Energy-Aware Computing

Lecture 6:Gate-level energy-saving techniques

Parameters

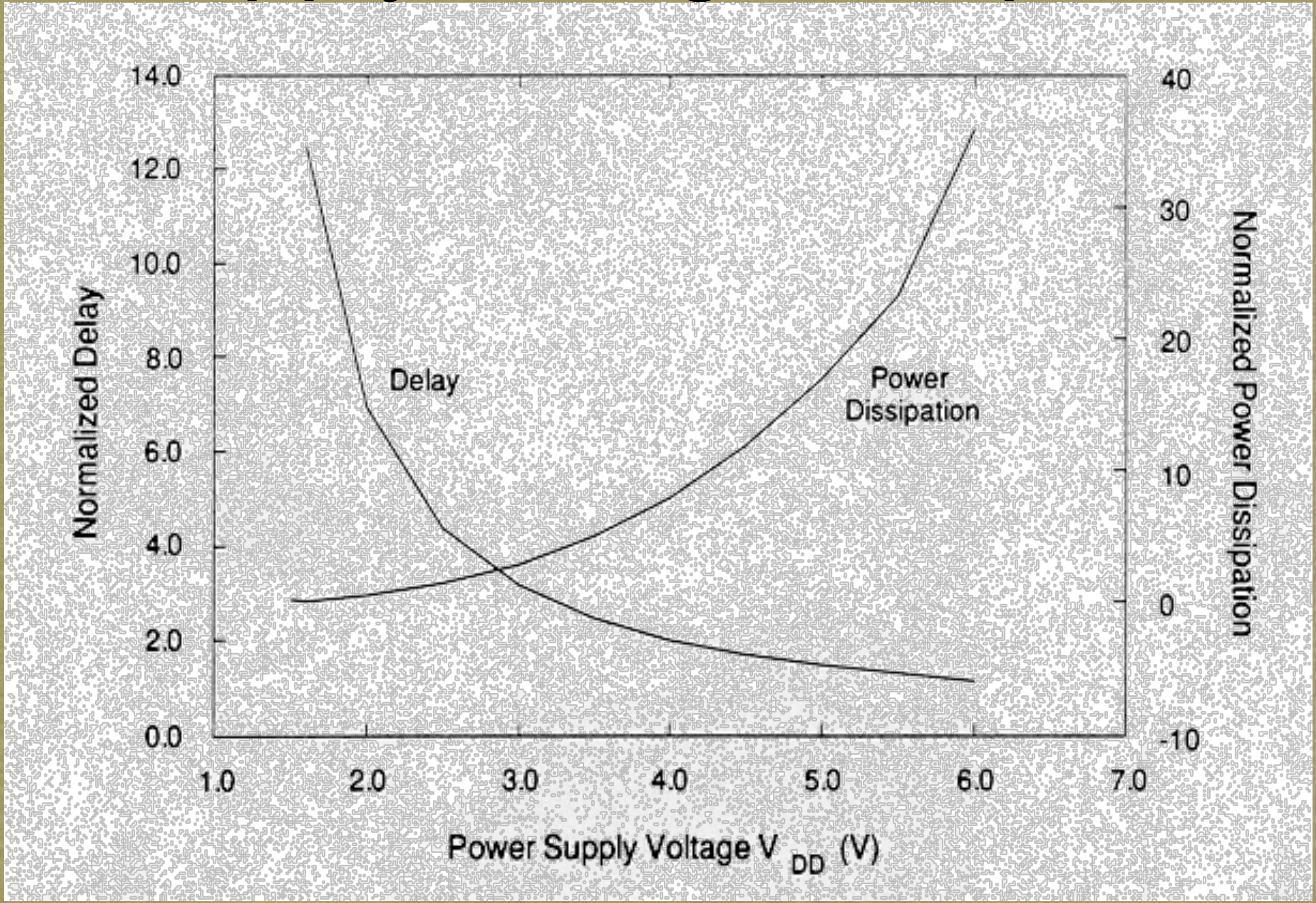
$$P = C_L \cdot V_{DD}^2 \cdot P_{0\to 1} \cdot f$$

- Frequency - clock rate
  - Ok for power, no effect for energy
- Supply voltage
  - Quadratic effect (!) but slows down circuit
- Capacitance
- Activity
  - “algorithm”/design
  - Other issues, e.g. glitches

Voltage

- Relation between voltage and speed
  - Threshold voltage
  - How low can we go?
- Dual supply voltage
- Low voltage signal swing
- Architecture driven voltage reduction
- Dynamic voltage scaling

