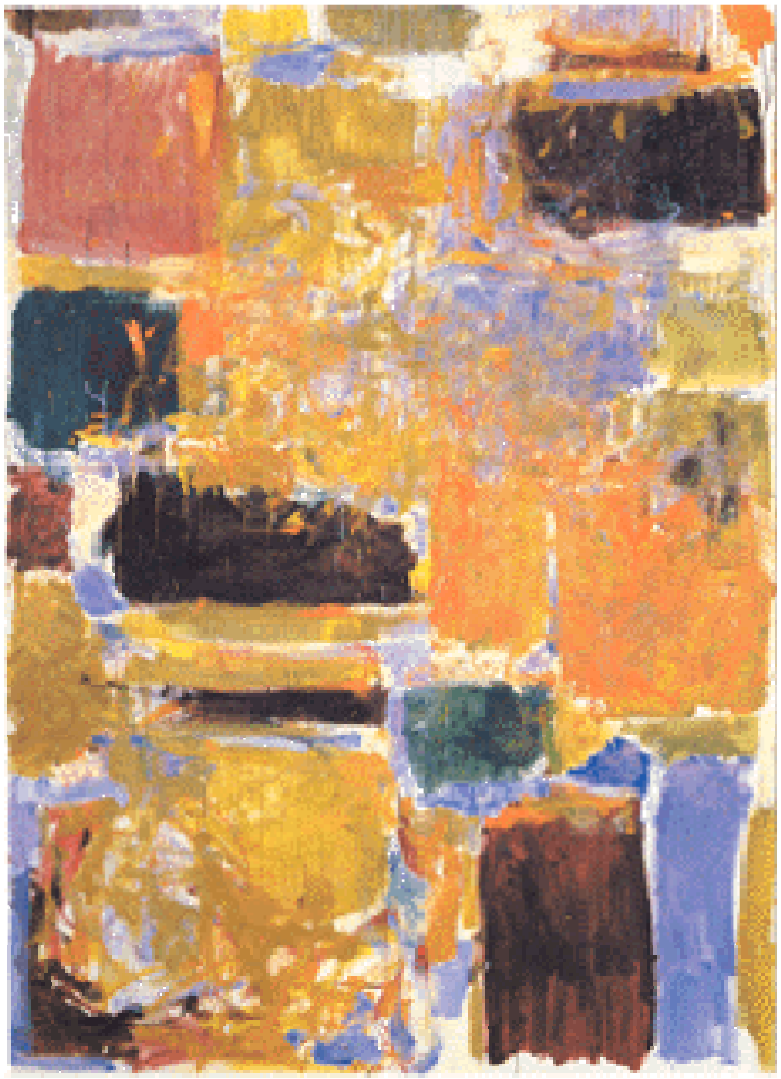


Alternative topologies (2)



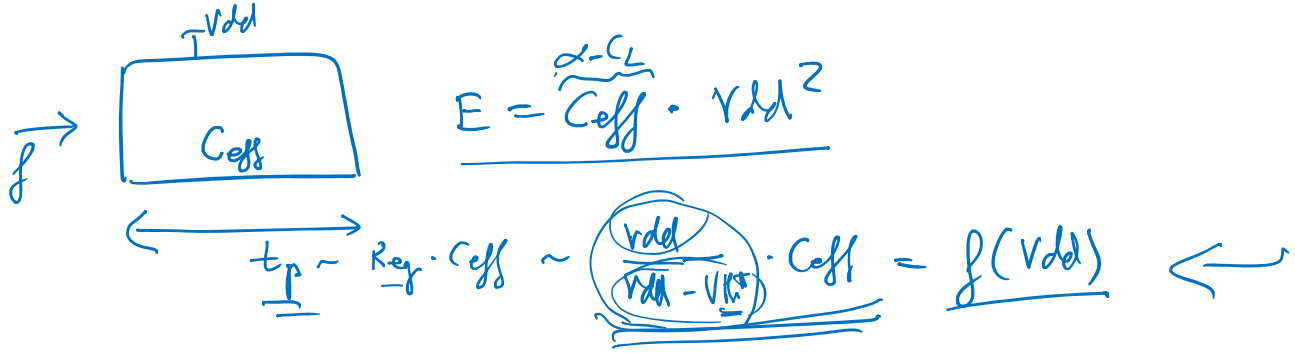
Discrete options (3)

