New Approaches to Total Power Reduction Including Runtime Leakage

Dennis Sylvester
University of Michigan, Ann Arbor
Electrical Engineering and Computer Science
http://vlsida.eecs.umich.edu
dennis@eecs.umich.edu
March 1, 2004

Colleagues on this work: Prof. David Blaauw, Ashish Srivastava, Dongwoo Lee, Harmander Deogun, Rajeev Rao, Saumil Shah

Components of power dissipation

- Increasing contribution of static (leakage) power
- Leakage is significant in both standby mode (mobile apps) and runtime (high-performance non-mobile parts)

Figure source: Intel
Reducing Power Dissipation

- Pressing need to reduce power dissipation
  - High-performance designs
    - Packaging / cooling costs
    - Power supply integrity
    - Reliability (temperature)
  - Mobile applications
    - In addition to above: Battery life

- Circuit performance is generally determined by a small fraction of the gates
  - Requires the availability of very high performance devices
    - Higher Vdd
    - Lower threshold voltage
    - Aggressive gate length

- All gates in the design contribute to power dissipation
  - Would like to use slower devices whenever possible (higher Vth, lower Vdd, possibly longer gate lengths)

Multiple Vth

- Exponential reduction in leakage power
- Cost : Additional masks
- Value of higher threshold
  - Tradeoff: Delay penalty ↔ Leakage reduction
- Can be easily incorporated into standard design flows
  - Multi-threshold library
  - Tradeoff: Library size ↔ runtime
  - Generally threshold selection is done at gate level
    - 2X library size
- Provides runtime leakage power reduction
  - Contrary to standby mode based approaches
Multiple Vdd

- Quadratic reduction in switching power
  \[ P_{\text{switching}} \sim \alpha_{sw} \cdot C_L \cdot V_{DD}^2 \cdot f \]
- Roughly cubic reduction in leakage power (DIBL, V*Ioff)
- Value of lower Vdd
  - 0.6 – 0.7 times V_{DD}^{high}
  - 0.5*V_{DD}^{high} in dual-Vth processes

![Power vs. VDDL graph]

Ref: Usami

Multiple Vdd - Topological Constraint

- V_{DD}^{low} cells cannot be directly connected to V_{DD}^{high} cells
  - PMOS does not turn off
  - Results in static current

- Level converters (LCs) are used to up-convert a low Vdd signal to a high Vdd signal
  - Incurs delay and energy overhead
**Multiple Vdd – 2 General Approaches**

- Clustered Voltage Scaling (CVS)
  - Only one voltage transition along a path
  - Level conversion only at flip-flops

- Extended CVS (ECVS)
  - Multiple voltage transitions along a path
  - Level conversion using asynchronous LC's
  - 40 - 50% improvement in power observed

**Other Issues in Multi-Vdd**

- Generation of additional voltage supplies
- Impact on power grid design
- Hard to use standard design tools
  - Simple Power Compiler based approach found to provide only a 6% power reduction
  - Cell layout must change
  - Increase in routing costs
Outline

- Concurrent Vdd/Vth assignment and sizing algorithm
- Standby mode leakage reduction using state, Vth, and Tox assignment
- Runtime leakage reduction with bus encoding + novel Vth assignment strategies

Our Approach: Overview

- Seek to maximize total power reduction in a dual Vdd/Vth design
- Uses Vdd, Vth, and sizing: VVS
  - VVS is a two-pass approach
    - Uses sensitivity metrics to minimize power in each pass
  - 1st pass: CVS with concurrent up-sizing
    - Generates slack and allows for a larger fraction of gates to be set to low Vdd
  - 2nd pass: Move back towards primary outputs (POs), setting gates to high Vth and re-setting gates to high Vdd or resizing to recover slack
    - Continue while total power dissipation is found to decrease
Gate Level Vdd/Vth Assignment

