
Low Power Design

Power Consumption

- Why is power consumption important?
- Battery powered devices
 - Maximize battery life
- Minimize cost of wall-powered systems
 - Plastic packaging is 10x cheaper than ceramic packaging but can only dissipate 1 - 2 W
 - What happens if the chip gets too hot?
 - Need a fan to cool somewhere above 10 W
 - Difficult to air cool at all somewhere above 50-100 W
 - Cost of power supply
- `Green' systems
 - Minimize pollution by reducing demand from power stations

Power Consumption

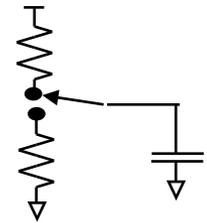
- Power is increasingly the MAJOR constraint on system performance
 - The major constraint used to be #transistors/die. Now it is power.
 - Wireless systems
 - System Performance is limited by how much data can be communicated and processed on a single battery charge
 - Want this capacity to permit a battery life of more than a day
 - Desktop systems
 - Without attention, power consumption of circa 2008-2010 CPUs would be 1kw or more → Impossible to air cool
 - Without attention, power consumptions of chips like HDTV decoders would be 100+W → Expensive to cool
 - Server farms and Supercomputers
 - Without attention, power consumption of next generation server farms would be > 20 kW → requires two power substations, not one

Power Consumption

- Static CMOS Circuits:
 - Static Power (when circuit not switching)
 - Leakage
 - Sub-threshold Drain to source
 - Gate
 - Some libraries are starting to include low-leakage cells, or cells that can be switching to a low leakage state
 - Dynamic Power (when logic transitions occur)
 - `through' current small during switching
 - Toggling power when output node changes logic state

CMOS Circuit

- Circuit during switching event
- E.g. Inverter driving a load:
 - Power dissipated in resistors in 010 cycle = potential energy stored and released on capacitor during that cycle



$$\begin{aligned}
 Q &= CV_{dd} \\
 E &= QV_{dd} = CV_{dd}^2 \\
 P &= E/T = CV_{dd}^2 / T \\
 &= N_{\text{switch}} CV_{dd}^2 f_{\text{clock}}
 \end{aligned}$$

$$\begin{aligned}
 Q &= CV_{out} \\
 \Rightarrow I &= C \frac{dV_{out}}{dt}
 \end{aligned}$$

- Alternative derivation:
 - When V_{out} $0 \rightarrow 1$, energy dissipated in top resistor:

$$E = \int_0^{V_{dd}} (V_{dd} - V_{out}) I dt = \int_0^{V_{dd}} (V_{dd} - V_{out}) C dV_{out} = \frac{CV_{dd}^2}{2}$$

Minimizing Power Consumption

- Power consumption in a CMOS module:

$$\text{Power} = \sum N_{\text{switch}} f V_{\text{cc}}^2 C_{\text{load}} + \text{leakage power}$$

- Sum over all nodes in circuit
- f = clock frequency
- N_{switch} = average % of clock periods in which node toggles (I.e. 010 or 101)
- C_{load} = capacitance of node
- Nswitch
 - Clock :
 - Maximum for glitch-free logic:
 - Logic typically has Nswitch ~ 0.1

Minimizing Energy

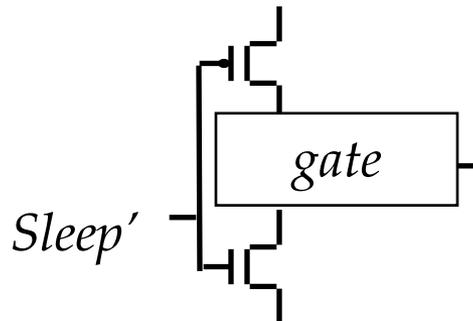
- Energy = $\int \text{Power} \cdot dt$
- Energy consumption in a CMOS module:
Energy = $\sum_{\text{cycles}} \sum_{\text{nodes}} N_{\text{switch}} f V_{\text{cc}}^2 C_{\text{load}} + \text{leakage power}$
 - f = clock frequency
 - N_{switch} = average % of clock periods in which node toggles (I.e. 010 or 101)
 - C_{load} = capacitance of node
- Note: Power reduction techniques do not save energy/complex operation if more cycles are needed to complete that operation
 - Important in energy-constrained (e.g. battery driven) systems

Approaches to minimizing power consumption

- Approaches to minimizing power consumption
 - Reduce Supply Voltage
 - E.g. Use low-Vdd cells in non critical paths
 - Reduce clock frequency
 - Does reducing the clock frequency reduce the energy required to perform a complex operation?
 - Battery-powered devices do not necessarily benefit from reducing clock frequency
 - Reduce “useless” toggling
 - Reduce clock frequency or stop clock when module is idle
 - Usually a system design, not module level design
 - Use designer knowledge to identify useless switching and redesign to reduce it
 - Use an algorithm that reduces total number of toggles required to compute a result

