EE141-Spring 2012 Digital Integrated Circuits

Lecture 24
Scaling + Energy

Administrativia

- Project Phase 2 due Today.
- Project Phase 3 to be launched today
- Assignment 9 posted today
  - One more assignmnet (#10) – will not be graded

CMOS Scaling

Technology Scaling

- Benefits of 30% "Dennard" scaling (1974):
  - Double transistor density
  - Reduce gate delay by 30% (increase operating frequency by 43%)
  - Reduce energy per transition by 65% (50% power savings @ 43% increase in frequency)
- Die size used to increase by 14% per generation (not any more)
- Technology generation spans 2-3 years

Full Scaling (Dennard, Long-Channel)

- W, L, t_{ox}: 1/S
- V_{DD}, V_T: 1/S

- Area: WL
- C_{ox}: 1/tox
- C_L: C_{ox}WL
- I_D: $C_{ox}(W/L)(V_{DD}-V_T)^2$
- R_{eq}: V_{DD}/I_{DSAT}

Full Scaling (Dennard, Long-Channel)

- W, L, t_{ox}: 1/S
- V_{DD}, V_T: 1/S

- t_p: $R_{eq}C_L$
- P_{avg}: $C_L V_{DD}^2/t_p$
- P_{avg}/A: $C_{ox}V_{DD}^2/t_p$

Scaling Relationships for Long Channel Devices

Parameter	Relation	Full Scaling	General Scaling	Fixed Voltage Scaling
W, L, t_{ox}		1/S	1/S	1/S
V_{DD}, V_T		1/S	1/U	1
N_{SUB}	V/W_{depl}^2	S	S^2/U	S^2
Area/Device	WL	$1/S^2$	$1/S^2$	$1/S^2$
C_{ox}	$1/t_{ox}$	S	S	S
C_L	$C_{ox}WL$	1/S	1/S	1/S
k_n, k_p	$C_{ox}W/L$	S	S	S
I_{av}	$k_{n,p} V^2$	1/S	S/U^2	S
t_p (intrinsic)	$C_L V / I_{av}$	1/S	U/S^2	$1/S^2$
P_{av}	$C_L V^2 / t_p$	$1/S^2$	S/U^3	S
PDP	$C_L V^2$	$1/S^3$	$1/SU^2$	1/S

Full Scaling (Dennard, Short-Channel)

- W, L, t_{ox}: 1/S
- V_{DD}, V_T: 1/S

- Area: WL
- C_{ox}: 1/tox
- C_L: $C_{ox}WL$
- I_D: $WC_{ox}v_{sat}(V_{DD}-V_T-V_{VSAT}/2)$
- R_{eq}: V_{DD}/I_{DSAT}

Full Scaling (Dennard, Short-Channel)

- W, L, t_{ox}: 1/S
- V_{DD}, V_T: 1/S

- t_p: $R_{eq}C_L$
- P_{avg}: $C_L V_{DD}^2/t_p$
- P_{avg}/A: $C_{ox}V_{DD}^2/t_p$

Transistor Scaling (Velocity-Saturated Devices)

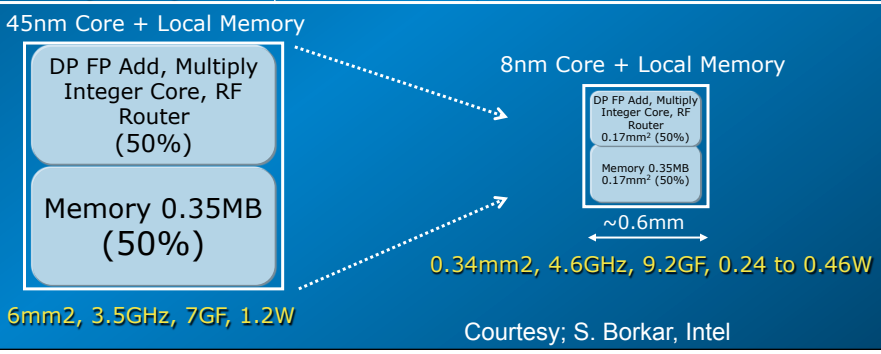
Parameter	Relation	Full Scaling	General Scaling	Fixed-Voltage Scaling
W, L, t_{ox}		$1/S$	$1/S$	$1/S$
V_{DD}, V_T		$1/S$	$1/U$	1
N_{SUB}	V/W_{depl}^2	S	S^2/U	S^2
Area/Device	WL	$1/S^2$	$1/S^2$	$1/S^2$
C_{ox}	$1/t_{ox}$	S	S	S
C_{gate}	$C_{ox}WL$	$1/S$	$1/S$	$1/S$
k_n, k_p	$C_{ox}W/L$	S	S	S
I_{sat}	$C_{ox}WV$	$1/S$	$1/U$	1
Current Density	$I_{sat}/Area$	S	S^2/U	S^2
Ron	V/I_{sat}	1	1	1
Intrinsic Delay	$R_{on}C_{gate}$	$1/S$	$1/S$	$1/S$
P	$I_{sat}V$	$1/S^2$	$1/U^2$	1
Power Density	$P/Area$	1	S^2/U^2	S^2

