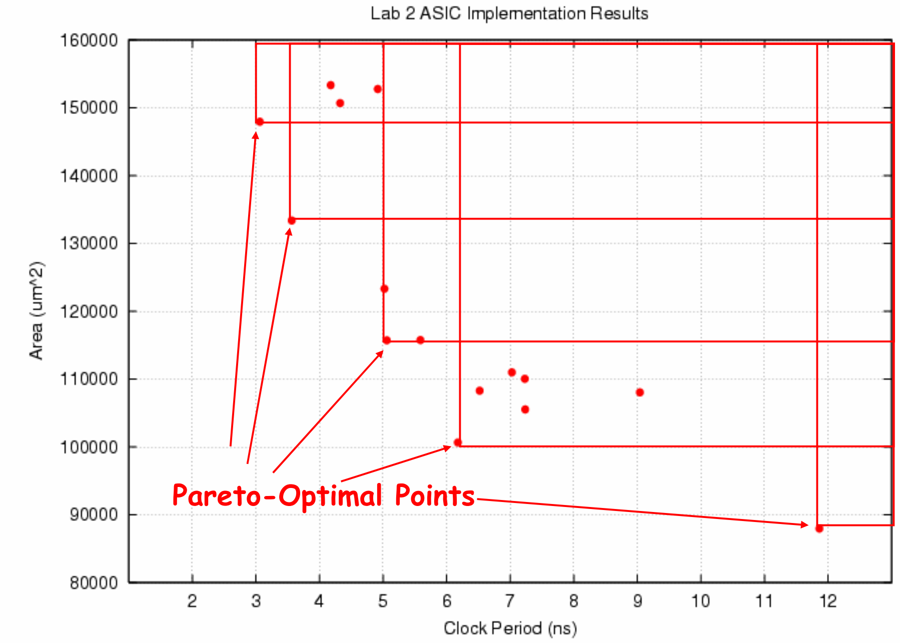
Power

Lab 2 Results



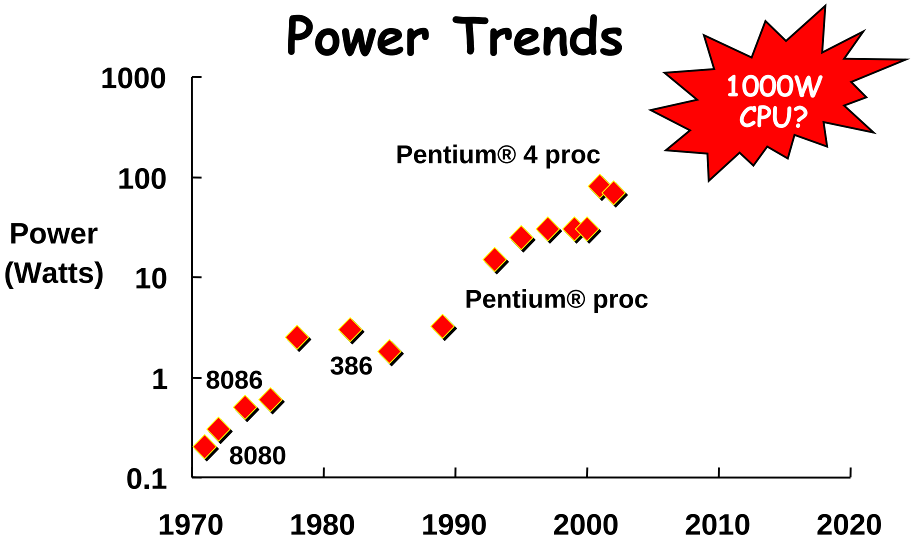
Standard Projects

- Two basic design projects
  - Processor variants (based on lab1&2 testrigs)
  - Non-blocking caches and memory system
  - Possible project ideas on web site
- Must hand in proposal before quiz on March 18th, including:
  - Team members (2 or 3 per team)
  - Description of project, including the architecture exploration you will attempt

Non-Standard Projects

- Must hand in proposal early by class on March 14th, describing:
  - Team members (2 or 3)
  - The chip you want to design
  - The existing reference code you will use to build a test rig, and the test strategy you will use
  - The architectural exploration you will attempt

