A Roadmap for Low-Power Design: Trends, Technology, Tools

Andrew B. Kahng
UCSD CSE and ECE Departments

abk@ucsd.edu
http://vlsicad.ucsd.edu
Agenda

- Devices / Circuits
- Design Technology
- Resilience
- Futures
- Low-Power Design

Adaptivity to Approximation
Agenda

TRENDS

LOW-POWER DESIGN

RESILIENCE TO APPROXIMATION

DESIGN TECHNOLOGY

FUTURES

DEVICES / CIRCUITS

LOW-POWER DESIGN
Trend 1: Race to the End of Roadmap

• $$$ for tech, design enablement: go big or go home
• Node pacing not slowing despite near-term “red bricks”
  • No EUV $\rightarrow$ multi-patterning
  • No Cu replacement $\rightarrow$ resistivity, rise of MOL / BEOL RC’s, variability
  • Reliability, layout restrictions $\rightarrow$ less benefit from node (20SOC “lost” like 45nm?)
  • Especially tough for early-adopter fabless
• Intrinsic mismatch of design-process time constants $\rightarrow$ margins!
  • Technology development, market definition, architectural design = O(years)
  • RTL-to-GDS implementation, reliability qualification = O(months)
  • Fab latency, cycles of yield learning, design re-spins = O(weeks)
  • Process tweaks, design ECOs = O(days)
  • Root cause of model-hardware miscorrelation, model guardbanding
• Paper to v1.0 SPICE models: 18 months $\rightarrow$ 12 months at N10
  • Will see how this works out…
Trend 2: Low Power Grand Challenge

- Drivers for semi growth share critical requirement: **LOW POWER**
  - Mobility
  - Big data, green datacenters, cloud
  - IoT
- Low-power design techniques increase design burden
- Added complexity of **system** + analysis + optimization
  - Multiple supply voltages
  - Multiple voltage domains
  - Extreme power, clock gating
  - DVFS
  - MTCMOS
  - Multi-Lgate
  - …
Power or Performance?

• Cannot escape basic “shape” of tradeoff
  • More power reduction (logic, Vt) available when freq ↓
  • ~Cubic relationship between power and frequency

• New designer mantras
  • “Highest performance at low power”
  • “Minimum V for any given throughput”

Energy vs. Delay: Near-Threshold Computing?

- Supply voltage at near-threshold region

>60X power reduction
6-8X energy reduction
Enables 3D integration

NTC Has Barriers …

- **Performance loss**
  - 45nm FO4 delay at NTC supply of 400mV vs. 1.1V: 10X slower

- **Increased performance variation**
  - Performance variation due to global process variation alone: ~30% at 1.1V, up to ~400% at 400mV

- **Increased functional failure**
  - Random dopant fluctuation, line edge roughness → positive feedback of device mismatch in SRAM

---

… Which Have Workarounds, but

- Performance loss
- Cluster-based architecture
- Device optimization
- Performance Variation
- Soft-edge clocking
- Body biasing
- Functional Failure
- Alternative SRAM Cells
- SRAM robustness analysis
- Reconfigurable cache designs
- *(FinFET changes the context!)*

---


---

Alternative 8 T SRAM cell, decoupling the read and write.

FIR filter with soft edge clocking compared to standard flip-flops (SFF); presented with and without useful skew.

Cluster-based architecture

Energy of SRAM topologies for 20-cycle L2 cache across voltages
Agenda

- Devices / Circuits
- Resilience
- Approximation
- TRENDS
- Low-Power Design
- Adaptivity
- Technology
Trend 3: High-Value Equivalent Scaling

- Device roadmap (FinFET, FDSOI) helps enormously
- Better electrostatic control
- Lower leakage current

![Graph showing ION current over years for BULK, FDSOI, and MGFET](image)

TSMC 28nm planar bulk [1]

ST 20nm FDSOI [2]

Intel 14nm FinFET [3]

FinFET vs. FDSOI for Low-Power Design

- **FinFET:** better subthreshold swing, DIBL
  
  \[\text{[Yeh 10]}\]
  
