Online Timing Variation Tolerance for Digital Integrated Circuits

Guihai Yan
Assistant Professor, Ph.D.
http://carch.ict.ac.cn/~yan

State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences (ICT, CAS)
Acknowledgement

- Prof. Xiaowei Li, Yinhe Han
- National Basic Research Program of China (973 Program)
- High-Tech Research and Development Program of China (863 Program)
- National Natural Science Foundation of China (NSFC)
Holistic perspective

Software

Applications

Architectural Level

Silicon

Hardware

Noise, Temperature instability, Process variation, Radiation, Defects……
Technology Trend

The end of Dennard Scaling

<table>
<thead>
<tr>
<th>Device or Circuit Parameter</th>
<th>Scaling Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device dimension tox, L, W</td>
<td>1/k</td>
</tr>
<tr>
<td>Doping concentration Na</td>
<td>k</td>
</tr>
<tr>
<td>Voltage V</td>
<td>1/k</td>
</tr>
<tr>
<td>Current I</td>
<td>1/k</td>
</tr>
<tr>
<td>Capacitance eA/t</td>
<td>1/k</td>
</tr>
<tr>
<td>Delay time per circuit VC/I</td>
<td>1/k</td>
</tr>
<tr>
<td>Power dissipation per circuit VI</td>
<td>1/k^2</td>
</tr>
<tr>
<td>Power density VI/A</td>
<td>1</td>
</tr>
</tbody>
</table>

Ramifications of timing variation

- Reliability
  - Delay fault, timing error, timing emergency
- Lifetime
  - Aging delay (NBTI, TDDB)
- Power efficiency
  - Timing margin
- Yield
  - Frequency binning
Pathology of timing variation (T)

- **PVT variation**
  - Dynamic: Voltage & Temperature fluctuations
  - Static: Process variation
- **Aging degradation**
  - NBTI, PBTI
  - TDDB
- **Soft errors (in non-regular logics)**
  - SEU & SET
T=f(Process variation)

- Sub-wavelength Lithography
  - “What you get is not what you want”
  - Systematic
- Random dopant fluctuations
  - $V_{th}$ variation
  - Random

P variation is time-independent, “DC component”
$T = f(Temperature\ variation)$

- Application-specific
- Slow-varying
  - Milliseconds
  - Typical thermal constant: 2ms

**T variation is slow-varying, “Low-frequency components”**

<table>
<thead>
<tr>
<th>benchmark</th>
<th>category</th>
<th>steady-state temperature (°C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>gzip</td>
<td>SPECint</td>
<td>70</td>
</tr>
<tr>
<td>mcf</td>
<td>SPECint</td>
<td>59</td>
</tr>
<tr>
<td>parser</td>
<td>SPECint</td>
<td>67</td>
</tr>
<tr>
<td>twolf</td>
<td>SPECint</td>
<td>67</td>
</tr>
<tr>
<td>mesa</td>
<td>SPECfp</td>
<td>65</td>
</tr>
<tr>
<td>swim</td>
<td>SPECfp</td>
<td>62</td>
</tr>
<tr>
<td>lucas</td>
<td>SPECfp</td>
<td>63</td>
</tr>
<tr>
<td>sixtrack</td>
<td>SPECfp</td>
<td>71</td>
</tr>
</tbody>
</table>
T = f(Voltage variation)

- Fast-changing
  - Inductive noise
    - a.k.a. L(di/dt) problem
  - IR-drop

Why it is harder to keep a constant voltage level?

**Example:**
Power budget: 100W,
Working voltage: 1V,
Current: 100A,
To keep voltage fluctuation between ±5%,
$R_{PDN} < 0.5$ mOhm

V variation is fast-changing, “High-frequency components”
Aging mechanisms

- NBTI (PMOS)
- PBTI (NMOS)
- TDDB
\[ T = f(\text{Soft errors}) \]

- **SEU (Single Event Upset)**
  - Unintentional bit-flip in storage cells
- **SET (Single Event Transient)**
  - Transient voltage pulse propagating in combinational logics

```
SET
SEU
……
……
```

**Clock edge**

- Fault free: 0
- Faulty: 1
- Faulty: 1

Soft error induced glitch
How to “cure” timing variation?

- Build PVT variation-mild environment
  - Architectural level solution
  - Hypervisor software support
  - Low cost

- Detect delay variation, then eliminate it with timing adaptation
  - Circuit level solution
  - Oblivious to software
  - Generally applicable
Focus on the essential **Timing** issue

Process variation

Voltage variation

Temperature variation

Timing variation

$$\Delta D(t) = D(t) - D_{spec} = \alpha P + \beta (V(t) - V_{spec}) + \gamma (T(t) - T_{spec})$$

Not Necessarily aggregated, but can cancel off each others in some cases. Hence, “Complementary”. 
Some terms

- **Timing emergency (TE):** delay exceeds a safe timing threshold
- **Emergency level (EL)**
  - “Density” of TE
  - Define: \( EL = \# \text{ of TE per 100 millions cycles} \)

<table>
<thead>
<tr>
<th>Voltage</th>
<th>Temperature</th>
<th>Process</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Violent</strong></td>
<td><img src="image1.png" alt="Waveform" /></td>
<td><img src="image2.png" alt="Temp" /></td>
</tr>
<tr>
<td><strong>Mild</strong></td>
<td><img src="image4.png" alt="Waveform" /></td>
<td><img src="image5.png" alt="Temp" /></td>
</tr>
</tbody>
</table>

---

14
How PVT Variations Complement each other?

Observation in time domain

- **Excessive headroom**
  - Core1: T. Mild, V. Mild
  - Core2: T. Violent, V. Violent
  - Time
  - Delay
  - Threshold

- **Emergency**
  - Core1: T. Mild, V. Violent
  - Core2: T. Violent, V. Mild
  - Time
  - Delay

What if exchange the threads on Core1 and Core2?

