Viper: Virtual Pipelines for Enhanced Reliability

Andrea Pellegrini, Joseph Greathouse, Valeria Bertacco

University of Michigan,
Advanced Computer Architecture Laboratory
Reliability Challenges with CMOS Scaling
Reliability Challenges with CMOS Scaling

**Manufacturing defects**
That escape testing
Reliability Challenges with CMOS Scaling

**Manufacturing defects**
That escape testing

**Transient faults**
Natural radiation, noise…

[Diagram of CMOS transistor with labels for source, gate, and drain]
Reliability Challenges with CMOS Scaling

**Manufacturing defects**
That escape testing

**Transient faults**
Natural radiation, noise…

**Age-related wearout**
Electromigration & gate-oxide breakdown
Reliability Challenges with CMOS Scaling

**Manufacturing defects**
That escape testing

**Transient faults**
Natural radiation, noise…

**Age-related wearout**
Electromigration & gate-oxide breakdown

Intel Cougar Point chipsets wearing out over time
[Estimated cost of 700M$ 2011]
Reliability Challenges with CMOS Scaling

**Manufacturing defects**
That escape testing

**Transient faults**
Natural radiation, noise...

**Age-related wearout**
Electromigration & gate-oxide breakdown

“Future technologies will make transistors less and less reliable” [Borkar, 2005]

Intel Cougar Point chipsets wearing out over time
[Estimated cost of 700M$ 2011]
Impact of Faults on Traditional CMPs
Impact of Faults on Traditional CMPs
Impact of Faults on Traditional CMPs
Impact of Faults on Traditional CMPs
Impact of Faults on Traditional CMPs
Faults Effects on CMP Throughput

- CMP system w/ 2 billion transistors fitting 128 cores – no caches
  - 15M transistors/core, similar to Intel Atom
Faults Effects on CMP Throughput

- CMP system w/ 2 billion transistors fitting 128 cores – no caches
  - 15M transistors/core, similar to Intel Atom
Limitations of Current $\mu$-Architectures
Limitations of Current μ-Architectures

- Single point of failures
Limitations of Current $\mu$-Architectures

Single point of failures
Limitations of Current $\mu$-Architectures

- Single point of failures
Limitations of Current μ-Architectures

Single point of failures

Diagram:
- Fetch
- Decoder
- Integer ALU
- LSQ
- Floating Point
- Back End

Single point of failures
Limitations of Current $\mu$-Architectures

- Single point of failures
- Rigidly connected pipeline stages
Limitations of Current $\mu$-Architectures

- Single point of failures
- Rigidly connected pipeline stages
Limitations of Current $\mu$-Architectures

- Single point of failures
- Rigidly connected pipeline stages
Limitations of Current $\mu$-Architectures

- Single point of failures
- Rigidly connected pipeline stages
Limitations of Current $\mu$-Architectures

- Single point of failures
- Rigidly connected pipeline stages
Limitations of Current $\mu$-Architectures

- Single point of failures
- Rigidly connected pipeline stages
- Centralized control logic
Limitations of Current $\mu$-Architectures

- Single point of failures
- Rigidly connected pipeline stages
- Centralized control logic
Limitations of Current μ-Architectures

- Single point of failures
- Rigidly connected pipeline stages
- Centralized control logic

![Diagram of Fetch, Decoder, Integer ALU, Floating Point LSQ, Back End]
Can We Overcome These Limitations?

- Single point of failures
- Rigidly connected pipeline stages
- Centralized control logic
Can We Overcome These Limitations?

- Single point of failures
  - ✓ Arrays of redundant hardware units
- Rigidly connected pipeline stages
- Centralized control logic
Can We Overcome These Limitations?

- Single point of failures
  - ✓ Arrays of redundant hardware units

- Rigidly connected pipeline stages
  - ✓ Loosely connected hardware modules

- Centralized control logic
Can We Overcome These Limitations?

