Leakage Mitigation Techniques in Smartphone SoCs

¹John Redmond

¹Broadcom
Smartphone Use Cases

Camera
Music player
Navigation
Gaming System
TV
Laptop
Access Point
Wallet

Device Convergence
Diverse Use Cases

Generalized Use Cases

Idle:
- Paging for call, text, & email

Low-Medium MIPS:
- MP3, voice call
- Video playback

High MIPS:
- Gaming, concurrent use-cases, benchmarks
Smartphone SoC

SoC Highlights

- Application Processor
- Modem
- Graphics
- Image Signal Processor
- Video
- Audio
- LPDDRx
Smartphone Use Cases

1. Idle
2. Low-Medium MIPS
3. High MIPS
Idle Mode

- Critical for standby time.
- Majority of the phone’s life is in idle mode.
- Leakage dominates overall power.
- Two phases:
  1. Deep-sleep leakage.
  2. Wake up to check for call.

- Key Challenge: *minimize leakage at all costs.*
- Key Focus Area: *architecture.*
Idle Leakage from 40nm->28nm

- 40nm – Gate off high-performance, high-leakage block.
- 28nm – Gate off everything that does not need to be on.

Note: FastLogic was power gated in 65nm->40nm transition
Switched Domain

Power Island Terminology (1/2)

Switched Domain

Power Switch

VDD

Combinatorial logic

Retention Flip Flop (RFF)

VDD_Switched

SRAM

1

ret

Bit cell array
OFF to ON domain crossing needs isolation for electrical and functional correctness
# Power Intent Methodology

<table>
<thead>
<tr>
<th>Flow</th>
<th>Power Management Cells Insertion</th>
<th>Power Architecture Change</th>
<th>Power/Ground Connections</th>
<th>Power Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legacy Flow</td>
<td>Technology dependent cells instantiated in RTL</td>
<td>RTL Change</td>
<td>Front-end specifies in spreadsheet or connected in RTL and communicated via e-mail.</td>
<td>Final PG Netlist</td>
</tr>
<tr>
<td>Power Intent Flow</td>
<td>UPF file specifies Power Intent and cells are inferred in flow</td>
<td>UPF Change</td>
<td>UPF</td>
<td>Starting at RTL through PG Netlist</td>
</tr>
</tbody>
</table>

| Benefit               | RTL is power architecture and technology | UPF easier to change | Lingua franca to ensure clean handoffs | Early and often verification |
ISO_DEVICE_MISSING: The crossover needs isolation, but has no isolation device present.
Power Intent Implementation

- UPF
- RTL

Power-aware libraries

Synthesis
- Insert isolation cells and level-shifters.
- Insert RFF.

Physical Design
- Insert power-switches.
- Route always-on signals.

Implementation Verification
- Structural LP checks.
- Custom power switch and RFF rules
- PG gate-level simulations.
Example Power Island Cost/Benefit

- Block has partial state retention with 10% of flops being retained.
- Cost is an additional 3.2% area for power switches and retention flip flops.
- Retention state leakage went from 5.1 mW to 45 uW achieving >100x leakage reduction.

Retention State Leakage Breakdown:

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Leakage (uW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributed power switch</td>
<td>2.1</td>
</tr>
<tr>
<td>Always-on buffers</td>
<td>2.6</td>
</tr>
<tr>
<td>ISO HIGH cells</td>
<td>0.04</td>
</tr>
<tr>
<td>RFF</td>
<td>4.4</td>
</tr>
<tr>
<td>Memory bit cells</td>
<td>36.6</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>45.74</strong></td>
</tr>
</tbody>
</table>
Other Considerations

• Decap leakage
  • Decap in unswitched routing channels.

• Minimize SoC memory that needs to be retained.
  • >50% of leakage budget.

• Ultra-low leakage considerations:
  • NWELL leakage.
  • RFF minimization – retain minimum set of state.
Idle Mode Leakage Mitigation

- Aggressive power islands.
  - UPF is required for implementation and verification of sophisticated power-island architecture.
- Reduce voltage for logic remaining on.
  - This is limited by the retention voltage of the memory bit-cells.
- Leakage is the top priority in implementation.
  - The trade-offs are in area, performance, and dynamic power for leakage.
- Always-on logic should be low frequency.
Smartphone Use Cases

1. Idle
2. Low-Medium MIPS
3. High MIPS
Low-Medium MIPS Leakage Mitigation

- Active power is a combination of dynamic and leakage power.
- Active power is optimized to extend usage time.
- User expectations:

<table>
<thead>
<tr>
<th>Task</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voice Call</td>
<td>&gt;10 hours</td>
</tr>
<tr>
<td>MP3 Playback</td>
<td>&gt;50 hours</td>
</tr>
<tr>
<td>Video Playback</td>
<td>&gt;10 hours</td>
</tr>
</tbody>
</table>