- Perform timing analysis and begin CVS
  - Initial circuit synthesized at $V_{dd\text{high}}$ and $V_{th\text{low}}$
- Obtain the candidate set of gates (front)
  - Do not serve as input to any high Vdd gate
  - If set to low Vdd will violate timing

Backward Pass

- Order candidates based on a metric
  - Slack, capacitance, etc.
- To meet timing size up gates
  - Gates to be sized up are obtained based on sensitivity
  - Size up until timing is again met
- Sensitivity=$\Delta D/\Delta \text{Area}$
  - \[ \Delta D = \sum_{\text{arcs}} (\Delta \text{delay}_{\text{arc}}(t) \times \frac{1}{(k + \text{Min (slack)} - \text{slack}_{\text{arc}})}) \]
  - $k$ is a small positive number
  - Weights arcs that impact critical paths
Backward Pass, cont.

- Stopping criterion
  - When a gate is set to low Vdd only a fixed number of gates are upsized
  - The total power dissipation measure is not used in the hope to get out of local minima
- The end of the pass is signaled when no candidate gates can be set to low Vdd
- The best seen solution is stored and is restored at the end of the pass

Forward Pass

- Now candidate gates which define the front are
  - Operating at low Vdd
  - Have all high Vdd as inputs
- Select gates on the front and set them to high Vdd/upsizen
  - Select gates to be set to high Vt
  - Commit these changes if total power is found to decrease
  - Stop when no available options for gate upsizing/high Vdd assignment
- The gates are set to high Vth based on their sensitivity
  - Sensitivities of the form $\Delta$Power/$\Delta$Delay
    - Weighted by slack
  - All gates are candidates to be set to high Vth (no topological constraints)
Results

- 0.13µm process, timing constraint is 20% slower than absolute fastest design point (optimally sized, all Vdd\text{high} and V\text{thlow})
- Vdd\text{high}=1.2V, V\text{thhigh}=0.23V
- Vdd\text{low}=0.6V, V\text{thlow}=0.12V

High switching activity at primary inputs
CVS+sizing (backward pass) does much better than just CVS

Impact of Circuit Activity

- For low activities the algorithm successfully steers toward a better solution by attacking leakage power more directly
  - In some benchmarks switching power is increased to minimize total power
  - Low activities \(\rightarrow\) converges dual-Vth + sizing
- VVS provides a single cohesive algorithm that seeks out best power reduction over a range of switching activities
  - Ex: across functional units in a design

Average power reduction by component across switching activities

<table>
<thead>
<tr>
<th>Activity</th>
<th>Static</th>
<th>Dynamic</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>High (3)</td>
<td>41%</td>
<td>31%</td>
<td>34%</td>
</tr>
<tr>
<td>Nominal (1)</td>
<td>69%</td>
<td>16%</td>
<td>45%</td>
</tr>
<tr>
<td>Low (1/3)</td>
<td>73%</td>
<td>7%</td>
<td>59%</td>
</tr>
</tbody>
</table>
Other results ...

- For high switching activities, VVS assigns many gates to low Vdd and low Vth combination to attack dynamic power.

- Exhaustive cutset enumeration was performed to find optimal results:
  - VVS performs close to optimal.
  - Least effective when optimal front lies in middle of circuit (more possibilities).

<table>
<thead>
<tr>
<th>Backoff</th>
<th>Initial Power (uW)</th>
<th>Final Power using VVS (uW)</th>
<th>Final Power using cutset enumeration (uW)</th>
<th>% Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2</td>
<td>117.10</td>
<td>91.70</td>
<td>91.70</td>
<td>0.00%</td>
</tr>
<tr>
<td>1.3</td>
<td>95.70</td>
<td>74.27</td>
<td>73.90</td>
<td>0.38%</td>
</tr>
<tr>
<td>1.4</td>
<td>78.60</td>
<td>57.84</td>
<td>56.90</td>
<td>1.20%</td>
</tr>
<tr>
<td>1.5</td>
<td>72.60</td>
<td>56.12</td>
<td>51.50</td>
<td>6.37%</td>
</tr>
<tr>
<td>1.6</td>
<td>66.70</td>
<td>48.40</td>
<td>46.80</td>
<td>2.40%</td>
</tr>
</tbody>
</table>

