Tabu Search Based Gate Leakage Optimization using DKCMOS Library in Architecture Synthesis

Saraju P. Mohanty  
Dept. of Comp. Science & Engineering  
University of North Texas, USA.  
Email: saraju.mohanty@unt.edu

Dhiraj K. Pradhan  
Dept of Computer Science  
University of Bristol, UK.  
Email: pradhan@compsci.bristol.ac.uk

Acknowledgment: This research is supported in part by NSF award numbers CCF-0702361 and CNS-0854182 and EPSRC grant EP/G032904/1.
Outline of the Talk

- Introduction
- Prior Related Research
- DKCMOS Technology
- Tabu Search Based Gate Leakage Optimization
- Architecture Component Library
- Experimental Results
- Conclusions
Introduction and Motivation
Why Low Power?

- Packaging costs
- Chip and system cooling costs
- Power supply rail
- Power affects
- Noise and reliability
- Environmental
- Battery life
Power Dissipation in CMOS

- **Power Dissipation**
  - **Static Dissipation**
    - Sub-threshold current
    - Gate Leakage
    - Reverse-biased diode Leakage
    - Contention current
  - **Dynamic Dissipation**
    - Capacitive Switching
    - Gate Leakage
    - Short circuit

ICIT 2009

UNIVERSITY OF NORTH TExAS™

Mohanty 5
Power Dissipation Redistribution

- Sub-Threshold
- Gate Leakage
- Gate Length
- Dynamic

Trajectory, With High-K

Normalized Power Dissipation

Chronological (Year) →

Source: Hansen Thesis 2004
Leakages in CMOS

$I_1$: reverse bias pn junction (both ON & OFF)
$I_2$: subthreshold leakage (OFF)
$I_3$: Gate Leakage current (both ON & OFF)
$I_4$: gate current due to hot carrier injection (both ON & OFF)
$I_5$: gate induced drain leakage (OFF)
$I_6$: channel punch through current (OFF)

Source: Roy 2003
Contributions of this Paper

- Introduces DKCMOS technology for architectural level gate leakage and delay tradeoffs.
- Presents an algorithm that schedules operations of a sequencing data flow graph (DFG) and maps the operations to RTL library for optimization.
- The algorithm minimizes the gate leakage of datapath circuits for given resource constraints and time constraints.
- The RTL library is constructed for classical SiO$_2$ device based modules, and two nonclassical high-K based modules.
Prior Related Research
**Related Research: RTL**

**Subthreshold Leakage:**
- Gopalakrishnan - ICCD2003: Dual-$V_{Th}$ approach for reduction of subthreshold current through binding.

**Gate Leakage:**
- Mohanty - VLSI Design 2006: Dual-$T_{Ox}$ approach for reduction of gate leakage current.
- Mohanty - ISQED 2006: Simulated annealing algorithms using dual-$K$ or dual-$T_{Ox}$. 
Related Research: Logic / Transistor Level Gate Leakage Reduction

- Lee - TVLSI 2004: Pin reordering to minimize gate leakage during standby positions of logic gates.
- Sirisantana - IEEE DTC Jan-Feb 2004: Use multiple channel lengths and multiple gate oxide thickness for reduction of leakage.
DKCMOS Technology: The Key Idea
SiO$_2$ CMOS Vs and High-K CMOS

Assumption: Constant L/T aspect ratio.
Dielectrics for Replacement of SiO$_2$

- Silicon Oxynitride (SiO$_x$N$_y$) (K=5.7 for SiON)
- Silicon Nitride (Si$_3$N$_4$) (K=7)
- Oxides of:
  - Aluminum (Al), Titanium (Ti), Zirconium (Zr), Hafnium (Hf), Lanthanum (La), Yttrium (Y), Praseodymium (Pr).
  - their mixed oxides with SiO$_2$ and Al$_2$O$_3$.
- HfO$_2$ has maximum permittivity (K) of 21.
The DKCMOS Technology

In this paper it is claimed that a mix of RTL units of type (a) and type (c) will serve gate leakage and performance trade-offs and will go well with industry trend.
Target Architecture

High-K and Low-K Islands
Tabu-Search Based Gate Leakage Optimization
Problem Formulation

- Given an unscheduled data flow graph $G_U(V,E)$, it is required to obtain the scheduled data flow graph $G_S(V,E)$ with appropriate resource binding such that the total gate leakage is minimized under specified resource and time constraints.

\[
\text{Minimize: } \sum_{v_i \in V} P_{\text{gate}}(v_i)
\]
\[
\sum_{v_i \in V_{cp}} T_i(v_i) \leq T_{con}
\]
\[
\text{Allocated}(FU_i(k, K)) \leq \text{Available}(FU_i(k, K))
\]
RTL Optimization for Gate Leakage

Input HDL

Compilation

Transformation

Data Flow Graph

Gate Leakage and Delay Estimator

RTL Scheduler for Gate Leakage Reduction

Resource Allocation and Binding

RTL Cell Library

Datapath and Control Generation

Datapath and Control Generation

RTL Description

Logic Synthesis

Physical Synthesis

Output Layout Description
The Tabu-Search based algorithm minimizes gate-leakage while performing simultaneous scheduling, allocation, and binding.

