ECE 571 – Advanced Microprocessor-Based Design Lecture 19

Vince Weaver

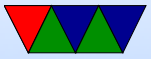
http://www.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

2 April 2013

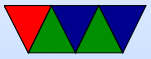
Project/HW Reminder

- Homework #4 was posted, due on Friday.
- Hardware issues with x86. Procure machine?

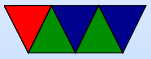


Metrics Clarification

- Energy delay = E*t (J*s), Energy delay squared = E*t*t (J*s*s), Smaller is better
- In related papers it is confusing, as no one shows formula (despite academic papers loving formulas). Often they use the inverse (so larger is better?) which also confuses things
- Other papers use MIPS/Watt = insn/J
  Not cross-platform. You really want to optimize for time,

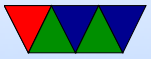


not for instructions which can vary for lots of reasons and doesn't exactly equate to time.



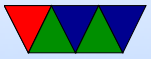
DVFS and other CPU Power/Energy Saving Methods

- A lot of related work
- Will focus on actual implementations rather than academic papers this time



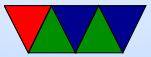
CMOS Energy Equation, Again

- $E_{tot} = [(C_{tot} V_{dd}^2 \alpha f) + (N_{tot} I_{leakage} V_{dd})]t$



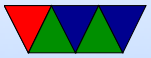
Delay, Again

- $T_d = \frac{C_L V_{dd}}{\mu C_{ox}(\frac{W}{L})(V_{dd}-V_t)}$
- Simplifies to $f_{MAX} \sim \frac{(V_{dd}-V_t)^2}{V_{dd}}$
- If you lower f, you can lower V_{dd}

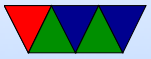


DVFS

- Voltage planes – on CMP might share voltage planes so have to scale multiple processors at a time
- DC to DC converter, programmable.
- Phase-Locked Loops. Orders of ms to change. Multiplier of some crystal frequency.
- Senger et al ISCAS 2006 lists some alternatives. Two phase locked loops? High frequency loop and have programmable divider?

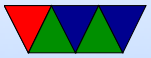


- Often takes time, on order of milliseconds, to switch frequency. Switching voltage can be done with less hassle.



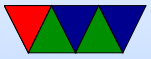
When can we scale CPU down?

- System idle
- System memory or I/O bound
- Poor multi-threaded code (spinning in spin locks)
- Thermal emergency
- User preference (want fans to run less)

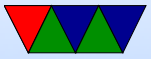


Adaptive Body Biasing

- Related to but not always considered part of DVFS
- Control voltage applied to body
- Change the threshold voltage
- Reduces leakage but slows performance

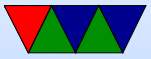


A History of Power Management on x86



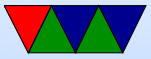
Halt Instruction

- Oldest power-saving interface on x86
- Tells CPU to stay idle until an interrupt comes in
- 486-DX4 and later enters low-power mode
- Ring 0. The OS does this when idle
- Similar instruction available on 65c816
- ARM has `wfi` in ARMv7 and maybe `hlt` in ARMv8?



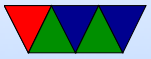
APM – Advanced Power Management

- For laptops
- Developed by Intel and Microsoft, 1992
- Made obsolete by ACPI
- Full On / APM Enabled / Standby / Suspend or Hibernate / Off
- Calls to BIOS. BIOS often buggy.

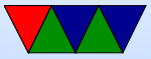


ACPI – Advanced Configuration and Power Interface

- http://www.acpi.info/presentations/ACPI_Overview.pdf
- Developed by Intel, Microsoft and Toshiba, 1996 Later HP and Phoenix
- Full ACPI interpreter needed.
- APM was a black box to Operating System. ACPI works with OS

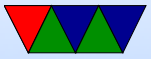


- ACPI code in theory provided by Intel or similar, no need for each manufacturer to implement (like APM)
- OS-directed power management
- Hardware registers for interface
- BIOS provides tables, motherboard initialization

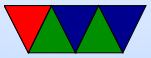


ACPI Sleep States

- G0/S0 – working
- G1 Sleeping
  - S1 – caches flushed, CPU stopped, CPU and RAM power maintained
  - S2 – CPU powered OFF
  - S3 – Standby, Sleep, Suspect to RAM. RAM still powered
  - S4 – Hibernate/Suspend to Disk – all memory stored to disk

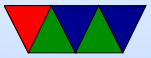


- G2 (S5) – “Soft Off” – power off, but power still supplied to power switch and wake on lan, etc
- G3 – “Mechanical Off” – all power removed



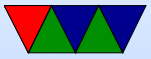
ACPI C-States

- C0 – operating
- C1 – Halt – processor not executing, but can start nearly instantaneously
- C2 – Stop-Clock – all state is stored, but might take some time to get going again
- C3 – Sleep – Processor does not keep cache coherent, but otherwise holds state

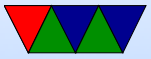


ACPI P-States

- actual values can sometimes be configured via MSR access.
- Some V/F combinations unstable/unsafe so BIOS only exports known good combinations
- P0 – max power and frequency
- P1 – less than P0, DVFS
- P2 – less than P1, DVFS

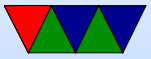


- Pn – less than P(n-1), DVFS



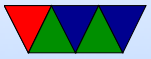
ACPI T-States

- throttling
- Linear reduction in power, linear reduction in performance
- Does not save Energy! (halve the frequency, double the time)
- Mostly used for passive cooling



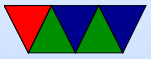
ACPI D-States

for devices such as modems, Cd-ROM, disk drive

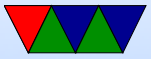


CPU Scaling

- Intel SpeedStep
- Enhanced speed step. Change V and F at different points. Slower to change frequency if V not changed first. Bus clock keeps running even as PLL shut down 10ms transition
- AMD PowerNow! (laptop) Cool'n'Quiet (desktop)
- VIA PowerSaver/LongHaul – Fine grained DVFS

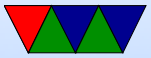


- p4-clockmod – mainly for thermal management, skip clocks, hurt performance without saving energy (throttling)
- IBM EnergyScale
- Transmeta LongRun – leakage varies due to process variation Longrun2 monitors performance/leakage and varies Vdd and Vt



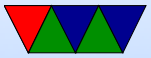
Governors

- ondemand – dynamically increase frequency if at 95% of CPU load
  introduced in 2.6.9
- performance – run CPU at max frequency
- conservative – increase frequency if at 75% of load
- powersave – run CPU at minimum frequency
- userspace – let the user (or tool) decide



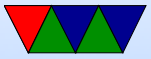
Governors – cont

- Various tunables under /sys/devices/system/cpu
- Can trigger based on ACPI events (power plug in, lid close)
- Laptop tools
- `cpufreq-info` and `cpufreq-set`
  Need to be root



User Governors

- typically can only update once per second
- ondemand people claim it reacts poorly to bursty behavior
- Powernowd – scale based on user and sys time
- cpufreqd
- Obsolete with introduction of “ondemand” governor?



Sources of Info for Governors

- System load
- performance counters
- input from user?