- Single point of failures
  - ✓ Arrays of redundant hardware units

- Rigidly connected pipeline stages
  - ✓ Loosely connected hardware modules

- Centralized control logic
  - ✓ Decentralized and redundant controls
Can We Overcome These Limitations?

- Single point of failures
  - ✓ Arrays of redundant hardware units
- Rigidly connected pipeline stages
  - ✓ Loosely connected hardware modules
- Centralized control logic
  - ✓ Decentralized and redundant controls

Service-oriented µ-architecture to tackle all three issues:
Can We Overcome These Limitations?

- Single point of failures
  - ✓ Arrays of redundant hardware units

- Rigidly connected pipeline stages
  - ✓ Loosely connected hardware modules

- Centralized control logic
  - ✓ Decentralized and redundant controls

Service-oriented μ-architecture to tackle all three issues:

**Virtual Pipelines for Enhanced Reliability**
Service-Oriented $\mu$-Architecture
Service-Oriented μ-Architecture

Renew Driving License:
Service-Oriented $\mu$-Architecture

Renew Driving License:
1. Check in
Service-Oriented \( \mu \)-Architecture

Renew Driving License:
- 1. Check in
- 2. Vision test
Service-Oriented $\mu$-Architecture

Renew Driving License:
- 1. Check in
- 2. Vision test
- 3. Take picture
Service-Oriented µ-Architecture

Renew Driving License:
- 1. Check in
- 2. Vision test
- 3. Take picture
- 4. Pay fee
Service-Oriented μ-Architecture

Renew Driving License:
- 1. Check in
- 2. Vision test
- 3. Take picture
- 4. Pay fee
- 5. Get license
Service-Oriented $\mu$-Architecture
Service-Oriented $\mu$-Architecture
Service-Oriented μ-Architecture
Service-Oriented $\mu$-Architecture
Service-Oriented $\mu$-Architecture
Service-Oriented $\mu$-Architecture
Service-Oriented μ-Architecture
Service-Oriented $\mu$-Architecture
Service-Oriented μ-Architecture
Service-Oriented $\mu$-Architecture
Service-Oriented μ-Architecture
Service-Oriented μ-Architecture
Service-Oriented $\mu$-Architecture
Service-Oriented μ-Architecture
Service-Oriented μ-Architecture
Service-Oriented $\mu$-Architecture
Viper - Overview

- HW units can **perform services** for instructions
Viper - Overview

- HW units can perform services for instructions

- Bundles are instruction sequences terminating with a control instruction (JMP)

4013c3: add %al, [%ebx]
4013c5: div cl
4013c8: jmp 40140a
Viper - Overview

- HW units can perform services for instructions

- Bundles are instruction sequences terminating with a control instruction (JMP)

- A Virtual Pipeline is the ordered sequence of HW units that can complete the instructions in a bundle

```assembly
4013c3: add %al, [%ebx]
4013c5: div cl
4013c8: jmp 40140a
```
Viper - Overview

- HW units can **perform services** for instructions

- **Bundles** are instruction sequences terminating with a control instruction (JMP)

- A **Virtual Pipeline** is the ordered sequence of HW units that can complete the instructions in a bundle

- **Bundle Scheduling Unit** allows instructions to use and be scheduled on the available HW units

```
4013c3: add %al, [%ebx]
4013c5: div cl
4013c8: jmp 40140a
```
Viper - Overview

- **HW units** can perform services for instructions

- **Bundles** are instruction sequences terminating with a control instruction (JMP)

- A **Virtual Pipeline** is the ordered sequence of HW units that can complete the instructions in a bundle

- **Bundle Scheduling Unit** allows instructions to use and be scheduled on the available HW units

- An **ISA** is defined as the set of services needed by its instructions

```
4013c3: add %al, [%ebx]
4013c5: div cl
4013c8: jmp 40140a
```
Viper Hardware Organization
Viper Hardware Organization

Sea of redundant HW modules
Viper Hardware Organization

Sea of redundant HW modules
Viper Hardware Organization

Sea of redundant HW modules

Single point of failure

___

___
Viper Hardware Organization

- Sea of redundant HW modules
- Single point of failure
- Homogenous module interconnect
Viper Hardware Organization