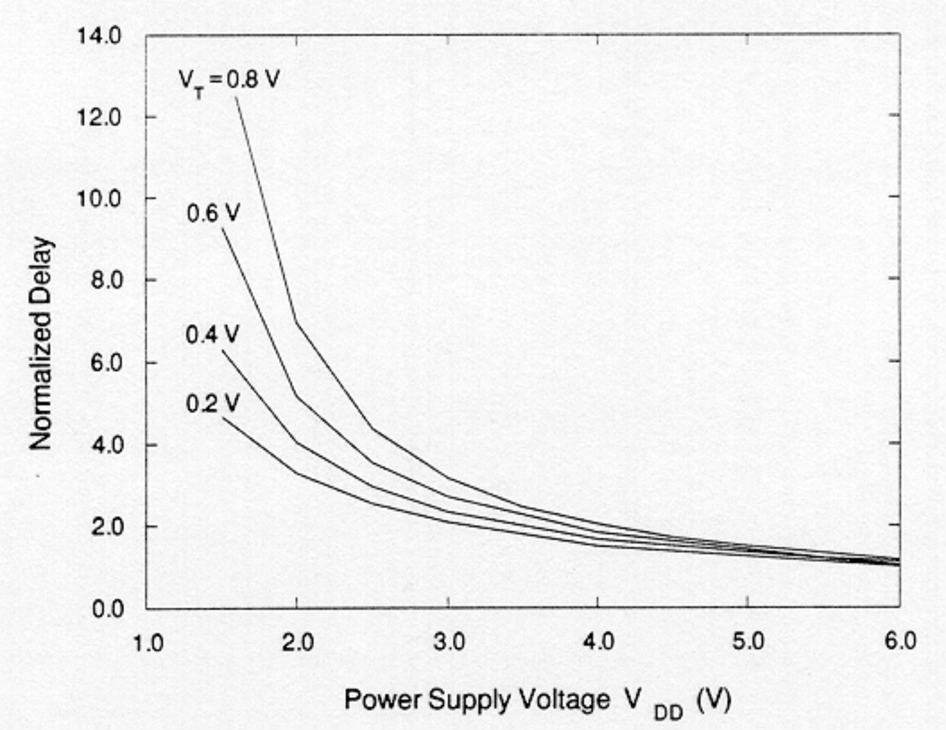
Supply voltage vs speed



Source: Mlynek, Leblebici, design of VLSI systems course, EPFL

Effect of threshold voltage

$$T_d \propto \frac{V_{dd}}{\left(V_{dd} - V_t\right)^2}$$



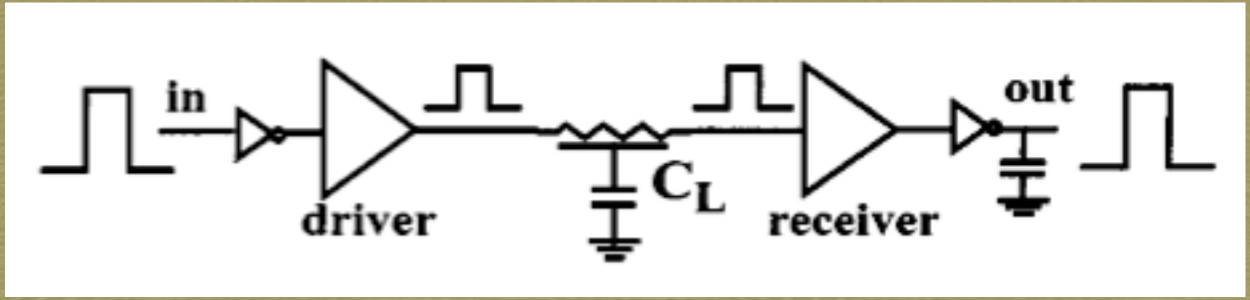
Source: Mlynek, Leblebici, design of VLSI systems course, EPFL

- Why not scale Vt (and Vdd) aggressively?
  - Leakage increases exponentialy with Vt
  - More on leakage in a future lecture

Dual-supply voltages

- Two (or more) supply voltages
  - High Vdd for gates on the critical path
  - Low Vdd for gates off the path
- Little if any speed loss
- Expensive
  - More design effort. E.g. level converters
  - More space: 2 power delivery systems
- Not popular

Low voltage swing



- Reduce the voltage swing of key signals
  - Signal recovery required at receiver
  - Differential signaling (2 wires per bit)
  - Single-rail with local reference voltage at receiver
- Can be combined with multi-value logic

Low voltage swing

$$P = C_{eff} \cdot V_{dd} \cdot V_{swing} \cdot f$$

- Good targets
  - High capacitance wires, e.g. busses, the clock
  - High activity wires, e.g. the clock
- Reduced immunity to noise
  - Signal may be lost
  - Noise is becoming a hard problem in deep sub-micron technologies