  → Performance less sensitive to Vdd
  → lower Vdd, less active power at same speed

- **FDSOI:** more V\text{t} control options
  
  \[\text{[Skotnicki10][PachaASX06][Biesemans]}\]
  
  → Metal gate stack changes work function, V\text{t}
  → Back-plane/gate doping
  → Back-gate biasing

<table>
<thead>
<tr>
<th></th>
<th>FinFET</th>
<th>FDSOI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Surface passivation</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>HK metal gate stack</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>BG biasing</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>DIBL</td>
<td>70mV/V</td>
<td>80mV/V</td>
</tr>
</tbody>
</table>
**Much Wider Operating Voltage Range**

- Supply voltage scaling is key low-power technique
- Enabled with FinFET, UTBB FDSOI
- Complexity explosion: modes, corners in timing closure

<table>
<thead>
<tr>
<th>Technology</th>
<th>28nm UTBB FDSOI</th>
<th>28nm LP Bulk</th>
<th>32nm Bulk</th>
<th>22nm Trigate</th>
<th>28nm UTBB FDSOI</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VDD operating range</strong></td>
<td>0.6V-1.2V</td>
<td>0.34V-1V</td>
<td>0.28V-1.2V</td>
<td>0.28V-1.1V</td>
<td>0.39V-1.3V</td>
</tr>
<tr>
<td><strong>Max measured Frequency</strong></td>
<td><a href="mailto:2.6GHz@0.6V">2.6GHz@0.6V</a></td>
<td>587MHz@1V</td>
<td><a href="mailto:915MHz@1.2V">915MHz@1.2V</a></td>
<td><a href="mailto:2.5GHz@1.1V">2.5GHz@1.1V</a></td>
<td><a href="mailto:460MHz@0.397V">460MHz@0.397V</a></td>
</tr>
<tr>
<td><strong>Total power consumption</strong></td>
<td>na</td>
<td>113mW@1V</td>
<td>400mW@1V</td>
<td>227mW@1V</td>
<td>370mW@1V</td>
</tr>
<tr>
<td><strong>Peak energy efficiency</strong></td>
<td>na</td>
<td>na</td>
<td>170pJ/cycle @0.45V</td>
<td>585GOPS/W @260mV</td>
<td>62pJ/op @0.53V</td>
</tr>
</tbody>
</table>

R. Wilson, et al., “A 460MHz at 397mV, 2.6GHz at 1.3V, 32b VLIW DSP, embedding FMAX tracking”, ISSCC 2014, pp. 452-454.
Trends 1, 2, 3 → Design Closure Nightmare

**OLD**
- 1 mode
- Setup-hold
- SI
- Cw only
- NLDM

**Modeling**
- LVF; BEOL/MOL
- Lib groups

**Analysis**
- MIS; SHPR; SI;
- PBA; -dynamic

**Signoff**
- Yield vs. Slack; MCMMP; TBC;
- AVS; Corner vs. Flat Margins

**Design Synthesis/Opt**
- Architecture; RTL;
- SP&R; Timing/Noise;
- ECOs

**Technology, Design Enablement**
- SPICE; ITF; Library/IP;
- Testchips

**Physical Implementation**

**NEW**
- MCMM
- AVS
- Power & Clock gating
- Multiple voltage domains
- DVFS
- MTCMOS
- Multi-Lgate
- Cell-POCV / LVF
- Dynamic IR
- Wide/exploding corners, corner reduction, cross-corners
  (BEOL Cw, Ccw, RCw, temp, VDD)
- Flat margin selection
- Noise closure
- Aging/AVS

Trends 1, 2, 3 → Design Closure Nightmare
Agenda

LOW-POWER DESIGN

DESIGN TECHNOLOGY

FUTURES

TRENDS

APPROXIMATION

RESILIENCE

ADAPTIVITY

DEVICES / CIRCUITS

LOW-POWER DESIGN
Alphabet Soup: DVFS, DCVS, AVS, PVS, SVS…

Dynamic Voltage / Frequency Scaling
Operating voltage and frequency change according to workload