- Sea of redundant HW modules
  - Single point of failure
- Homogenous module interconnect
  - Rigidly connected hardware modules
Viper Hardware Organization

- Sea of redundant HW modules
- Single point of failure
- Homogenous module interconnect
- Rigidly connected hardware modules
- Bundle Scheduling Units (BSU) to schedule HW modules
Viper Hardware Organization

- Sea of redundant HW modules
- Single point of failure
- Homogenous module interconnect
- Rigidly connected hardware modules
- Bundle Scheduling Units (BSU) to schedule HW modules
- Centralized control logic
Viper’s Execution Model

1. Building Virtual Pipelines
2. Inter-Module Data Dependencies
3. Handling Program Mispredictions
4. Handling Precise Exceptions
1. Building Virtual Pipelines

2. Inter-Cluster Dependencies

3. Mispredictions

4. Precise Exceptions

Fetch instruction
Register access
Execute
Write back/Commit
1. Building Virtual Pipelines

2. Inter-Cluster Dependencies

3. Mispredictions

4. Precise Exceptions

4013c3: add %al, [%ebx]

Fetch instruction
Register access
Execute
Write back/Commit
1. Building Virtual Pipelines

- Fetch instruction
- Register access
- Execute
- Write back/Commit

Instructions:
- `4013c3: add %al, [%ebx]`
- `4013c8: or %al, %bl`
1. Building Virtual Pipelines

BSU – Bundle Scheduling Unit

<table>
<thead>
<tr>
<th>4013c3: add %al, [%ebx]</th>
</tr>
</thead>
<tbody>
<tr>
<td>4013c8: or %al, %bl</td>
</tr>
</tbody>
</table>

Fetch instruction
Register access
Execute
Write back/Commit
1. Building Virtual Pipelines

**Fetch instruction**
**Register access**
**Execute**
**Write back/Commit**

**1. Building Virtual Pipelines**

**2. Inter-Cluster Dependencies**

**3. Mispredictions**

**4. Precise Exceptions**

---

**BSU – Bundle Scheduling Unit**

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**4013c3:** add %al, [%ebx]

**4013c8:** or %al, %bl
1. Building Virtual Pipelines

**BSU – Bundle Scheduling Unit**

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fetch instruction
Register access
Execute
Write back/Commit

**4013c3:**
```
add %al, [%ebx]
```

**4013c8:**
```
or %al, %bl
```
1. Building Virtual Pipelines

**BSU – Bundle Scheduling Unit**

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td></td>
<td>F0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fetch instruction
Register access
Execute
Write back/Commit

4013c3: `add %al, [%ebx]`
4013c8: `or %al, %bl`
1. Building Virtual Pipelines

BSU – Bundle Scheduling Unit

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td>F0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fetch instruction
Register access
Execute
Write back/Commit

4013c3: add %al, [%ebx]
4013c8: or %al, %bl
1. Building Virtual Pipelines

**BSU – Bundle Scheduling Unit**

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td></td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Fetch instruction**
**Register access**
**Execute**
**Write back/Commit**

```plaintext
4013c3: add %al, [%ebx]
4013c8: or %al, %bl
```
1. Building Virtual Pipelines

**BSU – Bundle Scheduling Unit**

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td></td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Fetch instruction**

**Register access**

**Execute**

**Write back/Commit**

**Instructions:**

- 4013c3: add %al, [%ebx]
- 4013c8: or %al, %bl
1. Building Virtual Pipelines

**BSU – Bundle Scheduling Unit**

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **4013c3**: `add %al, [%ebx]`
- **4013c8**: `or %al, %bl`

**Fetch instruction**

**Register access**

**Execute**

**Write back/Commit**
1. Building Virtual Pipelines

### BSU – Bundle Scheduling Unit

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fetch instruction
Register access
Execute
Write back/Commit

<table>
<thead>
<tr>
<th>4013c3: add %al, [%ebx]</th>
</tr>
</thead>
<tbody>
<tr>
<td>4013c8: or %al, %bl</td>
</tr>
</tbody>
</table>