Architecture-driven voltage scaling

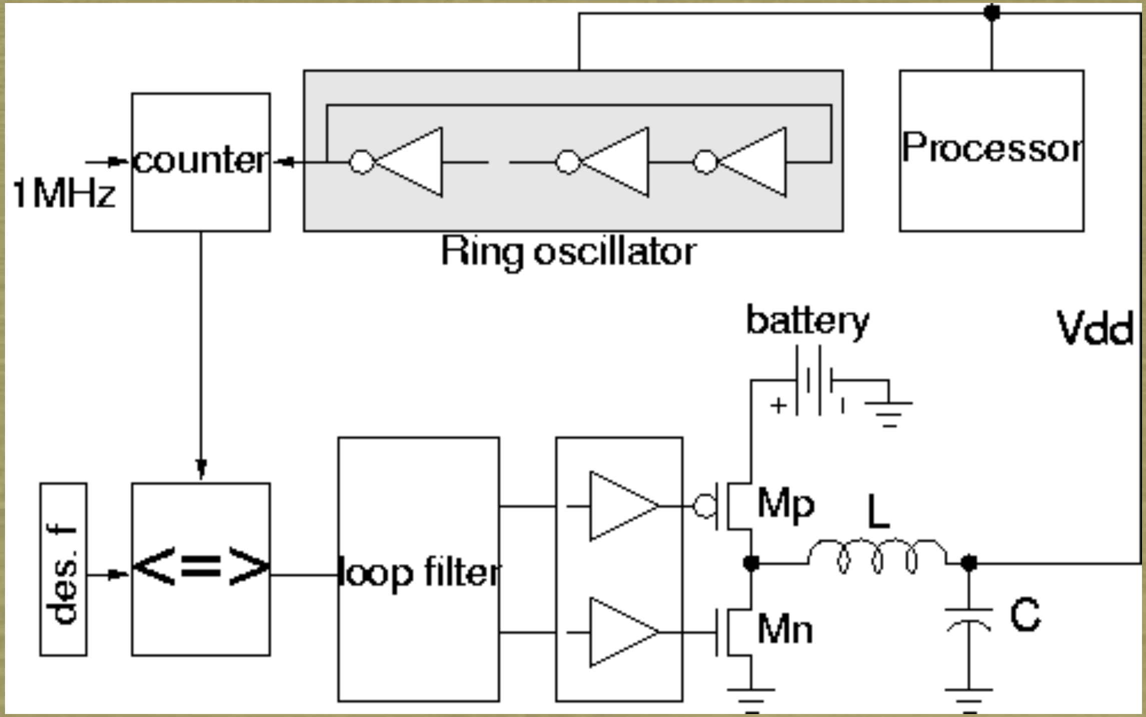
- Aggressively reduce the supply voltage
- Recover lost speed by parallelism
  - Extra pipeline stages and/or unit duplication
  - Mostly good for maintaining *throughput* rate
    - Operations start at a fast rate
  - *Latency* can still suffer
    - Results take longer to appear
  - Area increase, sometimes dramatic
- Adder-comparator example from Chandrakassan'95 (duplication)
  - $C_{par} = 2.15C_{ref}$, $V_{par} = 0.58V_{ref}$, $f_{par} = f_{ref}/2$
  - $P_{par} = 0.36P_{ref}$

Dynamic voltage scaling

- Allow run-time, dynamic adaptation of supply voltage and operating frequency
  - Multiple power/speed operation points
  - Software (OS) decides which is best
- Careful circuit design required
  - Some circuit types can fail
- Extra circuit, voltage regulator, required
  - Some energy loss incurred

Voltage regulator

- "User" sets required frequency (des. f), regulator finds corresponding voltage
- Ring oscillator matches critical path in all conditions



Src: Burd, Brodersen. Design Issues for Dynamic Voltage Scaling. ISLPED'00

Switched capacitance

- Logic style
  - Static vs dynamic logic gates
  - Transistor sizes
- Logic function
- Circuit topology
- Data statistics
- Sequencing of operations
  - E.g. encoding of FSM states

Static vs dynamic

- Dynamic circuits remove the PMOS stack with 1 transistor controlled by a “clock” signal
  - Precharge in every cycle
  - Discharge depending on input values
- Dynamic circuits have
  - Higher switching activities
  - Higher clock load
  - ✓ No spurious transitions (glitches)
  - ✓ Lower input capacitances
- No clear winner in general
- Dynamic circuits are hard to use in automated design flow

Low energy logic gates

- Transistor sizing
  - – Small transistors in logic gates out of critical path
  - – Lower input capacitances, smaller circuits, lower interconnect capacitances
- Dual threshold voltages
  - – Enables the use of low supply voltage
  - – Use low-Vt (fast) gates for critical path
  - – Use high-Vt (slow) gates for other paths

