Robust Design of Power-Efficient VLSI Circuits

Massoud Pedram
University of Southern California
Dept. of Electrical Engineering
Presentation at ISPD
March 28, 2011
Outline

• Background and Motivation
• Key Problem: Robust & Energy-Efficient Design
• Essential Elements of the Solution Approach
• Changing Landscape and New Opportunities
• Conclusion
More Functionality and Higher Performance

Intel's many-core Knights platform
ARM's Cortex-A series of applications processors
Nvidia's GeForceGTX 590 graphics processor
IBM supercomputer Watson wins Jeopardy!
Marvell’s ARMADA 1000 High-Def. Media Processor
Apple’s iphone 4 and ipad 2
Samsung’s Galaxy 4G Android Smartphone

M. Pedram
Power Efficiency

2010 ITRS’s Consumer Stationary Power Consumption

M. Pedram
Energy Cost

Sources: IDC data for installed base, shipments, and most popular models, and manufacturer data on power use for individual server models. Total expenditures assume US industrial electricity prices (2006 dollars)

1.2% of 2005 U.S. electricity sales, $2.7B/year

0.8% of estimated 2005 world electricity sales, $7.2B/year

Current Trends
Improved Operations

Source: US EPA 2007

M. Pedram
Variations

Normalized capacitance distribution

Line-edge roughness

Random dopant fluctuation

Gate oxide thickness fluctuation

Back-end variability

Sources: Kevin Nowka and Chandu Visweswariah
Variations

Example workload variations

Power supply droop map

CPU utilization of > 5,000 servers during a six-month period

M. Pedram

Sources: Sani Nassif, J. Fredrich, L.A. Barroso

Die thermal maps
Overarching Goal

Minimize power consumption of digital VLSI circuits, in presence of PVT, workload, and SLA variations
Shrinking Design Space and Increasing Uncertainty

- Double arrows indicate the desired scaling direction
- The design space bounded by the three curves is diminishing
- Region of uncertainty for designs is increasing

Reduce Switching Power
\[ P_{\text{switching}} = \text{const} \]
Reduce Leakage Power
\[ P_{\text{leakage}} = \text{const} \]
f_{\text{clk}} = \text{const}
Design space
Region of Uncertainty

M. Pedram
Required Components of a Global Solution

- Better characterizations, models, and calculators
- Multi-corner or statistical optimization
- Augment design for runtime adaptability
- Dynamic control based on in-situ sensing
A Current Source Model

\[ C_L \frac{\Delta V^o}{\Delta t} + C_o \frac{\Delta V^o}{\Delta t} + I(V_i, V_o) + C_M \frac{\Delta V^o}{\Delta t} - C_M \frac{\Delta V_i}{\Delta t} = 0 \]

- The non-linear behavior of the logic cell:
  - 2-D lookup table to store \( I(V_i, V_o) \)

- Parasitic effect of the logic cell:
  - 2-D lookup tables to store \( C_i(V_i, V_o) \), \( C_M(V_i, V_o) \) and \( C_o(V_i, V_o) \)

- Series of Spice simulation to pre-characterize the components of CSM model
Transition to “Physical” Gate Modeling: Controlled Current Source Models

Driver model

Load (pincap) model

$Z_{load}(s)$

$\text{Slew}_i$?

$\text{Slew}_o$?

$\text{Delay}_o$?
Data Explosion Problem

- Conventional logic cell delay calculation techniques ignore the actual shape of waveform
- Current Source Modeling (e.g., ECSM)
  - Two-dimensional table of voltage waveforms in terms of input slew and output capacitance
- Size of the CSM library is a serious concern
  - Data volume orders of magnitude greater than a .Lib library
  - Multiple Libraries in the Process, Temperature, Voltage (PVT) space
  - Additionally the CSM library may contain power, noise, and variability
- Goals
  - Reduce library size while maintaining accuracy
  - Parameterize all waveform data vs. slew, cap, and PVT
Variational Waveform Modeling

- Sources of variations as *input parameter*:
  - Such as supply voltage, temperature, $L_{\text{eff}}$ and $V_{\text{th}}$
- *Pre-alignment operations*
  - V-operators (shift and scale operations in the direction of the voltage axis)
Orthonormal Transformation