1. Building Virtual Pipelines
2. Inter-Cluster Dependencies
3. Mispredictions
4. Precise Exceptions
# 1. Building Virtual Pipelines

---

### BSU – Bundle Scheduling Unit

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
</tr>
<tr>
<td>2</td>
<td>4013c8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Fetch instruction**
- **Register access**
- **Execute**
- **Write back/Commit**

---

1. Building Virtual Pipelines
2. Inter-Cluster Dependencies
3. Mispredictions
4. Precise Exceptions

---

### Machine Code

- **4013c3**: `add %al, [%ebx]`
- **4013c8**: `or %al, %bl`

---

- **Fetch instruction**
- **Register access**
- **Execute**
- **Write back/Commit**
1. Building Virtual Pipelines

**BSU – Bundle Scheduling Unit**

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
</tr>
<tr>
<td>2</td>
<td>4013c8</td>
<td></td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Operational Code**

4013c3: `add %al, [%ebx]`

4013c8: `or %al, %bl`

**Fetch instruction**

**Register access**

**Execute**

**Write back/Commit**
Viper’s Distributed Control Logic

- HW units can negotiate their services with BSU through:
  - Queues
  - Proposal broadcasts
  - Tokens
- Resource starvation avoided if the oldest bundle is served first
Viper’s Distributed Control Logic

- HW units can negotiate their services with BSU through:
  - Queues
  - Proposal broadcasts
  - Tokens

- Resource starvation avoided if the oldest bundle is served first
Clusters might need operands generated by others

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
</tr>
<tr>
<td>2</td>
<td>4013c8</td>
<td></td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4013c3: add %al, [%ebx]

4013c8: or %al, %bl
2. Inter-Cluster Dependencies

Clusters might need operands generated by others

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
</tr>
<tr>
<td>2</td>
<td>4013c8</td>
<td></td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2. Inter-Cluster Dependencies

Clusters might need operands generated by others

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4013c8</td>
<td></td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4013c3: add %al, [%ebx]
4013c8: or %al %bl

1. Building Virtual Pipelines
2. Inter-Cluster Dependencies
3. Mispredictions
4. Precise Exceptions
2. Inter-Cluster Dependencies

Clusters might need operands generated by others

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4013c8</td>
<td></td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Additional storage required (768 bits/BSU for x86)

4013c3: add %al, [%ebx]

4013c8: or %al, %bl
### 2. Inter-Cluster Dependencies

Clusters might need operands generated by others

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>10</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>4013c8</td>
<td></td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Input Tags**

- F0
- R0
- E0
- W0

**Output Tags**

- F1
- R1
- E1
- W1

### Clusters might need operands generated by others

1. **Building Virtual Pipelines**
2. **Inter-Cluster Dependencies**
3. **Mispredictions**
4. **Precise Exceptions**

**Example Code**

```assembly
4013c3: add %al, [%ebx]
```

```assembly
4013c8: or %al, %bl
```
2. Inter-Cluster Dependencies

Clusters might need operands generated by others

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>10</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>4013c8</td>
<td></td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4013c3: add %al, [%ebx]

4013c8: or %al, %bl

1. Building Virtual Pipelines
2. Inter-Cluster Dependencies
3. Mispredictions
4. Precise Exceptions
2. Inter-Cluster Dependencies

Clusters might need operands generated by others

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
<th>Input Tags</th>
<th>Output Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>10</td>
<td>13</td>
<td>5  6  10</td>
</tr>
<tr>
<td>2</td>
<td>4013c8</td>
<td></td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4013c3: add %al, [%ebx]

4013c8: or %al, %bl
Clusters might need operands generated by others

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>10</td>
<td>13</td>
<td>5</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>4013c8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4013c3: add %al, [%ebx]

4013c8: or %al, %bl
2. Inter-Cluster Dependencies

Clusters might need operands generated by others

### Input Tags

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
</tr>
<tr>
<td>2</td>
<td>4013c8</td>
<td></td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Output Tags