Architecture and system transformations and optimizations reshape the E-D curves

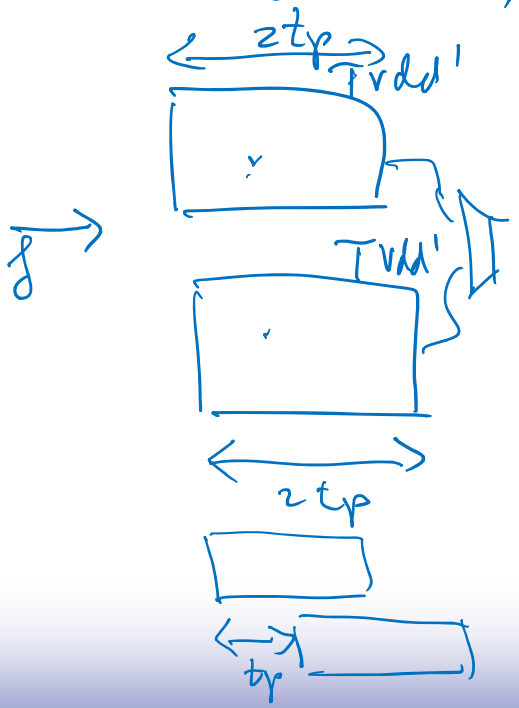


Voltage scaling: Parallelism and Pipelining

Supply scaling, delay and energy



$Vdd \downarrow \quad E \downarrow, \quad t_p \uparrow$



$t_p(Vdd') = 2 t_p(Vdd)$

$Vdd < Vdd$

$$\frac{Vdd'}{Vdd' - Vth} = 2 \frac{Vdd}{Vdd - Vth}$$

$Vdd' \rightarrow E' = C_{eff} \cdot Vdd'^2$

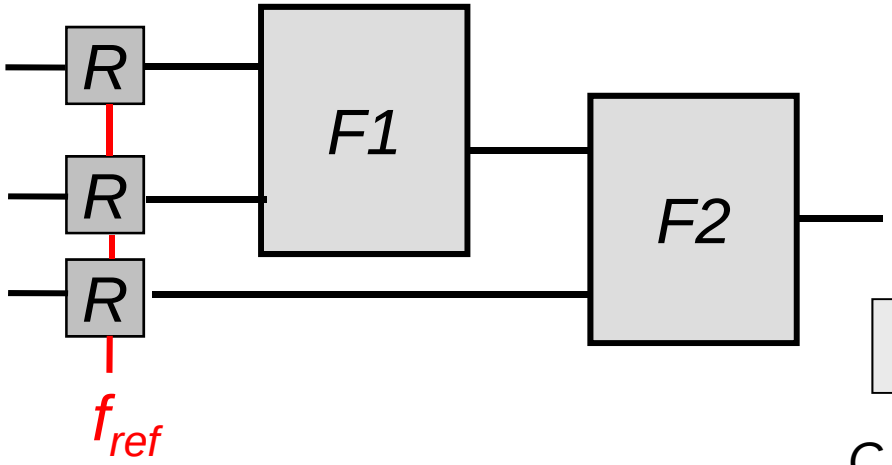
$\frac{E_{rf} - E'}{E_{rf}}$ ~ savings $= \frac{E'}{E_{rf}} = \frac{Vdd'^2}{Vdd^2}$

Reducing the Supply Voltage (while maintaining performance)

Concurrency:

trading-off clock frequency versus area to reduce power

Consider the following reference design



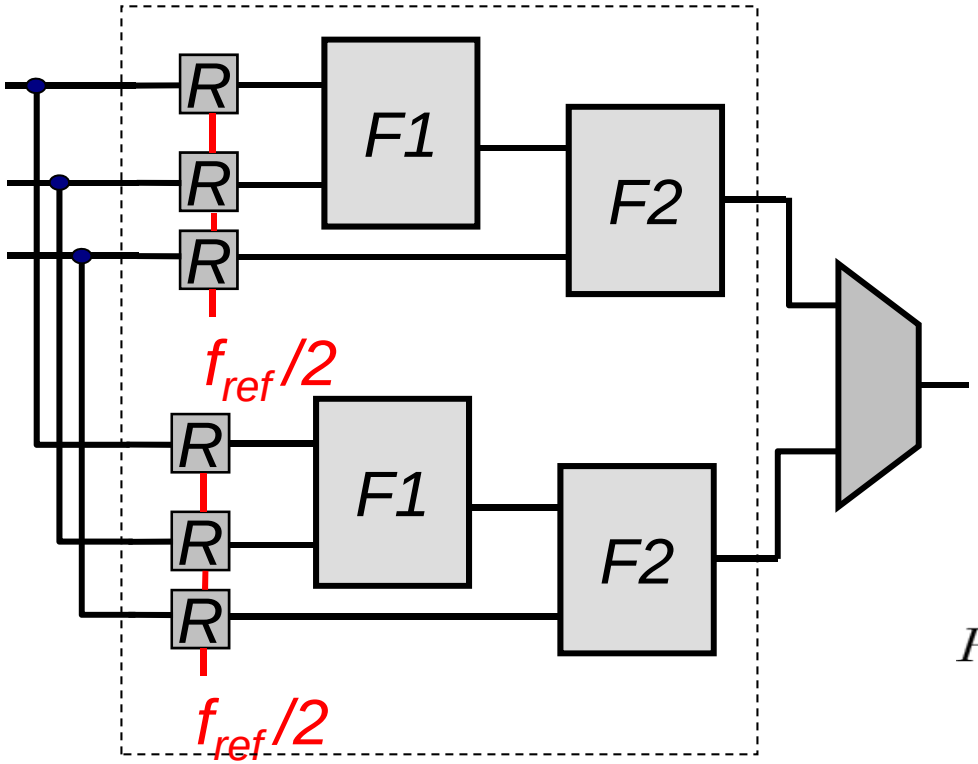
$$P_{ref} = C_{ref} \cdot V_{dd,ref}^2 \cdot f_{ref}$$

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

C_{ref}: average switching capacitance

[A. Chandrakasan, JSSC'92]

A Parallel Implementation



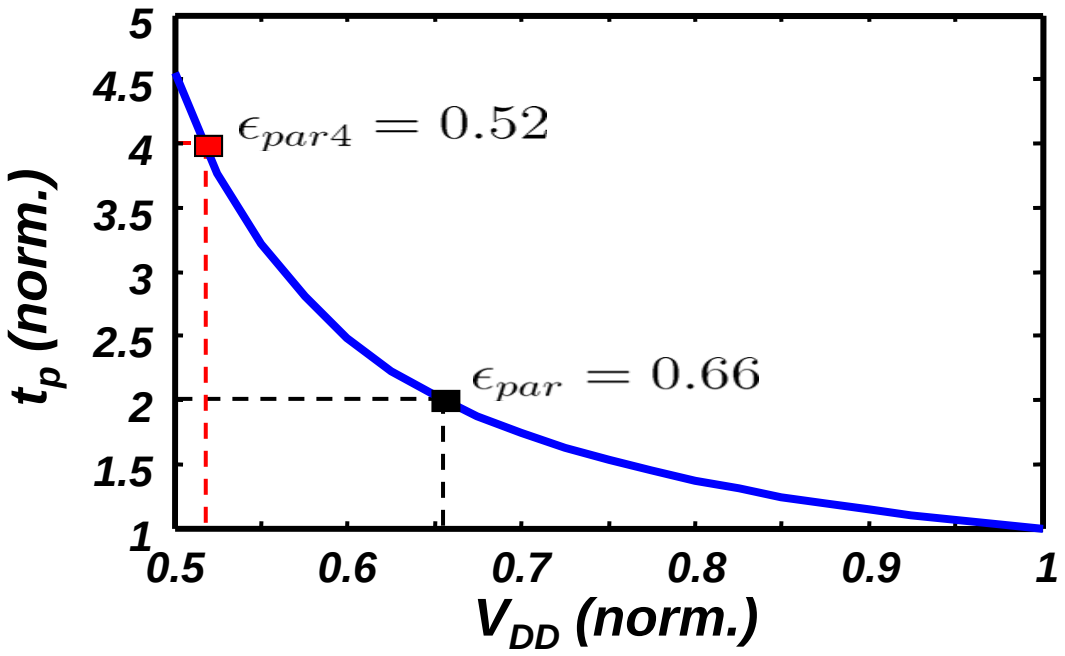
$$f_{par} = f_{ref}/2$$
$$C_{par} = (2 + ov_{par}) \cdot C_{ref}$$
$$V_{dd,par} = \epsilon_{par} \cdot V_{dd,ref}$$