Adaptive Voltage Scaling
Operating voltage can vary to meet the performance requirement
(also used to compensate process variation)

Multiple Voltage Islands (Multi-VDD)
Operating voltage is different, but, voltage has the fixed value
# Low-Power Design Roadmap in the ITRS

**ITRS: power and energy = the grand challenge for semiconductor industry**

→ low-power design roadmap (2011)

<table>
<thead>
<tr>
<th>Design Technology Improvement</th>
<th>Year</th>
<th>Dynamic</th>
<th>Static</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low Power Physical Libraries</td>
<td>Before 2011</td>
<td>1.50</td>
<td>1.50</td>
<td>Optimizing transistor size, layout style and cell topology for the standard-cell library</td>
</tr>
<tr>
<td>Back Biasing</td>
<td></td>
<td>1.00</td>
<td>1.35</td>
<td>Biasing wells of devices independently of the sources to shift the threshold voltage</td>
</tr>
<tr>
<td>Adaptive Body Biasing (ABB)</td>
<td></td>
<td>1.20</td>
<td>2.00</td>
<td>Delivering a positive or negative voltage below a transistor to reduce leakage</td>
</tr>
<tr>
<td>Power Gating</td>
<td></td>
<td>0.90</td>
<td>10.00</td>
<td>Turning off the power supplies to idle blocks for leakage reduction</td>
</tr>
<tr>
<td>Dynamic Voltage/Frequency Scaling (DVFS)</td>
<td></td>
<td>1.50</td>
<td>1.00</td>
<td>Dynamic management of supply voltage and operating frequency for power reduction</td>
</tr>
<tr>
<td>Multilevel Cache Architecture</td>
<td></td>
<td>1.00</td>
<td>1.20</td>
<td>Reduce amount of off-chip memory accesses for performance improvement and power reduction</td>
</tr>
<tr>
<td>Hardware Multithreading</td>
<td></td>
<td>1.00</td>
<td>1.30</td>
<td>Using multithreads to improve hardware utilization with leakage reduction</td>
</tr>
<tr>
<td>Hardware Virtualization</td>
<td></td>
<td>1.00</td>
<td>1.20</td>
<td>Using one physical server to support multiple guest operating systems simultaneously</td>
</tr>
<tr>
<td>Superscalar Architecture</td>
<td></td>
<td>1.00</td>
<td>2.00</td>
<td>Parallel instruction issuing and executing for performance improvement and power reduction</td>
</tr>
<tr>
<td>Symmetric Multiple Processing (SMP)</td>
<td></td>
<td>1.50</td>
<td>1.00</td>
<td>Lowering the frequency by using multiple processors and the parallel programming</td>
</tr>
<tr>
<td>Software Virtual Prototype</td>
<td>2011</td>
<td>1.23</td>
<td>1.20</td>
<td>Allow the programmer to develop software prior to silicon</td>
</tr>
<tr>
<td>Frequency Islands</td>
<td>2013</td>
<td>1.26</td>
<td>1.00</td>
<td>Designing blocks that operate at different frequencies</td>
</tr>
<tr>
<td>Near-Threshold Computing</td>
<td>2015</td>
<td>1.23</td>
<td>0.80</td>
<td>Lowering Vdd to 400 - 500 mV</td>
</tr>
<tr>
<td>Hardware/Software Co-Partitioning</td>
<td>2017</td>
<td>1.18</td>
<td>1.00</td>
<td>Hardware/software partitioning at the behavioral level based on power</td>
</tr>
<tr>
<td>Heterogeneous Parallel Processing (AMP)</td>
<td>2019</td>
<td>1.18</td>
<td>1.00</td>
<td>Using multiple types of processors in a parallel computing architecture</td>
</tr>
<tr>
<td>Many Core Software Development Tools</td>
<td>2021</td>
<td>1.20</td>
<td>1.00</td>
<td>Using multiple types of processors in a parallel computing architecture</td>
</tr>
<tr>
<td>Power-Aware Software</td>
<td>2023</td>
<td>1.21</td>
<td>1.00</td>
<td>Developing software using power consumption as a parameter</td>
</tr>
<tr>
<td>Asynchronous Design</td>
<td>2025</td>
<td>1.21</td>
<td>1.00</td>
<td>Total Non-clock driven design</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>4.66</strong></td>
<td><strong>0.96</strong></td>
<td></td>
</tr>
</tbody>
</table>
## Value of Low-Power Design Technology