Training set waveforms

Construct orthonormal basis using Principal Component Analysis

Bases for the new representational space

\[ W = \alpha_0 + \alpha_1 P_1 + \ldots + \alpha_n P_n \]

- Each normalized waveform between 0 and 1 is represented by coefficients: \( \alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_n \)
Library Data Compression

- In practice each normalized waveform between 0 and 1 is represented by using a fixed number e.g., 4 coefficients: $a_0, \ldots, a_3$
  - Time vector extraction from the cell library
  - Pre-alignment
  - Preprocessing including shifting, scaling, averaging and weighting,
  - Basis set extraction by using (Robust) Principal Component Analysis (R)PCA
  - Coefficient calculation for $m$ “most significant” basis vectors
Variational Waveform Modeling

- A 65nm ECSM library with 43141 waveforms
- Nominal process corner, 1.2 volts, and 25°C
- Each gate characterized for 7x7x5x5 (input slew, output capacitive load, supply voltage and temperature) combinations
- A voltage waveform modeled by 21 uniform point increments
- Used the first five coefficients of PCA (76% compression)

Variational and nominal waveform for an inverter

1st, 2nd, and 3rd basis vectors

Histogram of relative error for 43141 waveforms (5 bases)

M. Pedram
Required Components of a Global Solution

- Better characterizations, models, and calculators
- Multi-corner or statistical optimization
- Augment design for runtime adaptability
- Dynamic control based on in-situ sensing
Optimization flow: Multi-Corners + Multi-Modes

Mobile constraints

Corner 1
Lib

Optimized for corner 1

Optimized Mobile design

Corner 2
Lib

Optimized for corner 2

Un-optimized Design

Desktop constraints

Corner 3
Lib

Optimized for corner 3

Corner 4
Lib

Optimized for corner 4

Optimized Mobile design

Optimized Desktop design

M. Pedram
We consider multiobjective optimization problems:

\[
\begin{align*}
\text{minimize} & \quad \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_k(x) \end{bmatrix} \\
\text{subject to} & \quad x \in S,
\end{align*}
\]

where

\[ f_i : \mathbb{R}^n \to \mathbb{R} = \text{objective function} \]

\[ k (\geq 2) = \text{number of (conflicting) objective functions} \]

\[ x = \text{decision vector (of n decision variables } x_i) \]

\[ S \subset \mathbb{R}^n = \text{feasible region formed by constraint functions and} \]

``minimize`` = minimize the objective functions simultaneously
• A **decision maker** (DM) is needed to identify a final Pareto optimal (PO) solution. (S)he has insight into the problem and can express **preference relations**

• An **analyst** is responsible for the mathematical side

• **Solution process** = finding a solution

• **Final solution** = feasible PO solution satisfying the DM

• Ranges of the PO set: **ideal objective vector** $z^*$ (lower bounds of the PO set) and approximated **nadir objective vector** $z^{nad}$ (upper bounds of the PO set)

• **Utopian objective vector**, $z^{**}$, is strictly better than $z^*$

\[
\begin{align*}
    f_i^* &= \min_{x \in S} f_i(x) \\
    f_i^{nad} &= \max_{1 \leq j \leq k} f_i(x_j^*) \\
    x_j^* &= \arg\min_{x \in S} f_j(x)
\end{align*}
\]
Concepts cont.

- **Value function** $U: \mathbb{R}^k \rightarrow \mathbb{R}$ may represent preferences; other times DM is expected to be maximizing some value (or utility).
  - We use the notation $u: \mathbb{R}^n \rightarrow \mathbb{R}$ where $u(x) = U(f(x))$

- If $U(z_1) > U(z_2)$, then the DM prefers $z_1$ to $z_2$. If $U(z_1) = U(z_2)$ then $z_1$ and $z_2$ are equally good (indifferent).

- Decision making can be thought of being either value maximization or *satisficing*.

- An objective vector containing the *aspiration levels* $\hat{z}_i$ of the DM is called a *reference point*, $\hat{z} \in \mathbb{R}^k$.

