Resilient Microprocessor Design for Dynamic Variation Tolerance

Keith A. Bowman
Circuit Research Lab, Intel

keith.a.bowman@intel.com

Acknowledgements:
James Tschanz, Shih-Lien Lu, Paolo Aseron, Muhammad Khellah, Arijit Raychowdhury, Bibiche Geuskens, Carlos Tokunaga, Chris Wilkerson, Tanay Karnik, & Vivek De

June 8, 2010
UPC Seminar
Problem Statement:

- Variability is one of the primary challenges in the semiconductor industry
- Adversely impacts performance, power, yield, reliability, & time-to-market

Focus Area:

1) Resilient design for dynamic variation tolerance
Outline

• Review of Static Variations
• Resilient Microprocessor Core
• Error-Detection & Recovery Circuits
• Measurement Results
• Conclusion
• Future Research
Static Variations

- Technology trends amplify microprocessor performance & power variability

- Static Variations:
  - Within-die impacts $F_{\text{MAX}}$ mean & leakage median
  - Die-to-die impacts $F_{\text{MAX}}$ & leakage variances

- Adaptive circuits mitigate the impact of static variations on performance & power
Outline

- Review of Static Variations
- Resilient Microprocessor Core
- Error-Detection & Recovery Circuits
- Measurement Results
- Conclusion
- Future Research
Dynamic Variations

**$V_{CC}$ Droop**

![Graph showing $V_{CC}$ droop over time](image)

**Temperature**

![Heat map showing temperature variations](image)

**Aging**

![Graph showing path delay change over stress time](image)
Impact of Dynamic Variations on Conventional Design

Guardbands required to ensure correct operation within the presence of dynamic variations
Resilient Design

- Operate clock frequency ($F_{CLK}$) based on nominal conditions
- Resilient circuits detect and correct timing errors due to infrequent dynamic variations
- Throughput and energy benefits result from mitigating guardbands
Microprocessor Features

- 32-Bit Synthesized Core
  - Open-source RISC-style design
  - 7-stage in-order pipeline
  - Modified with resilient and adaptive circuits

- 16KB Instruction & Data Caches

- PLL-Based Clock Generator
Resilient & Adaptive Circuits

• Error Detection:
  1) Error-detection sequential (EDS)
  2) Tunable replica circuit (TRC)

• Error Control Unit (ECU) for Recovery:
  1) Instruction replay at $\frac{1}{2}F_{\text{CLK}}$
  2) Multiple issue instruction replay at $F_{\text{CLK}}$

• Adaptive Clock Controller:
  ➢ Adjust $F_{\text{CLK}}$ based on recovery cycles to maximize performance during persistent variations
Microprocessor Core Overview

refclk \rightarrow \text{PLL} \rightarrow \frac{1}{2}F_{CLK} \rightarrow \text{Duty Cycle}

16KB I$ \rightarrow \text{PC} \rightarrow \text{IF} \rightarrow \text{DE} \rightarrow \text{RA} \rightarrow \text{EX} \rightarrow \text{MEM} \rightarrow \text{X} \rightarrow \text{WB} \rightarrow \text{RF}

16KB D$
Microprocessor Core Overview

Errors are pipelined to write-back (WB) stage to invalidate erroneous instructions
Microprocessor Core Overview

- Error-control unit (ECU) enables recovery
Adaptive clock control enables dynamic $F_{CLK}$ change
Outline

• Review of Static Variations
• Resilient Microprocessor Core
• Error-Detection & Recovery Circuits
• Measurement Results
• Conclusion
• Future Research
Error-Detection Circuits

1) Error-Detection Sequential (EDS)

2) Tunable Replica Circuit (TRC)
Error-Detection Sequential (EDS)

- Detect timing error within error-detection window

Error-Detection Sequential (EDS)

Trade-off: Max-Delay ($t_{MAX}$) vs Min-Delay ($t_{MIN}$)

- Min-delay penalty increases by error-detection window
- Clock duty-cycle control required to maintain constant high-phase delay during low and high $F_{CLK}$
EDS Circuits

Double Sampling (Razor I) [1]-[3]

Transition Detector with Time Borrowing (TDTB) [5]

Razor II [4]

Double Sampling with Time Borrowing (DSTB) [5]

Error-Detection Sequential (EDS)

Implementation

- Contains additional scan-enabled latch for testing
  - mode=0: EDS
  - mode=1: FF
Error-Detection Sequential (EDS)

- EDS assigned during synthesis convergence
- EDS embedded in critical paths
- EDS inserted in 12% of core sequentials
Error-Detection Circuits

1) Error-Detection Sequential (EDS)

2) Tunable Replica Circuit (TRC)
Tunable Replica Circuit (TRC)

- TRC monitors critical path delays
- Non-intrusive design

Tunable Replica Circuit (TRC)

- TRC tuned to track critical paths per pipeline stage
- TRC must always fail if any critical path fails
- TRC error initiates pipeline error recovery
# EDS & TRC Overheads