<table>
<thead>
<tr>
<th>Design Technology Improvement</th>
<th>Year</th>
<th>Improvements</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Virtual Prototype</td>
<td>2011</td>
<td>1.23</td>
<td>1.20 Allow the programmer to develop software prior to silicon</td>
</tr>
<tr>
<td>Frequency Islands</td>
<td>2013</td>
<td>1.26</td>
<td>1.00 Designing blocks that operate at different frequencies</td>
</tr>
<tr>
<td>Extreme Power Gating</td>
<td>2015</td>
<td>0.90</td>
<td>1.00 Shutting down applications (Dark Silicon)</td>
</tr>
<tr>
<td>Hardware/Software Co-Partitioning</td>
<td>2017</td>
<td>1.18</td>
<td>1.00 Hardware/software partitioning at the behavioral level based on power</td>
</tr>
<tr>
<td>Heterogeneous Parallel Processing (AMP)</td>
<td>2019</td>
<td>1.18</td>
<td>1.00 Using multiple types of processors in a parallel computing architecture</td>
</tr>
<tr>
<td>Many Core Software Development Tools</td>
<td>2021</td>
<td>1.20</td>
<td>1.00 Using multiple types of processors in a parallel computing architecture</td>
</tr>
<tr>
<td>Power-Aware Software</td>
<td>2023</td>
<td>1.21</td>
<td>1.00 Developing software using power consumption as a parameter</td>
</tr>
<tr>
<td>Near-Threshold Computing</td>
<td>2025</td>
<td>1.23</td>
<td>0.80 Lowering Vdd to 400 - 500 mV</td>
</tr>
<tr>
<td>Asynchronous Design</td>
<td>2027</td>
<td>1.21</td>
<td>1.00 Total Non-clock driven design</td>
</tr>
</tbody>
</table>

**Total**: 3.47 0.96

**SOC consumer portable chip in 2028:**
- 2.5 billion logic gates
- Low-power DT reduces power from 48.8W to 9.1W
Roadmap: Big Gaps Ahead

• ITRS: power of mobile SOC-CP driver keeps increasing…
• … even if envisioned low-power innovations are developed and deployed on time
ADDED in 2013 Low-Power DT Roadmap

Approximate Computing
• Variable-accuracy computing (e.g., flexibly from 64b ↔ 16b)

4D Computing
• Reconfiguration on the fly

Adaptivity
• Recapture overdesign from wearout, variation margins

Power Gating Replacement
• HVT device as power switch hits headroom, area wall → ?

Extreme Heterogeneity
• “Coprocessor-dominated architectures”
  • (pervasive heterogeneity; energy-efficiency from specialization; HW accelerators)

Extreme Power Gating
• Reaching the limits of shut-off

Signoff At Typical
• Use adaptivity to recover margin, overdesign
Agenda

- Resilience
- Adaptivity to Approximation
- Devices / Circuits
- Design Technology
- Low-power Design
Resilience = “Long-Term Challenge” in ITRS

- Resilience = system product’s ability to mitigate variability, reliability phenomena
- Error detection and repair mechanisms
- Alternative guardbanding mechanisms for different abstractions: stochastic, approximate, …
- “Cross-layer resilience” = recent buzz-phrase
- Costs, benefits often hazy, difficult to quantify
Example Step: Minimize Cost of Resilience

- Additional circuits $\Rightarrow$ area and power penalties
- Recovery from errors $\Rightarrow$ throughput degradation
- Large hold margin $\Rightarrow$ short-path padding cost
- Want benefits (e.g., energy) to maximally outweigh costs