Interesting questions

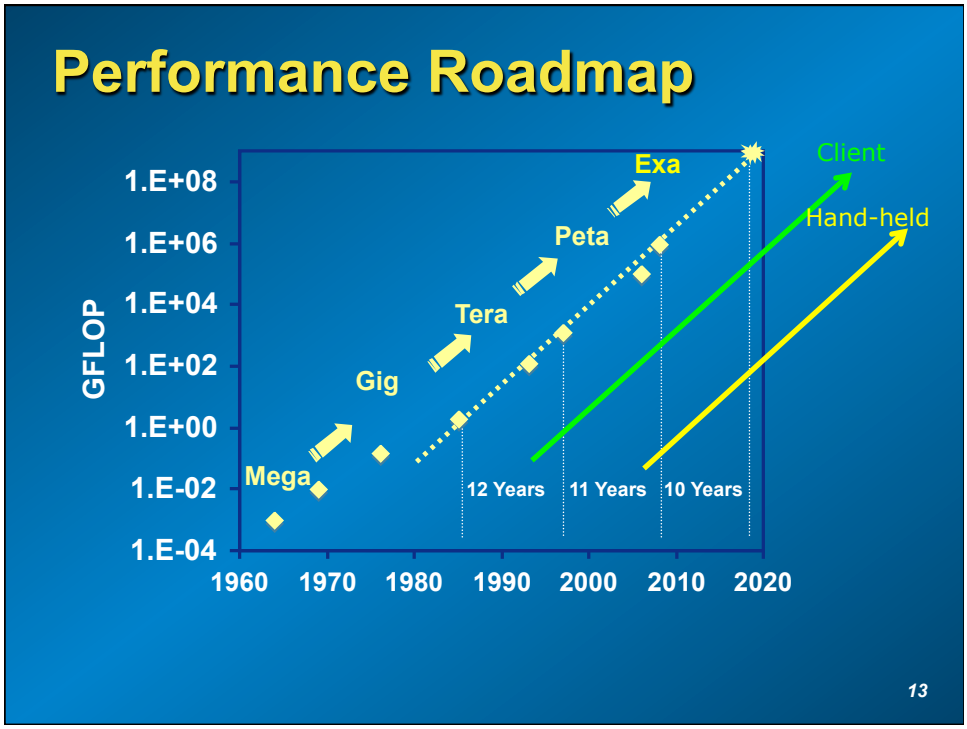
- What causes this model to break down?
  - Leakage set by kT/q
    - Temp. does not scale
    - V_T set to minimize power
  - Power actually increased
    - Leakage increased drastically
    - *f* increased faster than device speed
    - Hit cooling limit
  - Process Variation
    - Hard to build very small things accurately (less averaging)