<table>
<thead>
<tr>
<th>Circuit Blocks</th>
<th>EDS</th>
<th>TRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Detection &amp; Accumulation Area Overhead</td>
<td>2.2%</td>
<td>0.8%</td>
</tr>
<tr>
<td>ECU &amp; Clock Control Area Overhead</td>
<td>1.4%</td>
<td>1.4%</td>
</tr>
<tr>
<td>Min-Delay Buffer Insertion Area Overhead</td>
<td>0.2%</td>
<td>-</td>
</tr>
<tr>
<td>Total Area Overhead</td>
<td>3.8%</td>
<td>2.2%</td>
</tr>
<tr>
<td>Total Power Overhead (\text{(iso-F}<em>{\text{CLK}}, \text{iso-V}</em>{\text{CC}}))</td>
<td>0.9%</td>
<td>0.6%</td>
</tr>
</tbody>
</table>
Error-Recovery Circuits

1) Instruction Replay at $\frac{1}{2}F_{CLK}$
   - Clock divider generates $\frac{1}{2}F_{CLK}$ without PLL re-lock
   - Clock high-phase delay remains unchanged

2) Multiple Issue Instruction Replay at $F_{CLK}$
   - Does not require clock control
   - Issue *replica instructions* to setup pipeline registers
   - Last issue is a valid instruction
### Multiple Issue (MI) Replay

<table>
<thead>
<tr>
<th>Pipeline Stage</th>
<th>IF</th>
<th>DE</th>
<th>RA</th>
<th>EX</th>
<th>MEM</th>
<th>X</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>I1</td>
<td>I1</td>
<td>I1</td>
<td>I1</td>
<td>I1</td>
<td>I1</td>
<td>I2</td>
</tr>
<tr>
<td></td>
<td>I2</td>
<td>I2</td>
<td>I2</td>
<td>I2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>I3</td>
<td>I3</td>
<td>I3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>I4</td>
<td>I4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>I5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Clock Cycle**

1. Error occurs on instruction I2 in EX pipeline stage

**I2 Error in EX Stage**

**1) Error occurs on instruction I2 in EX pipeline stage**
## Multiple Issue (MI) Replay

<table>
<thead>
<tr>
<th>Pipeline Stage</th>
<th>Clock Cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1  2  3  4  5  6  7  8  9  10 11 12</td>
</tr>
<tr>
<td>IF</td>
<td>I1 I2 I3 I4 I5 I6 I7 I8</td>
</tr>
<tr>
<td>DE</td>
<td>I1 I2 I3 I4 I5 I6 I7 I8</td>
</tr>
<tr>
<td>RA</td>
<td>I1 I2 I3 I4 I5 I6 I7 I8</td>
</tr>
<tr>
<td>EX</td>
<td>I1 I2 I3 I4 I5 I6 I7 I8</td>
</tr>
<tr>
<td>MEM</td>
<td>I1 I2 I3 I4 I5 I6 I7 I8</td>
</tr>
<tr>
<td>X</td>
<td>I1 I2 I3 I4 I5 I6 I7 I8</td>
</tr>
<tr>
<td>WB</td>
<td>I1 I2 I3 I4 I5 I6 I7 I8</td>
</tr>
</tbody>
</table>

**I2 Error in EX Stage**

**Invalid: Does Not Affect Architecture State**

2) Invalidate errant instruction and subsequent instructions
### Multiple Issue (MI) Replay

#### Clock Cycle

<table>
<thead>
<tr>
<th>Pipeline Stage</th>
<th>IF</th>
<th>DE</th>
<th>RA</th>
<th>EX</th>
<th>MEM</th>
<th>X</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I1</td>
<td>I1</td>
<td>I1</td>
<td>I1</td>
<td>I1</td>
<td>I1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>I2</td>
<td>I2</td>
<td>I2</td>
<td>I2</td>
<td>I2</td>
<td>I2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>I3</td>
<td>I3</td>
<td>I3</td>
<td>I3</td>
<td>I3</td>
<td>I3</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>I5</td>
<td>I5</td>
<td>I5</td>
<td>I5</td>
<td>I5</td>
<td>I5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>I6</td>
<td>I6</td>
<td>I6</td>
<td>I6</td>
<td>I6</td>
<td>I6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>I7</td>
<td>I7</td>
<td>I7</td>
<td>I7</td>
<td>I7</td>
<td>I7</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>I8</td>
<td>I8</td>
<td>I8</td>
<td>I8</td>
<td>I8</td>
<td>I8</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>10</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>11</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>12</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

- **I2 Error in EX Stage**
- **Invalid: Does Not Affect Architecture State**

#### 3) Flush pipeline
## Multiple Issue (MI) Replay

### Clock Cycle

<table>
<thead>
<tr>
<th>Clock Cycle</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
</tr>
</thead>
</table>

### Pipeline Stage

<table>
<thead>
<tr>
<th>IF</th>
<th>DE</th>
<th>RA</th>
<th>EX</th>
<th>MEM</th>
<th>X</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

- **Invalid**: Clocks 17 and 18
- **Pipeline Flush**: Clocks 17, 18, 19, and 20
Multiple Issue (MI) Replay

