

---

# Viper: Virtual Pipelines for Enhanced Reliability

**Andrea Pellegrini, Joseph Greathouse, Valeria Bertacco**

---

University of Michigan,  
*Advanced Computer Architecture Laboratory*

---

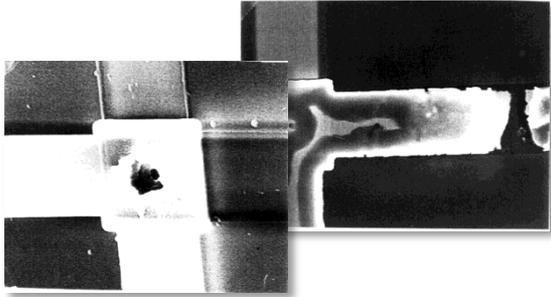
# Reliability Challenges with CMOS Scaling

---

# Reliability Challenges with CMOS Scaling

## **Manufacturing defects**

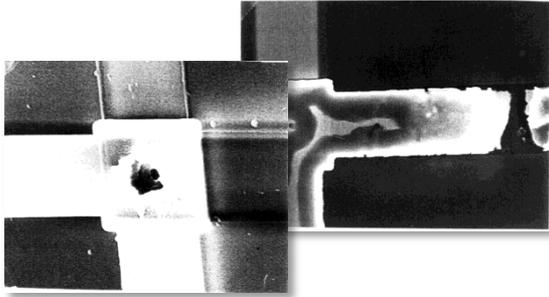
That escape testing



# Reliability Challenges with CMOS Scaling

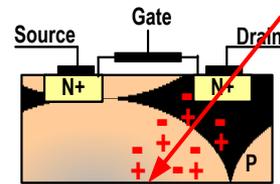
## Manufacturing defects

That escape testing



## Transient faults

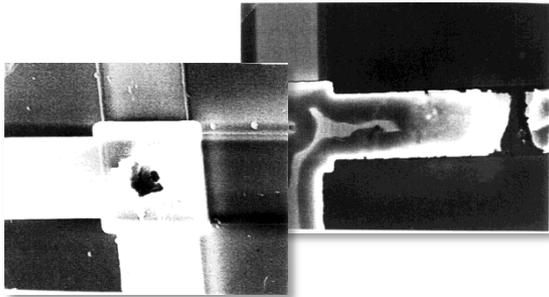
Natural radiation,  
noise...



# Reliability Challenges with CMOS Scaling

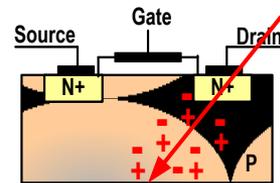
## Manufacturing defects

That escape testing



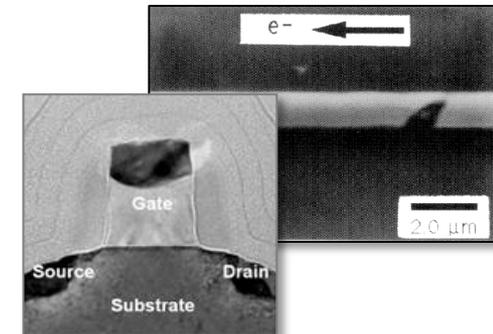
## Transient faults

Natural radiation,  
noise...



## Age-related wearout

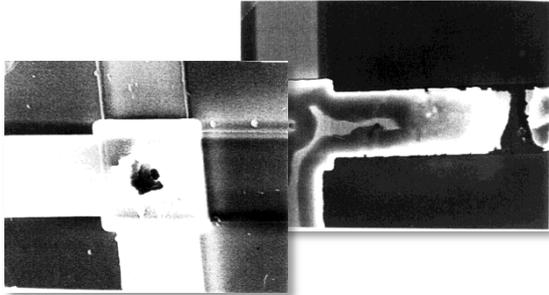
Electromigration &  
gate-oxide breakdown



