Low Power Design

Dr. Paul D. Franzon

Outline
1. Power consumption in CMOS
2. Strategies to reduce consumption

References
- Smith and Franzon, Chapter 11
- Weste and Harris, Principles of CMOS VLSI Design, A Systems Perspective
Objectives

- Identify why low power design is important.
- Understand the difference between low power design and low energy design.
- Understand static and dynamic power sources in static CMOS circuits.
- Identify techniques to reduce power consumption.
Power Consumption

Why is power consumption important?

- Battery powered devices
  - Maximize battery life
- Limit requirements of cooling
  - Temperature control of chip is important
    - Typically want chip transistor temperature to be ~90 C
      - Higher temperatures slow chips down, increases transistor leakage
      - On/off thermal cycles fatigue solder and other connections reducing reliability
  - Plastic packaging is 10x cheaper than ceramic packaging but can only dissipate 1 - 2 W
  - Fans, heat pipes etc. are need and get more expensive as heat flux increases
  - Becoming an issue even in mobile devices!
**Power Consumption**

...Why is power consumption important?

- As power requirements go up, the cost of delivering the power to the chips increases
  - Copper bus-bars
  - Power regulators/supplies
    - efficiency ~ 30% ➔ lose 30% of power even before it gets to the chip
    - Before that roughly ~30% of nations power consumption in the electrical grid is spent on overcoming distribution losses

- Green systems
  - About 1% of world's power goes to electronics, and is increasing
  - About 20% of home power goes to electronics, and is increasing
  - Greenhouse gas issues
Power Consumption

Power is increasingly the MAJOR constraint on system performance

- The major constraint used to be #transistors/die. Now it is power.

- Battery driven 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 2015 CPUs would be 1 kw or more \(\Rightarrow\) Impossible to air cool
  - Without attention, power consumptions of chips like HDTV decoders would be 100+W \(\Rightarrow\) Expensive to cool

- Server farms and Supercomputers
  - Without attention, power consumption of next generation server farms would be > 20 MW \(\Rightarrow\) requires multiple 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
    \[ Q = CV_{dd} \]
    \[ E = QV_{dd} = CV_{dd}^2 \]
    \[ P = \frac{E}{T} = CV_{dd}^2/T \]
    \[ = N_{\text{switch}} CV_{dd}^2 f_{\text{clock}} \]

• Alternative derivation:
  • When \( V_{out} \) \( \rightarrow 1 \), energy dissipated in top resistor:
    \[
    E = \int_0^{V_{dd}} (V_{dd} - V_{out})Idt = \int_0^{V_{dd}} (V_{dd} - V_{out})CdV_{out} = \frac{CV_{dd}^2}{2}
    \]
    \[ \Rightarrow I = C \frac{dV_{out}}{dt} \]
Minimizing Power Consumption

Power consumption in a CMOS module:

\[ \text{Power} = \sum N_{\text{switch}} f Vcc^2 C_{\text{load}} + \text{leakage power} \]

- Sum over all nodes in circuit
- \( f = \) clock frequency
- \( N_{\text{switch}} = \) average % of clock periods in which node toggles (I.e. 010 or 101)
- \( C_{\text{load}} = \) capacitance of node

\( N_{\text{switch}} \) trailer

- Clock :
- Maximum for glitch-free logic:
- Logic typically has \( N_{\text{switch}} \approx 0.1 \)
Minimizing Energy

Energy = \int \text{Power}.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_{\text{switch}} = average \% of clock periods in which node toggles (I.e. 010 or 101)
- C_{\text{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

- Reduce Supply Voltage
  - E.g. Use low-Vdd cells in non critical paths
  - Practical limit ~ 600 mV
    - When Vdd < 2*V_{threshold}, performance goes down quickly
- Reduce clock frequency
  - Reduce frequency, or even turn off clock, when block in “idle” mode
    - “clock gating” or regulating
  - Note, reduces power but not energy to complete a complex operation
- Reduce “useless” toggling
  - 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 at logic level

Example:

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

Possible ways to reduce power:

![Reducing Power Circuit Diagram](image.png)

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.
Other Alternatives

- 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 Cload on clock)
  - Not likely to be beneficial here
  - Might be beneficial for a larger design (e.g. multiplier)

Energy in MULT wasted if B does not select new mult output and inputs change

Overhead of FF acceptable here if B does not change a lot
System Level Power Reduction

Clock Gating
- Done ABOVE module level
- One clock per module ➔ Gated clock has to feed at least one module
- NOT done by designer inserting logic in clock tree
  - Most timing tools can not perform timing calculations with gated clocks
  - Increases clock skew if not done well
- INSTEAD use system level resources to create gated clocks
  - Done as part of clock sub-system design NOT logic design

Memory Management
- DRAM energy ~ 60 – 600 pJ/bit
  - Energy/bit very dependent on DRAM access patterns
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?