<table>
<thead>
<tr>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>6</td>
<td>10</td>
<td>13</td>
<td>5</td>
<td>6</td>
<td>10</td>
</tr>
</tbody>
</table>

#### 4013c3: add %al,[%ebx]

#### 4013c8: or %al,%bl
2. Inter-Cluster Dependencies

Clusters might need operands generated by others

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>4013c8</td>
<td></td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Tag creation is serialized

4013c3: add %al, [%ebx]

4013c8: or %al, %bl
Clusters might need operands generated by others

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>4013c8</td>
<td></td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4013c3: add %al, [%ebx]

4013c8: or %al, %bl
### 2. Inter-Cluster Dependencies

Clusters might need operands generated by others

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4013c8</td>
<td></td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
<td>13</td>
<td>5</td>
<td>6</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Input Tags**
- RA
- RB
- RC
- RD

**Output Tags**
- RA
- RB
- RC
- RD

#### Example Codes

- **4013c3**: `add %al, [%ebx]`
- **4013c8**: `or %al, %bl`
2. Inter-Cluster Dependencies

Clusters might need operands generated by others

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>4013c8</td>
<td></td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
<td>13</td>
<td>5</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4013c3:add %al, [%ebx]

4013c8:or %al, %bl
### Clusters might need operands generated by others

#### Input Tags

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>10</td>
<td>13</td>
<td>5</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>4013c8</td>
<td></td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
<td>13</td>
<td>5</td>
<td>6</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Output Tags

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>10</td>
<td>13</td>
<td>5</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>4013c8</td>
<td></td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
<td>13</td>
<td>5</td>
<td>6</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Example Operations

- **4013c3**: `add %al, [%ebx]`
- **4013c8**: `or %al, %bl`

#### Related Topics

1. Building Virtual Pipelines
2. Inter-Cluster Dependencies
3. Mispredictions
4. Precise Exceptions
Clusters might need operands generated by others

1. Building Virtual Pipelines
2. Inter-Cluster Dependencies
3. Mispredictions
4. Precise Exceptions

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>10</td>
<td>13</td>
<td>5</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>4013c8</td>
<td></td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
<td>13</td>
<td>5</td>
<td>6</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4013c3: `add %al, [%ebx]`

4013c8: `or %al, %bl`
### 2. Inter-Cluster Dependencies

Clusters might need operands generated by others

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
<th>Input Tags</th>
<th>Output Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
<td>1</td>
<td>5 6 10 13</td>
</tr>
<tr>
<td>2</td>
<td>4013c8</td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
<td>13</td>
<td>5</td>
<td>6 10</td>
</tr>
<tr>
<td>3</td>
<td>4013c3</td>
<td>4013c8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4013c3: `add %al, [%ebx]`

4013c8: `or %al, %bl`

1. Building Virtual Pipelines
2. Inter-Cluster Dependencies
3. Mispredictions
4. Precise Exceptions
2. Inter-Cluster Dependencies

Clusters might need operands generated by others

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>4013c8</td>
<td></td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
<td>13</td>
<td>5</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4013c3: add %al, [%ebx]

4013c8: or %al, %bl

1. Building Virtual Pipelines
2. Inter-Cluster Dependencies
3. Mispredictions
4. Precise Exceptions
Clusters might need operands generated by others

1. Building Virtual Pipelines
2. Inter-Cluster Dependencies
3. Mispredictions
4. Precise Exceptions

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>10</td>
<td>13</td>
<td>5</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>4013c8</td>
<td></td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
<td>13</td>
<td>5</td>
<td>6</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4013c3: add %al, [%ebx]

4013c8: or %al, %bl

Input Tags
Output Tags

Clusters might need operands generated by others
## 2. Inter-Cluster Dependencies

Clusters might need operands generated by others

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>10</td>
<td>13</td>
<td>5</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>4013c8</td>
<td></td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
<td>13</td>
<td>5</td>
<td>6</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Possible optimization: Tags based bundle ID
- Does not require serialization
- Much smaller storage needed
3. Handling Mispredictions