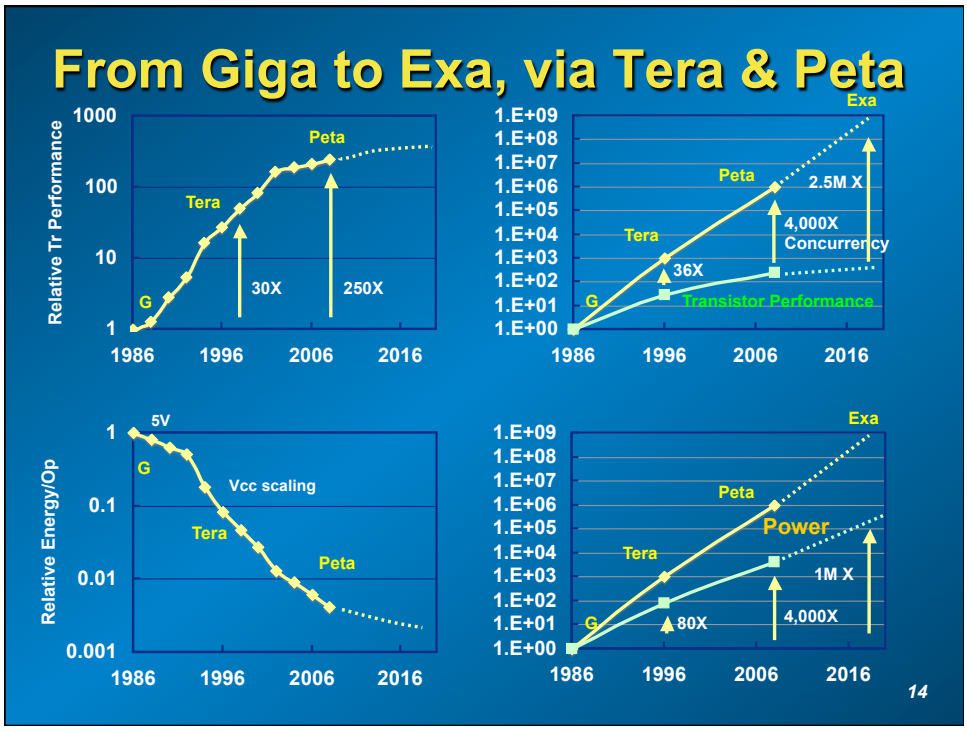
Scaling Assumptions

Technology (High Volume)	45nm (2008)	32nm (2010)	22nm (2012)	16nm (2014)	11nm (2016)	8nm (2018)	5nm (2020)
Transistor density	1.75	1.75	1.75	1.75	1.75	1.75	1.75
Frequency scaling	15%	10%	8%	5%	4%	3%	2%
Vdd scaling	-10%	-7.5%	-5%	-2.5%	-1.5%	-1%	-0.5%
Dimension & Capacitance	0.75	0.75	0.75	0.75	0.75	0.75	0.75
SD Leakage scaling/micron	1X Optimistic to 1.43X Pessimistic						



Courtesy; S. Borkar, Intel


Performance Roadmap
GFLOP
1.E+08
1.E+06
1.E+04
1.E+02
1.E+00
1.E-02
1.E-04
1960
1970
1980
1990
2000
2010
2020
Mega
Gig
Tera
Peta
Exa
Client
Hand-held
12 Years
11 Years
10 Years
13


From Giga to Exa, via Tera & Peta
Relative Tr Performance
1000
100
10
1
G
Tera
Peta
30X
250X
1986
1996
2006
2016
1.E+09
1.E+08
1.E+07
1.E+06
1.E+05
1.E+04
1.E+03
1.E+02
1.E+01
1.E+00
Exa
Peta
Tera
G
2.5M X
4,000X
Concurrency
36X
Transistor Performance
Relative Energy/Op
1
0.1
0.01
0.001
5V
G
Vcc scaling
Tera
Peta
Power
1M X
80X
4,000X
14

Wire Scaling

- Wire scaling model
  - C = WL/tox
  - R = ρL/(WH)
  - t_p = 0.38RC
  - E = C VDD^2

Energy/Power Revisited

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→1 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 vs. Dynamic)
  - Signal Statistics
  - Inter-signal Correlations
- "Dynamic" or timing dependent component
  - Circuit Topology
  - Signal Statistics and Correlations

Type of Logic Function: NOR

Example: Static 2-input NOR Gate

A	B	Out
0	0	1
0	1	0
1	0	0
1	1	0

Assume signal probabilities

$p_{A=1} = 1/2$

$p_{B=1} = 1/2$

Then transition probability

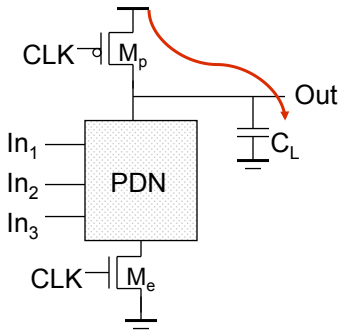
$p_{0\to1} = p_{Out=0} \times p_{Out=1}$