Outline

- Concurrent Vdd/Vth assignment and sizing algorithm

- Standby mode leakage reduction using state, Vth, and Tox assignment

- Runtime leakage reduction with bus encoding + novel Vth assignment strategies
Leakage Current Components

- **Subthreshold leakage** ($I_{\text{sub}}$)
  - Dominant when device is OFF
  - Enhanced by reduced $V_t$ from process scaling

- **Gate tunneling leakage** ($I_{\text{gate}}$)
  - Due to aggressive scaling of gate oxide thickness ($T_{\text{ox}}$)
  - A super-exponential function of $T_{\text{ox}}$
  - Comparable to $I_{\text{sub}}$ in 90nm technologies

---

Low Power Standby Mode

- Previous approaches to put a circuit into standby mode
  - State assignment [Halter, CICC1997]
  - Multi-threshold CMOS (MTCMOS) [Mutoh, JSSC1995]
  - Dual-$V_t$ assignment [Wei, DAC1998]
  - Simultaneous state and $V_t$ assignment [Lee, DAC2003]

- Only for subthreshold leakage reduction

- Proposed work
  - Leakage current reduction in standby mode
  - Minimize both $I_{\text{sub}}$ and $I_{\text{gate}}$
  - Simultaneous state, $V_t$ and $T_{\text{ox}}$ assignment
  - Gate leakage for PMOS
    - One order of magnitude smaller than NMOS
    - PMOS $I_{\text{gate}}$ is considered negligible in current analysis
Introduction – Dual $V_t$ and Dual $T_{ox}$

- Exploit dual oxide thickness technologies (becoming available)
  - Dual $T_{ox}$ – for $I_{gate}$ minimization
  - Dual $V_t$ – for $I_{sub}$ minimization

<table>
<thead>
<tr>
<th>Assignment</th>
<th>Normalized values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_t$</td>
<td>Oxide thickness</td>
</tr>
<tr>
<td>Low</td>
<td>Thin</td>
</tr>
<tr>
<td>High</td>
<td>Thin</td>
</tr>
<tr>
<td>Low</td>
<td>Thick</td>
</tr>
<tr>
<td>High</td>
<td>Thick</td>
</tr>
</tbody>
</table>

- $\Delta T_{ox} \sim 3A$, $\Delta V_t \sim 120mV$, $I_{gate}/I_{leak} = 36\%$
- Both high $V_t$ and thick $T_{ox}$: very large performance impact

Overview of Approach

- If input state is unknown
  - Cannot be predicted which transistors will be ON or OFF
  - Some transistors must be assigned to both high-$V_t$ and thick oxide

- Given a known input state
  - OFF device: $I_{gate}$ is small
    - Considered only for high-$V_t$
  - ON device: no impact on $I_{sub}$
    - Only needs to be considered for thick $T_{ox}$

- A transistor need not be assigned to both high-$V_t$ and thick $T_{ox}$
  - Significantly improved leakage/delay trade-off
- Only a subset of transistors need to be considered for high-$V_t$ or thick $T_{ox}$
Exploit Input Pin Re-ordering

- $I_{\text{gate}}$ dependence of input pin ordering [Lee, DAC2003]
  - $I_{\text{gate}}$ depends strongly on the position of ON/OFF transistors
  - Place off-transistor at bottom of stack

- Reduce performance penalty of thick-oxide transistors

Cell Library Options

- Library options
  - Trade-off points for a given gate
    - 4 vs. 2
    - Details in the paper (DATE04)
  - $V_t$ or $T_{\text{ox}}$ assignment control in a stack
    - Individual-based vs. stack-based
    - Both libraries have the same number of cells