- Key Challenge: *minimizing active power*
- Key Area: *design and implementation*
Multi-Vth and Gate Length Biasing

- Vth is a big knob trading off performance and leakage.
- Gate length is a small knob trading performance and leakage.
- Graphs are at the standard cell level.
Leakage Optimized Multi-Vt Flow

- Initial design is 100% High Vt cells.
- Low Vt cells are swapped in on critical paths to meet timing.
- This may not be the best for area and dynamic power.
- Histograms of endpoint slack (red paths are failing) as shown below:

<table>
<thead>
<tr>
<th>Vth Swap</th>
<th>Slack</th>
<th># of paths</th>
</tr>
</thead>
<tbody>
<tr>
<td>100% High Vt Cells</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mixed High and Low Vt Cells</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
AVS and DVFS

• Adaptive Voltage Scaling (AVS)
  – Adjusting voltage based the physical environment.
  – Process variation, temperature, silicon aging, and PMU inaccuracy.

• Dynamic Voltage and Frequency Scaling (DVFS)
  – Adjusting voltage and frequency based on workload.
Active Power Across Process and Temp

- Dynamic power decreases with faster silicon due to AVS.
- Leakage power increases with faster silicon.
- Leakage power increases significantly with fast silicon and hot temperature.
Optimizing Active Power (1/3)

- **10T design:**
  - Larger, faster cells.
  - Larger percentage of higher Vth cells.

- **8T design:**
  - Smaller, slower cells.
  - Larger percentage of lower Vth to meet performance.

*Which design is better?*
Optimizing Active Power (2/3)

- The bulk of the parts are typical silicon.
- Operating temperature is typically around 55°C for low-medium MIPS use-cases.
- Optimize for typical process and temperature or worst case process and temperature?
- **Optimize for typical** while ensuring that the worst case works.

<table>
<thead>
<tr>
<th>Design</th>
<th>10T</th>
<th>8T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Typical/55C</td>
<td>630</td>
<td>600</td>
</tr>
<tr>
<td>Fast/85C</td>
<td>710</td>
<td>810</td>
</tr>
</tbody>
</table>
Optimizing Active Power (3/3)

• 10T design:
  • Smaller leakage.
  • Higher active power at typical conditions.

• 8T design:
  • Larger leakage.
  • Lower active power at typical conditions.

Higher leakage can lead to overall lower active power.
Other Design Techniques

• Guardband reduction.
• Using low-power cells and macros in implementation.
• Retain until access memories:
  • Put memories in low-leakage retention state when not in use.
• Adaptive read/write assist techniques:
  • Boost voltage on memories when writing.
• Power down logic and memories when not in use.
Smartphone Use Cases

1. Idle
2. Low-Medium MIPS
3. High MIPS
High MIPS Leakage Minimization

• High-performance cores use Low Vth devices extensively and leakage is generally >40% of the power budget in high leakage corner.
• Due to high frequencies, dynamic power is also high.
• This can lead to leakage-induced thermal runaway.
  • Higher leakage current causes a temperature increase that further increases leakage.

• Key Challenge: *thermal considerations*
• Key Area: *runtime*
RFTS vs. DVFS (1/3)

- Run Fast Then Stop (RFTS) is a technique where the processor runs at the highest frequency until the job is finished, then it stops.
- DVFS runs “low and slow” to reduce dynamic power by $V^2$.
- RFTS:
  1. Clock Gating – core continues to leak.
  2. Power Gating – core is powered off and doesn’t leak.
RFTS vs. DVFS (2/3)

What is best?

1. DVFS

2. RFTS:  
   • clock gate

3. RFTS:  
   • power gate
Run Fast Then Stop vs. DVFS (3/3)

- “Break-even time” is defined as the time that the core needs to be powered off to compensate for save and restore energy.
- In high leakage situations, the power gating benefit is realized in a shorter time.

Lowest power technique is a run-time decision.
Other Considerations

• Co-adjacent heating:
  • Leakage increases in victim blocks due to aggressor block dynamic power.

• Thermal/leakage aware floorplan.
• Core hopping to put loads on cool cores.
• Leakage-aware power and thermal management algorithms.
Conclusions

- Leakage mitigation in idle mode:
  - Architect for leakage mitigation by aggressive power gating.
  - Leakage is the top design priority.
  - Use UPF to realize sophisticated power architectures.

- Leakage mitigation in low-mid MIPS modes:
  - Design for active power minimization.
  - Power down logic and memory when not in use.

- Leakage mitigation in high MIPS modes:
  - Runtime power and thermal management decisions that consider leakage.
Thank You!

Questions?