EECS251B : Advanced Digital Circuits and Systems

Lecture 21 – Power-saving techniques: DVFS and Clock Gating

Borivoje Nikolić, Vladimir Stojanović, Sophia Shao
Announcements

• Assignment 3 posted, due Wed 4/13
Dynamic Voltage-Frequency Scaling (DVFS)
## Power/Energy Optimization Space

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

Exploit Data Dependent Computation Times To Vary the Supply

from [Nielsen94]

(IEEE Transactions on VLSI Systems)
Processors for Portable Devices

- Eliminate performance ↔ energy trade-off

Dynamic Voltage Scaling

Processor Energy (Watt*sec)

Performance (MIPS)

PDAs

Pocket-PCs

Notebook Computers

Burd
ISSCC'00
Typical MPEG IDCT Histogram
**Processor Usage Model**

- **Preferred Throughput**
- **Maximum Processor Speed**

**System Optimizations:**
- Maximize Peak Throughput
- Minimize Average Energy/operation

Burd
ISSCC'00
Common Design Approaches (Fixed VDD)

**Compute ASAP:**
- Delivered Throughput
- Clock Frequency Reduction:
  - Excess throughput
  - Reduced $f_{CLK}$
  - Energy/operation remains unchanged ...
  - while throughput scaled down with $f_{CLK}$
Scale $V_{DD}$ with Clock Frequency

Constant supply voltage

~10x Energy Reduction

Reduce $V_{DD}$, slow circuits down.

Burd
ISSCC'00

Energy/operation

Throughput ($\propto f_{CLK}$)
CMOS Circuits Track Over $V_{DD}$

Normalized max. $f_{CLK}$

Delay tracks within +/- 10%
Dynamic Voltage Scaling (DVS)

• Dynamically scale energy/operation with throughput.
• Always minimize speed → minimize average energy/operation.
• Extend battery life up to 10x with the exact same hardware!

Burd
ISSCC’00
Operating System Sets Processor Speed

- DVS requires a voltage scheduler (VS).
- VS predicts workload to estimate CPU cycles.
- Applications supply completion deadlines.

\[
\frac{\text{CPU cycles}}{\Delta \text{time}} = F_{\text{DESIRE}}
\]

Processor Speed (MPEG)
Converter Loop Sets $V_{DD}$, $f_{CLK}$

• Feedback loop sets $V_{DD}$ so that $F_{ERR} \rightarrow 0$.
• Ring oscillator delay-matched to CPU critical paths.
• Custom loop implementation $\rightarrow$ Can optimize $C_{DD}$.
Power estimator example for HW-based DVFS control

- Receives ongoing counts from each CPU of “significant” events
  - e.g., how many vector instructions executed per cycle, aggregates these to a per-CPU instantaneous power estimate

- Sends per-CPU throttle requests as necessary.

- The number of “power events” over some period of time is tracked
  - If this is too high the fact is reported to the PMU and PMGR, and the frequency/voltage settings for the offending core(s) are reduced; and similarly if a core has been operating within spec for some period of time, its frequency/voltage is allowed to rise.

### Table: DVFM State

<table>
<thead>
<tr>
<th>DVFM State</th>
<th>Volt.</th>
<th>Freq.</th>
<th>Limit3</th>
<th>Limit4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>V0</td>
<td>F0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>V1</td>
<td>F1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>V2</td>
<td>F2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>V3</td>
<td>F3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3rdCoreMax</td>
<td>V3rd</td>
<td>F3rd</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4thCoreMax</td>
<td>V4th</td>
<td>F4th</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

### Diagram:

- **TgtSel**
- **44A**
- **46A**
- **CurCfg**
- **44B**
- **44C**
- **44D**
- **3Ac**
- **4Ac**
- **48A**
- **48B**
DVFS scheduling

• First stage – establish performance goals (achieve state X by time Y)
  • Use closed-loop performance controller to adjust DVFS to meet these performance goals.
  • E.g. performance described as a monotonic map from n-cores running at max frequency down to 1 core running at minimum frequency.
  • Move up/down this performance map depending on how the target is being met

• Second stage - energy and performance issues.
  • Track short-term power draw and temperature to back-off high-performance goals if needed
  • Track an energy per instruction metric and try to optimize this while not impairing the performance goals
**DVFS as a systems problem for the OS**

- Based on thread groups and matching target metrics to measured metrics, a map of the best performance schedule for the thread group is created (i.e. each thread gets mapped onto a P vs E core at certain voltage/frequency).

- These maps are then read by the thread scheduler (Thread Director on Adler Lake CPU, for example) which puts each thread into an E vs P queue, with the appropriate DVFS info attached to it. The queues are adjusted across different thread groups to ensure maximal occupancy of every core.
Design Over Wide Range of Voltages

- Circuit design constraints. (Functional verification)
- Circuit delay variation. (Timing verification)
- Noise margin reduction. (Power grid, coupling)
- Delay sensitivity. (Local power distribution)

Design verification complexity similar to high-performance processor design @ fixed $V_{DD}$
• Cannot use NMOS pass gates – fails for $V_{DD} < 2V_T$.
• Functional verification only needed at one $V_{DD}$ value.

Burd
ISSCC’00
Relative Delay Variation

Delay relative to ring oscillator

Four extreme cases of critical paths:

All vary monotonically with $V_{DD}$.

Dominated by:
- Gate
- Interconnect
- Diffusion
- Series

Timing verification only needed at min. & max. $V_{DD}$. 

