CMOS Power Consumption

Lecture 13

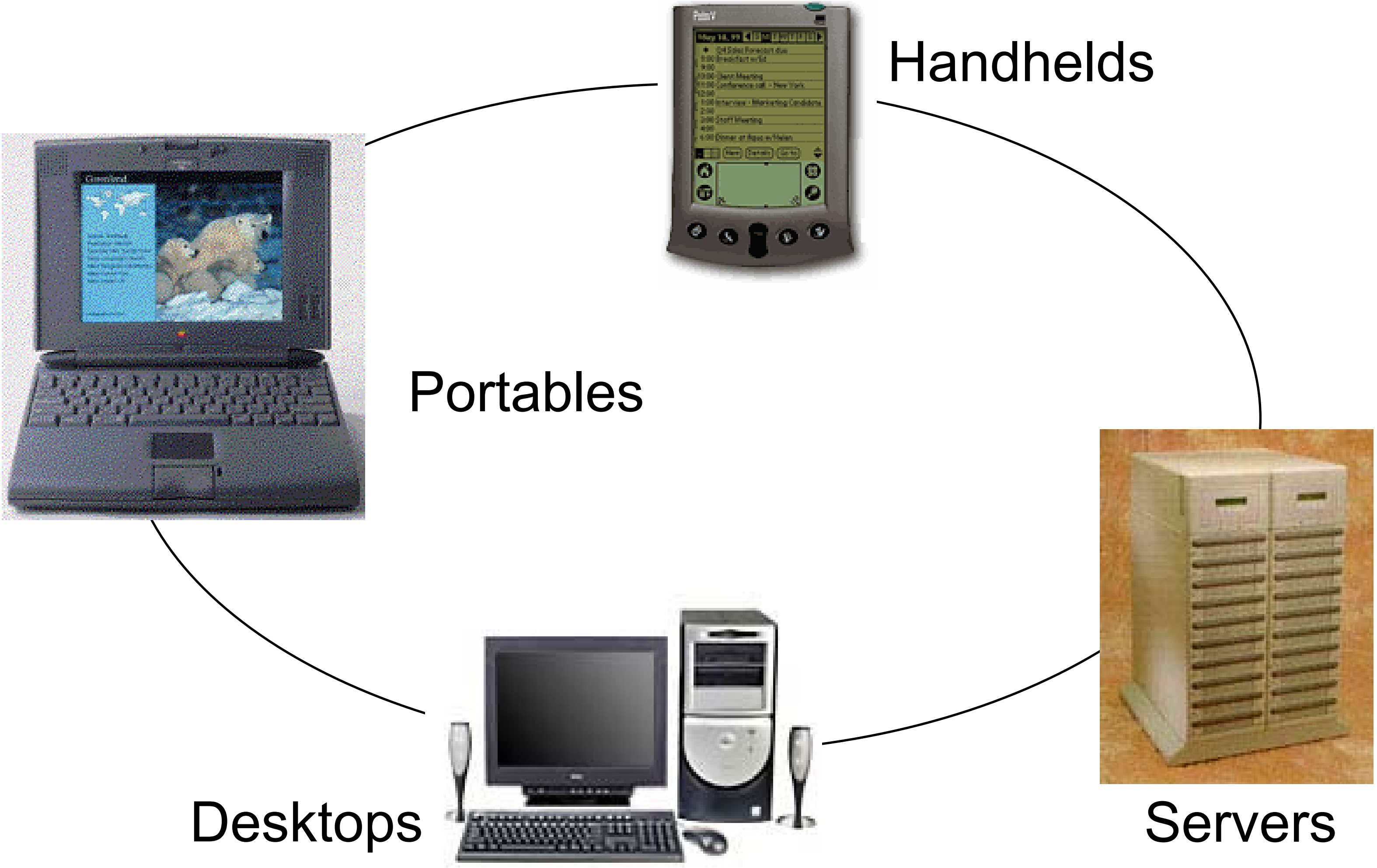
18-322 Fall 2003

Textbook: [Sections 5.5 5.6 6.2 (p. 257-263) 11.7.1]

Overview

- Low-power design
  - Motivation
  - Sources of power dissipation in CMOS
  - Power modeling
  - Optimization Techniques (a survey)

Why worry about power?
-- Heat Dissipation



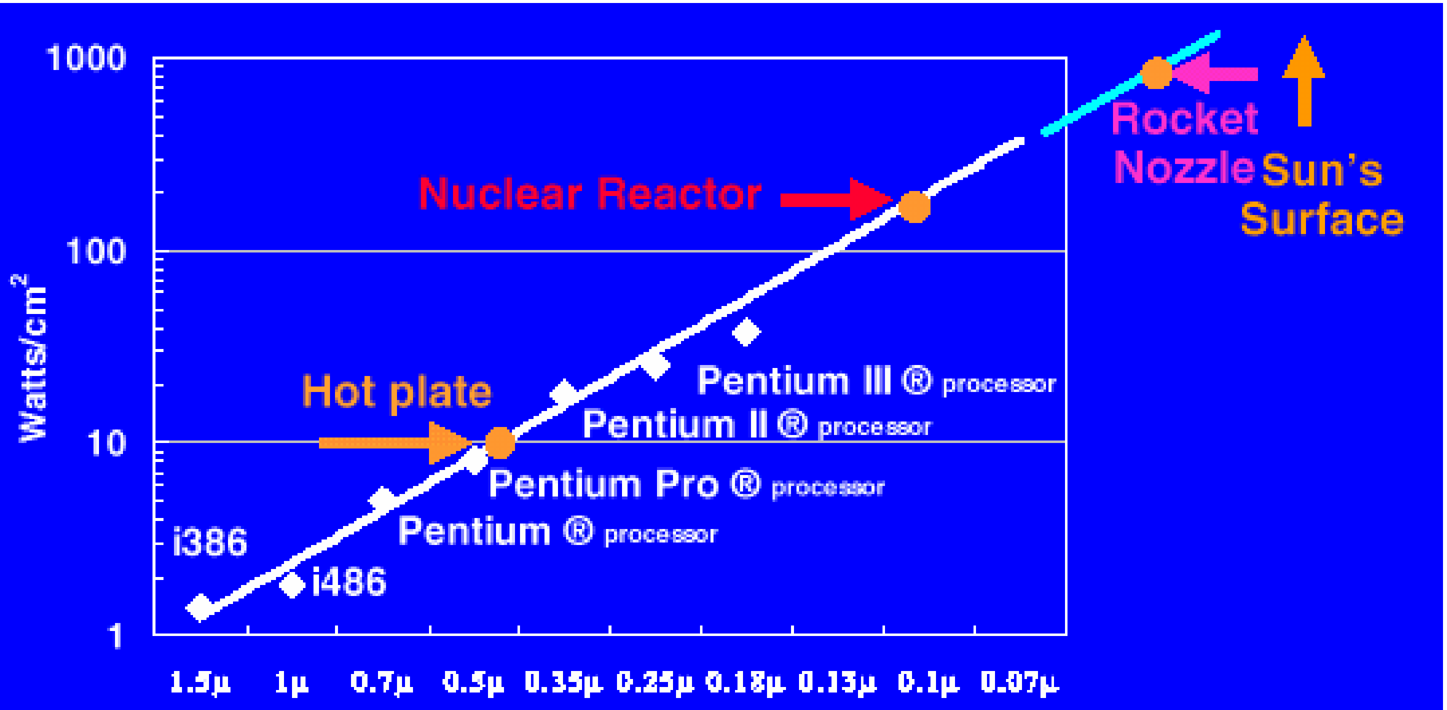
Handhelds

Portables

Desktops

Servers

Power Density Trends



Surpassed hot-plate power density in 0.5μ

Not too long to reach nuclear reactor

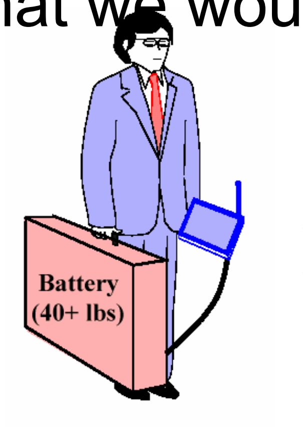
Courtesy of Fred Pollack, Intel
CoolChips tutorial, MICRO-32

