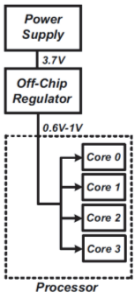
EE 156: Advanced Topics in Computer Architecture

Spring 2019
Tufts University

Instructor: Prof. Mark Hempstead
mark.hempstead@tufts.edu

Lecture 13: Dyanmic Voltage and Frequency Scaling (DVFS)

DVFS Intro – Dynamic Voltage Frequency Scaling



- Dynamic Power ~ αCV^2f
- Wouldn't it be nice to reduce the voltage
- Modern machines can do this with off chip regulators
  - Some systems have fast on-chip regulators

DVFS: How it works

- A voltage setting is transmitted from the core to the regulator. (chip is often stalled during the process)
- The regulator adjusts the voltage
- The on-chip PLL (phase-locked-loop) adjusts to the new frequency
- This is often slow (10 us)
- Newer designs in 100ns, like the plot below [Kim HPCA'08]

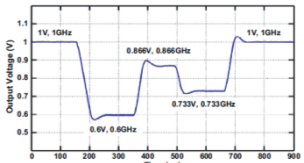
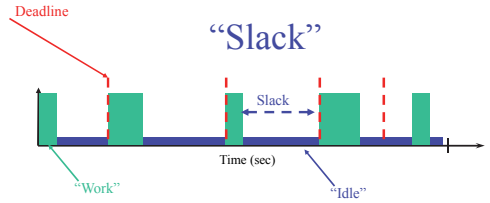


Figure 2. DVFS transition times with an on-chip regulator

Build it and they will come

- OK, we've argued that DVFS is useful
  - In fact, most CPU chips have it
- It's no good adding hardware if the software cannot use it reasonably well!
- The O/S controls DVFS via *P states* (for "Performance")
  - P0 = highest V/F
  - P1 = next highest
  - Pn = lowest V/F
- The O/S requests a given P state via the *Advanced Configuration and Power Interface*; the chip may or may not grant the request
  - The chip will not let itself exceed max power or overheat
- But how does the O/S decide what to request?

"Slack"



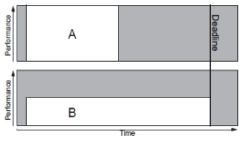
- DVFS approaches are all designed to exploit slack
- Types/sources of slack
  - Circuit timing slack (time between clock cycles that a circuit is not switching)
  - Microarchitectural slack (functional unit, or memory hierarchy)
  - Application slack (idle times between tasks)

DVFS Making Deadlines

- Problem: How to reduce voltage and frequency so that *deadlines* are not missed.
- Types of Deadlines
  - "Real-time" applications. Often found in embedded devices, statically scheduled off-line.
    - Audio encoding/decoding
    - Video
    - Control systems
    - Often single workloads or statically scheduled multi-programmed workloads
  - User perception deadlines. Flautner et al. [78]
    - Interactive tasks threshold is 50-100 ms

Using Deadlines to save energy

- With periodic user deadlines we can run at a lower voltage and frequency
- Analyzed applications to find periodic tasks and interactive tasks

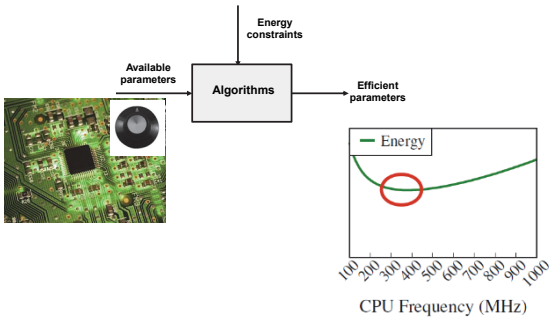


DVFS IN LINUX AND MODERN SYSTEMS

Controls on Linux

- Controlled by cpufreq subsystem
  - cpufreq module
  - CPU-specific drivers
  - In-kernel governors (policy)
- User-interface via sysfs
  - List of CPUs
  - min|max frequency
  - Available frequencies (read-only)
  - Current governor
  - Stats

Energy Management Algorithms



10

Linux Governors

- Implement simple DVFS policies
- Originally, 3 governors
  - Powersave
    - Run at lowest frequency
  - Performance
    - Run at highest frequency
  - Userspace
    - Run at user-specified frequency
- On-demand governor
  - Utilizes low-latency frequency switching
    - Intel Speedstep => 10us
  - Determines frequency based on CPU utilization