$= 3/4 \times 1/4 = 3/16$

If inputs switch every cycle

$\alpha_{0\to1} = 3/16$

Power Consumption of Dynamic Gates



Power only dissipated when previous Out = 0

Dynamic Power Consumption is Data Dependent

Dynamic 2-input NOR Gate

A	B	Out
0	0	1
0	1	0
1	0	0
1	1	0

Assume signal probabilities

$P_{A=1} = 1/2$

$P_{B=1} = 1/2$

Then transition probability

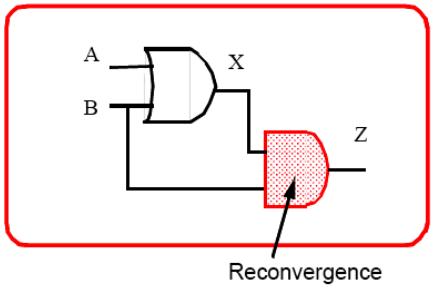
$P_{0\rightarrow1} = P_{out=0} \times P_{out=1}$

$= 3/4 \times 1 = 3/4$

Switching activity always higher in dynamic gates!

$P_{0\rightarrow1} = P_{out=0}$

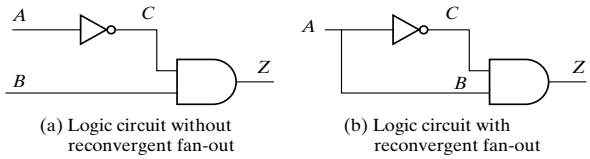
Problem: Reconvergent Fanout



$P(Z = 1) = P(B = 1) \cdot P(X = 1 \mid B=1)$

Becomes complex and intractable fast

Inter-Signal Correlations



(a) Logic circuit without reconvergent fan-out

(b) Logic circuit with reconvergent fan-out

Logic without reconvergent fanout

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

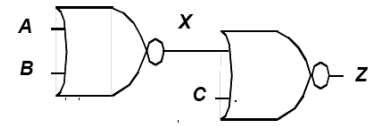
Logic with reconvergent fanout

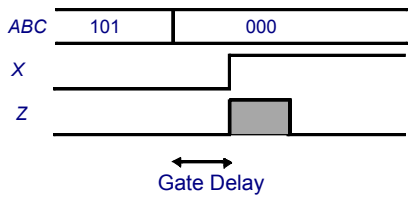
$P(Z = 1) = p(C=1 \mid B=1)\ p(B=1)$

$p_{0\to1} = 0$

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

Glitching in Static CMOS

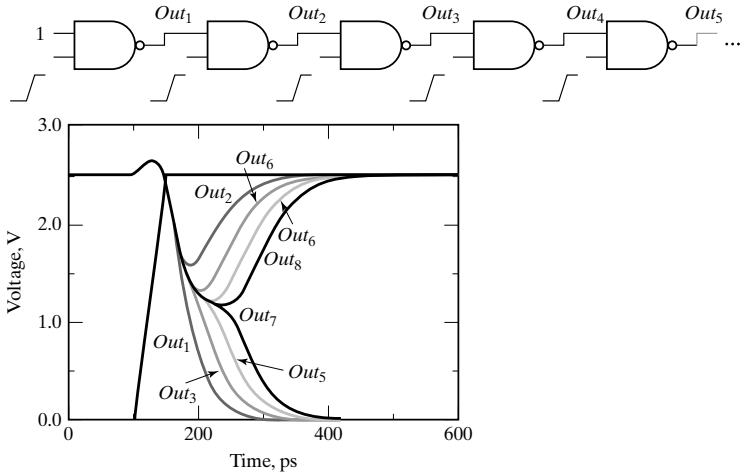




Also known as dynamic hazards

The result is correct,
but there is extra power dissipated

Example: Chain of NAND Gates



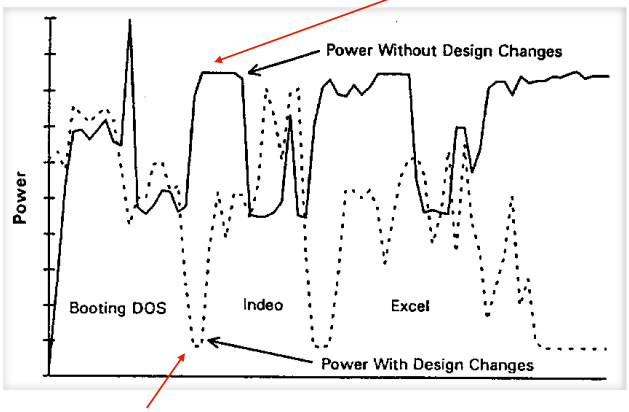
Principles for Power Reduction

- 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

Pentium-1: 15 Watt (5V - 66MHz)
Pentium-2: 8 Watt (3.3V- 133 MHz)

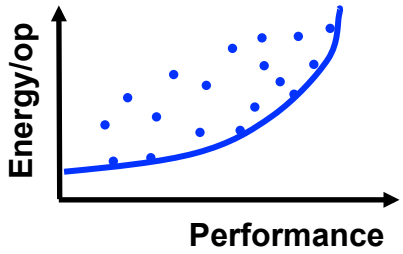
Processor in idle mode!