Power Trends



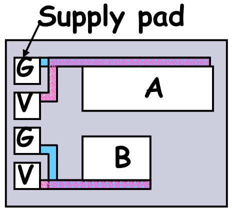
[Source: Intel]

- CMOS originally used for very low-power circuitry such as wristwatches
- Now some CPUs have power dissipation >100W

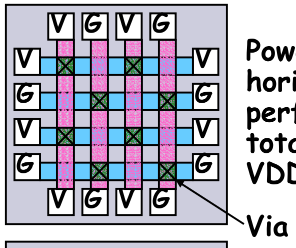
Power Concerns

- Power dissipation is limiting factor in many systems
  - battery weight and life for portable devices
  - packaging and cooling costs for tethered systems
  - case temperature for laptop/wearable computers
  - fan noise not acceptable in some settings
- Internet data center, ~8,000 servers,~2MW
  - 25% of running cost is in electricity supply for supplying power and running air-conditioning to remove heat
- Environmental concerns
  - ~2005, 1 billion PCs, 100W each => *100 GW*
  - 100 GW = 40 Hoover Dams

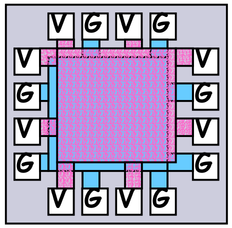
On-Chip Power Distribution



Routed power distribution on two stacked layers of metal (one for VDD, one for GND). OK for low-cost, low-power designs with few layers of metal.

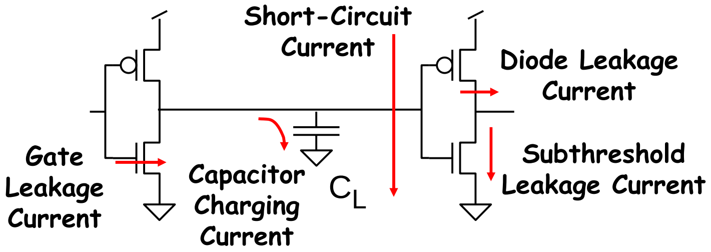


Power Grid. Interconnected vertical and horizontal power bars. Common on most high-performance designs. Often well over half of total metal on upper thicker layers used for VDD/GND.



Dedicated VDD/GND planes. Very expensive. Only used on Alpha 21264. Simplified circuit analysis. Dropped on subsequent Alphas.

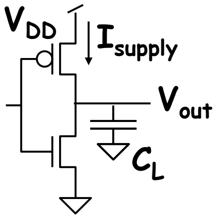
Power Dissipation in CMOS



Primary Components:

- ❑ Capacitor charging, energy is 1/2 CV^2 per transition
  - the dominant source of power dissipation today
- ❑ Short-circuit current, PMOS & NMOS both on during transition
  - kept to <10% of capacitor charging current by making edges fast
- ❑ Subthreshold leakage, transistors don't turn off completely
  - approaching 10-40% of active power in <180nm technologies
- ❑ Diode leakage from parasitic source and drain diodes
  - usually negligible
- ❑ Gate leakage from electrons tunneling across gate oxide
  - was negligible, increasing due to very thin gate oxides

Energy to Charge Capacitor



$$E_{0\to 1} = \int_0^T P(t)\,dt = V_{DD}\int_0^T I_{supply}(t)\,dt$$

$$= V_{DD}\int_0^{V_{DD}} C_L\,dV_{out} = C_L\,V_{DD}^2$$

- During 0->1 transition, energy $C_L V_{DD}^2$ removed from power supply
- After transition, 1/2 $C_L V_{DD}^2$ stored in capacitor, the other 1/2 $C_L V_{DD}^2$ was dissipated as heat in pullup resistance
- The 1/2 $C_L V_{DD}^2$ energy stored in capacitor is dissipated in the pulldown resistance on next 1->0 transition

Power Formula

Power = activity * frequency * (1/2 CV_{DD}^2 + $V_{DD}I_{SC}$)
+ $V_{DD}I_{Subthreshold}$
+ $V_{DD}I_{Diode}$
+ $V_{DD}I_{Gate}$