<table>
<thead>
<tr>
<th>Stack control</th>
<th>Individually</th>
<th>Uniform</th>
</tr>
</thead>
<tbody>
<tr>
<td># of trade-off points</td>
<td>2</td>
<td>✓</td>
</tr>
</tbody>
</table>
Heuristics

- Exact solution has search space size of $2^{n+2m}$ (where $n$ is # of PIs and $m$ is # of gates)
- Branch and bound approach used
- Heuristic 1
  - Both state & gate tree: only one downward traversal
  - Gate tree: pre-sorted by leakage
    - Tends to produce a fast high quality solution
- Heuristic 2
  - Gate tree: only one downward traversal
  - State tree: search w/time limit
- Results indicate
  - Heuristic 1: fast runtime
  - Heuristic 2: better results

Results

- Leakage current comparison between heuristics
  - 5% of maximum delay penalty
  - Baseline is avg of 10K random input vectors

<table>
<thead>
<tr>
<th></th>
<th>All Low $V_t$ &amp; thin $T_{ox}$</th>
<th>Heu1</th>
<th>Heu2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current (uA)</td>
<td>$I_{leak}$</td>
<td>X</td>
<td>Time</td>
</tr>
<tr>
<td>c432</td>
<td>24.5</td>
<td>6.9</td>
<td>3.6</td>
</tr>
<tr>
<td>c499</td>
<td>65.8</td>
<td>24.8</td>
<td>2.7</td>
</tr>
<tr>
<td>c880</td>
<td>50.1</td>
<td>8.7</td>
<td>5.7</td>
</tr>
<tr>
<td>c1355</td>
<td>70.8</td>
<td>15.4</td>
<td>4.6</td>
</tr>
<tr>
<td>c1908</td>
<td>56.7</td>
<td>14.7</td>
<td>3.9</td>
</tr>
<tr>
<td>c2670</td>
<td>104.7</td>
<td>14.7</td>
<td>7.1</td>
</tr>
<tr>
<td>c3540</td>
<td>128.5</td>
<td>21.6</td>
<td>6.0</td>
</tr>
<tr>
<td>c5315</td>
<td>221.2</td>
<td>31.1</td>
<td>7.1</td>
</tr>
<tr>
<td>c6288</td>
<td>346.8</td>
<td>114.7</td>
<td>3.0</td>
</tr>
<tr>
<td>c7552</td>
<td>270.0</td>
<td>32.6</td>
<td>8.3</td>
</tr>
<tr>
<td>alu64</td>
<td>260.0</td>
<td>42.2</td>
<td>6.2</td>
</tr>
<tr>
<td>AVG</td>
<td></td>
<td>5.3</td>
<td></td>
</tr>
</tbody>
</table>