11

On-demand Algorithm

```
for every CPU in the system
  every X milliseconds
    get utilization since last check
    if (utilization > UP_THRESHOLD)
      increase frequency to MAX
  every Y milliseconds
    get utilization since last check
    if (utilization < DOWN_THRESHOLD)
      decrease frequncy by 20%
```

12

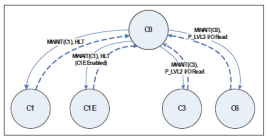
Adaptive Voltage Scaling

- No longer use static voltage tables
  - Frequencies are still fixed
  - Adaptive voltage
    - Determined using post-silicon testing & run-time temperature

13

Modern Machines

- Intel Enhanced SpeedStep (Core2 Processors) 2^7 different voltage settings from 1.5 V to 0 V with the VID Pins that can be set by the OS
- Besides voltage scaling the cores can power down. Both the Core2 and Core i7 have similar C states



While individual threads can request low power C-states, power saving actions only take place once the core C-state is resolved. Core C-states are automatically resolved by the processor. For thread and core C-states, a transition to and from C0 is required before entering any other C-state.

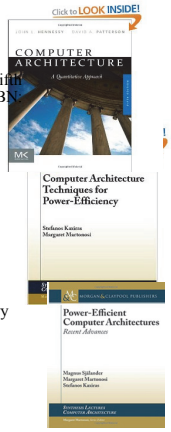
Core States (Intel)

- Core i7 States (http://download.intel.com/design/processor/datashts/322164.pdf)
  - C0: Normal Operating State
  - C1: low power state, but can still process bus snoops
  - C3: All caches are flushed, clocks are stopped for a core. Does not wake up for snoop traffic, but architected state is saved.
  - C6: Deep sleep, voltage is set to zero, all state saved to SRAM
- Core i7 Sandybridge C states:
  - http://www.hotchips.org/wp-content/uploads/hc_archives/hc23/HC23.19.9-Desktop-CPUs/HC23.19.921.SandyBridge_Power_10-Rotem-Intel.pdf (Hot'Chips 2011)

DVFS DESIGN ISSUES

Resources

- "Computer Architecture: A Quantitative Approach," Fifth Edition, John L. Hennessy and David A. Patterson, ISBN: 978-0-12-383872-8
- Two Additional References:
  - "Computer Architecture Techniques for Power-Efficiency"
    - By Stefanos Kaxiras, Margaret Martonosi, 2010
  - "Power-Efficient Computer Architectures: Recent Advances"
    - By Magnus Själander, Margaret Martonosi, Stefanos Kaxiras, Dec 2014
  - Part of the Synthesis Lectures on Computer Architecture Series. Available free from the Library http://library.tufts.edu:80/record=b2812045~S1
    - http://library.tufts.edu:80/record=b2812046~S1
- Research papers
  (will be available on the web)



DVFS Design Issues (pg 24)

- (1) At what level should the DVFS Control policies operate?
  - *System-level based on system slack*: change the whole processor or system based on the entire system load
  - *Program-level based on instruction slack*: make DVFS decisions based on programs current state. Change voltage based on program behavior. Can hide memory operations.
  - *Hardware-level*: change voltage dynamically to reflect the slack in the critical path of circuits (Razor)

DVFS Design Issues (pg 24)

(2) How will the DVFS settings be selected and orchestrated?

- Programmable registers (VIN) set by the OS or application
- When to decide on the (V, f) settings?
  - Offline: Compile time
  - Online: dynamic and reactive
- Hardware controllers making decisions *under the covers*

DVFS Design Issues (pg 24)

(3) What is the hardware granularity which the voltage and frequency can be controlled?

- Entire core at once
- Main memory / cache at the different voltage
- Multiple clock domains (MCD) on a single chip

(4) How do the implementation characteristics of DVFS approach being used affect the strategies to employ?

- How long to switch to a new (V,f)?
- Fast: Use dynamic control
- Slow: Try a static analysis technique. Because performance could be lost if settings are not correct

DVFS Design Issues (pg 24)

(5) How does the DVFS landscape change when considering parallel applications on multiple processors?

- Frequency of thread0 could make thread1 wait longer for dependant results

DVFS Cost (2014, pg 15)

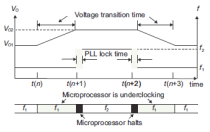
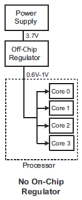


Figure 2.5: Voltage stepping vs. frequency stepping latencies [150].



