Techniques for Power Reduction
Polychronis Xekalakis
Intel Barcelona Research Center
Aula Empresa, Facultat d’Informàtica de Barcelona, February 2010
© Intel Corporation, 2010

What is Power for Physicists?
Power is the rate at which work is performed

P = \frac{W}{T}

Real-life experience go from home to UPC

“Real Life” Example
Go from Home to UPC

Home

UPC
“Real Life” Example

Go from *Home* to *UPC*

![Image of person going from Home to UPC]

\[ P = \text{KCal} \]

What is Power for Computer Scientists?

- Same as for physicists, only for computers
  - "Work" = # of performed operations
  - "Work" = Time idle (but powered-on)

\[ P = P_{\text{dyn}} + P_{\text{stat}} \]

Why Design for Low-Power?

- Most processors over-provision:
  - They only scarcely require all of the resources
- Goal: reduce power, without affecting perf.
- Low power means:
  - Increase battery life (laptop/mobile phone)
  - Better thermal behavior
  - Less money spent on electricity
  - But also: performance

Agenda

- Motivation
- Power Dissipation
- Saving Dynamic Power
- Dealing with Static Power
- Looking Ahead
- Concluding Remarks

Where does the Power Go?

- Two main components:
  - Dynamic power
  - Static power

![Diagram showing increase in power over years with labels for Dynamic and Static power]

Progress Bar

1. High
2. Medium
3. Low
**Dynamic Power**

- Dynamic power is proportional to four components:
  - Circuitry capacitance
  - Operating voltage
  - Activity factor
  - Operating frequency

\[ P_{\text{dyn.}} = C \times V^2 \times A \times f \]

**Static Power**

- Static/Leakage power depends on:
  - Operating voltage
  - Temperature
  - Threshold voltage
  - Transistor other characteristics

\[ P_{\text{stat.}} = V \times k \times e^{qV_{\text{th}}/(kT)} \]

**Agenda**

- Motivation
- Power Dissipation
- Saving Dynamic Power
- Dealing with Static Power
- Looking Ahead
- Concluding Remarks

**Techniques to Save Dynamic Power**

- Reduce any of \( C, V, A, f \)
  - Caveat: without increasing the others

\[ P_{\text{dyn.}} = C \times V^2 \times A \times f \]

**Reducing Activity**

- On every tick → add data

- When no data, clock still ticks consuming power...
**Clock Gating**
- Reduce activity significantly
- Simple technique: used in most processors!!

**Reducing Capacitance**
- Long wires \(\Rightarrow\) big capacitance
- What if signal does not have to go to everyone?

- We now only have to find which *switches* to activate!!

**Reducing Voltage and Frequency**
- Increasing operating frequency relies on increasing voltage ...
  - Need high currents to drive capacitances fast!!
- Some times there is slack in applications:
  - Detect that
  - Put core in lower freq/ voltage
  - Decrease slack in high-power mode

**Example: DVFS**

- \(A \quad B \quad \text{Slack} \quad C\)
  - HighFreq, HighVoltage
  - Time
  - \(A \quad B \quad \text{C}\)
  - LowFreq, LowVoltage
  - Time

**Agenda**
- Motivation
- Power Dissipation
- Saving Dynamic Power
- Dealing with Static Power
- Looking Ahead
- Concluding Remarks
Techniques to Save Static Power

- Reducing static power == reducing V, Vth, T
  - Caveat: without increasing dynamic power
  - In practice this never happens

Making V\textsubscript{th} Larger

- Primary reason for leakage
- But, making it larger:
  - Makes transistors slower
  - Makes capacitance (dynamic power) larger
- For some parts this could be OK (L2 cache)
- Adaptive circuit techniques:
  - Reverse body bias
  - Forward body bias
  - Gate Vss

Body Bias Techniques

- Primary reason for leakage
- But, making it larger:
  - Makes transistors slower
  - Makes capacitance (dynamic power) larger
- For some parts this could be OK (L2 cache)
- Adaptive circuit techniques:
  - Reverse body bias
  - Forward body bias
  - Gate Vss

Reducing Voltage when Idle

- Reducing operating voltage when idle is important
  - Tricky part: changing voltage takes time
- Usually proposed for caches:
  - Large component
  - Many data we will not use anymore
  - Lines not being recently used may not be used
- Technique shown to work for L1 data and instruction caches

Dealing with Temperature

- Temperature is tricky:
  - Positive feedback between power and temp
- We have to be more aggressive with leakage saving techniques when high temp
  - Even at the expense of increased dynamic power
  - i.e. decay faster
Trading Dynamic for Static

Pays-off to do aggressive leak opts

\[ P \] vs \[ \text{Temperature} \]

- Dynamic
- Static

Agenda

- Motivation
- Power Dissipation
- Saving Dynamic Power
- Dealing with Static Power
- Looking Ahead
- Concluding Remarks

Free Lunch is Over

Progress Bar

Looking Ahead

- Power consumption first-class constraint
- Low-power is the only way to go!!
- Many things to improve at many levels:
  - Circuit level (smart gating, fast dvfs)
  - Architecture (detect when over-provisioning)
  - Compilation (parallelization)

Agenda

- Motivation
- Power Dissipation
- Saving Dynamic Power
- Dealing with Static Power
- Looking Ahead
- Concluding Remarks

Conclusions

- Excessive power consumption comes from over-provisioning
- Performance is important, but power is as well
- Today we described:
  - Why we need to look at it
  - What are the different types of power consumption
  - Techniques used to mitigate them
  - Some current/future trends
- Expect to see a lot of exciting research in this area!!