Delay with all low $V_t$ & thin $T_{ox}$

- 0%
- 5%
- 10%
- 25%

Delay with all high $V_t$ & thick $T_{ox}$

- 100%
## Results

- Leakage current comparison vs. previous work
  - At 25% delay penalty

<table>
<thead>
<tr>
<th></th>
<th>All low $V_t$ &amp; thin $T_{ox}$</th>
<th>$V_t$ &amp; State</th>
<th>$V_t$, $T_{ox}$ &amp; State</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$I_{leak}$</td>
<td>X</td>
<td>$I_{leak}$</td>
</tr>
<tr>
<td>c432</td>
<td>24.5</td>
<td>8.2</td>
<td>3.0</td>
</tr>
<tr>
<td>c499</td>
<td>65.8</td>
<td>23.8</td>
<td>2.8</td>
</tr>
<tr>
<td>c880</td>
<td>50.1</td>
<td>16.2</td>
<td>3.1</td>
</tr>
<tr>
<td>c1355</td>
<td>70.8</td>
<td>23.9</td>
<td>3.0</td>
</tr>
<tr>
<td>c1908</td>
<td>56.7</td>
<td>18.2</td>
<td>3.1</td>
</tr>
<tr>
<td>c2670</td>
<td>104.7</td>
<td>30.0</td>
<td>3.5</td>
</tr>
<tr>
<td>c3540</td>
<td>128.5</td>
<td>40.3</td>
<td>3.2</td>
</tr>
<tr>
<td>c5315</td>
<td>221.2</td>
<td>70.6</td>
<td>3.1</td>
</tr>
<tr>
<td>c6288</td>
<td>346.8</td>
<td>112</td>
<td>3.1</td>
</tr>
<tr>
<td>c7552</td>
<td>270.0</td>
<td>84.2</td>
<td>3.2</td>
</tr>
<tr>
<td>alu64</td>
<td>260.0</td>
<td>75.3</td>
<td>3.5</td>
</tr>
<tr>
<td>AVG</td>
<td>3.1</td>
<td>9.1</td>
<td></td>
</tr>
</tbody>
</table>

- Leakage current comparison between cell library options
  - At 5% delay constraint

<table>
<thead>
<tr>
<th></th>
<th>All low $V_t$ &amp; thin $T_{ox}$</th>
<th>4-option individually</th>
<th>2-option individually</th>
<th>4-option uniform stack</th>
<th>2-option uniform stack</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$I_{leak}$</td>
<td>X</td>
<td>$I_{leak}$</td>
<td>X</td>
<td>$I_{leak}$</td>
</tr>
<tr>
<td>c432</td>
<td>24.5</td>
<td>6.9</td>
<td>3.6</td>
<td>7.5</td>
<td>3.3</td>
</tr>
<tr>
<td>c499</td>
<td>65.8</td>
<td>24.8</td>
<td>2.7</td>
<td>27.6</td>
<td>2.4</td>
</tr>
<tr>
<td>c880</td>
<td>50.1</td>
<td>8.7</td>
<td>5.7</td>
<td>9.0</td>
<td>5.6</td>
</tr>
<tr>
<td>c1355</td>
<td>70.8</td>
<td>15.4</td>
<td>4.6</td>
<td>17.0</td>
<td>4.2</td>
</tr>
<tr>
<td>c1908</td>
<td>56.7</td>
<td>14.7</td>
<td>3.9</td>
<td>15.2</td>
<td>3.7</td>
</tr>
<tr>
<td>c2670</td>
<td>104.7</td>
<td>14.7</td>
<td>7.1</td>
<td>12.2</td>
<td>8.6</td>
</tr>
<tr>
<td>c3540</td>
<td>128.5</td>
<td>21.6</td>
<td>6.0</td>
<td>23.9</td>
<td>5.4</td>
</tr>
<tr>
<td>c5315</td>
<td>221.2</td>
<td>31.1</td>
<td>7.1</td>
<td>30.7</td>
<td>7.2</td>
</tr>
<tr>
<td>c6288</td>
<td>346.8</td>
<td>114.7</td>
<td>3.0</td>
<td>120.6</td>
<td>2.9</td>
</tr>
<tr>
<td>c7552</td>
<td>270.0</td>
<td>32.6</td>
<td>8.3</td>
<td>31.2</td>
<td>8.7</td>
</tr>
<tr>
<td>alu64</td>
<td>260.0</td>
<td>42.2</td>
<td>6.2</td>
<td>42.3</td>
<td>6.2</td>
</tr>
<tr>
<td>AVG</td>
<td>5.28</td>
<td>5.27</td>
<td>4.91</td>
<td>4.77</td>
<td></td>
</tr>
</tbody>
</table>
Outline

- Concurrent Vdd/Vth assignment and sizing algorithm
- Standby mode leakage reduction using state, Vth, and Tox assignment
- Runtime leakage reduction with bus encoding + novel Vth assignment strategies