The Tabu-Search based algorithm uses more aggressive approach than other algorithms.

The Tabu-Search based algorithm skips inferior solutions and gets out of local optimization easily.

The Tabu-Search provides useful solution to the problems in a reasonable time.
1. Preprocess given behavioral description to construct a sequencing DFG.
2. Perform simulations to estimate gate leakage and delay of RTL units.
3. Construct resource allocation table and available resource table based on input resource constraints.
4. Obtain ASAP and ALAP schedules of the input DFG.
5. Determine the number of different resources for each K using the resource allocation table.
Optimization Algorithm Flow

6. Modify both ASAP and ALAP schedules obtained above using the number of resources found in previous step.

7. Construct the mobility graph based on above schedules.

8. Fix the total number of clock cycles as the maximum of modified ASAP and ALAP schedules’ control step.

9. Assume initial schedule as modified ASAP schedule and initial binding as high-K resource for each vertex.

10. Call the Tabu-Search based algorithm for optimal solution.

11. Estimate leakage and delay of the final solution.

12. Postprocess final DFG to obtain the RTL description.
Tabu-Search Based Algorithm

Tabu-Search Algorithm (UDFG, Resource/Time Constraints)
(01) Consider S as initial solution and $P_{gate-S}$ as estimate.
(02) Initialize the number of iterations as Counter = 0.
(03) While (Counter < Max-Iteration) do
(04) Increment Counter.
(05) Generate a neighborhood solution $S^*$ for constraint $T_{con}$.
(06) Estimate gate leakage $P_{gate-S^*}$ for that solution.
(07) If (Solution $S^*$ is not visited in previous iteration) then
(08) If ($P_{gate-S^*} < P_{gate-S}$) then Update S with new solution $S^*$.
(09) Else
(10) Discard the new solution $S^*$.
(11) End if.
(12) End If.
(13) End If.
(14) End While.
Datapath Components Library
It is observed that a NAND gate has least gate leakage compared to all other basic logic gates. Therefore we constructed datapath components using NAND gates.
First the NAND gate is characterized using analog simulations and then the functional units.

It is assumed that there are total $n_{total}$ NAND gates in the network of NAND gates constituting an $n$-bit functional unit out of which $n_{cp}$ are in the critical path of the logic netlist.

The effect of interconnect wires are not considered and the focus is on the gate leakage dissipation and propagation delay of the active units only.
Datapath Component Library ...
(NAND Gate)

<table>
<thead>
<tr>
<th>Input Configuration</th>
<th>Circuit Diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>![Circuit Diagram for Input 00]</td>
</tr>
<tr>
<td>01</td>
<td>![Circuit Diagram for Input 01]</td>
</tr>
<tr>
<td>10</td>
<td>![Circuit Diagram for Input 10]</td>
</tr>
<tr>
<td>11</td>
<td>![Circuit Diagram for Input 11]</td>
</tr>
</tbody>
</table>
The gate leakage current for a specific state of a logic gate is then calculated by:

$$I_{\text{gate Logic}_{\text{state}}} = \sum_{\forall MOS_i} I_{\text{gate MOS}[i]}$$

The gate leakage of a n-bit RTL unit is calculated as:

$$I_{\text{gate}_R} = \sum_{j=1}^{n_{\text{total}}} \text{Pr ob}(\text{state}) I_{\text{gate NAND}_j_{\text{state}}}$$

The propagation delay of an n-bit functional unit is:

$$T_{pd_R} = \sum_{i=1}^{n_{cp}} T_{pd_{\text{NAND}_i}}$$
Datapath Component Library …
(Inverter Showing Components)

- **Low Input**: Input supply feeds tunneling current.
- **High Input**: Gate supply feeds tunneling current.

Low Input: $V_{in} = V_{low}$

- $V_{out} = V_{high}$
- $I_{gd}$
- $I_{gs}$
- $I_{gcs}$

High Input: $V_{in} = V_{high}$

- $V_{out} = V_{low}$
- $I_{gd}$
- $I_{gs}$
- $I_{gcs}$

NOTE: Gate to body component found to be negligible.
Datapath Component Library ...