Burd
ISSCC'00
Multiple Path Tracking

A. Drake, ISSCC’07

Path delay variation at nominal voltage

<table>
<thead>
<tr>
<th>ps</th>
<th>adder</th>
<th>nor</th>
<th>nand</th>
<th>pass-gate</th>
<th>wire</th>
</tr>
</thead>
<tbody>
<tr>
<td>core 0: σ</td>
<td>3.78</td>
<td>3.45</td>
<td>3.91</td>
<td>3.28</td>
<td>2.82</td>
</tr>
<tr>
<td>core 1: σ</td>
<td>3.55</td>
<td>3.16</td>
<td>2.25</td>
<td>3.55</td>
<td>4.21</td>
</tr>
<tr>
<td>chip: σ</td>
<td>4.09</td>
<td>4.03</td>
<td>3.90</td>
<td>4.80</td>
<td>4.10</td>
</tr>
<tr>
<td>core 0: Δ</td>
<td>10</td>
<td>9</td>
<td>11</td>
<td>10</td>
<td>7</td>
</tr>
<tr>
<td>core 0: Δ</td>
<td>12</td>
<td>9</td>
<td>6</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>core 0: Δ</td>
<td>13</td>
<td>11</td>
<td>14</td>
<td>16</td>
<td>15</td>
</tr>
</tbody>
</table>
Multiple Path Tracking

[Tunable replica circuit]

Path type  Tunable delay  Polarity

selectable INV/NAND/NOR/INV delay

tunable INV delay

rise/fall

TDC[15] TDC[14] ... TDC[0]

DQ# DQ ...

DQ

CORE CLK
Tracking with SRAM in Critical Path

Mismatch between logic and SRAM

$V_{DD} = 0.6V$

$V_{DD} = 1.1V$

$\Delta V_{BL} = \frac{V_{DD}}{2 \cdot n}$

$V_{OS} (= 0.15V)$

SRAM multiplicative replica

Niki, JSSC’11
Alternative: Error Detection

Bull, ISSCC'2010

EECS251B L21 DVFS, CLOCK GATING
Design for Dynamically Varying VDD

- Static CMOS logic.
- Ring oscillator.
- Dynamic logic (& tri-state busses).
- Sense amp (& memory cell).

Max. allowed $|dV_{DD}/dt| \rightarrow \text{Min. } C_{DD} = 100\text{nF (0.6}\mu\text{m)}$

Circuits continue to properly operate as $V_{DD}$ changes
Static CMOS robustly operates with varying $V_{DD}$.

$V_{in} = 0$, $V_{out} = V_{DD}$

$r_{ds|PMOS}$

$C_L$

$V_{out}$

max. $\tau = 4\text{ns}$

0.6μm CMOS: $|dV_{DD}/dt| < 200\text{V/μs}$

• Static CMOS robustly operates with varying $V_{DD}$. 
Ring Oscillator

Simulated with $dV_{DD}/dt = 20V/\mu s$

- Output $f_{CLK}$ instantaneously adapts to new $V_{DD}$. 

EECS251B L21 DVFS, CLOCK GATING
Dynamic Logic

Errors

- False logic low: $\Delta V_{DD} > V_{TP}$
- Latch-up: $\Delta V_{DD} > V_{be}$

$0.6 \mu m$ CMOS: $|dV_{DD}/dt| < 20V/\mu s$

- Cannot gate clock in evaluation state.
- Tri-state busses fail similarly → Use hold circuit.
• Dynamic operation can increase energy efficiency > 10x.

Burd
ISSCC’00
V_{DD}-Hopping

Application slicing and software feedback guarantee real-time operation.

Two hopping levels are sufficient.

Normalized power

Transition time between f levels = 200\mu s

MPEG-4 encoding

Time

n-th slice finished here

Next milestone

# of frequency levels

1 2 3 \infty

EECS251B L21 DVFS, CLOCK GATING
Dithering Between Supply Levels

- Done with switched-capacitor DC-DC converters which efficiently work only at discrete levels

Keller et al, ESSCIRC’16
Dithering Between Supply Levels

• Dithering fills in between fixed DC-DC modes

Keller et al, ESSCIRC’16
Clock Gating
## Power/Energy Optimization Space

<table>
<thead>
<tr>
<th>Energy</th>
<th>Constant Throughput/Latency</th>
<th>Variable Throughput/Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Design Time</td>
<td>Sleep Mode</td>
</tr>
<tr>
<td><strong>Active</strong></td>
<td>Logic design, Scaled $V_{DD}$, Trans. sizing, Multi-$V_{DD}$</td>
<td>Clock gating, DFS, DVS</td>
</tr>
<tr>
<td><strong>Leakage</strong></td>
<td>Stack effects, Trans sizing, Scaling $V_{DD}$ + Multi-$V_{Th}$</td>
<td>Sleep T’s, Multi-$V_{DD}$ Variable $V_{Th}$ + Input control</td>
</tr>
</tbody>
</table>
• Lots of flops don’t change their state very frequently

Wimer TVLSI’14
Clock gating

• Selective shut-down of a part of a clock tree
Clock gating cell types

• Latch Free

• Latch Based
Value-based clock gating

- If both input operands for add have all zeros in their top 48 bits, these bits do not have to be latched and sent to the functional units.
- Zeros can be multiplexed onto the top 48 bits of the result bus, rather than computed via the adder.
- Low 16 bits are always latched normally.
- High 48 bits are selectively latched based on zero48 signal that accompanies the input operand from the reservation stations or the bypass network.
Data-driven clock gating (activity-driven)

• Adds timing overhead to Tclk for 10-20% power savings

Wimer TVLSI'12
Clock gating does not come for free

- Increases the number of critical paths

Wimer TVLSI'14
Next Lecture

- More power-saving techniques: Multiple thresholds, sleep modes