High End Power Consumption

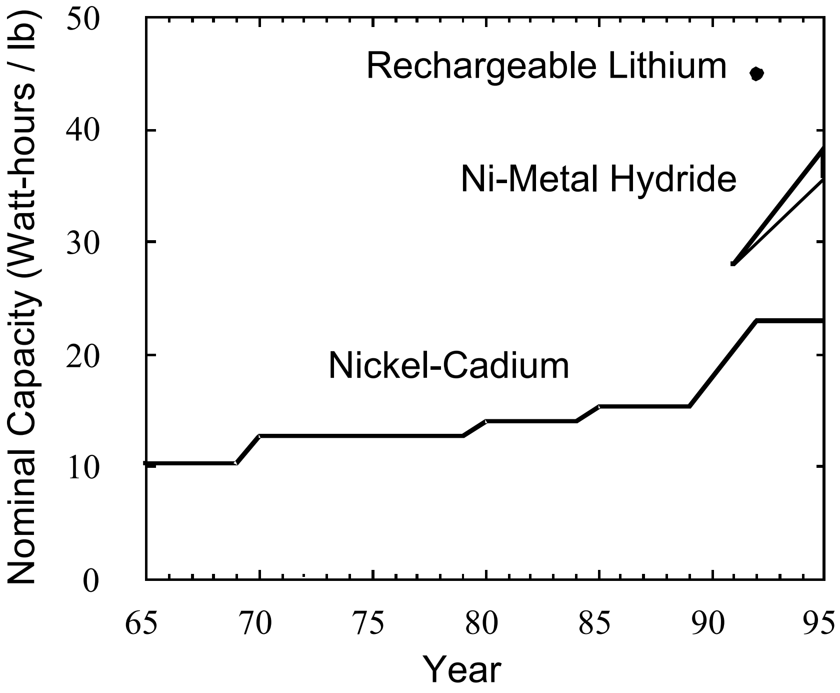
- While you can probably afford to pay for 100-200W of power for your desktop…
- Getting that heat off the chip and out of the box is expensive

A Booming Market: Portable Devices

What we would need…

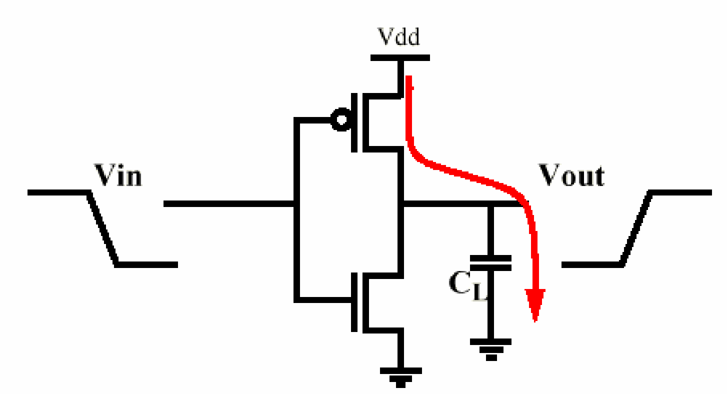


- What we'd like…
  - Video decompression
  - Speech recognition
  - Protocols, ECC, ...
  - Handwriting recognition
  - Text/Graphics processing
  - Java interpreter
- Up to 1 month of uninterrupted operation!



Expected Battery Lifetime increase over next 5 years: 30-40%

Where Does Power Go in CMOS?



- **Switching power**: due to charging and discharging of output capacitances:

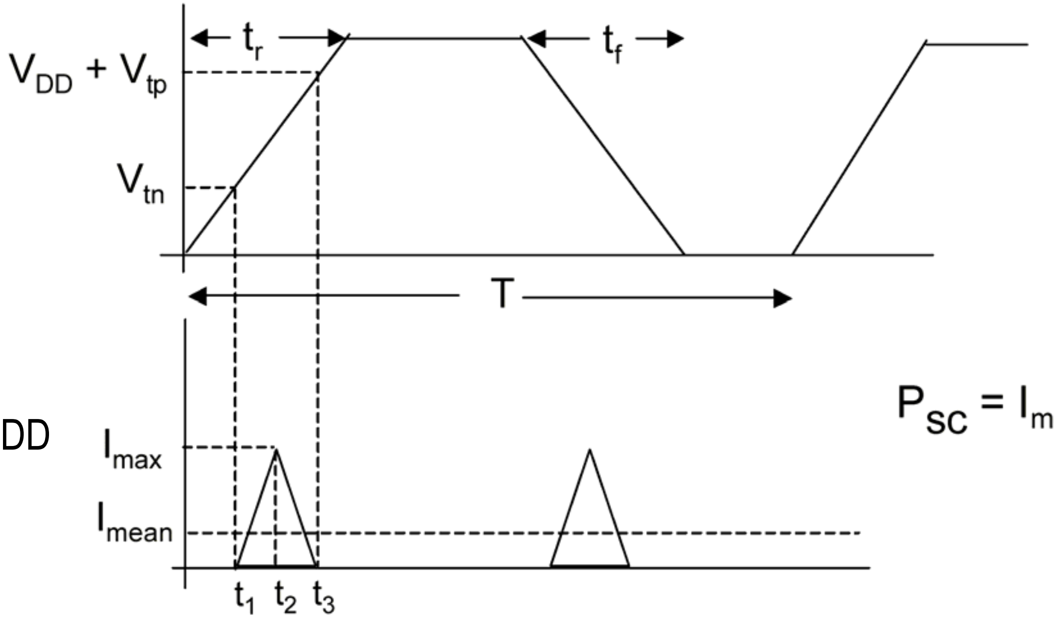
$$\text{Energy/transition} = C_L * V_{dd}^2$$

$$\text{Power} = \text{Energy/transition} * f = C_L * V_{dd}^2 * f$$

- Short-circuit power: due to non-zero rise/fall times
- Leakage power (important with decreasing device sizes)
  - ☒ Typically between 0.1nA - 0.5nA at room temperature

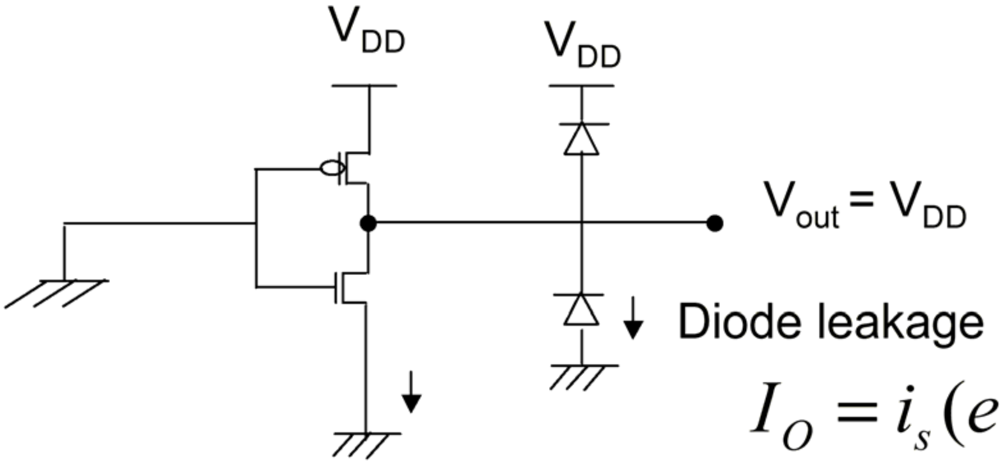
Short-Circuit Power

- Inputs have finite rise and fall times
  - Depends on device sizes
- Direct current path from V_{DD} to GND while PMOS and NMOS are ON simultaneously for a short period