# Reliability Challenges with CMOS Scaling

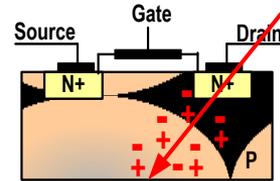
## Manufacturing defects

That escape testing



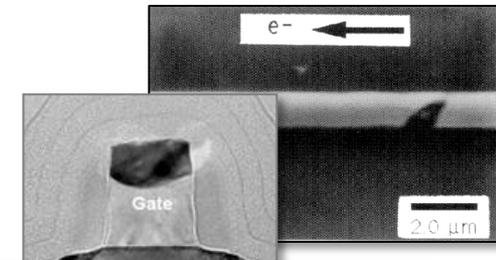
## Transient faults

Natural radiation,  
noise...



## Age-related wearout

Electromigration &  
gate-oxide breakdown

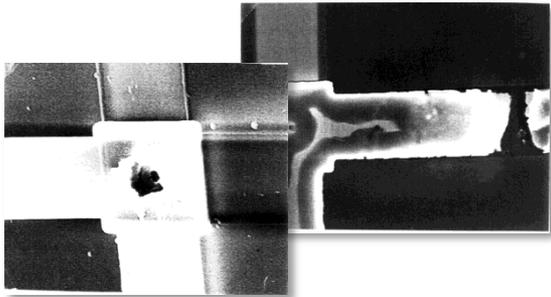


Intel Cougar Point chipsets  
wearing out over time  
[Estimated cost of 700M\$ 2011]

# Reliability Challenges with CMOS Scaling

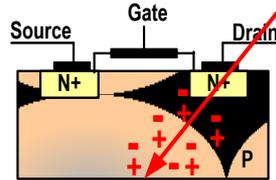
## Manufacturing defects

That escape testing



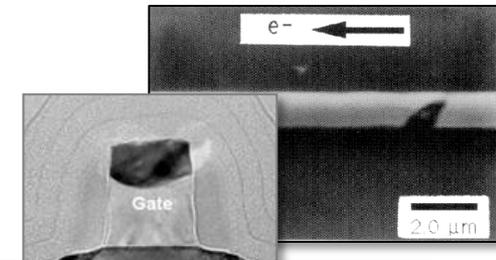
## Transient faults

Natural radiation,  
noise...



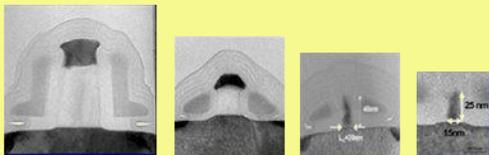
## Age-related wearout

Electromigration &  
gate-oxide breakdown

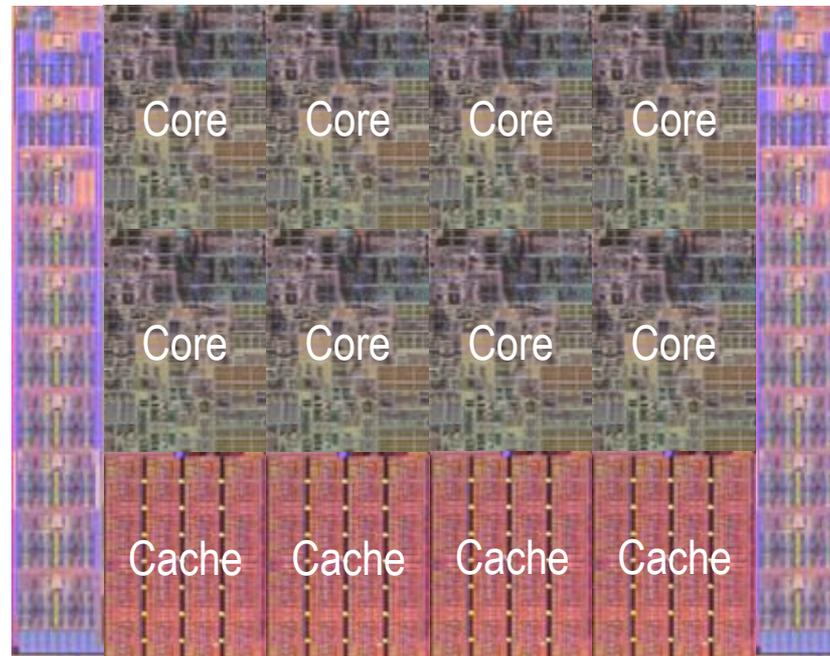


Intel Cougar Point chipsets  
wearing out over time  
[Estimated cost of 700M\$ 2011]

