Inductive Noise: Sources, Problems, & Solutions

Raj Parihar
Aaron Carpenter

Presented for ACAL
Who cares?

- Power distribution system is essentially resistive and inductive in nature
  - For proof, see any paper or book by Prof. Friedman

- At high frequency
  - $\text{IR} \ll \text{L} \cdot \frac{\text{d}I}{\text{d}t}$
  - High freq, high load, high inductance $\Rightarrow$ more timing/voltage errors
Outline

- What is inductive (or di/dt) noise?
- How does it affect modern processors?
- Can we fix it at circuit/package level?
- Can we reduce it?
- Can we correct/recover from it?
What is inductive noise?

- Voltage drop because of inductance ($Z_L = j\omega L$)
  - As frequency increases, so does the inductive component
- $V = L \frac{di}{dt}$
  - On-chip inductance is unavoidable
  - $di/dt$ noise increases with switching speed, current load
- Voltage power delivery system has noise margins (typically 5-10\% of nominal $V_{dd}$)

---

E. Grochowski, D. Ayers, and V. Tiwari, “Microarchitectural $di/dt$ Control,” DATC 03.
D. Ernst et al, “Razor: Circuit-Level Correction of Timing Errors for Low-Power Operation,” MICRO03
How does it affect modern processors?

- **Mid-Frequency di/dt noise**
  - Power supply resonance 50-200MHz

- **High-Frequency di/dt noise**
  - Single cycle, large current swing
  - Can happen at any time
  - Can’t eliminate a resonance
Can we fix it at circuit/package level?

- Decoupling Capacitors
  - Offset the inductive load
  - Keep area, cost, energy low
  - Place decaps equally OR
  - Determine current draw during design, place decaps where load determines

Floorplanning Fixes

- Floorplanning
  - Self & correlated weighting on modules
  - Iteratively decide where to put them to reduce load on Vdd power pins Reduce need for decaps

Architectural Reduction

- Gradual wake-up, sleep signals
  - More time, less current change
  - Decrease performance

- Pipeline Damping or Muffling
  - Stop pipeline from issuing to stop high current draw
  - Insert dummy instructions to keep resources busy to stop big low swings

- Noise controller
  - Decay counter – only turn off after 16 cycles of idleness
  - Queue-based – priority for certain modules
  - Pre-emptive gating – make sure not to turn off and get turned back on

Reduced di/dt - Costs

- **Goal of body of work:**
  - Keep current swings low, reducing di per cycle
- **All have tradeoffs in performance and/or power and energy**
  - Decreasing the pipeline throughput, throttling performance
  - Inserting instructions to raise energy
  - The end goal is to keep consistency
- **Tried to reduce the physical noise**
  - Instead, reduce *the effects* of noise on the architecture
Next Level: Architectural Tricks

- Architectural techniques
  - Mainly targets low- or mid- frequency di/dt noise
  - More efficient solutions
    - Compare to “pure” circuit based techniques

- Change the way we look at the noise
  - Treat as voltage (noise margin) or timing violation
  - Avoid the errors from happening
  - Or accept the errors and recover from them
RAZOR Flip Flops

- Timing critical Flip Flops are augmented with shadow latch

- Shadow (backup) latch
  - Conservative timing
  - Verifies the results

- If timing violation detected
  - Store results from shadow latch

- In RAZOR
  - Only 3% FFs require backup
  - Failure is not an option!