Almost cancels

$$P_{par} = \epsilon_{par}^2 \cdot \left(\frac{2 + ov_{par}}{2}\right) \cdot P_{ref}$$

Running slower reduces required supply voltage
Yields quadratic reduction in power

Example: 90nm Technology

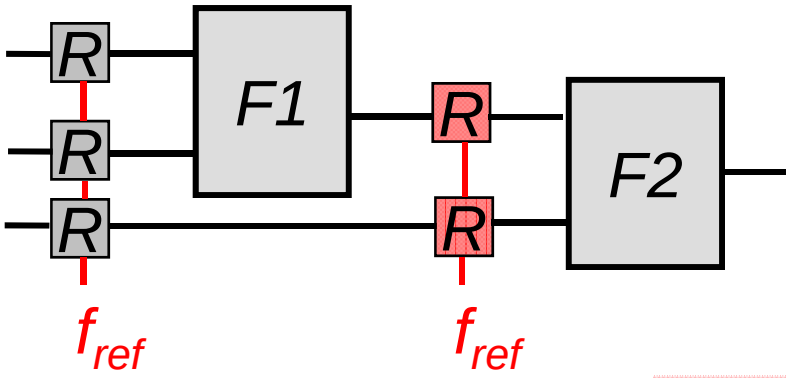


Assuming $ov_{par} = 7.5\%$

$$P_{par} = 0.66^2 \times \frac{2.075}{2} P_{ref} = 0.45 Pref$$

$$P_{par4} = 0.522 \times \frac{4.15}{2} P_{ref} = 0.28 Pref$$

A Pipelined Implementation



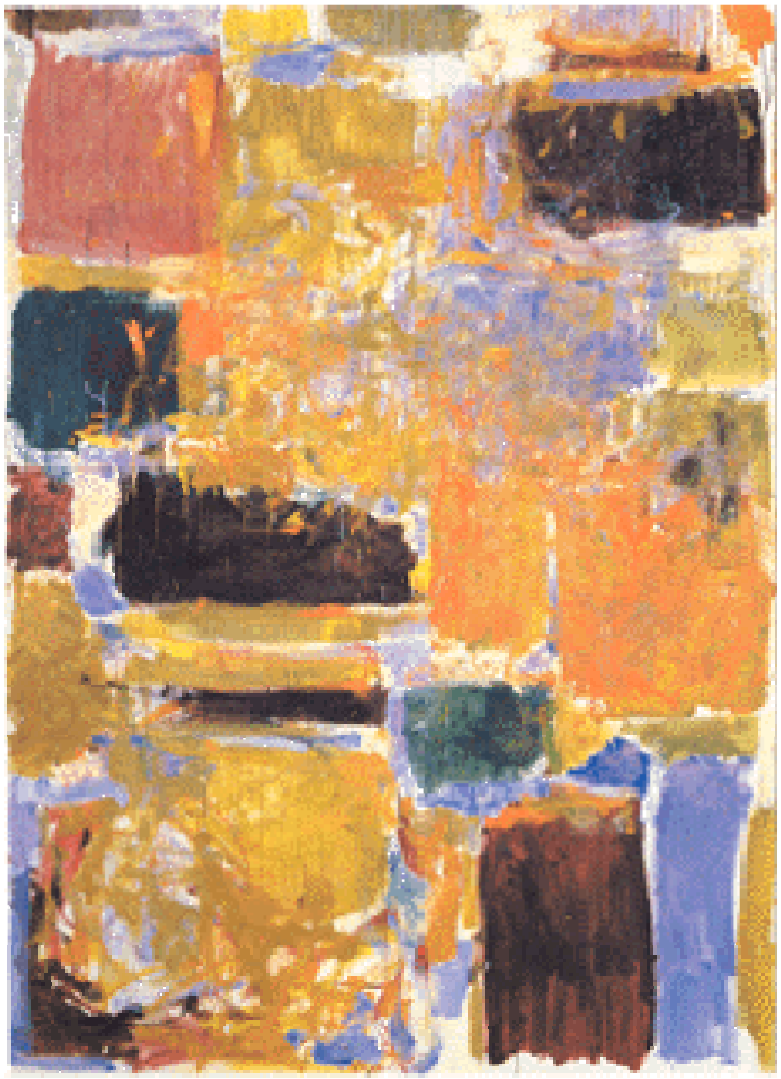
$$f_{pipe} = f_{ref}$$
$$C_{pipe} = (1 + ov_{pipe}) \cdot C_{ref}$$
$$V_{dd,pipe} = \epsilon_{pipe} \cdot V_{dd,ref}$$

$$P_{pipe} = \epsilon_{pipe}^2 \cdot (1 + ov_{pipe}) \cdot P_{ref}$$

Shallower logic reduces required supply voltage
(this example assumes equal V_{dd} for par / pipe designs)