- Problems are usually solved by *scalarization*, where a real-valued objective function is formed (depending on parameters). Then, single objective optimizers can be used!
A Posteriori Methods: Weighting and $\varepsilon$-Constraint Methods

- Generate the PO set (or a part of it); Present it to the DM; Let the DM select one

- Problem

  \[
  \begin{align*}
  \text{minimize} & \quad \sum_{i=1}^{k} w_i f_i(x) \\
  \text{subject to} & \quad x \in S, \\
  \text{where} & \quad w_i \geq 0 \text{ for all } i = 1, \ldots, k, \\
  & \quad \sum_{i=1}^{k} w_i = 1.
  \end{align*}
  \]

- Problem

  \[
  \begin{align*}
  \text{minimize} & \quad f_\ell(x) \\
  \text{subject to} & \quad f_j(x) \leq \varepsilon_j, \text{ for all } j = 1, \ldots, k, j \neq \ell \\
  & \quad x \in S.
  \end{align*}
  \]
A Priori Methods: Goal Programming

- The DM must specify an aspiration level \( \bar{z}_i \) for each objective function
- \( f_i \) and aspiration level = a goal. Deviations from aspiration levels are minimized \( (f_i(x) - \delta_i = \bar{z}_i \) where \( \delta_i \) may be positive or negative)
- The deviations can be represented as overachievements \( \delta_i > 0 \) if \( f_i(x) \leq \bar{z}_i \)
- Weighted approach:
  - With \( x \) and \( \delta_i \) (i=1,...,k) as variables

Weights from the DM

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{k} w_i \delta_i \\
\text{subject to} & \quad f_i(x) - \delta_i \leq \bar{z}_i, \quad i = 1, \ldots, k, \\
& \quad \delta_i \geq 0, \quad i = 1, \ldots, k, \\
& \quad x \in S
\end{align*}
\]
Interactive Methods: Satisficing Trade-Off Method

- Idea: To classify the objective functions:
  - Functions to be improved ($I^<$)
  - Functions whose values can be relaxed ($I^>$)
  - Acceptable functions ($I^= $)
- Notice that $I^< \cup I^> \cup I^= = \{1,\ldots,k\}$

- Assumptions
  - Trade-off information is available in the KT-multipliers

- Aspiration levels for functions in $I^<$ given by the DM, upper bounds for function in $I^>$ from the KT-multipliers
- Satisficing decision making is emphasized
Satisficing Trade-Off Method cont.

- Problem

\[
\begin{align*}
\text{minimize} & \quad \max_{1 \leq i \leq k} \left[ \frac{f_i(x) - z_i^{***}}{\bar{z}_i - z_i^{***}} \right] \\
\text{subj. to} & \quad x \in S
\end{align*}
\]

or

\[
\begin{align*}
\text{minimize} & \quad \max_{1 \leq i \leq k} \left[ \frac{f_i(x) - z_i^{***}}{\bar{z}_i - z_i^{***}} \right] + \rho \sum_{i=1}^{k} \frac{f_i(x)}{\bar{z}_i - z_i^{***}},
\end{align*}
\]
Some Results – Transistor Sizing

- Optimization in 2.4V
- Optimization in 2.1V
- Multi-objective optimization
- Optimization in 1.8V