1. Building Virtual Pipelines
2. Inter-Cluster Dependencies
3. Mispredictions
4. Precise Exceptions

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>Next</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>4013c0</td>
<td>4013c3</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>4013c8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3. Handling Mispredictions

1. Building Virtual Pipelines
2. Inter-Cluster Dependencies
3. Mispredictions
4. Precise Exceptions

**Code Snippet:**

```
4013c0: jmp 4013eb
4013c3: add %al, [%ebx]
4013c8: or %al, %bl
```

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>Next</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>4013c0</td>
<td>4013c3</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>4013c8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3. Handling Mispredictions

1. Building Virtual Pipelines
2. Inter-Cluster Dependencies
3. Mispredictions
4. Precise Exceptions

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>Next</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>4013c0</td>
<td>4013c3</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>4013c8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4013c0: jmp 4013eb
4013c3: add %al, [%ebx]
4013c8: or %al, %bl
3. Handling Mispredictions

1. Building Virtual Pipelines
2. Inter-Cluster Dependencies
3. Mispredictions
4. Precise Exceptions

### Table: Pipeline Stages

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>Next</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>4013c0</td>
<td>4013c3</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>4013c8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Diagram: Pipeline Stages

- BSU 0: PC 4013c0
- BSU 1: PC 4013c3
- BSU 2: PC 4013c8

### Code Listings:

- `4013c0: jmp 4013eb`
- `4013c3: add %al, [%ebx]`
- `4013c8: or %al, %bl`
3. Handling Mispredictions

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>Next</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>4013c0</td>
<td>4013c3</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>4013c8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fetch instruction
Register access
Execute
Write back/Commit

4013c0: jmp 4013eb
4013c3: add %al, [%ebx]
4013c8: or %al, %bl
3. Handling Mispredictions

1. Building Virtual Pipelines
2. Inter-Cluster Dependencies
3. Mispredictions
4. Precise Exceptions

### BSU Table

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>Next</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>4013c0</td>
<td>4013c3</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>4013c8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Branch mispredicted! Correct NPC: 4013eb**

- **Register access**
- **Execute**
- **Write back/Commit**

### Code Examples

- `4013c0: jmp 4013eb`
- `4013c3: add %al, [%ebx]`
- `4013c8: or %al, %bl`
3. Handling Mispredictions

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>Next</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>4013c0</td>
<td>4013eb</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>4013c8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Branch mispredicted! Correct NPC: 4013eb

Fetch instruction
Register access
Execute
Write back/Commit
3. Handling Mispredictions

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>Next</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>4013c0</td>
<td>4013eb</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>4013c8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- 4013c0: jmp 4013eb
- 4013c3: add %al, [%ebx]
- 4013c8: or %al, %bl
3. Handling Mispredictions

1. Building Virtual Pipelines
2. Inter-Cluster Dependencies
3. Mispredictions
4. Precise Exceptions

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>Next</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>4013c0</td>
<td>4013eb</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>4013c8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4013c0: jmp 4013eb
4013c3: add %al, [%ebx]
4013c8: or %al, %bl
3. Handling Mispredictions

1. Building Virtual Pipelines
2. Inter-Cluster Dependencies
3. Mispredictions
4. Precise Exceptions

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>Next</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>4013c0</td>
<td>4013eb</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>4013c8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4013c0: jmp 4013eb
4013c3: add %al, [%ebx]
4013c8: or %al, %bl

1. Building Virtual Pipelines
2. Inter-Cluster Dependencies
3. Mispredictions
4. Precise Exceptions
3. Handling Mispredictions

1. Building Virtual Pipelines
2. Inter-Cluster Dependencies
3. Mispredictions
4. Precise Exceptions

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>Next</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>4013c0</td>
<td>4013eb</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>4013c8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Assembly Code

- `4013c0: jmp 4013eb`
- `4013c3: add %al, [%ebx]`
- `4013c8: or %al, %bl`
3. Handling Mispredictions