Floating Point Unit and Cache powered down when not in use

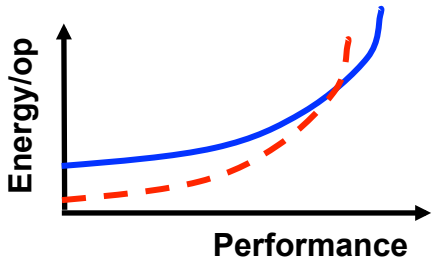
[Source: Intel]

Energy – Performance Space



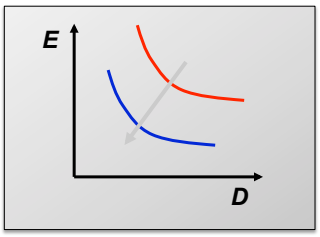
- Plot all possible designs on a 2-D plane
  - No matter what you do, can never get below/to the right of the solid line
- This line is called "Pareto Optimal Curve"
  - Usually (always) follows law of diminishing returns

Optimization Perspective

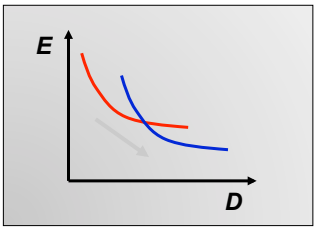


- Instead of metrics like EDP, this curve often provides information more directly
  - Ex1: What is minimum energy for XX performance?
  - Ex2: Over what range of performance is a new technique (dotted line) actually beneficial?

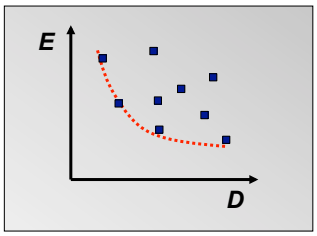
Expanding the Playing Field



Removing inefficiencies (1)



Alternative topologies (2)



Discrete options (3)

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

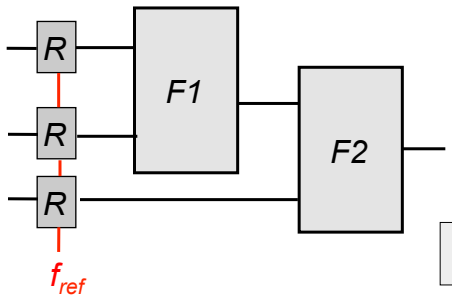
Reducing the Supply Voltage

(while maintaining performance)

Concurrency:

trading off clock frequency versus area to reduce power

Consider the following reference design



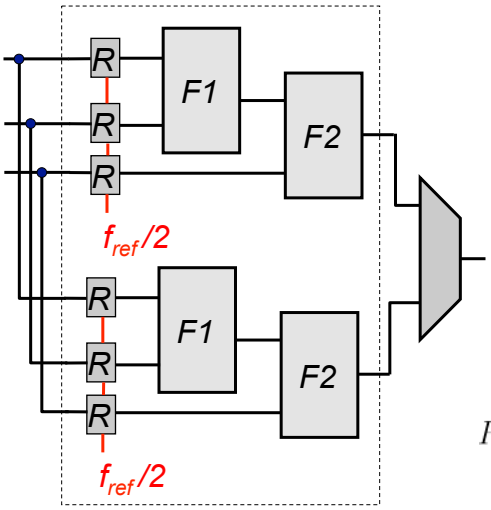
$$P_{ref} = C_{ref} \cdot V_{dd,ref}^2 \cdot f_{ref}$$

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

C_{ref}: average switching capacitance

[A. Chandrakasan, JSSC'92]