$P_{SC} = I_{mean}.V_{DD}$

Leakage Current



Diode leakage

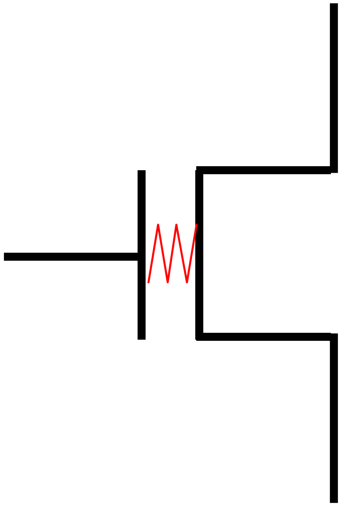
$$I_O = i_s(e^{Vq/kT} - 1)$$

Sub-threshold current

$$I_D = K \cdot e^{(V_{gs} - V_t)q/nkT}(1 - e^{V_{ds}q/kT})$$

New Problem: Gate Leakage

- Now about 20-30% of all leakage, and growing
- Gate oxide is so thin, electrons tunnel thru it…
- NMOS is much worse than PMOS



Gate/Circuit-Level Power Estimation

- It is a very difficult problem
  - Challenges
    - V_{DD}, f_{clk}, C_L are known
      - Actually, the layout will determine the interconnect capacitances
    - Need *node-by-node* accuracy
      - Power dissipation is highly data-dependent
    - Need to estimate switching activity accurately
      - Simulation may take days to complete

Dynamic Power Consumption - Revisited

Power = Energy/transition * transition rate

$$= C_L * V_{dd}^2 * f_{0\rightarrow 1}$$

$$= C_L * V_{dd}^2 * P_{0\rightarrow 1} * f$$

$$= C_{EFF} * V_{dd}^2 * f$$

$$P = C_L(V_{dd}^2/2)\, f_{clk}\, sw$$

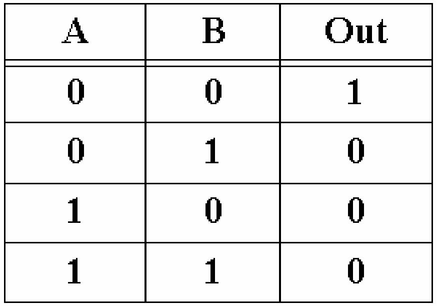
Switching activity (factor) on a signal line

C_{EFF} = Effective Capacitance = $C_L * P_{0\rightarrow 1}$

Power Dissipation is Data Dependent

Function of *Switching Activity*

Example: Static 2 Input NOR



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

Truth Table of 2 input NOR gate

Assume:

$P(A=1) = 1/2$

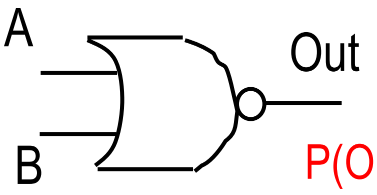
$P(B=1) = 1/2$

Then:

$P(Out=1) = 1/4$ (this is the *signal probability*)

$$P(0 \rightarrow 1) = P(Out = 0) \cdot P(Out = 1)$$
$$= 3/4 \times 1/4 = 3/16$$ (this is the *transition probability*)

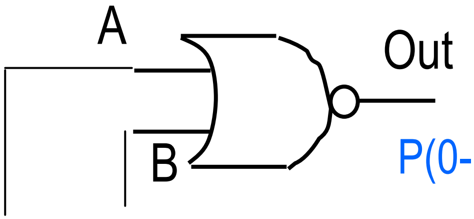
$$C_{EFF} = 3/16\ C_L$$



P(Out =1) = ?

P(0->1) = ?

Power Consumption *is* Data Dependent



P(0->1) = ?

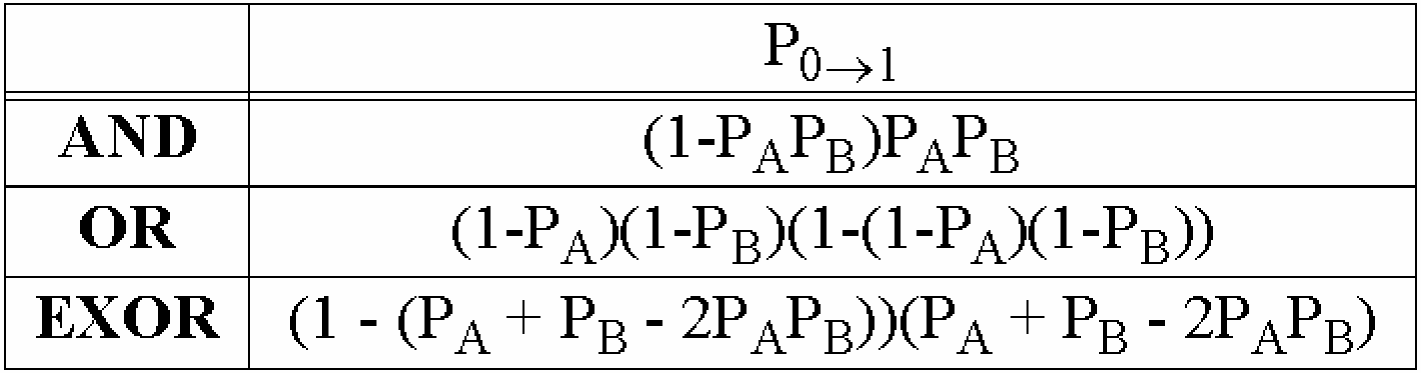
0→0	0→0	1→1
0→0	0→1	1→0
0→0	1→0	0→1
0→0	1→1	0→0
0→1	0→0	1→0
0→1	0→1	1→0
0→1	1→0	0→0
0→1	1→1	0→0
1→0	0→0	0→1
1→0	0→1	0→0
1→0	1→0	0→1
1→0	1→1	0→0
1→1	0→0	0→0
1→1	0→1	0→0
1→1	1→0	0→0
1→1	1→1	0→0

Suppose now that only patterns 00 and 11 can be applied (w/ equal probabilities). Then:

0→0	0→0	1→1	
0→1	0→1	1→0	
1→0	1→0	0→1	=> P(0->1) = 1/4
1→1	1→1	0→0	

Similarly, suppose that every 0 applied to the input A is immediately followed by a 1 while every 1 applied to B is immediately followed by a 0. P(0->1) = ?

Transition Probabilities for Basic Gates

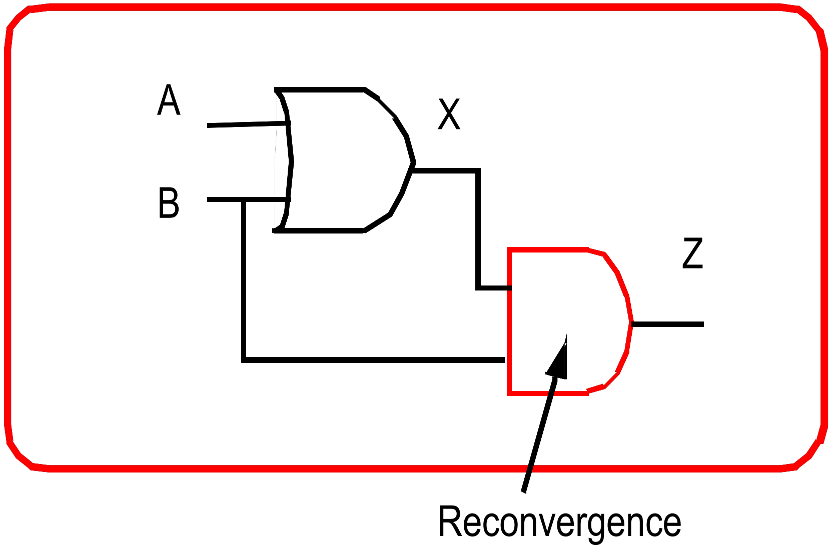


	$P_{0\rightarrow1}$
AND	$(1-P_AP_B)P_AP_B$
OR	$(1-P_A)(1-P_B)(1-(1-P_A)(1-P_B))$
EXOR	$(1 - (P_A + P_B - 2P_AP_B))(P_A + P_B - 2P_AP_B)$

Switching Activity for Static CMOS

$$\mathbf{P_{0\rightarrow1} = P_0 \cdot P_1}$$

(Big) Problem: Re-convergent Fanout



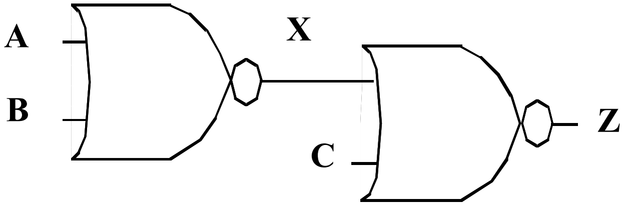
In this case, Z = B as it can be easily seen. The previous analysis simply fails because the signals are not independent!