Assuming
ov_{pipe} = 10%

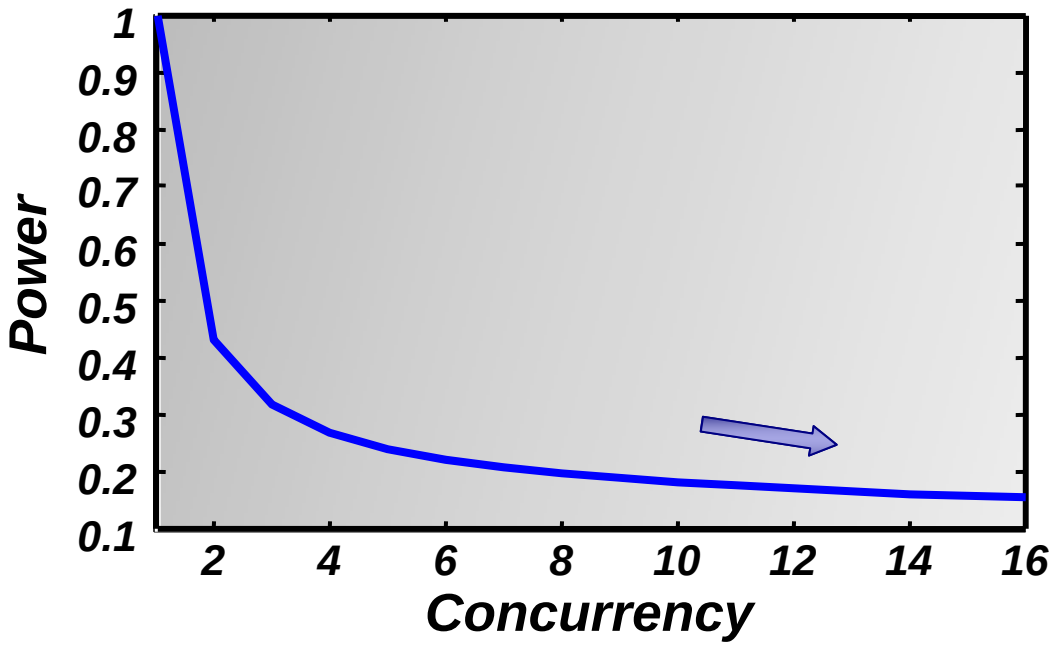
$$P_{pipe} = 0.66^2 \cdot 1.1 \cdot P_{ref} = 0.48 P_{ref}$$
$$P_{pipe4} = 0.52^2 \cdot 1.1 P_{ref} = 0.29 P_{ref}$$



Leveraging Concurrency

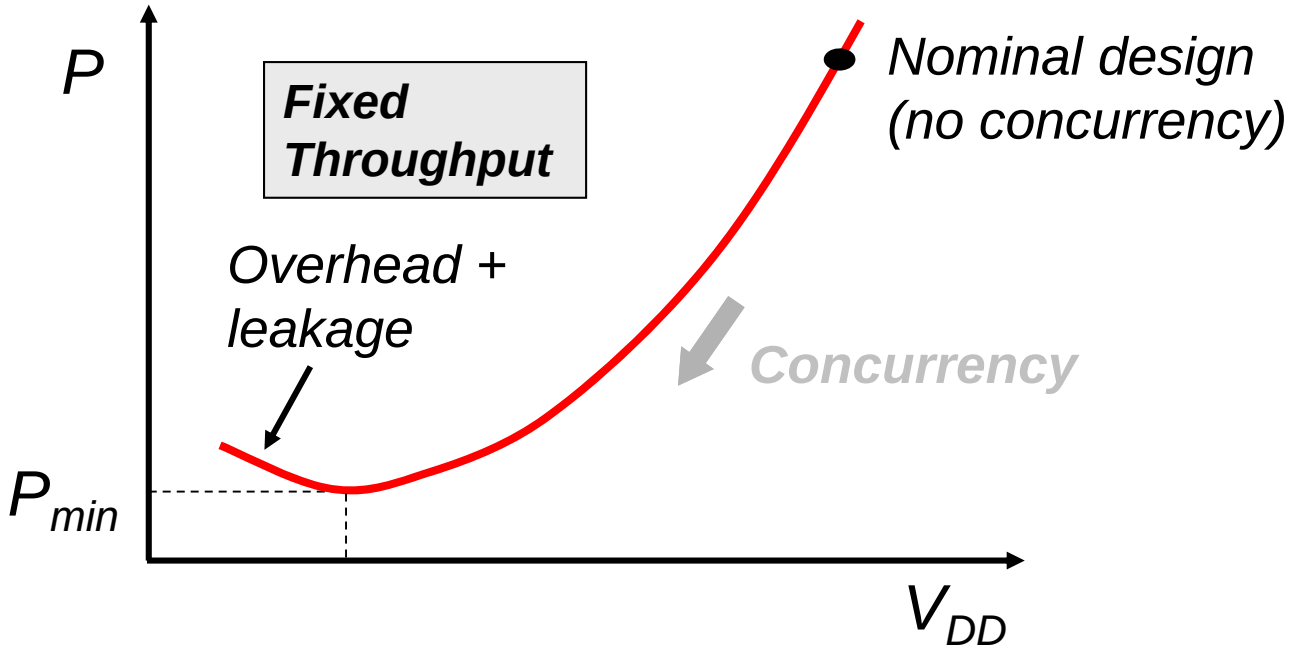
Increasing use of Concurrency Saturates

- Can combine parallelism and pipelining to drive V_{DD} down
- But, close to process threshold overhead of excessive concurrency starts to dominate