Delay in ns

Voltage Corners
Simulation results for the adder

<table>
<thead>
<tr>
<th>non-convex/convex</th>
<th>Method</th>
<th>Using of gradient</th>
<th>( \text{power} \times 10^{-1} )</th>
<th>Single-obj Power Optimization</th>
<th>Single-obj Delay Optimization</th>
<th>Multiobjective Optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convex</td>
<td>WS</td>
<td>w/o grad</td>
<td>Power 48.011</td>
<td>48.932</td>
<td>48.719</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Delay 11.992</td>
<td>10.463</td>
<td>10.537</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>w grad</td>
<td>Power 47.924</td>
<td>48.908</td>
<td>48.657</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Delay 12.533</td>
<td>10.279</td>
<td>10.389</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CP</td>
<td>w/o grad</td>
<td>Power 47.924</td>
<td>48.633</td>
<td>48.439</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Delay 12.533</td>
<td>10.806</td>
<td>11.018</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>w grad</td>
<td>Power 48.019</td>
<td>48.908</td>
<td>48.148</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Delay 12.808</td>
<td>10.279</td>
<td>11.317</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WS</td>
<td>w/o grad</td>
<td>Power 48.085</td>
<td>48.495</td>
<td>48.432</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Delay 13.001</td>
<td>10.647</td>
<td>10.658</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>w grad</td>
<td>Power 47.809</td>
<td>48.451</td>
<td>48.335</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Delay 13.642</td>
<td>10.306</td>
<td>10.381</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CP</td>
<td>w/o grad</td>
<td>Power 47.809</td>
<td>48.464</td>
<td>48.123</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Delay 13.542</td>
<td>10.403</td>
<td>11.408</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>w grad</td>
<td>Power 47.809</td>
<td>48.451</td>
<td>48.102</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Delay 13.642</td>
<td>10.306</td>
<td>10.791</td>
<td></td>
</tr>
</tbody>
</table>

M. Pedram
Required Components of a Global Solution

- Better characterizations, models, and calculators
- Multi-corner or statistical optimization
- **Augment design for runtime adaptability**
- Dynamic control based on in-situ sensing
On-Chip Power Delivery Network

- A voltage regulator module (VRM) converts and regulates a DC power source
  - A VRM can typically produce one of a number of distinct voltage levels based on user-specified voltage identification (VID) code
  - A VRM must meet various voltage tolerances such as voltage droops, output ripple and noise, dynamic load limit, etc.
    - Depending on these tolerances, the response time of the VRM can change

An example of voltage droops
A PDN configuration is defined by a VRM tree, which is a rooted tree with:

- A root node representing the main power source (battery)
- Internal nodes denoting various VRM’s
- Leaf nodes representing functional blocks (FB’s) with their supply voltage and peak current levels
Energy-Delay Optimal Pipeline Design

- Key idea: Allow the data to pass through a flip flop during some transparency window, instead of on the triggering edge of a clock.
- Benefit: Enable opportunistic time borrowing across adjacent pipeline stages.
  - The goal is to provide the timing-critical stages with more time to complete their computations.
    - Thus, reducing the probability of timing errors.
- Determine values of:
  - operating frequency, supply voltage level, transparency window sizes of the individual soft-edge FF-sets.
- Problem Formulations:
  - stage delays are modeled as random variables due to process variations.
  - allow timing violations to take place but then implement a mechanism to detect and fix it.

![Diagram of pipeline stages with borrowed data.]
Soft Edge Flip Flops

\[
\begin{align*}
\tau_{s,ij} &= t_s \left( w_i, v_j \right) = a_1(v_j)w + a_0(v_j) \\
\tau_{h,ij} &= t_h \left( w_i, v_j \right) = b_1(v_j)w + b_0(v_j) \\
t_{cj,q} &= t_{cj}(v_j) \\
t_{dq,j} &= t_{dq}(v_j)
\end{align*}
\]

\[P_{SEFF} = k_3(v) \frac{w}{T_{clk}} + k_2(v) \cdot w + k_1(v) \frac{1}{T_{clk}} + k_0(v)\]
Design of power-delay optimal soft pipeline (OSP)

• minimize total power-delay (energy) of an $N$-stage pipeline, by finding optimal values of:
  – global supply voltage
  – pipeline clock period
  – transparency windows of SEFF sets

• Solution:
  – enumerate all possible values for $v$,
  – for each $v$, optimally solve a quadratic program, OSP-FV:

![Graph showing %PDP Saving with respect to Baseline and Base+VS for multiple benchmarks]

M. Pedram
SEFF with Built-in Error Detection

• A shadow latch resamples the input data with some phase delay
  – Need a phase-shifted global clock signal, PS-Clk
• The two sampled data are compared to one another to detect and flag errors
Error-Tolerant Statistical Power-Delay Optimal Soft Pipeline (ESOSP)

- minimizes total power-delay
- aggressively scales down the pipeline clock period to improve performance
- employing “SEFF with error detection”
- statistical timing constraints
- variables:
  - global supply voltage
  - pipeline clock period
  - transparency windows of SEFF sets