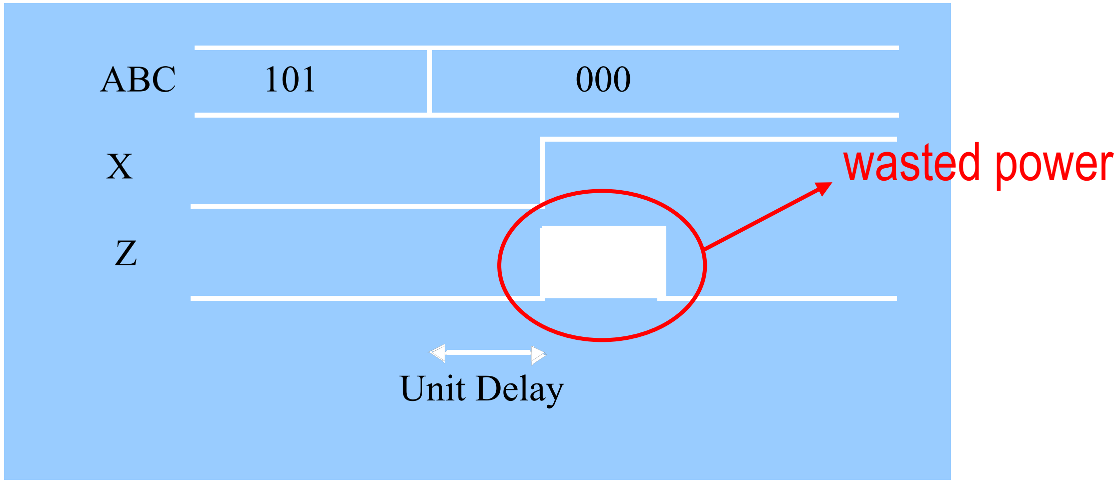
$$\mathbf{P(Z=1) = P(B=1) \cdot P(X=1 \mid B=1) = P(B=1)}$$

Main issue: Becomes complex and intractable real fast!

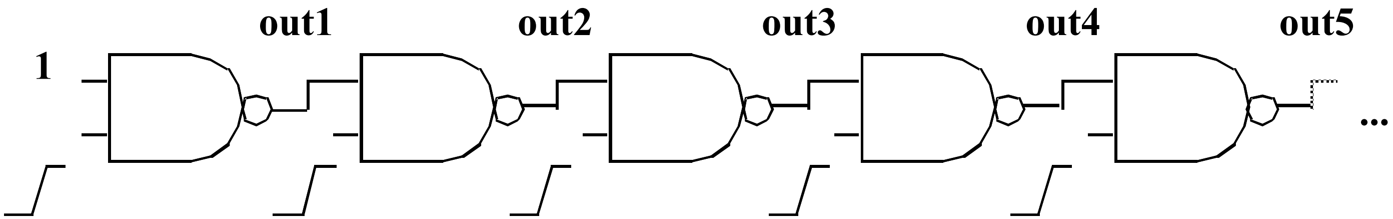
Another (Big) Problem: Glitching in Static CMOS