<table>
<thead>
<tr>
<th></th>
<th>Razor</th>
<th>Razor-Lite</th>
<th>TIMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power penalty</td>
<td>30% [Das08]</td>
<td>$\sim$0% [Kim13]</td>
<td>100% [Choudhury09]</td>
</tr>
<tr>
<td>#recovery cycles</td>
<td>5 [Wan09]</td>
<td>11 [Kim13]</td>
<td>0 [Choudhury09]</td>
</tr>
</tbody>
</table>

![Razor Circuit](image1)

![Razor-Lite Circuit](image2)

![TIMBER Circuit](image3)
Tradeoff: Resilience Cost vs. Datapath Cost

#Razor FFs (resilience cost)

Power/area of fanin circuits

Tradeoff

Can minimize total energy via this tradeoff
Selective-Endpoint Optimization (SEOpt)

• Optimize fanin cone of an endpoint w/ tighter constraints  
  ⇒ Allows replacement of Razor FF w/ normal FF

• Pick endpoints based on heuristic sensitivity functions

Candidate Sensitivity Functions

\[ SF1 = |\text{slack}(p)| \]
\[ SF2 = |\text{slack}(p)| \times \text{numcri}(p) \]
\[ SF3 = |\text{slack}(p)| \times \frac{\text{numcri}(p)}{\text{numtotal}(p)} \]
\[ SF4 = |\text{slack}(p)| \times \sum_{\text{celfanin}(p)} \text{Pwr}(c) \]
\[ SF5 = \sum_{\text{celfanin}(p)} |\text{slack}(c)| \times \text{Pwr}(c) \]

\[ p \quad = \text{negative slack endpoint} \]
\[ c \quad = \text{cells within fanin cone} \]
\[ \text{numcri} \quad = \text{number of negative-slack cells} \]
Clock Skew Optimization (SkewOpt)

- Increase slacks on timing-critical and/or frequently-exercised paths
  1. Generate sequential graph
  2. Find cycle of paths with minimum total weight
     → adjust clock latencies
     → contract the cycle into one vertex
  3. Iterate Step 2 until all endpoints are optimized

\[ W' = \text{average weight on cycle} \]

\[ W_{pq} = \frac{\text{Slack}_{pq}}{1 + \beta \times TG(p, q)} \]

Configuring the clock skews requires adjusting the clock latencies to optimize the setup slacks of the critical paths. The weight of a cycle is calculated as the average weight on the cycle, which is used to adjust the clock skews and minimize the setup time for the critical paths.
Overall Optimization Flow

- Iteratively optimize with **SEOpt** and **SkewOpt**

- **Initial placement** (all FFs = error-tolerant FFs)
- **OR-tree insertion**
- **Margin insertion on $K$ paths based on sensitivity function**
- **Replace error-tolerant FFs w/ normal FFs**
- **Activity aware clock skew optimization**
- **Energy < min energy?**
- **Save current solution**

![Graph with initial design, after SEOpt, and after SkewOpt]

[GLSVLSI14; ACM TODAES, to appear]
Benefit of Low-Cost Resilience

- Reference flows
  - Pure-margin (PM): conventional method w/ only margin insertion
  - Brute-force (BF): use error-tolerant FFs for timing-critical endpoints
- Proposed method (CO) achieves up to 21% energy reduction compared to reference methods
- Resilience benefits increase with larger process variation

![Energy Comparison Graph]

Small/medium/large margin $\Rightarrow 1\sigma/2\sigma/3\sigma$ for SS corner
Increased Benefit of Resilience with AVS

- Adaptive voltage scaling allows a lower supply voltage for resilient designs, thus reduced power.
- Proposed method trades off between timing-error penalty vs. reduced power at a lower supply voltage.
- Proposed method achieves an average of 17% energy reduction compared to pure-margin designs.

⇒ Resilience benefits increase in the context of AVS strategy.