- expected value of power-delay:
  \[ \Phi = (1 - q_j)P_j T_{cik,j} + q_j (P_j + P_{p,j})\gamma T_{cik,j} \]

- Solution:
  - enumerate all possible values for \( \nu \),
  - for each \( \nu \), optimally solve a quadratic program, ESOSP-FV:
Multi-Threshold CMOS

- High-\(V_{th}\) power switches are connected to low-\(V_{th}\) logic gates
  - Achieves high performance due to low-\(V_{th}\) logic gates
  - Reduces leakage power dramatically due to the series-connected high-\(V_{th}\) power switch
- Typically only a header or a footer sleep transistor is used, not both
- A single sleep transistor may be shared among several logic gates
Tri-Modal MTCMOS Switch

<table>
<thead>
<tr>
<th>SLEEP/DROWSY</th>
<th>Multi-Mode Switch Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0X</td>
<td>Active</td>
</tr>
<tr>
<td>10</td>
<td>Sleep</td>
</tr>
<tr>
<td>11</td>
<td>Drowsy</td>
</tr>
</tbody>
</table>
Multimodal Power-Gated Pipeline Architecture

- Use different TM (Tri-Modal) switches for pipeline registers and for combinational logic gates so as to enable different power-gating modes for these circuit elements.
Experimental Results

• We designed and implemented a 16×16 pipelined carry save multiplier (CSM) using TSMC 0.18um CMOS
  – The circuit is divided into two pipeline stages
  – The 46-bit output of the first stage is latched into the pipeline registers (46 FF’s)
  – The first 16 bits out of these 46 bits make the first 16 bits of the product and are passed to the output directly
  – The last 30 bits are passed to the second stage making the last 16 bits of the product
Results, cont’d

- Four circuits, which are in the standby modes, are compared:
  - CMOS
  - Deep-sleep MTCMOS: all the cells (including FF’s and logic gates) are in the sleep mode
  - Drowsy MTCMOS: all the cells (including FF’s and gates) are in the drowsy mode
  - Data-retentive MTCMOS: Logic cells are in sleep mode and pipeline FF’s are in drowsy mode

<table>
<thead>
<tr>
<th>Circuit Type</th>
<th>Leakage (nA)</th>
<th>Ground-Bounce (mV)</th>
<th>Wakeup/Ready Latency (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMOS</td>
<td>63</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Deep-Sleep</td>
<td>0.10</td>
<td>473</td>
<td>19.32</td>
</tr>
<tr>
<td>Drowsy</td>
<td>48</td>
<td>143</td>
<td>4.83</td>
</tr>
<tr>
<td>Data-Retentive</td>
<td>2.85</td>
<td>441</td>
<td>19.32</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Circuit Type</th>
<th>Stage delay (ns)</th>
<th>Cell area (um²)</th>
<th>Wire length (um)</th>
<th>Wire length (um)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>$n_f=1$</td>
<td>$n_f=2$</td>
</tr>
<tr>
<td>CMOS</td>
<td>4.54</td>
<td>54720</td>
<td>54402.6</td>
<td>54402.6</td>
</tr>
<tr>
<td>MTCMOS</td>
<td>4.83</td>
<td>55710</td>
<td>59008.4</td>
<td>56077.2</td>
</tr>
<tr>
<td>% Increase</td>
<td>6.4</td>
<td>1.8</td>
<td>8.5</td>
<td>3.1</td>
</tr>
</tbody>
</table>

M. Pedram
Required Components of a Global Solution

- Better characterizations, models, and calculators
- Multi-corner or statistical optimization
- Augment design for runtime adaptability
- Dynamic control based on in-situ sensing
Architectures That Tolerate Uncertainty

- Due to variations and computational constraints, there will be significant uncertainty about the real state of a circuit as a result of an optimization decision.
- Need solution techniques that can learn from their mistakes and/or successes on similar problem instances encountered in the past to improve the quality of their decision making.
- Utilize partially-observable (semi-)Markov decision process model.
• To avoid over constraining a system in terms of its power dissipation or temperature, one must adopt online control and supervision solutions that change the system behavior on the fly so as to maximize performance without violating the constraints (or provide the required performance while minimizing energy consumption).
Idleness, State Transitions, and Power Savings