- Mild + Violent

T. Mild, V. Mild
- Large margin, low EL

T. Violent, V. Violent
- Little margin, High EL
Frequency domain analysis

Migrate threads = “Graft” V component
Frequency domain analysis (cont.)

- Relative frequency spectrum deviations on 2GHz quad-core processor.
  - P: 0-100Hz, T: 100Hz-1MHz, V: 1MHz-250MHz.
- Potential
  - Core3 and Core4 are mild
- Strategy
  - exchange threads on Core1 and Core4, Core2 and Core3
TEA-TM Architecture

Timing Emergency Aware + Thread Migration

TEA-TM

- Reduce 30% timing emergency
- Improving 70% fairness
Let’s dive into the circuits

We need a new model for timing variation!
Stability Violation

What's “Stability Violation”? Signal transitions occur in Stable Period.

Stable Period vs. Variable Period

What's “Stability Violation”?
Signal transitions occur in Stable Period.

Stable Period vs. Variable Period
In what situations would SVs occur?

- Delay defects (introduced in manufacturing processes)
- Aging (Wearout) induced performance degradation

Thus, delay faults caused stability violation does not differ too much from “setup time violation”

• But, can soft errors be modeled by SV? **YES!**
Detect stability violations for errors

Circuit Implementation

Self-test Deployment

Logic

Output Compactor

Upstream flip-flops

Downstream flip-flops

Timing Non-critical Signals

Sensor 1

Sensor n

VDD

CLK

M1

M2

M3

M4

M5

M6

M7

M8

S1

S2

S3

S4

S5

S6

S7

S8

GND

VDD

GND

Aging alarm

Stability Checker

Stability Checker

Stability Checker

Output Latch

Aging alarm

Circuit Implementation.
Error detection overhead

- **Implementation**
  - SVFD fully protected SPARC-T1 FPU
  - Using 65nm PTM, Hspice Simulation

---

**Silicon area comparison**

<table>
<thead>
<tr>
<th>SCAN</th>
<th>DMR</th>
<th>SEFF</th>
<th>LOWCOST</th>
<th>ARSC</th>
<th>CWSPFF</th>
<th>SVFD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1.8</td>
<td>1.6</td>
<td>1.4</td>
<td>1.6</td>
<td>1.8</td>
</tr>
</tbody>
</table>

**Power comparison**

<table>
<thead>
<tr>
<th>SCAN</th>
<th>DMR</th>
<th>SEFF</th>
<th>LOWCOST</th>
<th>ARSC</th>
<th>SVFD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1.8</td>
<td>1.6</td>
<td>1.4</td>
<td>1.6</td>
</tr>
</tbody>
</table>
Other applications

- **Dynamic margin reduction**
  - MicroFix: an application to DVFS
- **Aging tolerance**
  - ReviveNet: Fine-grained aging delay tolerance
MicroFix operational principle

(a) Traditional DVFS
- Reduce frequency from F to F'
- Reduce voltage from V to V'
- Increase frequency from F' to F
- Increase voltage from V' to V

(b) MicroFix enhanced DVFS
- Reduce frequency from F to F'
- Reduce voltage from V to V'
- Increase frequency from F' to F
- Increase voltage from V' to V

Tighten runtime voltage margin!
Dynamic margin reduction

- Feedback-control loop
- No error recovery support engaged
Case study results

- Apply to a FPU
- 32nm PTM models

TH = 0.2 ~ 0.3 is an optimal choice!
Efficiency Improvement: 35% EDP
Aging Tolerance

- The chance for aging adaptation
  - We have chance to “act before it’s too late”

Aging delay over time

- Tolerant aging delay
- Cycle period
- Alarm threshold

Aging adaptation

Extended lifetime

Aging tolerant design

Aging resilient design
Nudge for timing margin

- Dynamic time borrowing
  - Path-grained, NOT stage-grained
Trail-based adaptation

- Trial-till-success

Sensor

# of test case: $2^4 - 1$

Round-Robin Trial Adaptation (K)
01. The $K^{th}$ Agent receives an aging emergency
02. FOR each adaptation state candidate
03. Conduct a trial adaptation
04. IF the emergency is eliminated
05. THEN break (*Adaptation succeeded!*)
06. ELSE
07. Recover this trial adaptation
08. IF all the adaptation states have been reached
09. THEN break (*Adaptation failed!*)
10. END FOR
Implementation

- False-alarm filter
- Sharing filters to reduce overhead
Conclusion

- Dynamic timing variation is increasingly critical
- Online timing variation detection and tolerance is a promising approach to dynamic variation
  - Detect in-field transient errors
  - Improve power efficiency
  - Enhance lifetime reliability
- Holistic solution can be more cost-effective


Thanks for your attention!

For more information, please visit http://carch.ict.ac.cn/~yan