Technology: foundry 28nm
Agenda

- Resilience
- Adaptivity to Approximation
- Future Trends
- Devices/Circuits
- Design Technology
- Low-Power Design
A Story of Adaptivity…

Step 1. Design team signs off chip at 1.4 GHz with worst-case (slow silicon) timing corner

Step 2. Chip comes back from fab (typical silicon) and runs at 1.8GHz

[Management is unhappy: pessimism in signoff wasted area, power and design time]

[Cross-functional tiger teams are formed to work on (A) signoff corner pessimism (margin) reduction and (B) improved model-hardware correlation]

Step 3. Scale down supply voltage so chip runs at 1.4 GHz with as little power as possible (= adaptive voltage scaling)
Adaptive Voltage Scaling Approaches

### Power

#### Open Loop AVS
- Pre-characterize LUT [Martin02]

#### Closed-Loop AVS
- Post-silicon characterization
- Generic monitor
- Design dependent replica
- In-situ monitor

#### Application Driven AVS
- Application-driven $V_{dd}$ and frequency scaling [Lin09]

#### Error Detection System
- Process-aware AVS
- Post-silicon characterization [Tschanz03]

- Process and temperature-aware AVS
- Generic on-chip monitor [Burd00]

- Design-dependent monitor
  - [Elgebaly07, Drake08, Chan12]

- In-situ performance monitor
  - Measure actual critical paths [Hartman06, Fick10]

- Error detection and correction system
  - $V_{dd}$ scaling until error occurs [Das06, Tschanz10]
“Process-aware Voltage Scaling” (PVS) [ICCAD-2012]

• Monitor design considerations
  • Critical path can be difficult to identify (IP from 3rd party)
  • Multiple modes/voltages: F_{max} calibration requires long test time

• UCSD: generic, tunable monitor
  • RO-based monitor with V_{min_ro} > V_{min} for any data path at any process condition (generic \Leftrightarrow overdesign)
  • Monitor is tunable based on F_{max} of sample chips to recover design margin (calibrate once)

• Abstracts voltage scaling property instead of matching critical path
  • Keys: (1) PMOS-, NMOS-dominated paths determine V_{min}; (2) tune ROs with series resistance (pass gates)
Voltage Scaling Basic Concepts

- **Process distance**: process-induced frequency shift relative to target frequency
- **Scaling rate**: frequency shift ($\Delta f$) per unit voltage difference ($\Delta V$)
- $V_{\text{min}}$: Minimum $V_{dd}$ to meet target frequency
  - Calculated from process distance and scaling rate

![Diagram showing process distance and scaling rate](image_url)
Process-Aware Voltage Scaling Concept

- Use $V_{\text{min}}$ of ring-oscillator (RO) as a reference
- Design ROs with worst-case voltage scaling properties $\rightarrow$ an arbitrary circuit will meet target frequency at $V_{\text{min_ro}}$

**Process corner A**

Max. $V_{\text{min}}$ of ROs $>$ Max. $V_{\text{min}}$ of paths

EDPS-2015 Keynote 150424
Experimental Results on Tunability

Aggressive config.
→ $V_{\text{min\_est}} < V_{\text{min\_chip}}$
→ Some chips will fail

Optimized config.
• Increase % high resistance passgates
• $V_{\text{min\_est}} \approx V_{\text{min\_chip}}$

Default config.
• Low resistance passgates
• Guardband for worst-case
• $V_{\text{min\_est}} > V_{\text{min\_chip}}$
• 13mV margin

---Tunable ROs
— Normal ROs
Experimental Results on Tunability

**Aggressive config.**

$V_{\text{min\_est}} < V_{\text{min\_chip}}$

$\rightarrow$ Some chips will fail

**Optimized config.**

- Increase % high resistance passgates
- $V_{\text{min\_est}} \approx V_{\text{min\_chip}}$

**Default config.**

- Low resistance passgates
- Guardband for worst-case
- $V_{\text{min\_est}} > V_{\text{min\_chip}}$
- 13mV margin

**Benefits of tunability**

- Compensate for difference between model vs. silicon
- Recover margin when variation is reduced due to improved process

**Experimental Results on Tunability**
Error-Resilience

• Error-freeness guarantees cost {margin, power, $$$}
• Unnecessary in some contexts
  • Machine learning, data mining, search
  • Signal processing: image, video, speech
  • Optimization
• Paradigms for error-resilience
  • Approximate computing
  • Stochastic computing
  • Probabilistic computing
  • …
• In what contexts, with what knowledge?