- *Activity is average number of transitions per clock cycle (clock has two)*

Switching Power

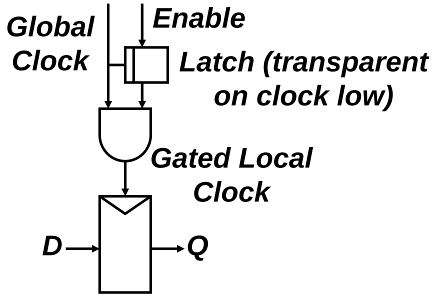
Power $\propto$ activity * 1/2 CV^2 * frequency

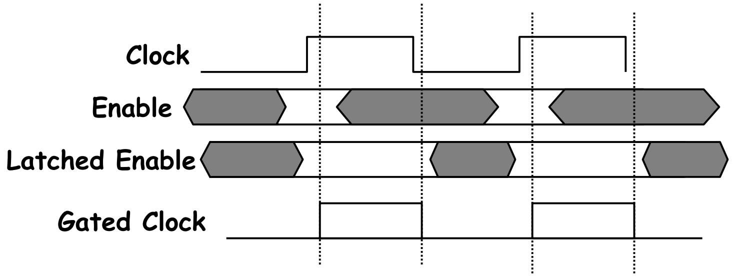
- Reduce activity
- Reduce switched capacitance C
- Reduce supply voltage V
- Reduce frequency

Reducing Activity with Clock Gating

Clock Gating

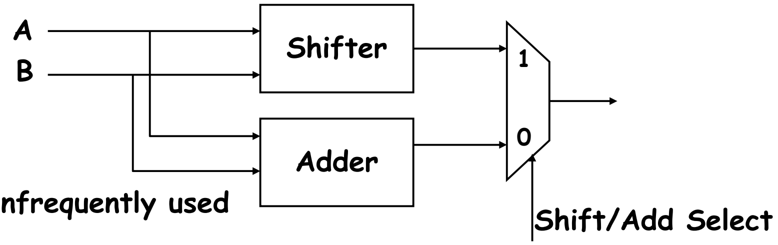
- – don't clock flip-flop if not needed
- – avoids transitioning downstream logic
- – enable adds to control logic complexity
- – Pentium-4 has hundreds of gated clock domains



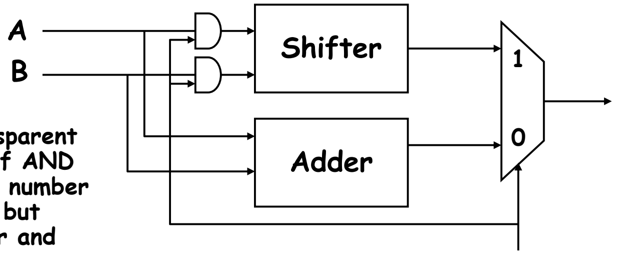


Reducing Activity with Data Gating

Avoid data toggling in unused unit by gating off inputs



Shifter infrequently used



Could use transparent latch instead of AND gate to reduce number of transitions, but would be bigger and slower.

Other Ways to Reduce Activity

Bus Encodings

- choose encodings that minimize transitions on average (e.g., Gray code for address bus)
- compression schemes (move fewer bits)

Freeze "Don't Cares"

- If a signal is a don't' care, then freeze last dynamic value (using a latch) rather than always forcing to a fixed 1 or 0.
- E.g., 1, X, 1, 0, X, 0 ===> 1, X=1, 1, 0, X=0, 0

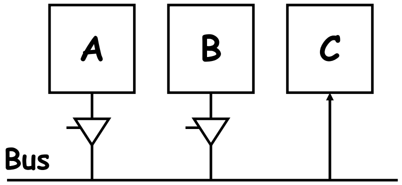
Remove Glitches

- balance logic paths to avoid glitches during settling

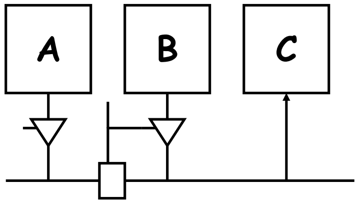
Reducing Switched Capacitance

Reduce switched capacitance C

- Careful transistor sizing (small transistors off critical path)
- Tighter layout (good floorplanning)
- Segmented structures (avoid switching long nets)