Logic function

- The probability of a 0→1 transition at the output depends on the logic function
- E.g. 2-input NOR with all possible input combinations equally likely. p(0)=3/4, p (1) = 1/4
  - $P_{0\to 1} = p(0)\,p(1) = 3/16$
- 2-input XOR
  - $P_{0\to 1} = 1/2 \cdot 1/2 = 1/4$
- Extended to networks of logic gates

Signal probabilities

- Signal probability:
  - Average fraction of clock cycles when signal is high
- Propagate from inputs using Shannon's decomposition:

$$y = f(x_0, \ldots, x_n) = x_i f_{x_i} + \bar{x}_i f_{\bar{x}_i}$$

$$f_{x_i} = f(x_0, \ldots, 1, \ldots, x_n)$$

$$f_{\bar{x}_i} = f(x_0, \ldots, 0, \ldots, x_n)$$

$$P(y) = P(x_i f_{x_i}) + P(\bar{x}_i f_{\bar{x}_i})$$

$$= P(x_i)P(f_{x_i}) + P(\bar{x}_i)P(f_{\bar{x}_i})$$

Transition density

- For power/energy we need to find the probability of transition from 0 to 1
- **Transition density** D(y) **-** average number of transitions per unit time
  - Includes both 0-1 and 1-0 transitions
  - $D(y) = 2 \cdot P_{0 \to 1}(y) \cdot f$

- For a transition on x to cause a transition on y

$$\partial y / \partial x_i = f_{x_i} \oplus f_{\bar{x}_i} = 1$$

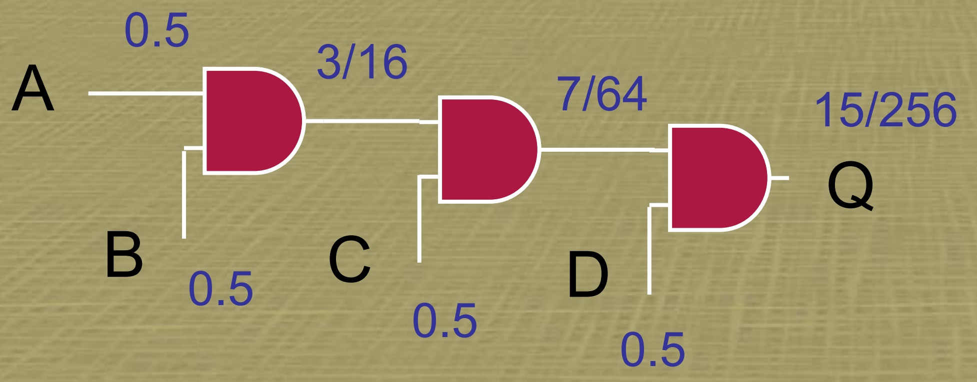
- Total density of y: $D(y) = \sum P(\partial y / \partial x_i) D(x_i)$
- Average power:

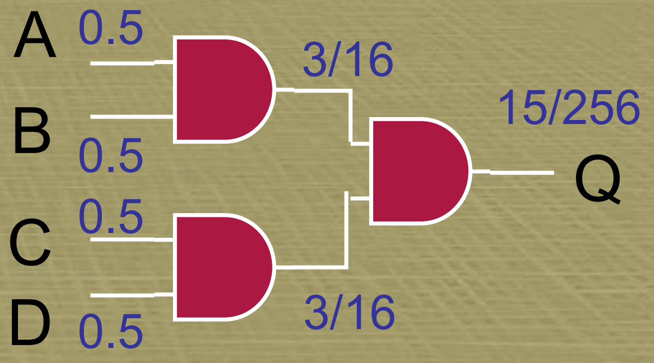
$$P = \sum 0.5 C_y V_{dd}^2 D(y)$$

Circuit topology

- Two components of switching activity
  - Static - only up to 1 toggle per gate
    - Timing behaviour is ignored
  - Dynamic
    - Spurious transitions, glitches
- Balancing the delay paths reduces glitching

Circuit topology example





- Chain topology has lower combined static switching activity
- Tree topology is better for reducing glitches
- Similarly it is better to move signals with high transition rate as close to the output as possible

Glitches

- Estimated to account for 15-20% of dynamic power
- Not present in dynamic logic circuits
- Glitches depend on the *logic depth* of the circuit
  - Max number of logic gates from input to output
  - Arrival times spread due to delay imbalances
- Pipelining reduces logic depth, reduces power due to glitches
  - But more *clock load* - capacitance driven by the clock signal in every cycle

Summary

- Voltage reduction
  - Relationship with speed
  - Dual supply, dual Vt
  - Low swing, architecture-driven scaling, dynamic scaling
- Effective capacitance
  - Logic style
  - Logic function
  - Probabilistic estimation of power
  - Glitches