(A CMOS Transistor)

- Calculated by evaluating both the source and drain components

- For a MOS, \( I_{\text{gate}} = (|I_{gs}| + |I_{gd}| + |I_{gcs}| + |I_{gcd}| + |I_{gb}|) \)

- Values of individual components depend on states, ON or OFF
The effect of varying dielectric material was modeled by calculating an equivalent oxide thickness \( T_{ox}^* \) according to the formula:

\[
T_{ox}^* = \left( \frac{K_{gate}}{K_{ox}} \right) T_{gate}
\]

Here, \( K_{gate} \) is the dielectric constant of the gate dielectric material other than SiO\(_2\), (of thickness \( T_{gate} \)), while \( K_{ox} \) is the dielectric constant of SiO\(_2\).
Experimental Results
Experimental Results ...

- While calculating the gate leakage current for single thickness, we used a nominal 1.4nm thickness and SiO$_2$(K=3.9) is used as a nominal dielectric value from BSIM4.4.0 model.

- Two pairs of dual dielectric are considered:
  (i) SiO$_2$(K=3.9) – SiON (K=5.7)
  (ii) SiO$_2$(K=3.9) – Si$_3$N$_4$(K=7)

- The results take into account the gate leakage current, area and propagation delay of functional units, interconnect units, and storage units present in the datapath circuit.
### Experimental Results

<table>
<thead>
<tr>
<th>$D_T$</th>
<th>$\text{SiO}_2 (K=3.9) - \text{SiON}(K=5.7)$</th>
<th>$\text{SiO}_2 (K=3.9) - \text{Si}_3\text{N}_4 (K=7)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$P_{\text{gate}_{DK}}$ ($\mu W$)</td>
<td>$T_{\text{CP}_{DK}}$ ($ns$)</td>
</tr>
<tr>
<td><strong>Base Case:</strong> $P_{\text{gate}<em>{SK}} = 4632.74 \mu W, T</em>{\text{CP}_{SK}} = 308.9 ns$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>1.0</td>
<td>2729.4</td>
</tr>
<tr>
<td>R</td>
<td>1.1</td>
<td>1862.3</td>
</tr>
<tr>
<td>F</td>
<td>1.2</td>
<td>1741.9</td>
</tr>
</tbody>
</table>

| **Base Case:** $P_{\text{gate}} = 3655.68 \mu W, T_{\text{CP}_{SK}} = 290.1 ns$ |
| B | 1.0 | 1582.9 | 290.1 | 56.7 | 1144.2 | 290.1 | 68.7 |
| P | 1.1 | 1414.7 | 310.7 | 61.3 | 1082.0 | 290.1 | 70.4 |
| F | 1.2 | 1257.5 | 343.5 | 65.6 | 979.9 | 341.7 | 73.2 |

| **Base Case:** $P_{\text{gate}} = 4159.12 \mu W, T_{\text{CP}_{SK}} = 308.9 ns$ |
| D | 1.0 | 1879.9 | 308.9 | 54.8 | 1439.0 | 308.9 | 65.4 |
| C | 1.1 | 1813.3 | 308.9 | 56.3 | 1339.2 | 308.9 | 67.8 |
| T | 1.2 | 1522.2 | 341.7 | 63.4 | 1172.8 | 308.9 | 71.8 |

| **Base Case:** $P_{\text{gate}} = 2726.78 \mu W, T_{\text{CP}_{SK}} = 498.4 ns$ |
| E | 1.0 | 1497.0 | 498.4 | 45.1 | 1167.0 | 498.4 | 57.2 |
| W | 1.1 | 1385.2 | 531.2 | 49.2 | 107.0 | 530.6 | 59.4 |
| F | 1.2 | 1107.9 | 584.5 | 59.3 | 839.8 | 582.2 | 69.2 |
Experimental Results

The graph shows the average percentage reduction for different benchmark circuits: ARF, BPF, DCT, and EWF. The reduction is compared between SiO2-SiON and SiO2-Si3N4 materials. The percentage reduction ranges from 0% to 80%.
Conclusions and Future Research
Conclusions

- This paper presents a new process driven technique called DKCMOS for reduction of gate leakage during RTL synthesis.
- The Tabu-Search based algorithm performs scheduling and assignment for gate leakage reduction for different resource/time constraints.
- Experimental results reveal significant reductions in gate leakage with the use of this technology, thus proving its effectiveness.
Future Research

- Further exploration of this technique is the incorporation of process variation.
- The effectiveness of DKCMOS for subthreshold leakage needs investigation.
- The ultimate objective is to extend the research on gate leakage current to provide a broader solution to the problem of power dissipation in all its forms at the RTL.
- The area overhead due to the use two separate islands (high-K and nominal-K) will also be investigated.
Thank You !!!

The presentation is available at:
http://www.cse.unt.edu/~smohanty