What If We Knew… (switching activity, workload)

**Error-Tolerant Design**

*CPU, heal thyself...*

Errors are detected and corrected with redundancy technique

---

**Problem:**

- Many paths have near-critical slack → *wall of (critical) slack*
- Scaling beyond the critical operating point causes massive errors that cannot be corrected

---

**Reshape slack distribution for gracefully increasing error rate**

Frequently-exercised paths: upsize cells
Rarely-exercised paths: downsize cells

---

Scale voltage further
Recovery-Driven Design for Error-Tolerance [TCAD12]

- Minimize power for a target error rate
- Slack redistribution based on functional information

**Voltage Scaling**
reduce voltage until the error rate exceeds a target

**Path Optimization**
optimize frequently exercised, negative slack paths

**Power Reduction**
reduce power without affecting error rate

![Graph showing power consumption and voltage scaling with 22% power savings]
What If We Knew … (accuracy requirements)

Problem:
• Accuracy requirement can change during runtime → benefits of approximation could be reduced

Adapt to changing requirements with runtime accuracy configuration

[DAC 2012] “accuracy-configurable approximate adder”

What is the square root of 10?
“a little more than three”
“3.162278…”
Approximation could be faster and more powerful

lower power consumption higher accuracy
Accuracy-Configurable Approximate Adder

- Power reduction when accuracy requirement varying

- Accuracy configuration with pipelined adder

\[\text{Accuracy} = \text{Avg.} \left(1 - \frac{|\text{result} - \text{reference}|}{\text{reference}} \right)\]

Average 30% power savings vs. no accuracy configuration
Stochastic Computing

• Conventional computation circuits rely on parallel bits aligned by clock → timing errors can be fatal (e.g., if occurring in MSB)

• Stochastic Circuit (SC) paradigm replaces parallel bits with serial bit-stream → resilient to voltage scaling

Example: \[ Z = \frac{1}{4} + \frac{1}{2} \cdot X_1 \cdot X_2 \]
Agenda

RESILIENCE

ADAPTIVITY TO APPROXIMATION

TRENDS

DEVICES / CIRCUITS

DESIGN TECHNOLOGY

LOW-POWER DESIGN

FUTURES

DESIGN TECHNOLOGY

LOW-POWER DESIGN
Better Optimizations

- Better MCMM gate sizing
  - Gate sizing with single timing view can induce timing violations in other timing view ⇒ multi-corner-multi-mode (MCMM) optimization is needed

- Better FinFET fin discreteness-aware optimization
  - E.g., PlaceOpt to comprehend and avoid change in diffusion height induced by different number of fins

- Better design-technology co-optimization
  - E.g., BEOL stack optimizations, FEOL-BEOL gear ratio and library co-optimization for PPA, …
System Thinking: Design Synergy

• Low power implementation for the modern system on chip (SOC) requires a holistic and concurrent approach which includes collaboration between:

<table>
<thead>
<tr>
<th>Section</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>System level design</td>
<td></td>
</tr>
<tr>
<td>Architectural design</td>
<td></td>
</tr>
<tr>
<td>Software optimization and SW-HW co-design</td>
<td></td>
</tr>
<tr>
<td>Power aware RTL implementation/synthesis (front end design)</td>
<td></td>
</tr>
<tr>
<td>Physical design (chip/block level) (back end design)</td>
<td></td>
</tr>
<tr>
<td>IP design: Circuit design, Physical implementation of the IP</td>
<td></td>
</tr>
<tr>
<td>Process selection ((Bulk CMOS, SOI, FinFET) and device definition (nmos, pmos etc). Process optimization and DFM.</td>
<td></td>
</tr>
<tr>
<td>Adaptive design (on die sensors, process aware voltage scaling, process, power and temperature monitors). Power models (upf/cpf)</td>
<td></td>
</tr>
<tr>
<td>Power and thermal verification and modeling</td>
<td></td>
</tr>
<tr>
<td>Silicon characterization, power models validation, silicon to model correlation</td>
<td></td>
</tr>
</tbody>
</table>