4) Issue errant instruction N times: N-1 issues setup pipeline register values; Nth issue may change architecture state.
4) Issue errant instruction N times: N-1 issues setup pipeline register values; Nth issue may change architecture state
Answer: An outline is shown with the following sections:

- Review of Static Variations
- Resilient Microprocessor Core
- Error-Detection & Recovery Circuits
- Measurement Results
- Conclusion
- Future Research
## Characteristics & Measurements

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology</td>
<td>45nm CMOS</td>
</tr>
<tr>
<td>Die Area</td>
<td>13.64 mm²</td>
</tr>
<tr>
<td>Core Area</td>
<td>0.39 mm²</td>
</tr>
<tr>
<td>Core $F_{MAX}$</td>
<td>1.45GHz at 1.0V</td>
</tr>
<tr>
<td>Core Power</td>
<td>135mW at 1.0V</td>
</tr>
</tbody>
</table>

- Programs compiled from C code
- Caches and settings loaded via JTAG scan
Measured Throughput (TP) vs $F_{CLK}$

- 16% TP gain with EDS

![Graph showing Measured Throughput (TP) vs Clock Frequency ($F_{CLK}$). The graph illustrates the normalized throughput and recovery cycles (%) in relation to clock frequency. The Resilient: EDS Max TP and Conventional Max TP are highlighted, with a significant gain at $V_{CC} = 1.0V$ and 10% $V_{CC}$ droop. The figure also indicates that EDS leads to a 16% TP gain.]
Measured Throughput (TP) vs $F_{CLK}$

- **Normalized Throughput**

- **Recovery Cycles (%)**

- **Clock Frequency (GHz)**

- **Conventional Max TP**

- **Resilient: EDS Max TP**

- **Resilient: TRC Max TP**

- **V_{CC}=1.0V 10% V_{CC} Droop**

- **16% TP gain with EDS**

- **12% TP gain with TRC**
EDS exploits path-activation differences across programs

- EDS throughput benefits range from 15% to 20%
- TRC throughput benefits remain at 12%
TRC TP gains exceed EDS TP gains at low $V_{CC}$

Error-detection window determines EDS & TRC TP benefits

Min-delay limits EDS error-detection window
Measured Average Recovery Cycles

Replay at $\frac{1}{2}F_{CLK}$ & Multiple Issue (MI) Replay at $F_{CLK}$

- Multiple issue replay:
  - ~46% reduction in average recovery cycles
  - Does not require clock control
TRC & EDS resilient circuits enable:

- 41% throughput gain at equal energy
- 22% energy reduction at equal throughput
Adaptive Clock Control Measurement

- Adaptive $F_{CLK}$ compensates for persistent variations
- Maintains optimum recovery rate for maximum throughput
- Core operates through PLL lock – Jitter errors corrected
Outline

• Review of Static Variations
• Resilient Microprocessor Core
• Error-Detection & Recovery Circuits
• Measurement Results
• Conclusion
• Future Research
Conclusion

- Microprocessor core employs resilient circuits to mitigate dynamic variation guardbands

- Error-detection circuits:
  - Error-detection sequential (EDS)
  - Tunable replica circuit (TRC)

- Error-recovery circuits:
  - Instruction replay at $\frac{1}{2}F_{CLK}$
  - Multiple issue instruction replay at $F_{CLK}$

- Silicon measurements indicate:
  - 41% throughput gain at iso-energy
  - 22% energy reduction at iso-throughput

- Resilient & adaptive circuits enable the microprocessor to adjust to operating variations for maximum efficiency
References


Outline

- Review of Static Variations
- Resilient Microprocessor Core
- Error-Detection & Recovery Circuits
- Measurement Results
- Conclusion
- Future Research
Wide Dynamic Operation Range

- $V_{\text{MAX}}$: Limited by reliability or power constraints
- $V_{\text{MIN}}$: Limited by circuit failures
Resilient design expands the operating range.
# Wide Range of Platform Segments

<table>
<thead>
<tr>
<th>Platform</th>
<th>Power</th>
<th>Perf.</th>
<th>Cores</th>
<th>Thermal</th>
<th>Ambient</th>
<th>RAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>High</td>
<td>Very High</td>
<td>Active</td>
<td>Controlled</td>
<td>Very High</td>
<td></td>
</tr>
<tr>
<td>Desktop</td>
<td>Med</td>
<td>High</td>
<td>Fan</td>
<td>Controlled</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>Mobile</td>
<td>Low</td>
<td>Med</td>
<td>Fan or Fan-less</td>
<td>Uncontrolled</td>
<td>Med</td>
<td></td>
</tr>
<tr>
<td>MID</td>
<td>Very Low</td>
<td>Low</td>
<td>SOC</td>
<td>Fan-less</td>
<td>Uncontrolled</td>
<td>Low</td>
</tr>
</tbody>
</table>

- Few designs must support many segments
- Resilient design to satisfy various platform targets
Future Research

- Resilient Design for Wide Operation Range:
  - Explore error-detection & recovery capabilities throughout system hierarchy
  - Optimize resilient features at the system level across unique platform segments
  - Opportunities & challenges for validation & test
  - Opportunities for CAD
Q&A