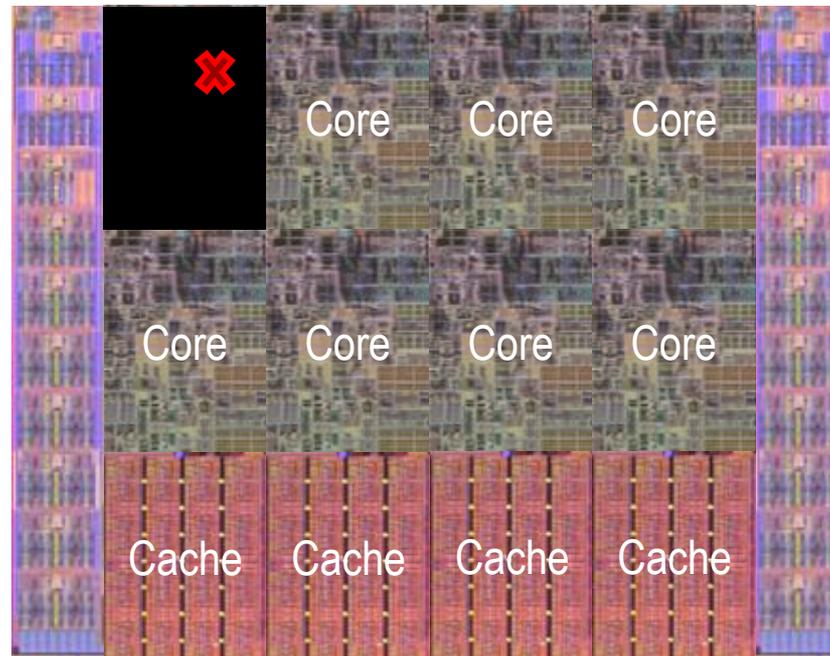
**“Future technologies will make transistors  
less and less reliable” [Borkar, 2005]**