A Parallel Implementation



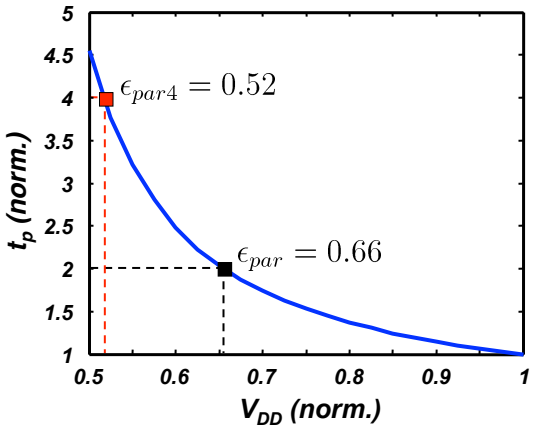
$$f_{par} = f_{ref}/2$$
$$C_{par} = (2 + ov_{par}) \cdot C_{ref}$$
$$V_{dd,par} = \epsilon_{par} \cdot V_{dd,ref}$$

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

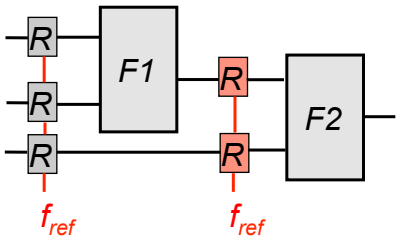


Assuming ov_{par} = 7.5%

$$P_{par} = 0.66^2 \cdot \frac{2.15}{2} \cdot P_{ref} = 0.47 P_{ref}$$

$$P_{par4} = 0.52^2 \cdot \frac{4.3}{4} \cdot P_{ref} = 0.29 P_{ref}$$

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)*

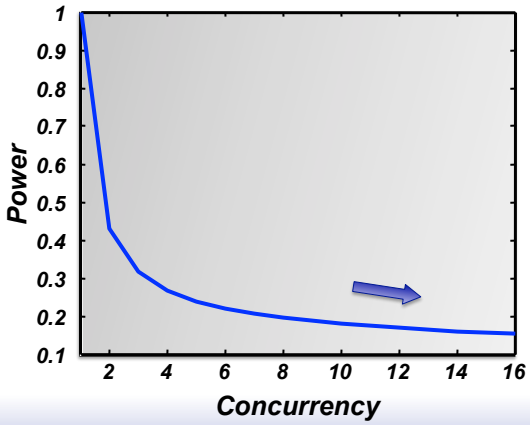
Assuming ov_{pipe} = 10%

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

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

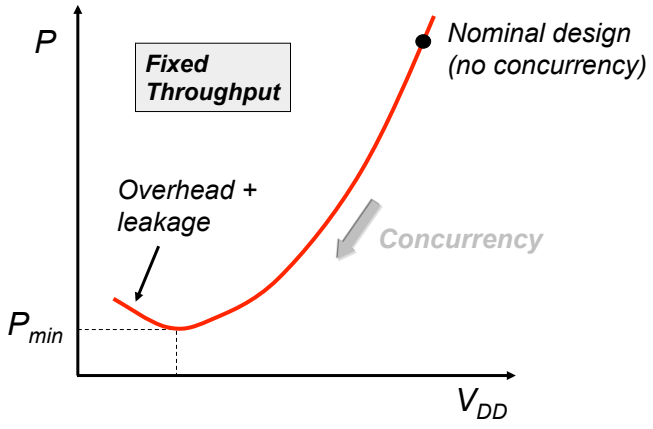
Increasing use of Concurrency Saturates

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



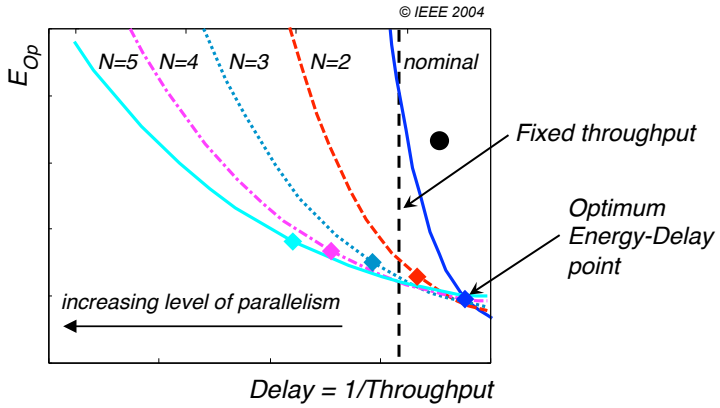
Assuming constant % overhead

Increasing use of Concurrency Saturates



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

Mapping into the Energy-Delay Space



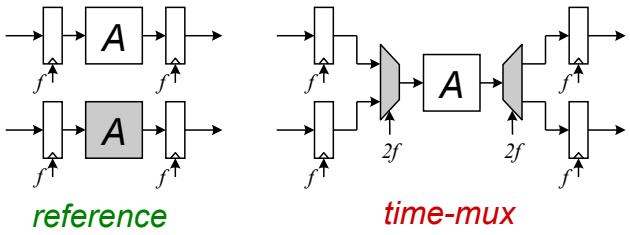
- For each level of performance, optimum amount of concurrency
- Concurrency only energy-optimal if requested throughput larger than optimal operation point of nominal function

[Ref: D. Markovic, JSSC'04]

What if the Required Throughput is Below Minimum?