Shared bus driven by A or B when sending values to C



Insert switch to isolate bus segment when B sending to C

Reducing Frequency

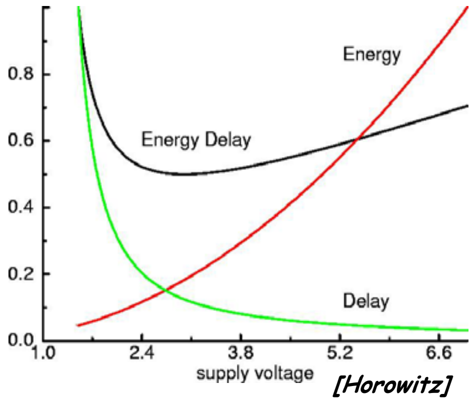
Doesn't save energy, just reduces rate at which it is consumed (lower power, but must run longer)

- Get some saving in battery life from reduction in rate of discharge

Reducing Supply Voltage

Quadratic savings in energy per transition (1/2 CV_{DD}^{2})

- *Circuit speed is reduced*
- *Must lower clock frequency to maintain correctness*



[Horowitz]

$$T_d = \frac{CV_{DD}}{k(V_{DD} - V_{th})^{\alpha}}$$

$$\alpha = 1 - 2$$

Delay rises sharply as supply voltage approaches threshold voltages

Voltage Scaling for Reduced Energy

- Reducing supply voltage by 0.5 improves energy per transition by ~0.25
- Performance is reduced – need to use slower clock
- Can regain performance with parallel architecture
- Alternatively, can trade surplus performance for lower energy by reducing supply voltage until "just enough" performance

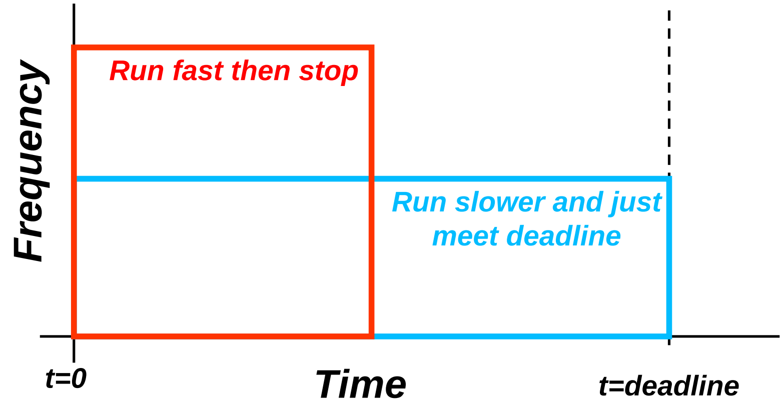
Dynamic Voltage Scaling

Parallel Architectures Reduce Energy at Constant Throughput

- 8-bit adder/comparator
  40MHz at 5V, area = 530 kμ^2
  Base power Pref
- Two parallel interleaved adder/compare units
  20MHz at 2.9V, area = 1,800 kμ^2 (3.4x)
  Power = 0.36 Pref
- One pipelined adder/compare unit
  40MHz at 2.9V, area = 690 kμ^2 (1.3x)
  Power = 0.39 Pref
- Pipelined and parallel
  20MHz at 2.0V, area = 1,961 kμ^2 (3.7x)
  Power = 0.2 Pref

Chandrakasan et. al. "Low-Power CMOS Digital Design",
IEEE JSSC 27(4), April 1992

"Just Enough" Performance



❑ Save energy by reducing frequency and voltage to minimum necessary

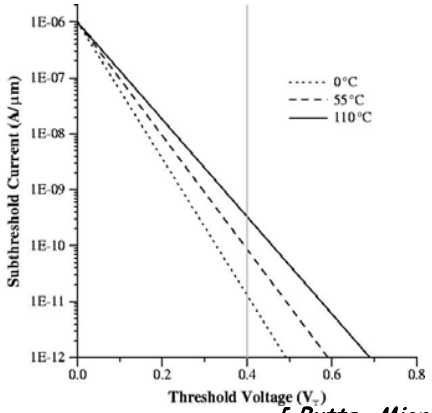
Voltage Scaling on Transmeta Crusoe TM5400