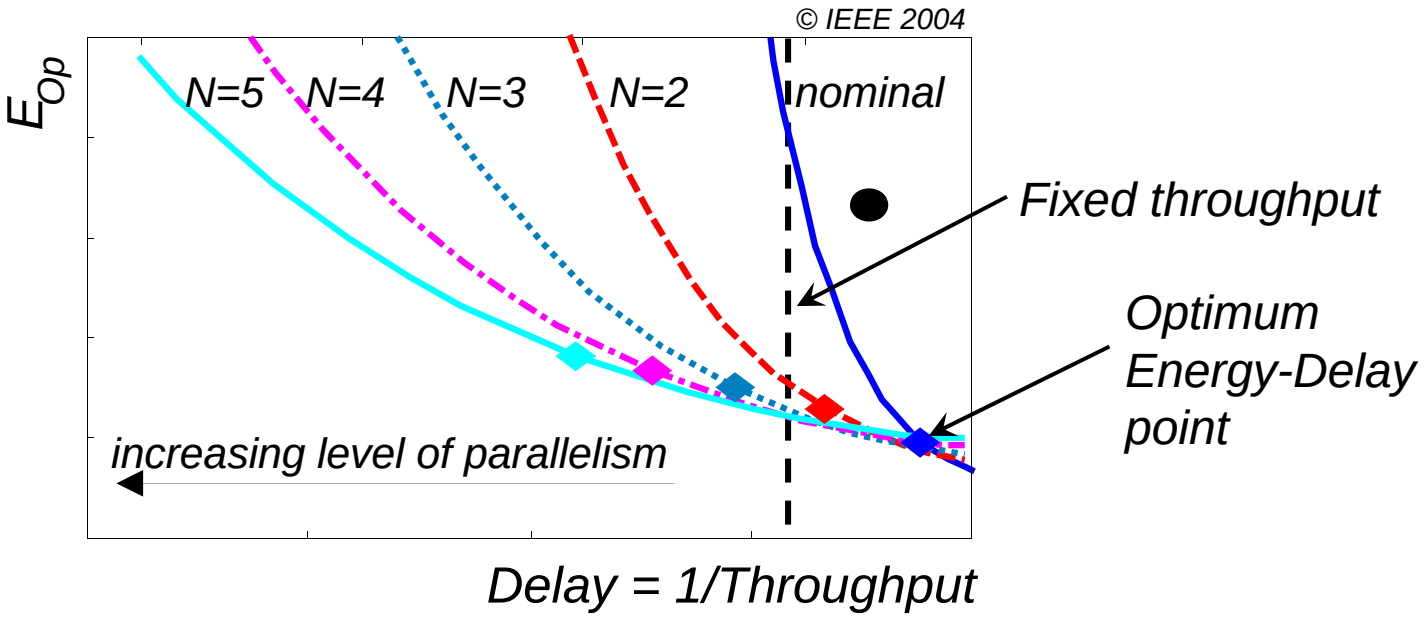
Assuming constant % overhead

Reducing V_{TH}



Only option: Reduce V_{TH} as well!
But: Must consider Leakage …

Mapping into the Energy-Delay Space



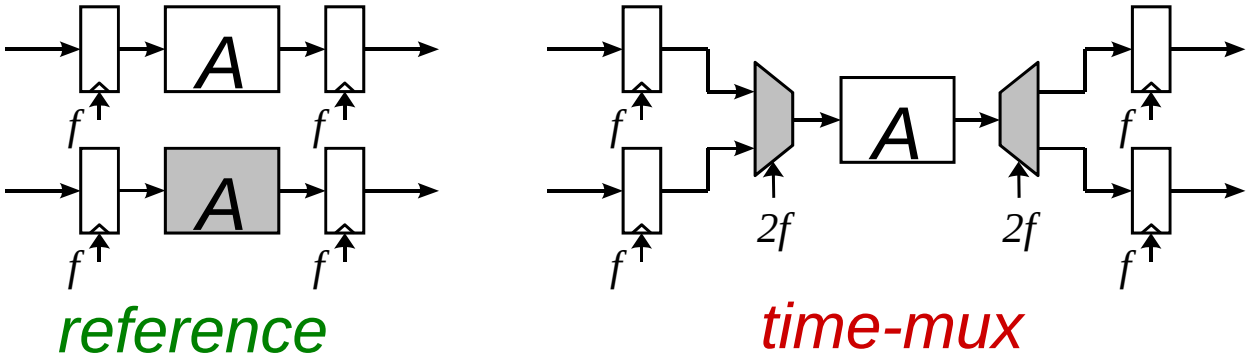
- For each level of performance, optimum amount of concurrency
- Concurrency only energy-optimal if requested throughput larger than optimal operation point of nominal function

[Ref: D. Markovic, JSSC'04]

What if the Required Throughput is Below Minimum?

(that is, at no concurrency)

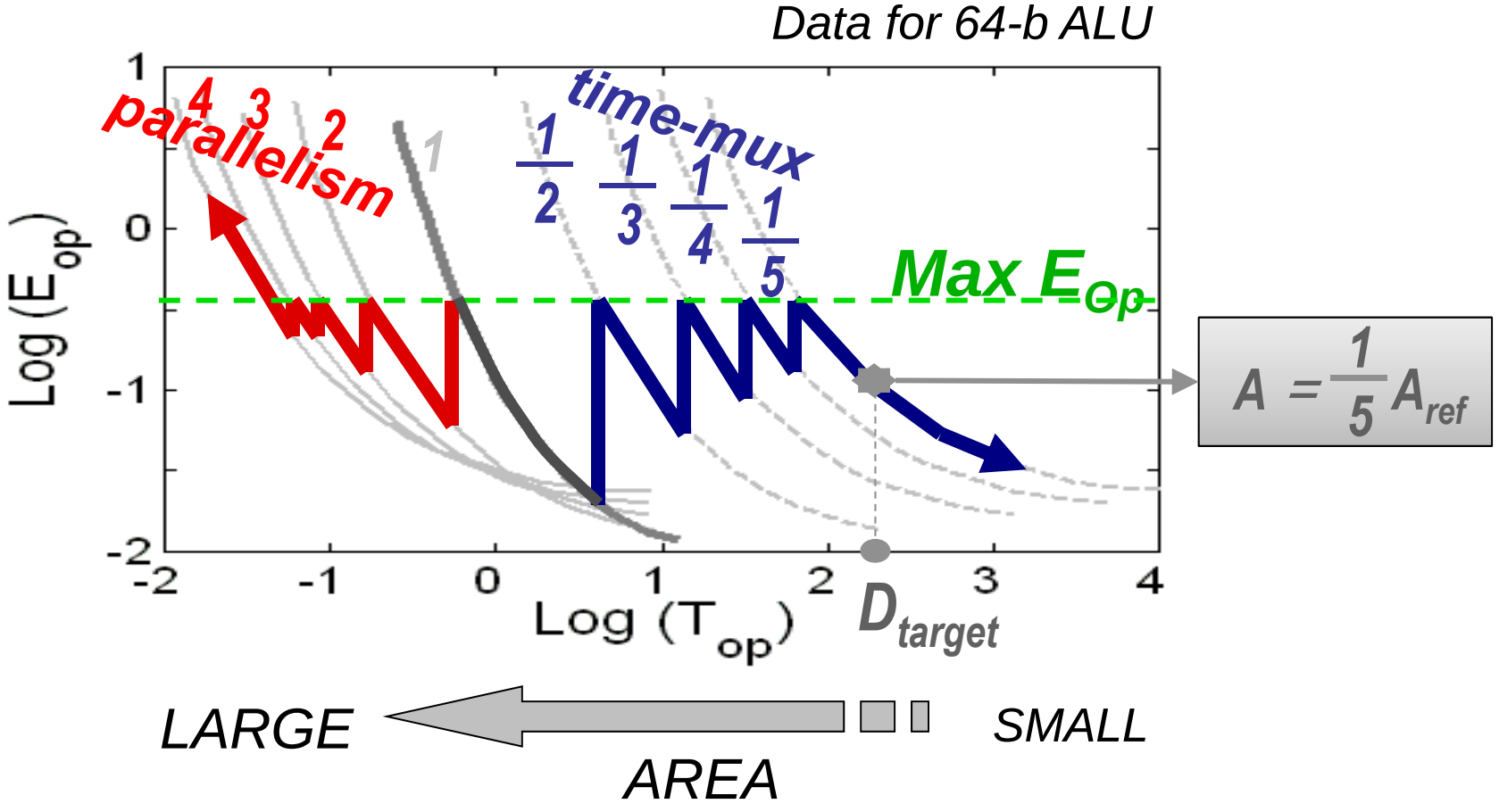
Introduce Time-Multiplexing!