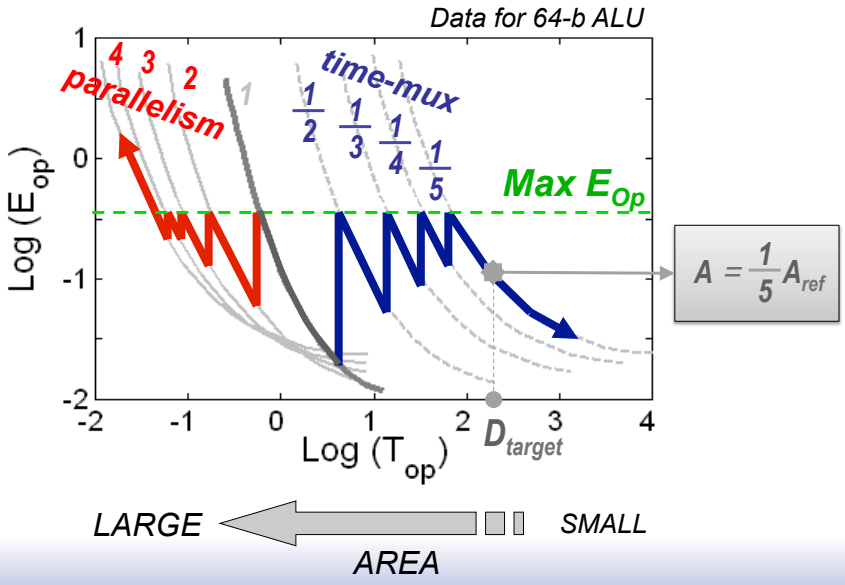
(that is, at no concurrency)

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)

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
  - Remember constant ratio between dynamic and static power …
- Challenge – how to disable unit most effectively given that no ideal switches are available

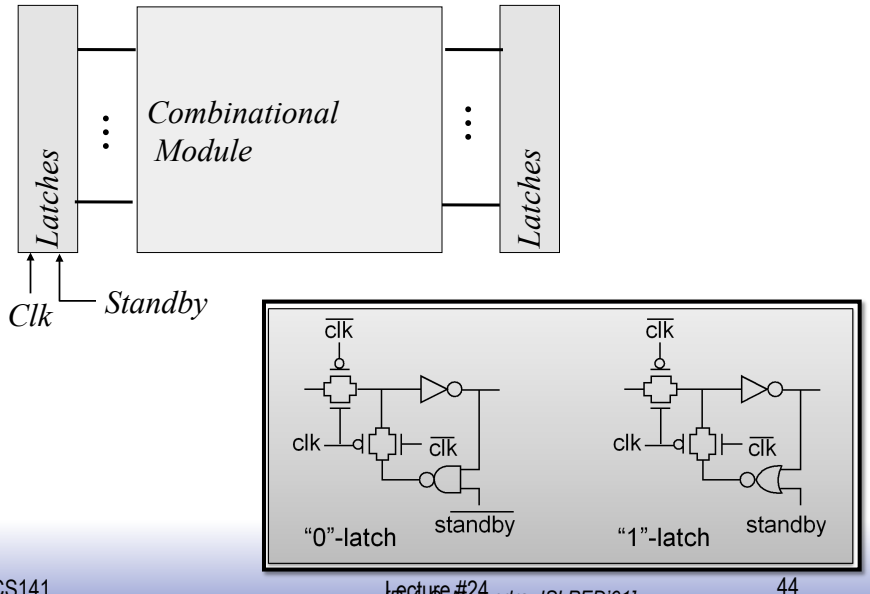
Standby Static Power Reduction Approaches