S. Dobre, UCSD lecture, 2015.
Example: Power and Performance Meters

- Power and performance meters:
  - Hardware–software solution
  - Measures performance and power in real time for different sub-systems integrated in the system on chip
- Provides feedback to the system for:
  - Power management
  - Thermal management
  - Workload optimization

S. Dobre, UCSD lecture, 2015.
What If We Knew…(scenarios, duty cycles)

- DVFS allows adaptation to workloads, operating conditions
- DVFS processor operates at multiple power/performance points with different lifetimes
- Lifetime energy can be different in each scenario ($R \times X$)

Different duty cycle ($R$)

$e.g.,$ talk mode

- Low Performance ($1 - R$)
- High Performance ($R$)

$e.g.,$ standby mode

Different frequency scaling ($X$)

$$X = \frac{\text{clock frequency of high-perf mode}}{\text{clock frequency of low-perf mode}}$$
Minimize **lifetime energy** based on modes, duty cycles

- CTL module has **12%** energy savings through replication
- Processor-level: selective replication gives **12% total** energy savings with 10% area overhead
Lower Power With 3DIC

- Power = key value proposition for 3DICs (shorter / wider connections)
- Recent work (DAC-2015): 3DIC power reduction at 28nm foundry FDSOI libraries and estimation of 3D power benefits only from 2D implementations
- Power benefit with 3D varies with testcases and implementation styles
  - Percentage delta benefit ranges from -5.1% (i.e., power increases in 3DIC) to 16.0%

3DIC Implementation and Modeling flow

3DIC % delta power benefit relative to 2DIC Implementation
Clock (well-known, but still on table)

- Clocking = ~30-40% of total power
- SOC complexity: 1000+ clock domains
- Clock architecture: Many frequencies, local dividers to reduce frequency
- SP&R: planning, placement, buffering of “top-level”: CGCs, MUXes, dividers
- CTS: MCMM skew reduction, skew variation reduction (high voltage is wire-dominated but low voltage is gate-dominated, large hold buffering costs, …)
- Routing: NDRs, routing with distributed drivers, long common paths (reduces wirelength, driver sizes and number of clock buffers)

Clock (Emerging)

- Reduced-swing, half-swing clocks
  - Reduces clock power when $V_{tn}$ (resp., $V_{tp}$) is less (resp. greater) than $\frac{1}{2} V_{dd}$
  - On-off characteristics of NMOS (resp. PMOS) remain unchanged

**Half-Swing Clock** [1]

- Globally Asynchronous, Locally Synchronous (GALS); more asynchrony

Sources:
A Few Nutshells…

- **AVS** mandatory to address variation at low voltage
- **NTC** in use as “ultra-low voltage mode”; standard design enablement works out of box in FinFET nodes (e.g., 0.46V – 1.25V)
- **FinFET transition** has large benefits → takes some pressure off of low-power design in near term (?)
- **Next-generation optimizers** needed for DVFS/MCMM, wide voltage corners, multi-patterning, FinFET discreteness, …
- **Next-generation analysis tools/flows** (thermal, dynamic IR, reliability wearout, stress = “next loops to close”) to further squeeze design margin
- **Active power** regains focus → clock power reduction, active leakage cost, design for min V at given throughput
- **System-level low-power design** continues as both (complexity) challenge and (cross-layer) opportunity
- **3DIC** will turn the corner for power-performance envelope
- **Approximate computing** will take longer to turn the corner …
THANK YOU!