- There is a delay when changing Voltage and Frequency
- Regulators must switch voltage
- Phase-lock loops take a while to transition
- On-chip regulators have been proposed to speed this up

Predicting DVFS performance Interval Model (2014 pg 17)

- Performance (execution time) does not scale linearly with frequency
- The execution of a program it split into intervals of different IPC
  - Steady state intervals: IPC limited by issue-width and program dependencies
  - Miss-intervals: introduce stalls in the processor. Start with a cache miss and last until the processor pipeline is full again

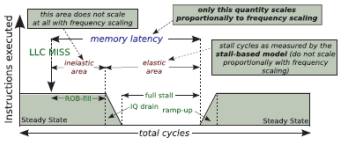
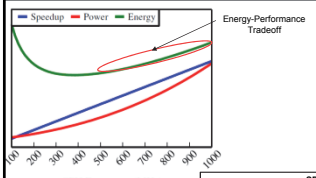


Figure 2.8: Interval model for f scaling [106].

d

DVFS IN MULTICORE AND MULTICOMPONENT

Dynamic Voltage and Frequency Scaling



Energy-Performance Tradeoff

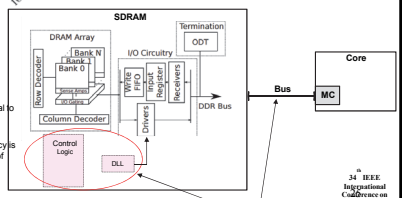
DRAM DFS

Only Frequency Scaling

Performance and power are proportional to DRAM frequency

Increase in energy with DRAM frequency is due to scalable and non-scalable parts of DRAM

SDRAM Frequency Scaling



1)David, Howard, et al. international conference on Autonomic computing. ACM, 2011.

34th IEEE International Conference on Computer Design ICCD 2016

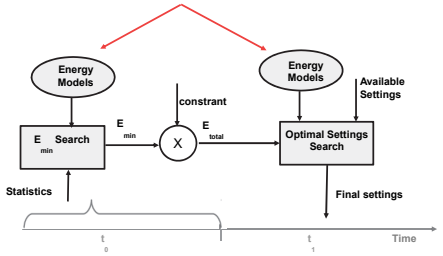
CPU DVFS and Memory DFS

- Managing Systems - a challenging task

- CPU intensive applications – higher CPU frequency
- Interplay of performance and energy of CPU and memory frequency scaling is complex
- Even more power-performance "knobs" are planned
  - GPU power scaling
  - Accelerator power scaling
  - Big + Little Cores
  - Display + I/O

26

System Design



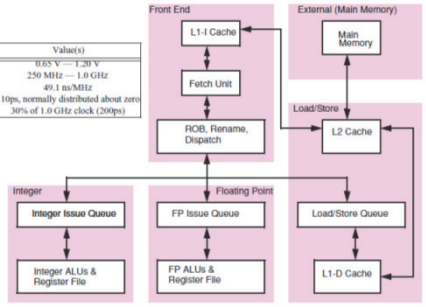
Open Areas of Research

- Power models, multicomponent models
- Predictors
- Energy constrants

27

MCD DVFS – Semeraro et al [199, 200]

Parameter	Value(s)
Domain Voltage	0.65 V — 1.20 V
Domain Frequency	250 MHz — 1.0 GHz
Frequency Change Rate	49.1 ns/MHz
Domain Clock Jitter	110ps, normally distributed about zero
Synchronization window	30% of 1.0 GHz clock (200ps)



Semeraro et al. and online DVFS [199]

- Showed that the hardware can control DVFS of each domain by watching the occupancy of *issue queues* (inputs) between stages
- Collect occupancy during intervals of time
- Two modes
  - *Attack:* significant change in occupancy between intervals therefore **significant frequency change (up or down)**
  - *Decay:* small change between intervals, **small frequency reduction**
- 19% reduction in energy/instruction, 16.7% EDP improvement, 3.2% increase in CPI

Multi-threaded DVFS (2014, pg 25-29)

- Ongoing research challenge, how to schedule DVFS for multithreaded workloads
  - One thread could depend on data from another thread
  - Should use "critical path" analysis to avoid performance issues
- Scheduling algorithms (p26)
  - Some approaches set real-time deadlines and create a static schedule
  - Other dynamic approaches use synchronization points (e.g. barriers) to determine the DVFS schedule