- **DIVA** vs RAZOR
  - Full fledge as oppose to selective
Sensor based Throttling

- "Voltage emergency"
  - Coarse grain phenomenon
    - 10s’ of clock cycles
  - Due to sudden rush of activities
    - Branch misprediction
    - L2 cache miss, Pipeline flushing

- Sensor based control
  - Sensor: Detect the droop/surge
  - Actuator: Control clock gating, functional units, data L1 cache

- Downsides
  - Inherent sensor delay (1-2 clks)
  - Sensor error – false alarm

- ~20% performance/energy ↑
Software based Approach

- Identification of loops/code sequences which cause the voltage “emergencies”

- Findings/Insights
  - 2-5 loops are responsible for ~ 75% of the total voltage emergencies per application

- Software based solutions
  - L2 miss: Better (balanced) prefetching
  - Long latency INS: Better code scheduling
  - Branch Misprediction: Perfect prediction
  - Loop unrolling: Performance vs inductive noise

- All above optimizations together can
  - Reduce 10-60% of the emergencies

Reasons of voltage emergencies

Consider di/dt noise in all above optimizations!
Voltage Emergency Prediction

- "Predictor" replaces the threshold sensor

- Voltage "emergencies" are
  - Consequence of control flow and micro-architectural events
  - Easy to accurately predict (~90% of the time)

- Signature based throttling
  - Predictor "learns" the signatures
  - Eventually predicts the emergency in advance
  - 13.5% higher performance compared to sensor-based throttling

- Fail-Safe mechanism
  - Checkpoint based recovery

- Sufficient lead time to activate the control

3/7/2012

DeCoR: **Delayed Commit & Rollback**

- Doesn’t require fast sensors or actuation mechanisms
- Machine architecture divided into
  - Rollback (RB) protected zone
    - Performance enhancing parts
    - i.e. ROB, issue logic, LSQ etc
  - Timing margin (TM) protected zone
    - Employs improved circuit techniques
    - i.e. Retirement Register File, L1
- Delayed Commit
  - Verify the noise speculative state
- Rollback
  - If sensor detects the emergency, flush all the speculative states
- **Sensor delay doesn’t penalize!**

---

Tribeca: PVT Variations

- A fine-grain distributed local recovery (LR) scheme
  - Per-unit voltage settings
  - Error detection unit (EDU)
    - Transition of each stage
    - Replay using buffers

- Comparison: Max clk speed possible
  - Worst case design: upto ~75% of $F_{\text{MAX}}$
  - Tribeca design: upto ~ 91% of $F_{\text{MAX}}$

- Area overhead
  - IBM POWER6: Global recovery unit
    - 15% of baseline design; without RU
  - Area overhead: 1% of POWER6

Some Other Advancements

• **RAZOR II**
  - Detection happens in shadow flip flop
  - For correction a global recovery unit is used

• **Bullet proof pipeline**
  - BIST for whole pipeline
  - Results are validated after each stage

• **Event-Guided approach voltage noise in processors**
  - Monitor “hot loops” i.e. loops with L2 misses and pipeline flushing

• **IBM POWER6 reliability**
  - Ships with a global recovery unit
References


References (cont…)

• D. Ernst et al, “Razor: Circuit-Level Correction of Timing Errors for Low-Power Operation”, IEEE MICRO 2004

• Joseph, R. et al, “Control Techniques to Eliminate Voltage Emergencies in High Performance Processors”, HPCA-03

• Reddi, V. et al, “Voltage Emergency Prediction: Using Signatures to Reduce Operating Margins”, HPCA-09


• Gupta, M. et al, “DeCoR: A Delayed Commit and Rollback Mechanism for Handling Inductive Noise in Processors”, HPCA-08


References (not covered)

- Process Variation Tolerance/Timing Tolerance
Questions?
Backup Slides
Where is it a problem?

- **Mid-frequency**
  - Current load transitions operate @ power supply resonance (50-200MHz)

- **High-frequency**
  - Large current swing in a single clock cycle
  - Exacerbated by clock gating

M. Popovich and E. Friedman, “Decoupling capacitors for multi-voltage power distribution systems,” TVLSI Vol 14, No 3, March 2006
Tribeca: Tackle PVT Variations

- PVT variations
  - Vary from part to part of chips
  - Adding them together leads to excessive conservative design
  - Differ significantly in temporal and spatial scales

- Fine grain mechanism to control various part of microarchitecture
  - Global recovery mechanism maybe wasteful