Frequency (MHz)	Relative Performance (%)	Voltage (V)	Relative Energy (%)	Relative Power (%)
700	100.0	1.65	100.0	100.0
600	85.7	1.60	94.0	80.6
500	71.4	1.50	82.6	59.0
400	57.1	1.40	72.0	41.4
300	42.9	1.25	57.4	24.6
200	28.6	1.10	44.4	12.7

Leakage Power

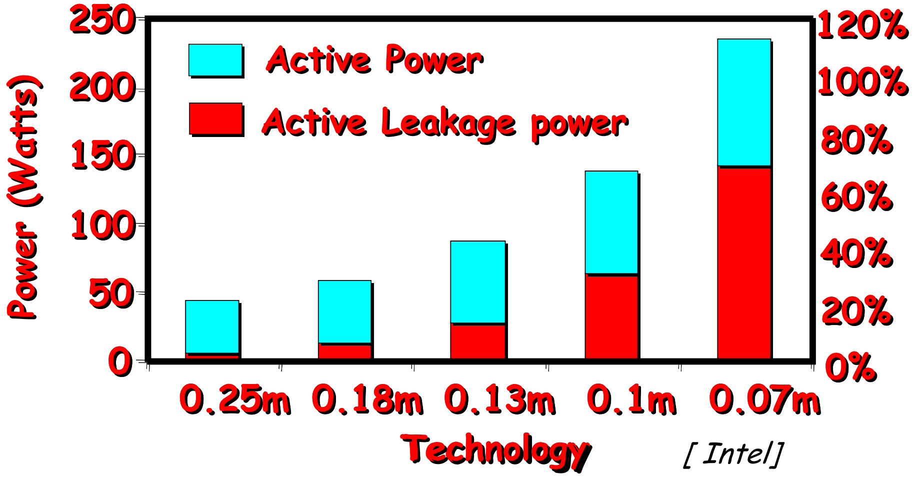
- Under ideal scaling, want to reduce threshold voltage as fast as supply voltage
- But subthreshold leakage is an exponential function of threshold voltage and temperature

$$I_{subthreshold} = k\, e^{\frac{-q V_T}{a\, k_B\, T}}$$



[Butts, Micro 2000]

Rise in Leakage Power



[Intel]

Design-Time Leakage Reduction

Use slow, low-leakage transistors off critical path

- leakage proportional to device width, so use smallest devices off critical path
- leakage drops greatly with stacked devices (acts as drain voltage divider), so use more highly stacked gates off critical path
- leakage drops with increasing channel length, so slightly increase length off critical path
- dual V_T - process engineers can provide two thresholds (at extra cost) use high V_T off critical path (modern cell libraries often have multiple V_T)

Critical Path Leakage

Critical paths dominate leakage after applying design-time leakage reduction techniques

Example: PowerPC 750

5% of transistor width is low Vt, but these account for *>50%* of total leakage

Possible approach, run-time leakage reduction

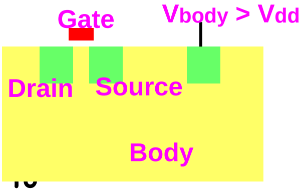
– switch off critical path transistors when not needed

Run-Time Leakage Reduction

- Body Biasing

  V_t increase by reverse-biased body effect

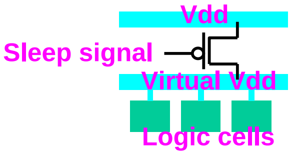
  Large transition time and wakeup latency due to well cap and resistance



- Power Gating

  Sleep transistor between supply and virtual supply lines

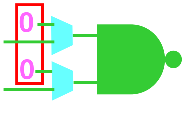
  Increased delay due to sleep transistor