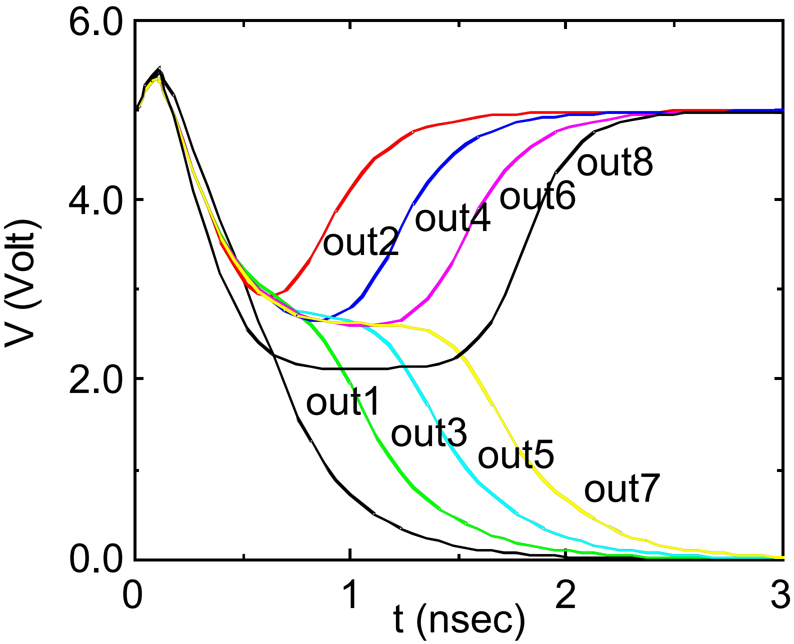
also called: dynamic hazards



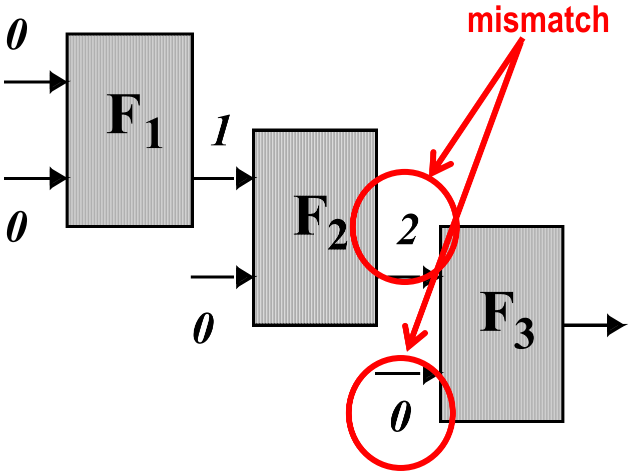


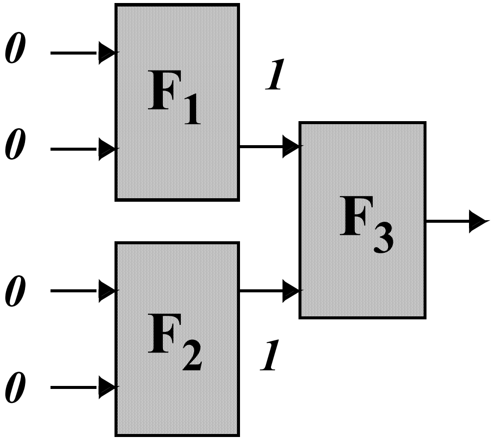
Example: A Chain of NAND Gates





Glitch Reduction Using Balanced Paths

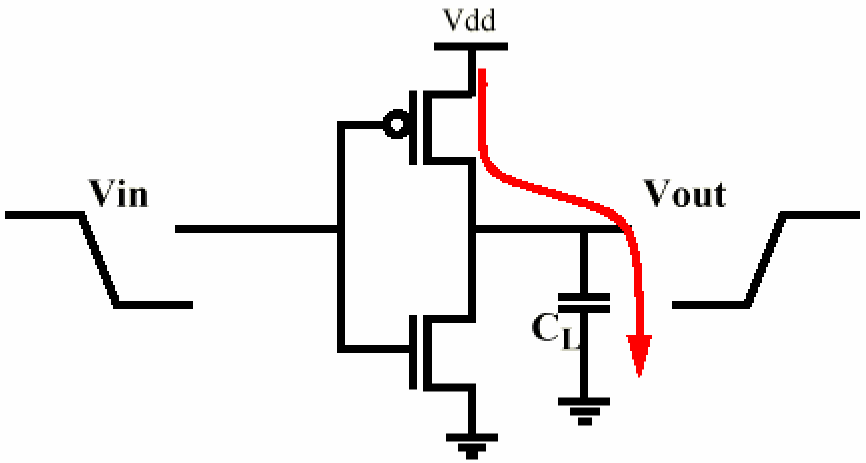




Equalize Lengths of Timing Paths Through Design

Delay is important: Delay vs. V_{DD} and V_T

Think about (Power × Delay) product!

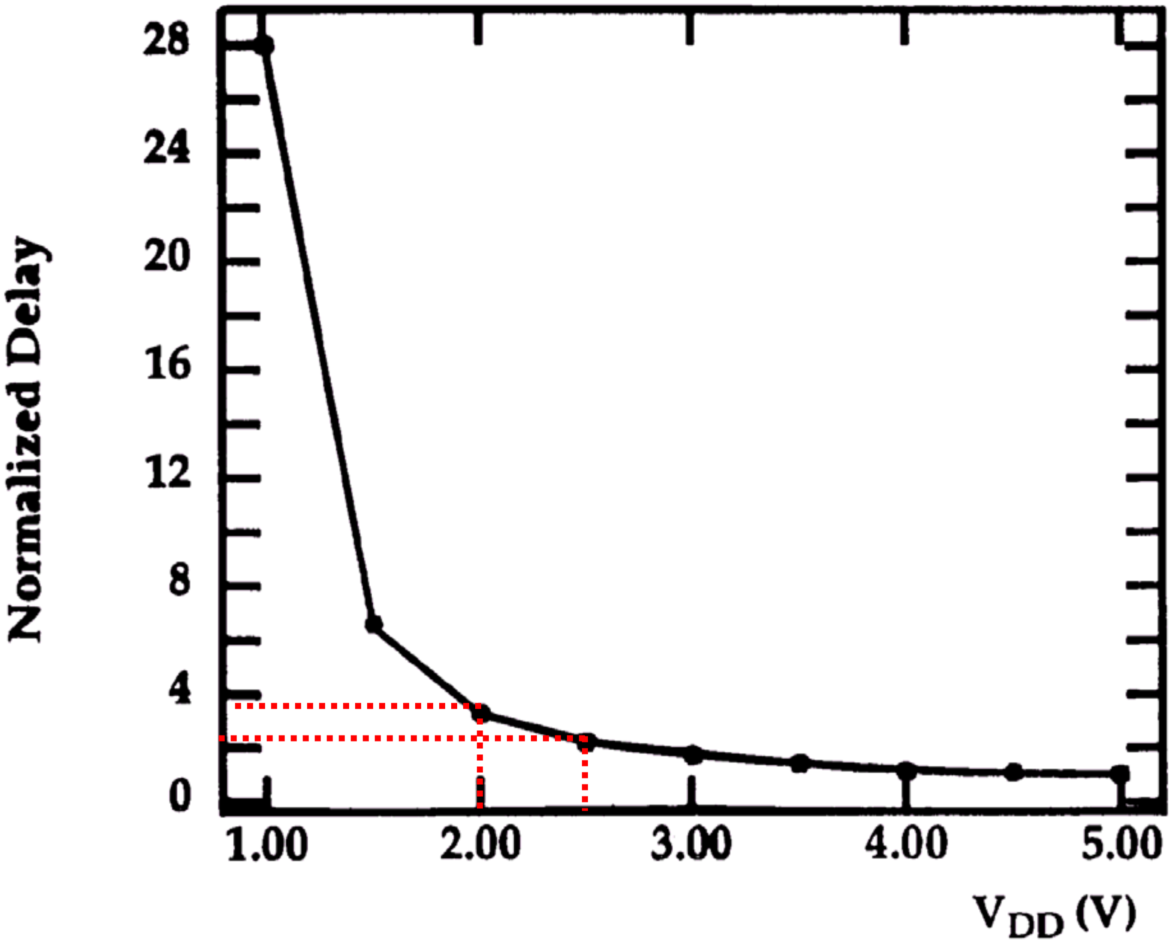


- Delay for a 0->1 transition to propagate to the output:

$$t_{pLH} = \frac{C_L V_{DD}}{k_n \left(V_{DD} - V_{Tn}\right)^2}$$

  - Similar for a 1->0 transition

Delay vs. V_{DD}



Power-Performance Trade-offs

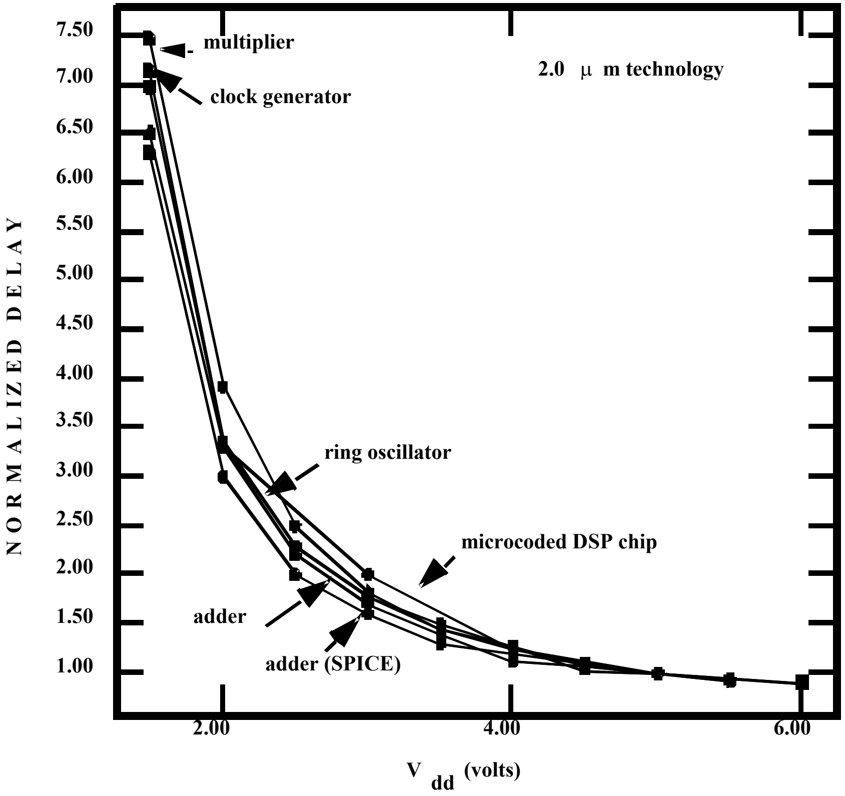
- Prime choice: V_{DD} reduction
  - In recent years we have witnessed an increasing interest in supply voltage reduction (e.g. Dynamic Voltage Scaling)
    - High V_{DD} on critical path or for high performance
    - Low V_{DD} where there is some available slack
  - Design at very low voltages is still an open problem (0.6 – 0.9V by 2010!)
    - Ensures lower power
    - … but higher latency – loss in performance

- Reduce switching activity
  - Logic synthesis
  - Clock gating
- Reduce physical capacitance
  - Proper device sizing
  - Good layout