Approaches to minimizing power consumption

- Static Power
 - Significant issue at 65 nm transistor sizes and smaller
 - When performance is not an issue, use a low leakage cell library
 - Use low leakage cells in non-critical paths
 - Use cells with sleep transistors to reduce leakage in modules that are idle for long periods

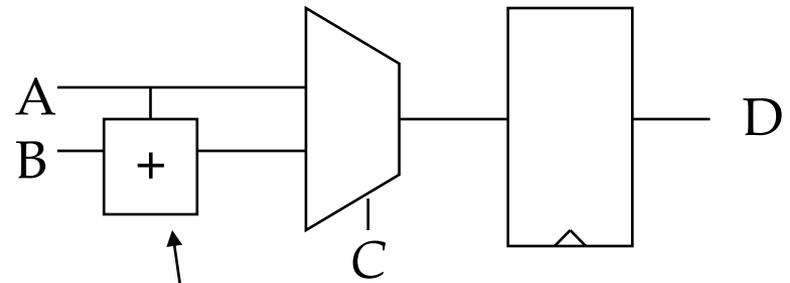


Reducing “Useless” toggling

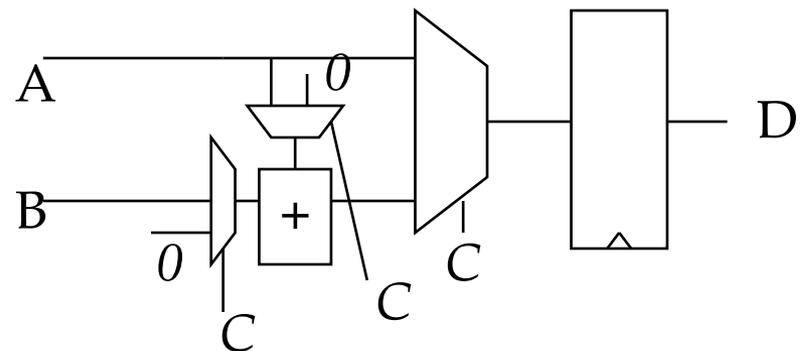
- **Example:**

```
reg [31:0] A, B, D;  
always@(posedge clock)  
begin  
    if (C) D <= A+B;  
    else D <= A;  
end
```

Possible ways to reduce power:

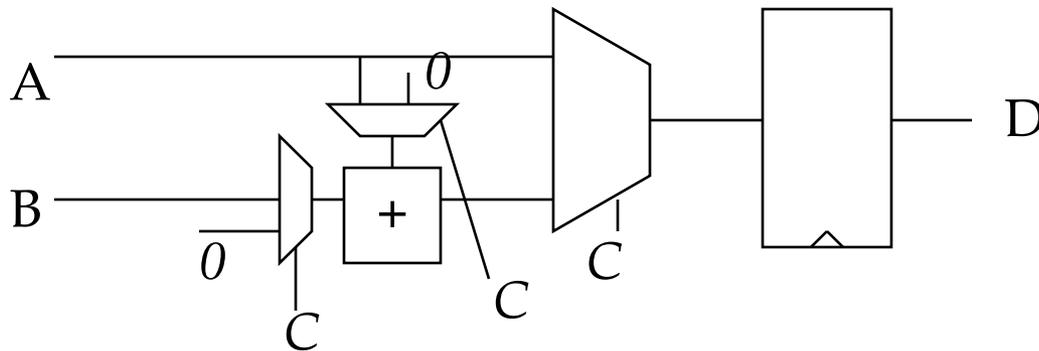


Wasted power when C low



Toggling Reduction

- If C is low a lot...



```
assign E=C?A:0;  
assign F=C?B:0;  
always @(posedge clock)  
if (C) D <= E+F;  
else D <= A;
```

Only useful if C is low more than 50% of the time.

Power Reduction

- Other Alternatives:
 - Gate the clock to register D
 - Smallest overhead
 - Complicates clock design and timing
 - Usually gated clocks only done at “block level” (.e.g an FPU)
 - Store previous value of A and B in a register
 - Used instead of 0 input to mux
 - Must consider power overhead of register (including extra Clod on clock)
 - Not likely to be beneficial here
 - Might be beneficial for a larger design (e.g. multiplier)

Summary

- What determines power consumed in a CMOS circuit?
- What strategies can you use to reduce power consumption?
- If Energy is the issue rather than power, what strategies are available to you?

Summary

- Complementary MOS transistors gives dense circuits and lower power than other circuit families

- Standard Cell designs use Static CMOS
- Transistor speed approximated using 'on resistance'
- Ron proportional to electron/hole mobility and W/L
 - Hole mobility = half electron mobility
 - Inverter $W_p = 2 W_n$ to make $t_{rise} = t_{fall}$
 - To drive larger loads, increase transistor width proportionally

- Power consumption important in many designs

$$\text{Power} = \sum N_{\text{switch}} f V_{cc}^2 C_{\text{load}}$$

- Lowering voltage by one-half, quarters the power but halves the speed
- Turn clock frequency down when performance not needed
- Reduce N_{switch} through good design