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

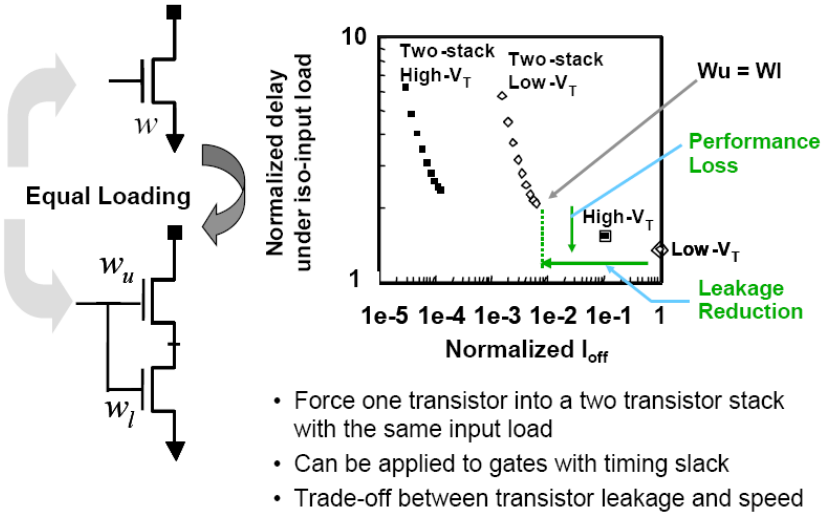
Transistor Stacking

- Off-current reduced in complex gates (see leakage power reduction @ design time)
- 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

Transistor Stacking



[Ref: S. Narendra, ISLPED'01]

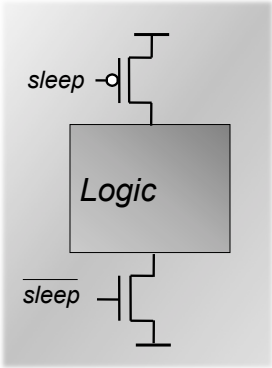


- Force one transistor into a two transistor stack with the same input load
- Can be applied to gates with timing slack
- Trade-off between transistor leakage and speed

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

[Ref: S. Narendra, ISLPED'01]

Power Gating



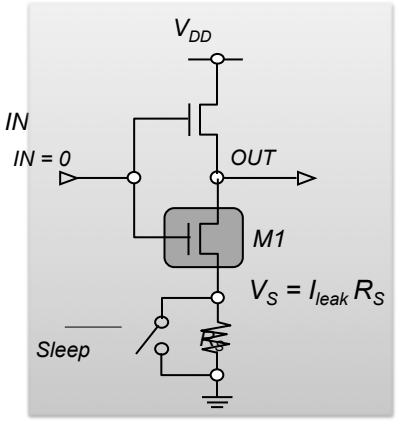
Disconnect module from supply rail(s) during standby

- *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*

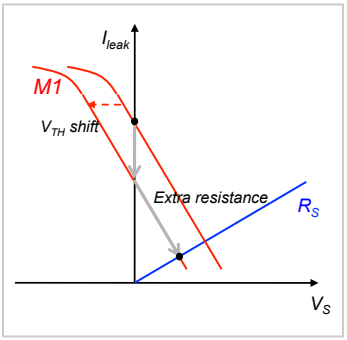
Very often called "MTCMOS" (when using high- and low- threshold devices)

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

Power Gating — Concept

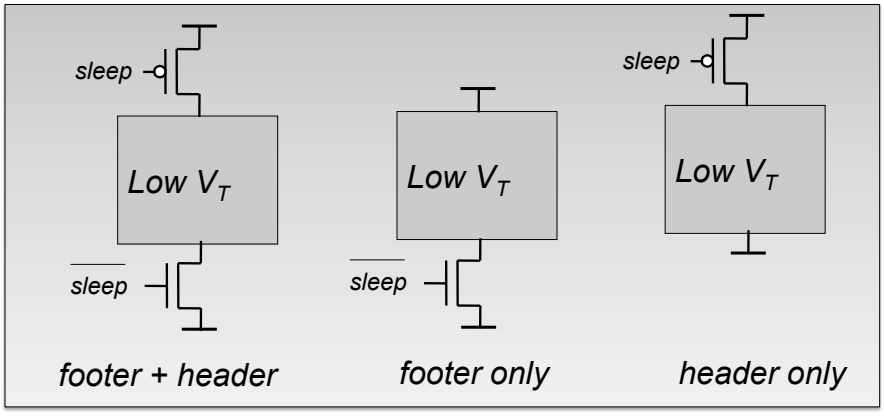


Leakage current reduces because
- Increased resistance in leakage path
- Stacking effect introduces source biasing



(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*