EE241B : Advanced Digital Circuits
Lecture 22 – Reducing Leakage
Borivoje Nikolić

Announcements
* Assignment 4 due next Friday.
* Reading
  * Rabaey, LPDE, Chapter 8

Sweetfarm.org/goat-2-meeting:
invite a goat or a llama to a zoom meeting
https://www.sweetfarm.org/goat-2-meeting

Outline
* Module 5
  * Clock gating
  * Leakage reduction during design time and runtime

5.5 Reducing Switching Activity Through Logic Design

Power/Energy Optimization Space

<table>
<thead>
<tr>
<th></th>
<th>Constant Throughput/Latency</th>
<th>Variable Throughput/Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy</td>
<td>Design Time</td>
<td>Sleep Mode</td>
</tr>
<tr>
<td>Active</td>
<td></td>
<td>Clock gating</td>
</tr>
<tr>
<td>Leakage</td>
<td>Stack effects</td>
<td>Sleep T’s</td>
</tr>
<tr>
<td></td>
<td>Trans sizing</td>
<td>Multi-Vth</td>
</tr>
<tr>
<td></td>
<td>Scaling Vth</td>
<td>+ Input control</td>
</tr>
<tr>
<td></td>
<td>+ Multi-Vth</td>
<td></td>
</tr>
</tbody>
</table>

Basic Idea
* $E \sim qCV^2$
* Reduce switching activity, $q$, through logic and architectural transformations
* Many options
  * Switching activity lower with deeper logic
  * Pipelining has significant effect
  * Reduce the number of clocked devices in a flip-flop
    * e.g. group generation of clk_b
  * A few logic ideas follow

Circuit-Level Activity Encoding

Conditional Inversion Coding for Interconnect

Number Representation
* Input signals are noise most of the time

Two’s Complement

Sign Magnitude

- Sign-extension activity significantly reduced using sign-magnitude representation
5.H Clock Gating

Clock Gating

Requires a bit more complex gate... Well handled in today's EDA tools

Clock Gating Efficiently Reduces Power

Without clock gating

30.6mW

With clock gating

8.5mW

90% of F/F’s were clock-gated.

70% power reduction by clock-gating alone.

MPEG4 decoder

Courtesy M. Ohashi, Matsushita, ISSCC 2002

Local Clock Gating

Clock Gating

• Enabling clock needs to be synchronized

ARM Cortex-A9 Technical Reference Manual:

Dynamic high level clock gating activity:

When dynamic high level clock gating is enabled, the clock of the integer core is as in the following cases:

• no integer core is empty and there is no instruction ready causing a load/R
• the integer core is empty and there is no instruction ready causing a load/R
• the integer core is full and there is a data port emptying a load/R
• the integer core is full and data stores are stalled because the bus buffers are full.

When dynamic clock gating is enabled, the clock of the system control block is as in the following cases:

• there are no system control expressions (exceptions) being unmasked
• there are no system control expressions present in the pipeline
• performance modes are not enabled
• debug is not enabled.

When dynamic clock gating is enabled, the clock of the data engine is as when there is an data engine instruction in the data engine and no data engine instruction in the pipeline.
**Power / Energy Optimization Space**

<table>
<thead>
<tr>
<th></th>
<th>Constant Throughput/Latency</th>
<th>Variable Throughput/Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Energy</strong></td>
<td>Design Time</td>
<td>Sleep Mode</td>
</tr>
<tr>
<td><strong>Active</strong></td>
<td>Logic design</td>
<td>Clock gating</td>
</tr>
<tr>
<td></td>
<td>Scaled $V_{DD}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Trans. sizing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Multi-$V_{DD}$</td>
<td></td>
</tr>
<tr>
<td><strong>Leakage</strong></td>
<td>Stack effects</td>
<td>Sleep T’s</td>
</tr>
<tr>
<td></td>
<td>Trans sizing</td>
<td>Multi-$V_{TH}$</td>
</tr>
<tr>
<td></td>
<td>Scaling $V_{DD}$</td>
<td>Variable $V_{TH}$</td>
</tr>
<tr>
<td></td>
<td>+ Multi-$V_{TH}$</td>
<td>+ Input control</td>
</tr>
</tbody>
</table>

**Technology Options**

- Multiple thresholds, each spaced 50-100mV apart (5-10x less leakage)

**Typical Technologies**

- 2-3 Thresholds
  - To choose from 4-6 in a node
  - In bulk and finfet, but not in FDSOI (unless doped)
- Threshold voltage diff ~5-10x in leakage

**Using Multiple Thresholds**

- Cell-by-cell $V_T$ assignment (not block level)
- Allows us to minimize leakage
- Achieves all-low-$V_T$ performance

---

**Plan For the Rest of the Semester**

- 4 more lectures:
  - Finish low power (2 lectures)
  - Supplies, clocks and their interaction
- Homework 4 due on April 24th
  - Quiz 4 on April 28th
- Final on April 30th
  - 80 minutes, open everything
- Final presentations, May 4
- Final reports due on May 4
5.1 Lowering Leakage During Design: Longer Channels

---

**Power/Energy Optimization Space**

<table>
<thead>
<tr>
<th>Constant Throughput/Latency</th>
<th>Variable Throughput/Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy</td>
<td>Design Time</td>
</tr>
<tr>
<td>Active</td>
<td>Logic design</td>
</tr>
<tr>
<td></td>
<td>Scaled ( V_{DD} )</td>
</tr>
<tr>
<td></td>
<td>Trans. sizing</td>
</tr>
<tr>
<td></td>
<td>Multi-( V_{DD} )</td>
</tr>
<tr>
<td>Leakage</td>
<td>Stack effects</td>
</tr>
<tr>
<td></td>
<td>Trans sizing</td>
</tr>
<tr>
<td></td>
<td>Scaling ( V_{DD} )</td>
</tr>
<tr>
<td></td>
<td>+ Multi-( V_{th} )</td>
</tr>
</tbody>
</table>

---

**Longer Channels**

- 10% longer gates reduce leakage by 35% (in 130nm)
- Increases switching energy by 21% with \( W/L = \text{const.} \)
- Attractive when don’t have to increase \( W \) (memory)
- Doubling \( L \) reduces leakage by 3x (in 0.13um)
- Much stronger effect in 28nm!
- Effect improves with shorter channel devices

---

**Poly Bias**

- 28FDSOI example

---

**Stack Effect**

Reduction (in 0.13\( \mu \)):

- 1.9X: 56.7%
- 1.5X: 36.6%
- 1.3X: 26.5%
- 1.2X: 16.4%
- 1.0X: 6.7%

---

**Narendra, ISLPED’01**
Stack Forcing

Tradeoffs:
• W/2 = 1/3 of drive current, same loading
• 1.5W = 3x loading, same drive current

Next Lecture
• Low-power design
• Power gating
• Dynamic thresholds
• Optimal supplies and thresholds