Absorb unused time slack by increasing clock frequency (and voltage …)

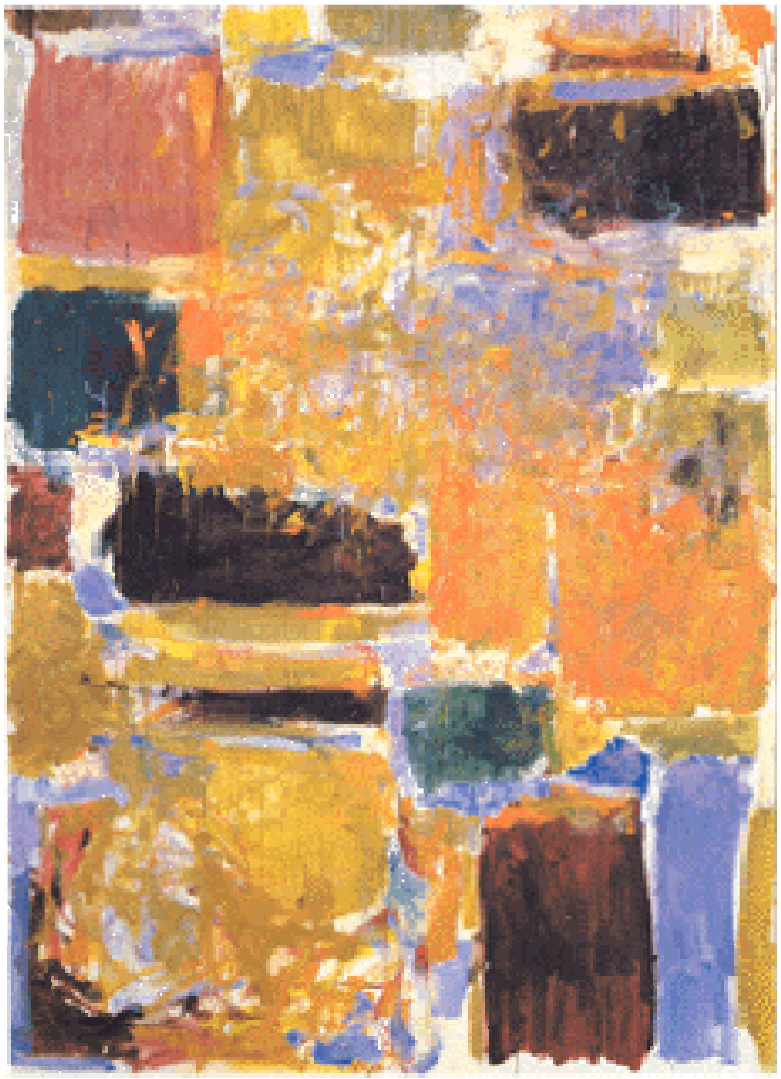
Again comes with some area and capacitance overhead!