Thread packing (to cores) | Job packing (to machines)

Time scale

ns ms sec min hour day

Better active states

Components sleeping

More sleep states

Machines sleeping

Source: W-D. Weber

M. Pedram
Reality: Non-Energy-Proportional Servers

Power dissipation of Intel Xeon 5400 series server at high (2.3GHz) and low (2GHz) settings

Here $P_1$ denotes server power dissipation at 100% utilization, whereas $P_U$ is power at utilization level of $U$.

An energy proportional system will have an Energy Inefficiency Factor (EIF) of one at all utilization levels.

$$EIF_U = \frac{P_U}{P_1 U}$$
Minimum Energy CMP Design with Core Consolidation and DVFS

- Chip Multiprocessor system with $M$ identical cores
  - Per-core DVFS with $N$ (voltage-frequency) configurations
  - Local Queue
- Power Management Unit
- Global Queue
- Task Dispatcher
- Tasks
  - Expected execution time, $\tau$
  - Expected instructions per cycle

Objective: Minimize the total Energy consumption of cores in a multicore system

- Constraints: Minimum required throughput, IPSreq
- Variables:
  - Number of active cores – total number of cores: $M$
  - $v$-$f$ settings of cores – total settings: $J$
  - Task distribution– total number of tasks: $K$
A Hierarchical Solution

- Determine the number of ON cores
  - ON cores are in C0 when active or C2 (halt or sleep) when idle
- Determine operating frequency of ON cores
  - A feedback-based control method is adopted for DVFS setting
    - This is needed due to inherent uncertainty and variability of task characteristics
    - PI Controller: controller adjusts the v-f setting to match the required throughput based on the observed error
    - Feedback control loop determines a single optimum frequency for all cores and then the Quantizer would translate $f_{opt}$ to the available DVFS levels
- Find a feasible assignment of tasks to the ON cores
Simulation Results

- Comparison to a relatively energy-efficient baseline PM
  - It implements the same power reduction techniques as our method
  - It utilizes open loop DVFS, and does not support smart wakeup and shutdown
- The figure compares the frequency setting and throughput
  - The baseline PM always runs with all cores ON
  - The closed loop frequency is always lower for the same throughput
  - In the second 100ms, the baseline chooses lower frequency with four cores running while our proposed method uses only three cores
Changing Landscape: Smart Grid and Dynamic Pricing

Ameren offers power by the hour

Users can track price of electricity

BY MIKE HELENTHAL
Commercial-News

DANVILLE — Illinois customers with a day-trader’s attitude can save nearly 15 percent on their electricity bills under a new program offered by Ameren.

The two-year-old Power Smart Pricing program was created after Illinois legislators — at the urging of power industry watchdog Citizens Utility Board — required state utility companies offer pricing programs rewarding customers for “green” diligence.

Area Ameren customers received information on the program this month’s bill, but so far only 5,000 Illinois customers have signed up. Ameren subsidiaries serve some 2.4 million electric customers in Illinois and Missouri.

“I don’t think that many people know about it yet,” said Jim Chilsen, CUB communications manager. “It can be a big money-saver for the right customer and there are very specific things you should consider. It’s a good program, but it’s not for everyone.”

The program offers customers the ability to track in real time, via the Web, the day-ending regional commodity price of electricity. And as the rate fluctuates, participants can adjust their usage to avoid peak rates the following day.

“You don’t have to turn everything off and you don’t have to sit around in the dark,” said Stephanie Folk, a spokeswoman for CNT Energy.
Electrical Energy Storage Systems
Conclusion

• These are exciting times with many new opportunities and challenges due to planned upgrades to the Power Grid, introduction of renewable sources of energy, smart metering and dynamic energy pricing, people’s awareness of environmental issues, etc.

• A holistic, cross-layer approach to energy efficiency and robustness is needed, which spans
  – Application efficiency and energy management, micro-architecture and system design, storage and networks, resource management and scheduling
  – Synthesis and physical design
  – Library characterization and cell design