1. Building Virtual Pipelines
2. Inter-Cluster Dependencies
3. Mispredictions
4. Precise Exceptions

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>Next</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>4013c0</td>
<td>4013eb</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>4013c8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```
4013c0: jmp 4013eb
4013c3: add %al, [%ebx]
4013c8: or %al, %bl
```
3. Handling Mispredictions

1. Building Virtual Pipelines
2. Inter-Cluster Dependencies
3. Mispredictions
4. Precise Exceptions

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>Next</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>4013c0</td>
<td>4013eb</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4013c3</td>
<td>4013c8</td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>4013c8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- 4013c0: jmp 4013eb
- 4013c3: add %al, [%ebx]
- 4013c8: or %al, %bl
4. Precise Exceptions

1. Building Virtual Pipelines
2. Inter-Cluster Dependencies
3. Mispredictions
4. Precise Exceptions

```
4013c3: add %al, [%ebx]
4013c5: div cl
4013c8: jmp 40140a
```

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>Next</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>4013c3</td>
<td></td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4. Precise Exceptions

1. Building Virtual Pipelines
2. Inter-Cluster Dependencies
3. Mispredictions
4. Precise Exceptions

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>Next</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>4013c3</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Exception! Division by 0

BSU

4013c3: add %al,[%ebx]
4013c5: div cl
4013c8: jmp 40140a
4. Precise Exceptions

1. Building Virtual Pipelines
2. Inter-Cluster Dependencies
3. Mispredictions
4. Precise Exceptions

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>Next</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>4013c3</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **4013c3**: `add %al, [%ebx]`
- **4013c5**: `div cl`
- **4013c8**: `jmp 40140a`
4. Precise Exceptions

1. Building Virtual Pipelines
2. Inter-Cluster Dependencies
3. Mispredictions
4. Precise Exceptions

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>Next</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>4013c3</td>
<td>4013c5</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>4013c5</td>
<td></td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4013c3: add %al, [%ebx]
4013c5: div cl
4013c8: jmp 40140a
4. Precise Exceptions

1. Building Virtual Pipelines
2. Inter-Cluster Dependencies
3. Mispredictions
4. Precise Exceptions

4013c3: add %al, [%ebx]
4013c5: div cl
4013c8: jmp 40140a

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>Next</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>4013c3</td>
<td>4013c5</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>4013c5</td>
<td></td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4. Precise Exceptions

1. Building Virtual Pipelines
2. Inter-Cluster Dependencies
3. Mispredictions
4. Precise Exceptions

4013c3: add %al, [%ebx]
4013c5: div cl
4013c8: jmp 40140a

<table>
<thead>
<tr>
<th>BSU ID</th>
<th>Next</th>
<th>PC</th>
<th>NPC</th>
<th>Fetch</th>
<th>Reg</th>
<th>Exec</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>4013c3</td>
<td>4013c5</td>
<td>F0</td>
<td>R1</td>
<td>E0</td>
<td>W0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4013c5</td>
<td>Exc. Handler</td>
<td>F1</td>
<td>R0</td>
<td>E1</td>
<td>W1</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Exc. Handler</td>
<td>4013c8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Impact of Faults in Viper
Impact of Faults in Viper
Impact of Faults in Viper
Impact of Faults in Viper
Impact of Faults in Viper

Handling runtime failures:
1. Periodic full system checkpoint
Impact of Faults in Viper

Handling runtime failures:
1. Periodic full system checkpoint
2. Detected through hardware self-tests or SW symptoms
Impact of Faults in Viper

Handling runtime failures:
1. Periodic full system checkpoint
2. Detected through hardware self-tests or SW symptoms
3. If a fault is detected:
Impact of Faults in Viper

Handling runtime failures:
1. Periodic full system checkpoint
2. Detected through hardware self-tests or SW symptoms
3. If a fault is detected:
   a. Faulty component is diagnosed and disabled
Impact of Faults in Viper

Handling runtime failures:
1. Periodic full system checkpoint
2. Detected through hardware self-tests or SW symptoms
3. If a fault is detected:
   a. Faulty component is diagnosed and disabled
   b. System state is restored to the previous checkpoint
Impact of Faults in Viper

Handling runtime failures:
1. Periodic full system checkpoint
2. Detected through hardware self-tests or SW symptoms
3. If a fault is detected:
   a. Faulty component is diagnosed and disabled
   b. System state is restored to the previous checkpoint
   c. Program execution is restarted
Experimental Setup

- **Viper Configuration**
  - 6 services – Fetch, Decode, Tag Generation, Execute, Commit, WriteBack
  - 4 copy of 5 Clusters
  - 4 cycles latency crossbar / 1 cycle cluster communication latency

- **Baseline CMP**
  - 4 OoO cores: 32k D$ and I$ / 12 stage pipeline / 128 entry ROB / 5 RS entry per FU
  - 6 In-order cores: 32k D$ and I$ / 12 stage pipeline

- **Microarch simulation**
  - Gem5 / timing accurate / system emulation mode
  - SPEC2006 and MIBench
Viper Enables Reliable OoO Execution

Mean +87%
Mean +69%

Relative IPC
Viper Enables Reliable OoO Execution

Mean +87%

Mean +69%

Relative IPC

MiBench

SPEC 2006
Viper is Competitive vs Unprotected OoO
Viper is Competitive vs Unprotected OoO

Mean +87%

Mean +69%

Relative IPC

MiBench

SPEC 2006
## Comparison With Other Solutions

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reconfig. Granularity</strong></td>
<td>Core</td>
<td>Functional units</td>
<td>Functional units</td>
<td>Pipeline stages</td>
<td>Arbitrary</td>
</tr>
<tr>
<td><strong>Control logic</strong></td>
<td>Central</td>
<td>Central</td>
<td>Central</td>
<td>Central</td>
<td>Decoupled</td>
</tr>
<tr>
<td><strong>Interconnect</strong></td>
<td>NoC</td>
<td>Specialized</td>
<td>Specialized</td>
<td>Stage-specific</td>
<td>Loose</td>
</tr>
<tr>
<td><strong>Slowdown</strong></td>
<td>~3%</td>
<td>~5%</td>
<td>~18%</td>
<td>~20%</td>
<td>~24%</td>
</tr>
</tbody>
</table>

Low fault rate  \hspace{1cm} High fault rate
Performance degradation

- CMP tile w/ 2 Billion transistors
- Much more graceful performance degradation
Conclusions

- Viper: scalable, service-oriented µ-architecture

- Hardware reconfiguration granularity as a design choice

- Much more graceful performance degradation
  - Can exploit available hardware to improve performance
Service-Oriented μ-Architecture

MicroProcessor

[Diagram of microprocessors with various colored components]

37
Service-Oriented μ-Architecture
Service-Oriented µ-Architecture
Service-Oriented $\mu$-Architecture
Service-Oriented µ-Architecture
Service-Oriented $\mu$-Architecture
Service-Oriented µ-Architecture
Service-Oriented µ-Architecture
Service-Oriented $\mu$-Architecture
Service-Oriented μ-Architecture
Viper Enables Reliable OoO Execution
Viper Enables Reliable OoO Execution
Viper is Competitive vs Unprotected OoO
Viper is Competitive vs Unprotected OoO

![Graph showing relative IPC for Viper and OoO across various benchmarks]

- **MiBench**
  - basicmath_small
  - cjpeg
  - crc
  - dijkstra_small
  - fft
  - gs
  - gsort_small
  - rawaudio
  - rawdjpeg
  - rijndael
  - say
  - susan
  - toast
  - untoast
  - GeomMean

- **SPEC 2006**
  - astar
  - bwaves
  - bzip2
  - cactusADM
  - dielife
  - GemsFDTD
  - h264ref
  - hmmer
  - lesle3d
  - libquantum
  - mcf
  - milc
  - namd
  - omnetpp
  - specrand_i
  - specrand_f
  - GeomMean