Concurrency and Multiplexing Combined



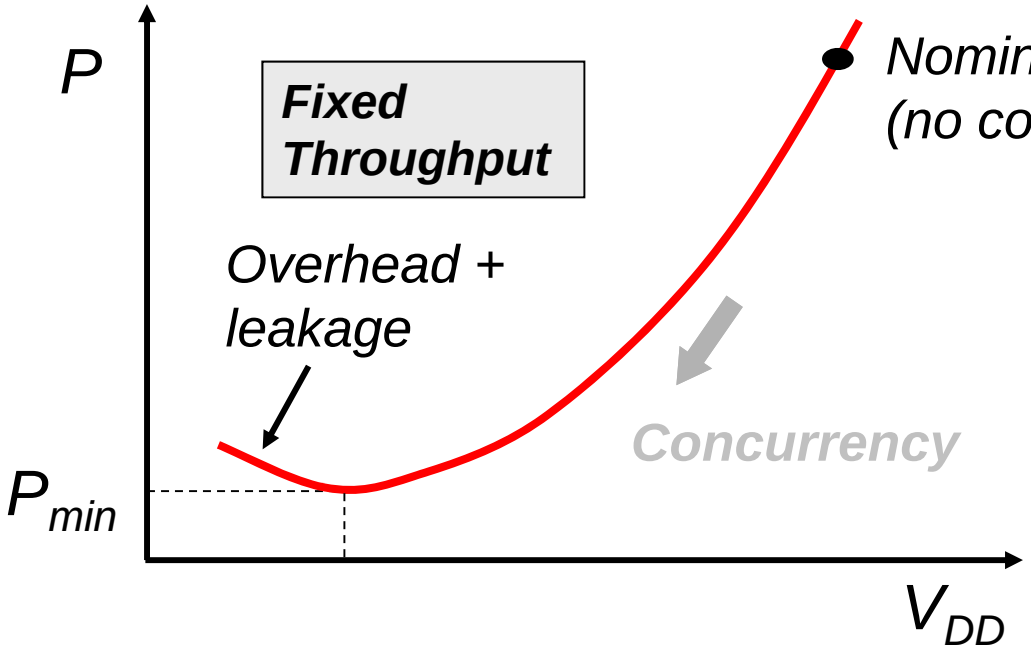
Some Energy-Inspired Design Guidelines

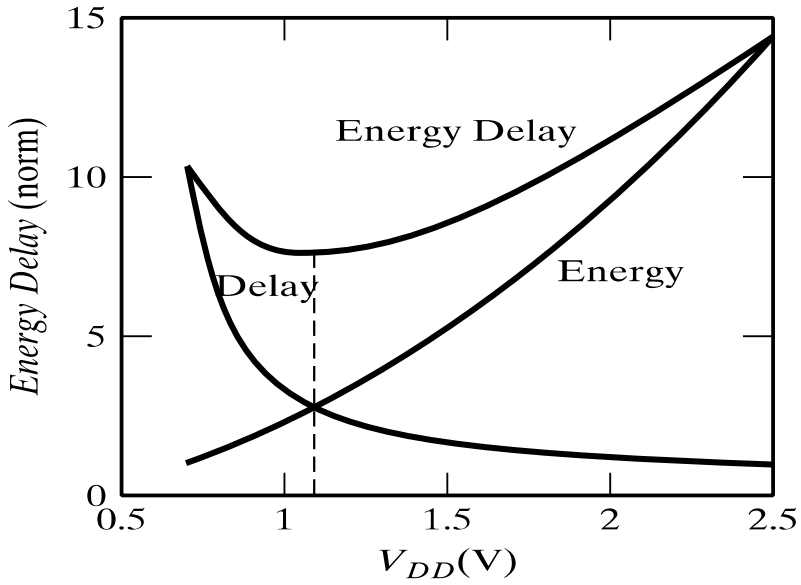
- **For maximum performance**
  - Maximize use of concurrency at the cost of area
- **For given performance**
  - Optimal amount of concurrency for minimum energy
- **For given energy**
  - Least amount of concurrency that meets performance goals
- **For minimum energy**
  - Solution with minimum overhead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



Managing Leakage

Leakage: Game-over for energy-delay trade-off





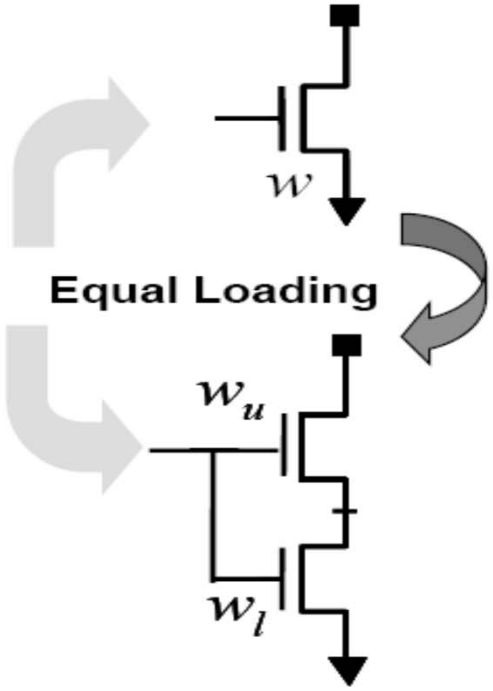
The Leakage Challenge – Power in Standby

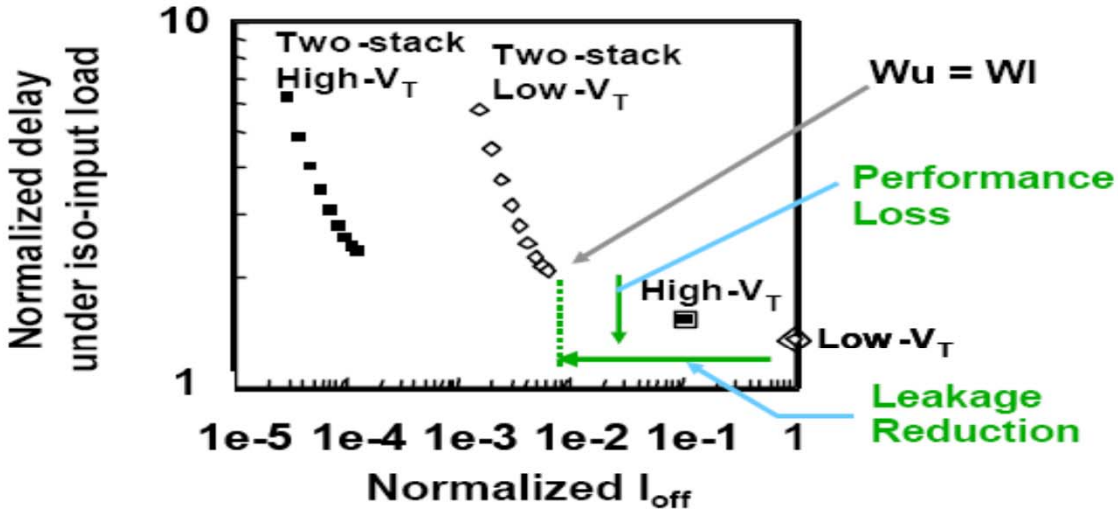
- With clock-gating employed in most designs, leakage power has become the dominant standby power source
- With no activity in module, leakage power should be minimized as well
  - Constant ratio between dynamic and static power desirable
- Challenge – how to disable unit most effectively given that no ideal switches are available

Standby Static Power Reduction Approaches

- Transistor stacking
- Power gating
- Body biasing
- Supply voltage ramping

Forced Transistor Stacking





- Force one transistor into a two transistor stack with the same input load
- Can be applied to gates with timing slack
- Trade-off between transistor leakage and speed

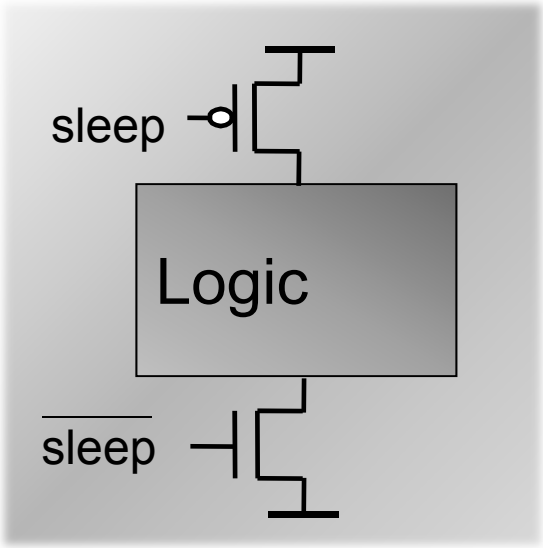
Useful for reducing leakage in non-critical shallow gates (in addition to high V_{TH})

[Ref: S. Narendra, ISLPED'01]

Transistor Stacking

- Off-current reduced in complex gates
- Some input patterns more effective than others in reducing leakage
- Effective standby power reduction strategy:
  - Select input pattern that minimizes leakage current of combinational logic module
  - Force inputs of module to correspond to that pattern during standby
- Pro's: Little overhead, fast transition
- Con: Limited effectiveness