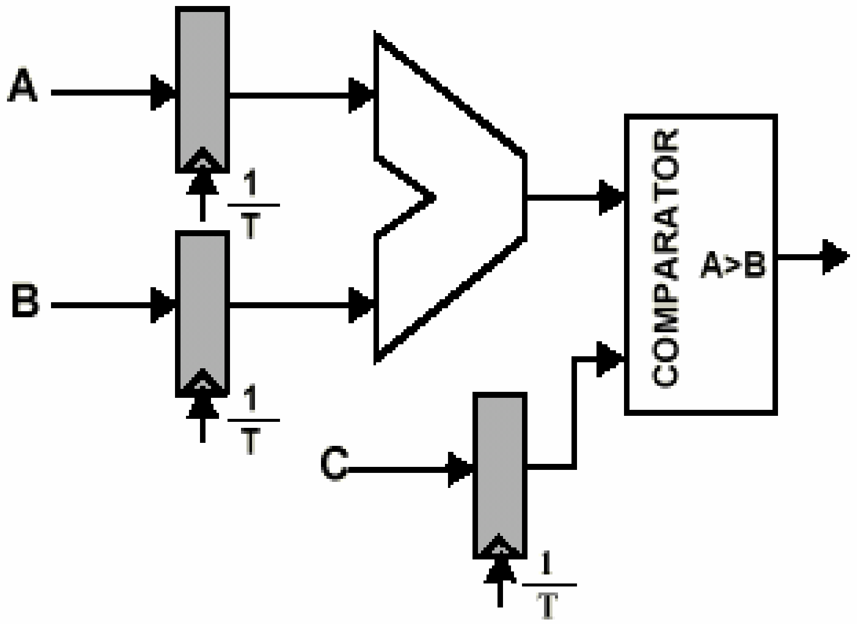
How about POWER?
Ways to reducing power consumption

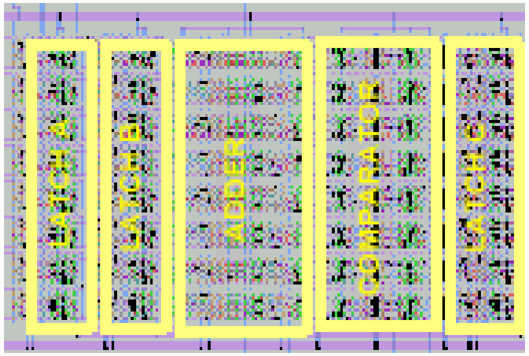
- Load capacitance (C_L)
  - ☒ Roughly proportional to the chip area
- Switching activity (avg. number of transitions/cycle)
  - ☒ Very data dependent
  - ☒ A big portion due to glitches (real-delay)
- Clock frequency (f)
  - ☒ Lowering only f decreases average power, but total energy is the same and throughput is worse

- Voltage supply (V_{DD})
  - – Biggest impact



Using parallelism (1)



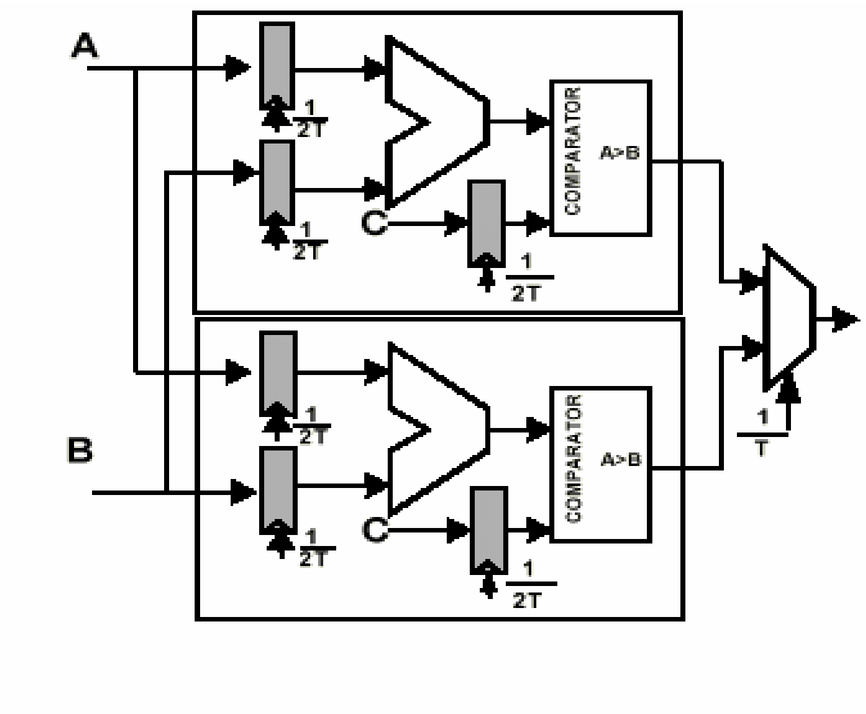


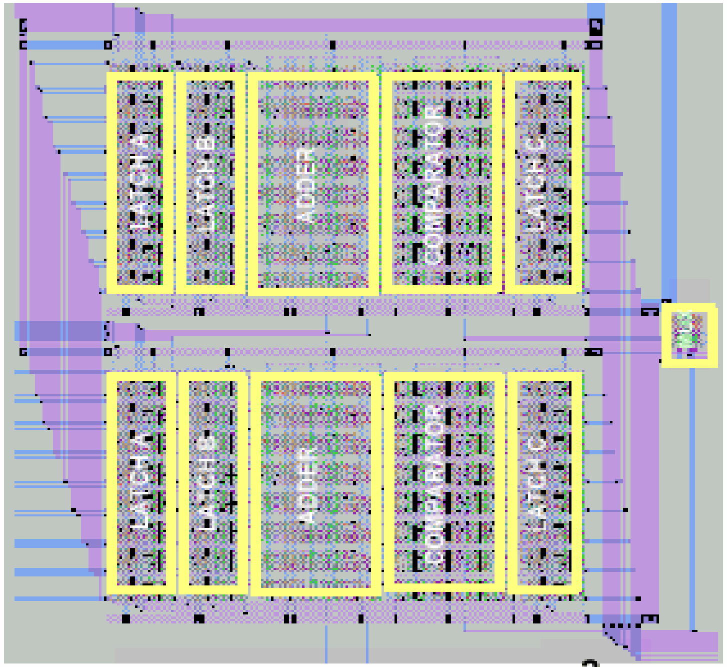
Area = 636 x 833 μ^2

$$P_{ref} = C_{ref} V_{DD}^2 f_{ref}$$

Assume: t_p = 25ns (worst-case, *all* modules) at V_{DD} = 5V

Using parallelism (2)