- Sleep Vector

  Input vector which minimizes leakage

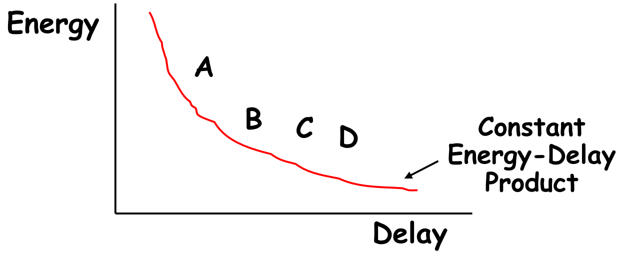
  Increased delay due to mux and active energy due to spurious toggles after applying sleep vector



Power Reduction for Cell-Based Designs

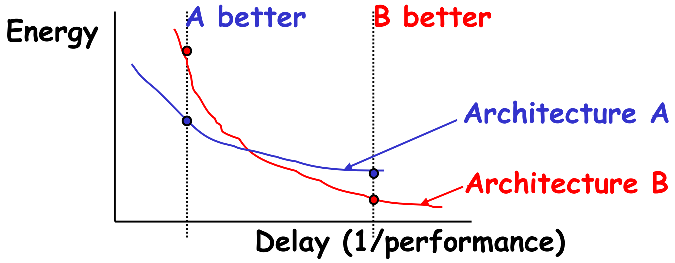
- Minimize activity
  - Use clock gating to avoid toggling flip-flops
  - Partition designs so minimal number of components activated to perform each operation
  - Floorplan units to reduce length of most active wires
- Use lowest voltage and slowest frequency necessary to reach target performance
  - Use pipelined architectures to allow fewer gates to reach target performance (reduces leakage)
  - After pipelining, use parallelism to further reduce needed frequency and voltage if possible
- Always use energy-delay plots to understand power tradeoffs

Energy versus Delay



- Can try to compress this 2D information into single number
  - Energy*Delay product
  - Energy*Delay^2 – gives more weight to speed, mostly insensitive to supply voltage
- Many techniques can exchange energy for delay
- Single number (ED, ED^2) often misleading for real designs
  - usually want minimum energy for given delay or minimum delay for given power budget
  - can't scale all techniques across range of interest
- To fully compare alternatives, should plot E-D curve for each solution

Energy versus Delay

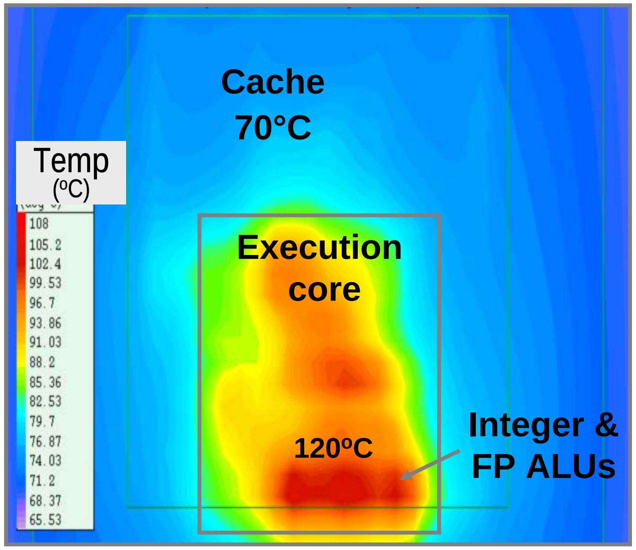


- Should always compare architectures at the same performance level or at the same energy
- Can always trade performance for energy using voltage/frequency scaling
- Other techniques can trade performance for energy consumption (e.g., less pipelining, fewer parallel execution units, smaller caches, etc)

Temperature Hot Spots

- Not just total power, but power density is a problem for modern high-performance chips
- Some parts of the chip get much hotter than others
  - Transistors get slower when hotter
  - Leakage gets exponentially worse (can get thermal runaway with positive feedback between temperature and leakage power)
  - Chip reliability suffers
- Few good solutions as yet
  - Better floorplanning to spread hot units across chip
  - Activity migration, to move computation from hot units to cold units
  - More expensive packaging (liquid cooling)

Itanium Temperature Plot



[Source: Intel]