Power Gating



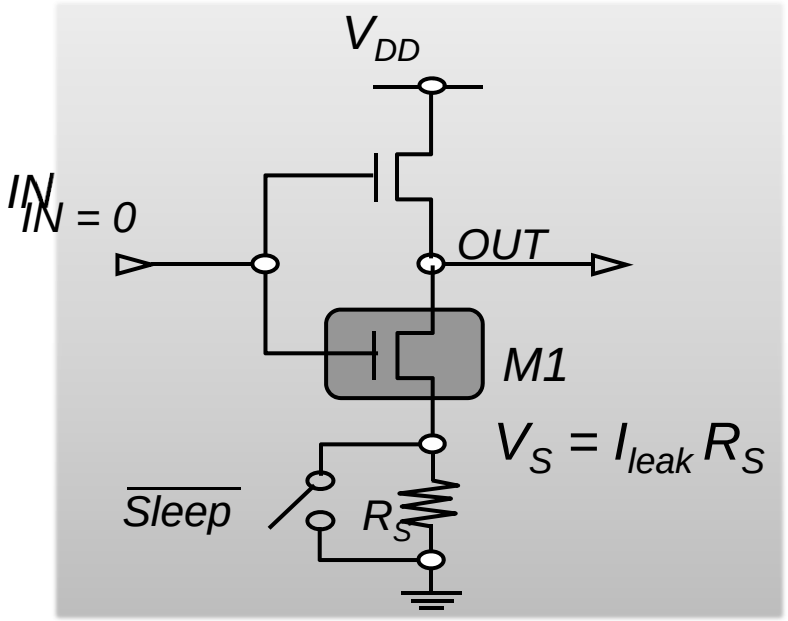
Disconnect module from supply rail(s) during standby

- Footer or header transistor, or both
- Most effective when high V_T transistors are available
- Easily introduced in standard design flows
- But … Impact on performance

Very often called “MTCMOS” (when using high- and low- threshold devices)

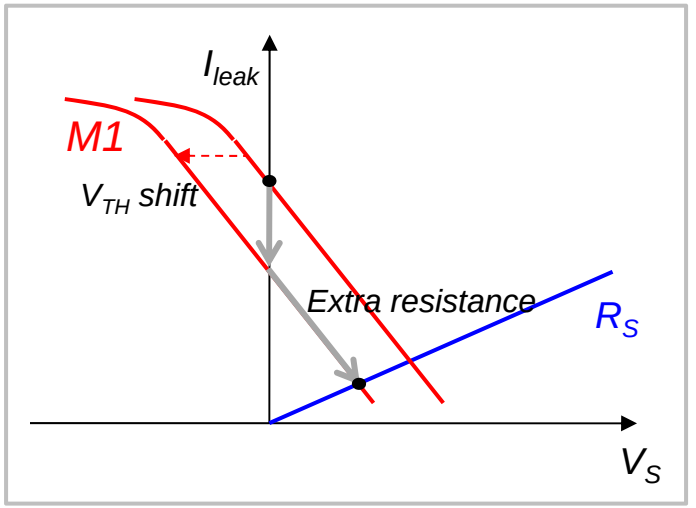
[Ref: T. Sakata, VLSI’93; S. Mutoh, ASIC’93]

Power Gating — Concept



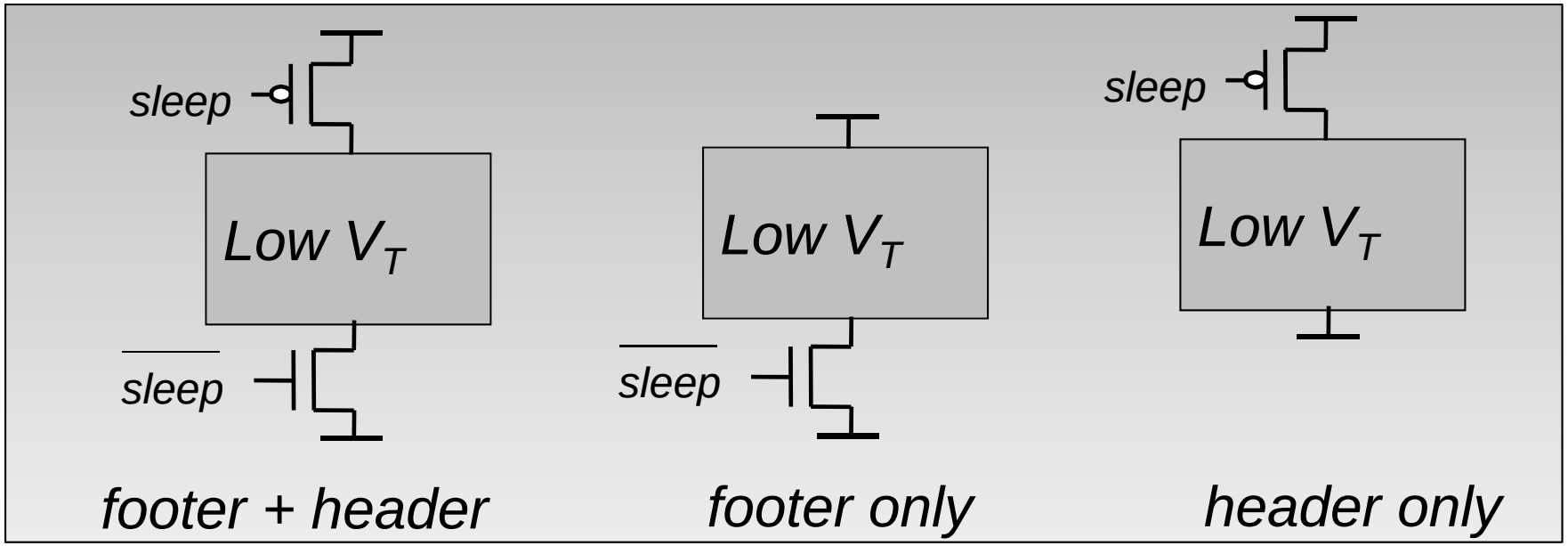
Leakage current reduces because

- Increased resistance in leakage path
- Stacking effect introduces source biasing



(similar effect at PMOS side)

Power Gating Options



- NMOS sleeper transistor more area efficient than PMOS
- Leakage reduction more effective (under all input patterns) when both footer and header transistors are present