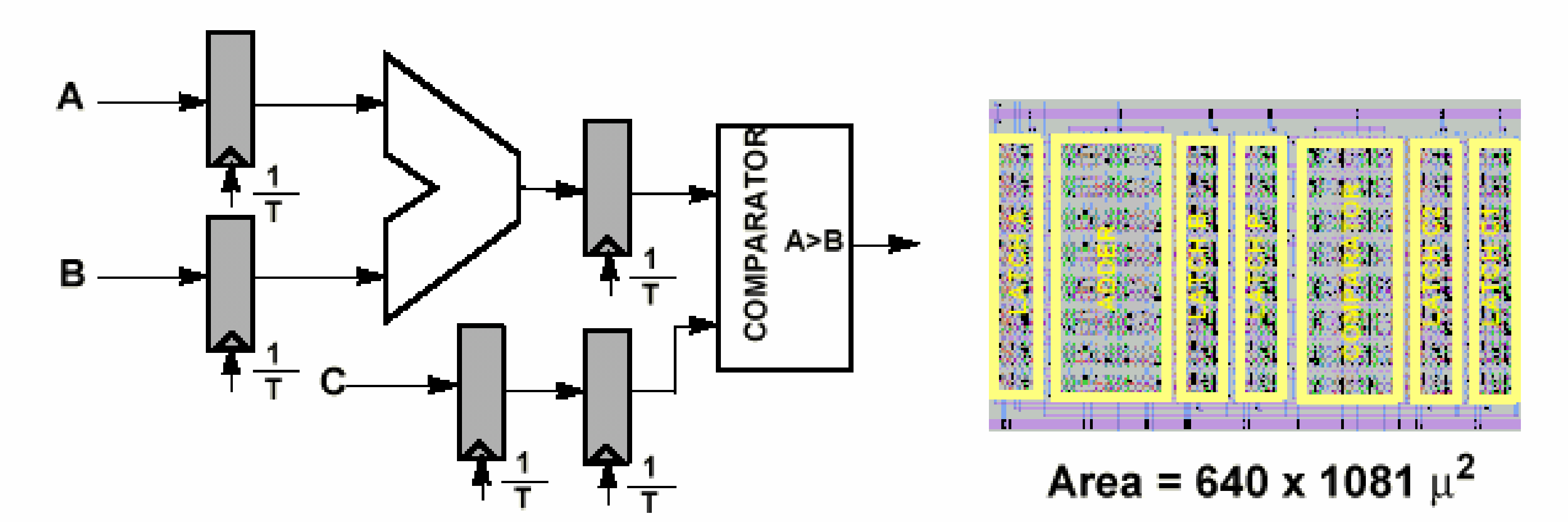


Area = 1476 x 1219 μ^2

Area increases about 3.4 times!

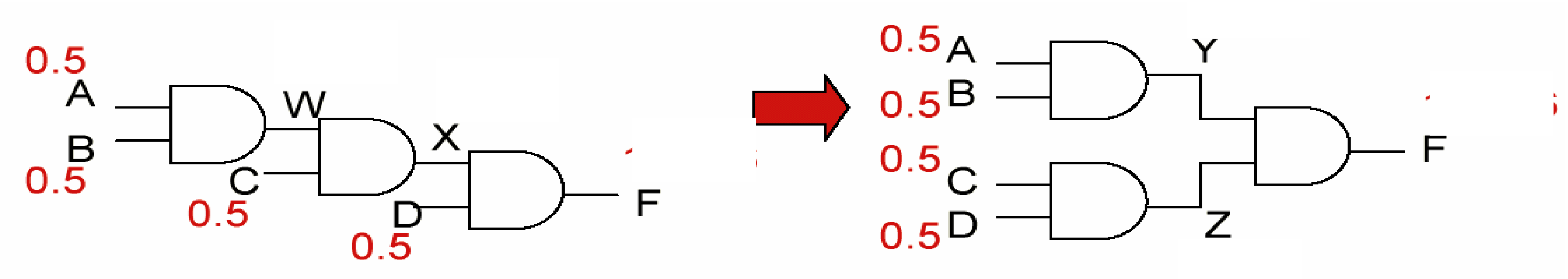
- C_{par} = 2.15C (extra-routing needed)
- f_{par} = f/2 ($t_{p,new}$ = (50)ns => V_{DD} ~ 2.9V; $V_{DD,par} = 0.58\ V_{DD}$)
- $P_{par} = C_{par} V_{DD}^2 f_{par} = 0.36\ P_{ref}$

Using pipelining



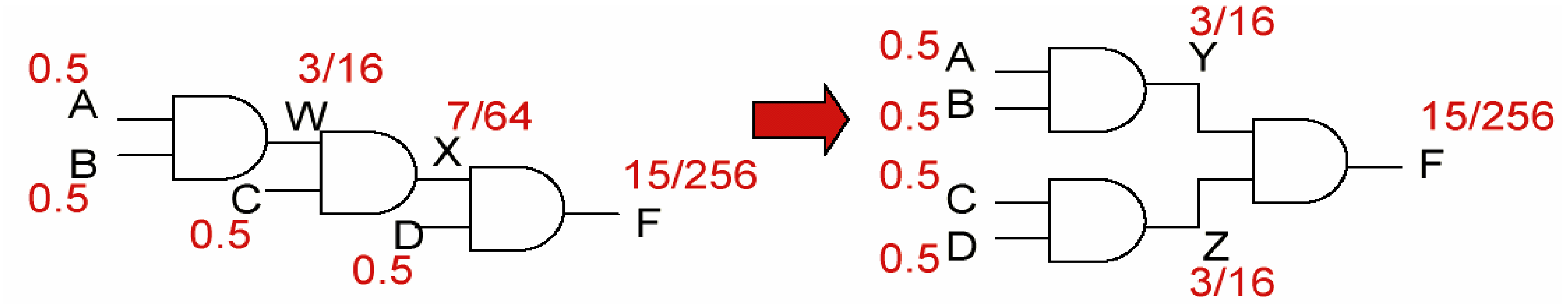
- C_{pipe} = 1.15C
- Delay decreases 2 times ($V_{DD,pipe}$ = 0.58 V_{DD})
- P_{pipe} = 0.39 P

Chain vs. balanced design



- *Question for you*:
  - Which of the two designs is more energy efficient?
    - Assume:
      - Zero-delay model
      - All inputs have a signal probability of 0.5
    - Hint: Calculate $p_{0\to1}$ for W, X and F

Chain vs. balanced design

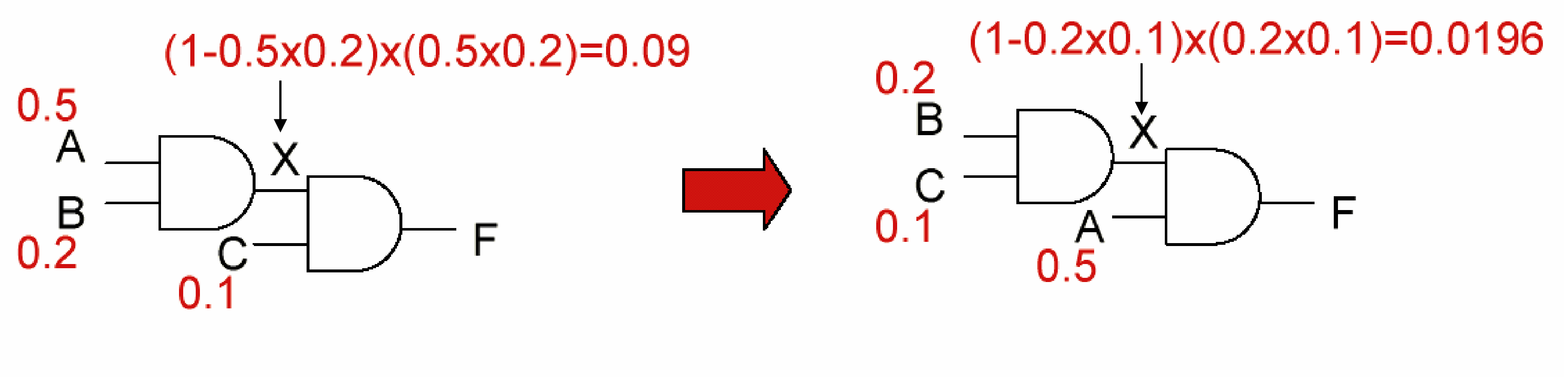


- For the zero-delay model
  - Chain design is better
  - But ignores glitching
    - Depending on the gate delays, the chain design may be worse