Runtime leakage in buses

- 50% of total chip leakage in inverters/buffers
  - Much of this in repeaters which are:
    - Very wide
    - Growing in #
    - Heavily speed constrained so often use low Vth
    - Do not experience stack effect as multi-input gates do

- Standby leakage reduction relatively easy compared to runtime
  - We can absorb a delay penalty when we know that no new data is coming
  - In runtime, data can come at any time; must be ready to process as fast as possible

- What can we do besides dual-Vth?
Staggered Vth bus design

- Selective use of high-Vth devices yields the possibility of low leakage in runtime
  - Stagger them along the wire to create a very low leakage state
  - Delay (or dynamic energy) penalty is much lower than all high-Vth
  - We cannot dictate state in runtime so this does not help in general
    - Unless we can dictate state

Encoding to enforce proper state

- Choose a 3→4 encoding, also eliminate worst-case crosstalk
- Exact encoding selected to minimize total power
  - Requires anticipated state and transition probabilities
    - Ex: what is the most common state, what is the most common transition
  - Also consider the encode/decode logic complexity
Reducing encoding complexity

- We consider all possible encodings (mappings from input states to actual transmitted encoded states) within T% of minimal
- Then use logic complexity as tiebreaker
- Results in 1-2% power penalty with 13% fewer gates/area overhead

Results (includes delay overhead)

- 0.13um CMOS at 105C, 64-bit Alpha architecture running 9 applications (address bus)
- 26% total power savings on average, 42% leakage reduction
- Maximal switching activity case (Test_1), total power still reduced
- Compared to previous crosstalk-aware approaches, we save 54% total power (nearly all of it in leakage)
Alternate Repeater Vth Assignments

- Other possibilities of Vth assignment in repeaters can help reduce leakage in runtime

- **Separate NMOS/PMOS Vth (SPNVt)**
  - All PMOS are low-Vth, all NMOS are high-Vth
  - Advantages: predictable leakage (state independent), balances fast/slow paths through the repeater chain, easy to manufacture

- **Mixed Vth**
  - Wide devices such as in repeaters are split into parallel fingers, separated by a contacted pitch
  - Assign a fraction, \( \alpha \), of total width to low-Vth (1 - \( \alpha \) is then high-Vth)
  - Effectively a third Vth with speed and leakage behavior intermediate to high/low Vth
    - No manufacturing costs for this 3rd Vth, no area penalties since parallel fingers are spaced out significantly already

Vth assignment scheme results

- Mixed config: \( \alpha = 0.3 \)
- Achievable speed is best for mixed, also good for SPNVt
- Runtime leakage of \( \alpha = 0.3 \) is 54% lower than low-Vth with small dynamic energy penalty
- Total average power reduction is 14%
  - Switching behavior taken from 11 benchmark applications, address bus
  - Strongly depends on ratio of static to dynamic power

Hybrid approaches are possible; upper bits in 64-bit address buses are usually zeroes
  
  Stagger to favor low-leakage 0s
Conclusions

- Need to leverage “multi-everything” to address the power management gap
  - EDA must enable simultaneous sizing, Vdd, and Vth assignment; the 3 major knobs in power reduction
  - Total power reductions on the order of 35-60% are achievable

- Standby mode leakage can be effectively reduced by combining state assignment with Vth and Tox assignment
  - Sizable leakage reductions (5-9X) with modest delay penalties (3-15% vs. all low Vt and thin Tox)
  - Much less overhead than MTCMOS, body biasing

- Runtime leakage in global interconnect repeaters can be addressed using Vth assignment schemes (sometimes with encoding)
  - 40-54% leakage reductions with small dynamic power penalty
  - Total power savings depends heavily on static/dynamic ratio
    - Implies these techniques improve with scaling
  - Mixed Vth provides pseudo-continuous Vth assignment, opening up a range of new optimizations in the energy/delay design space