Low energy gates – transistor sizing

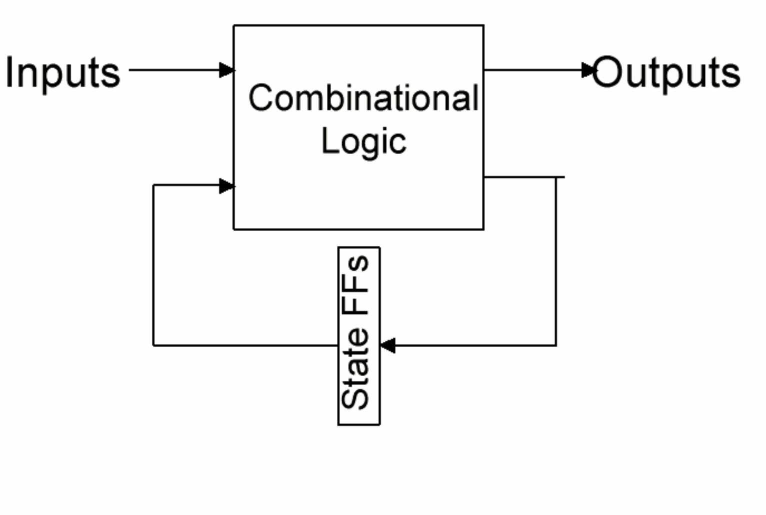
- Use the *smallest transistors* that satisfy the delay constraints
  - Increasing transistor size improves the speed but it also increases power dissipation (since the load capacitances increases)
    - Slack time - difference between required time and arrival time of a signal at a gate output
      - Positive slack - size down
      - Negative slack - size up
- Make gates that toggle more frequently smaller

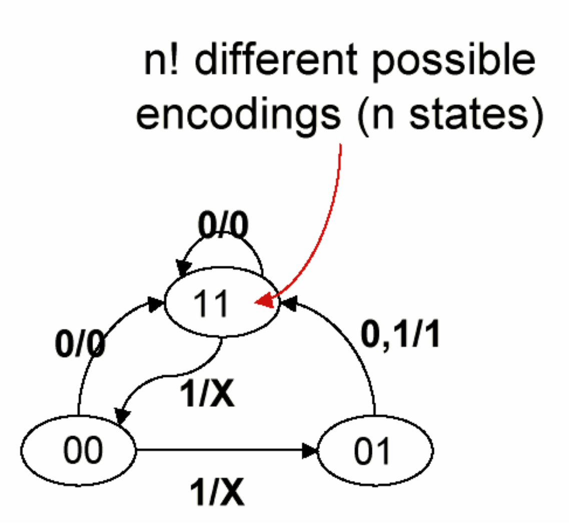
Low energy gate netlists – pin ordering



- Better to postpone the introduction of signals with a high transition rate (signals with signal probability close to 0.5)

Control circuits





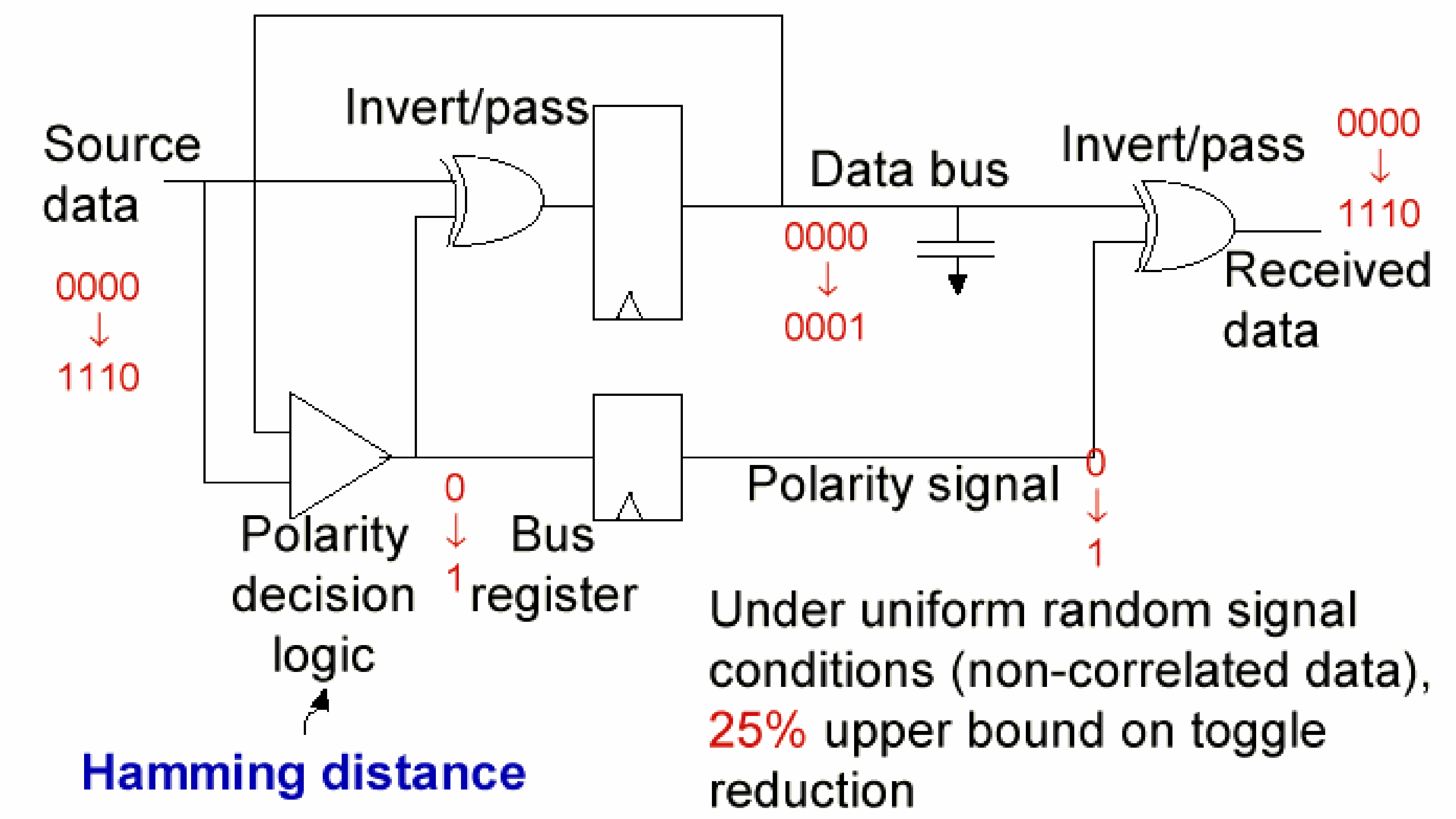
- State encoding has a big impact on the power efficiency
- Energy driven -> try to minimize number of bit transitions in the state register
  - Fewer transitions in state register
  - Fewer transitions propagated to combinational logic

Bus encoding

- Reduces number of bit toggles on the bus
- Different flavors
  - Bus-invert coding
    - Uses an extra bus line *invert*:
      - if the number of transitions is < *K*/2, invert = 0 and the symbol is transmitted as is
      - if the number of transitions is > *K*/2, invert = 1 and the symbol is transmitted in a complemented form
  - Low-weight coding
    - Uses *transition* signaling instead of *level* signaling

Encoder
Bus
Decoder

Bus invert coding



Under uniform random signal conditions (non-correlated data), 25% upper bound on toggle reduction

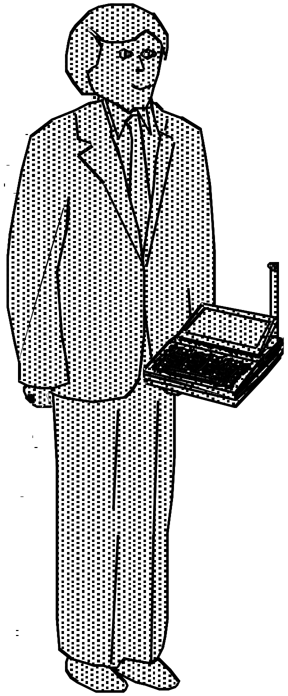
Source: M.Stan et al., 1994

Summary

- Power Dissipation is already a prime design constraint

- Low-power design requires operation at lowest possible voltage and clock speed

- Low-power design requires optimization at all levels of abstraction



Announcements

- Project M1:
  - Check off in lab session
  - Report by Friday
- Exam Review Session:
  - Monday Oct 13, 4:30-6:30pm
  - PH 125C