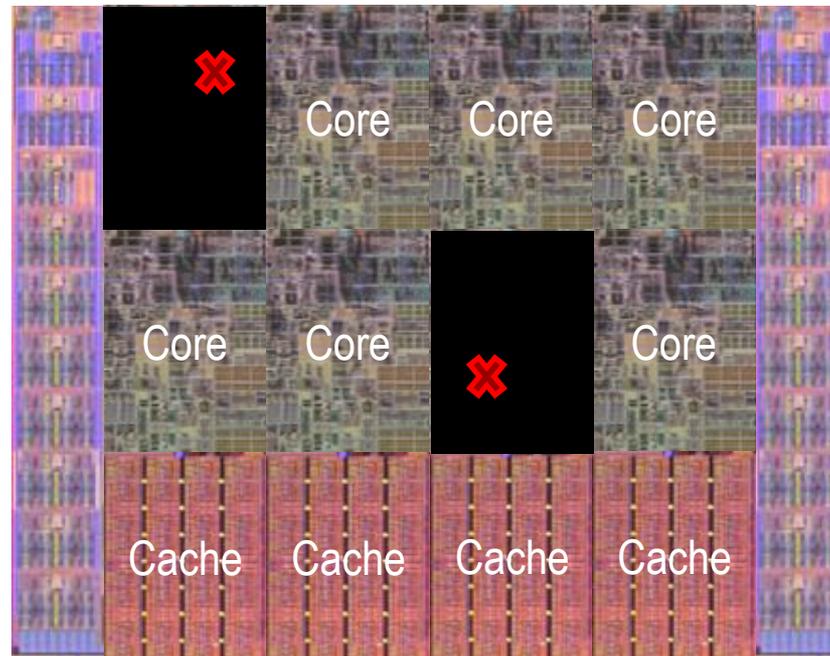
# Impact of Faults on Traditional CMPs



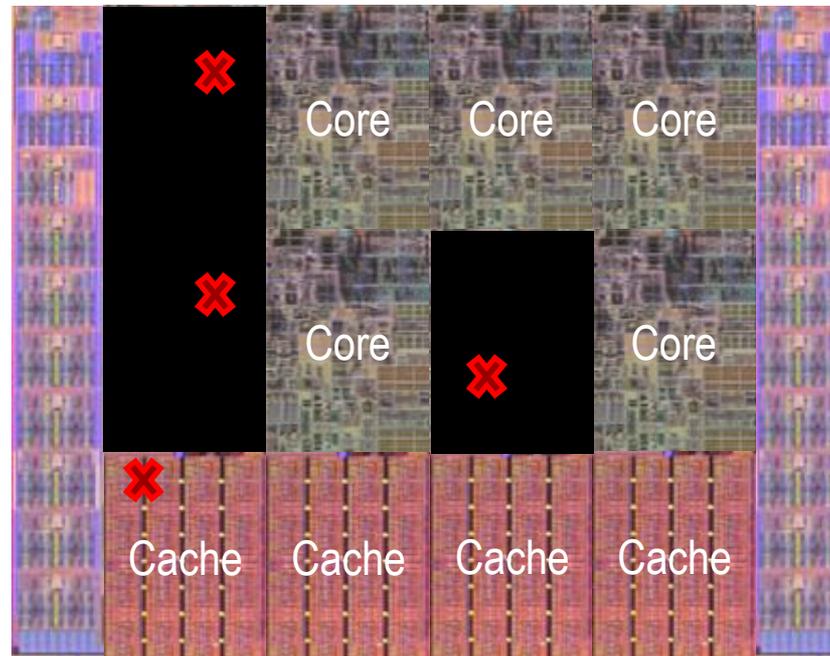
# Impact of Faults on Traditional CMPs



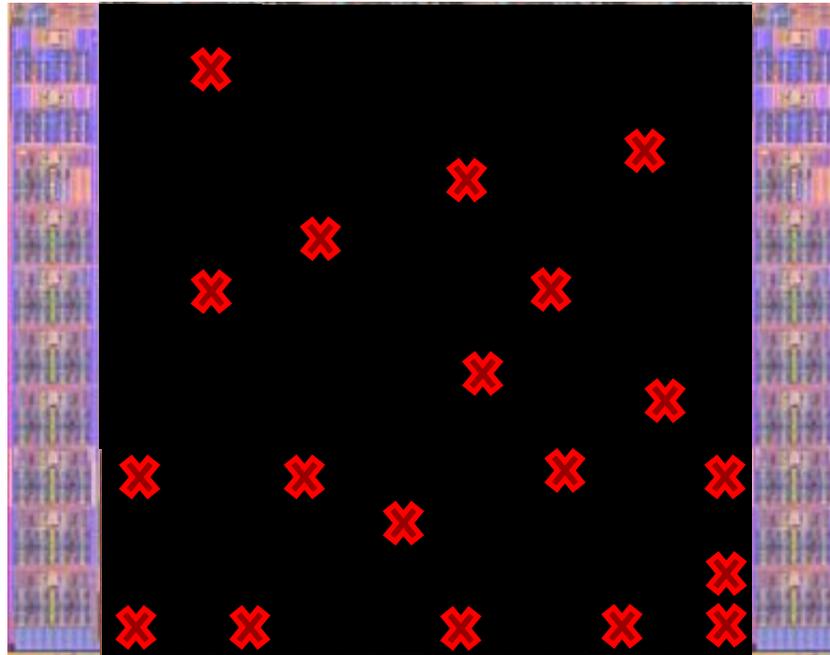
# Impact of Faults on Traditional CMPs



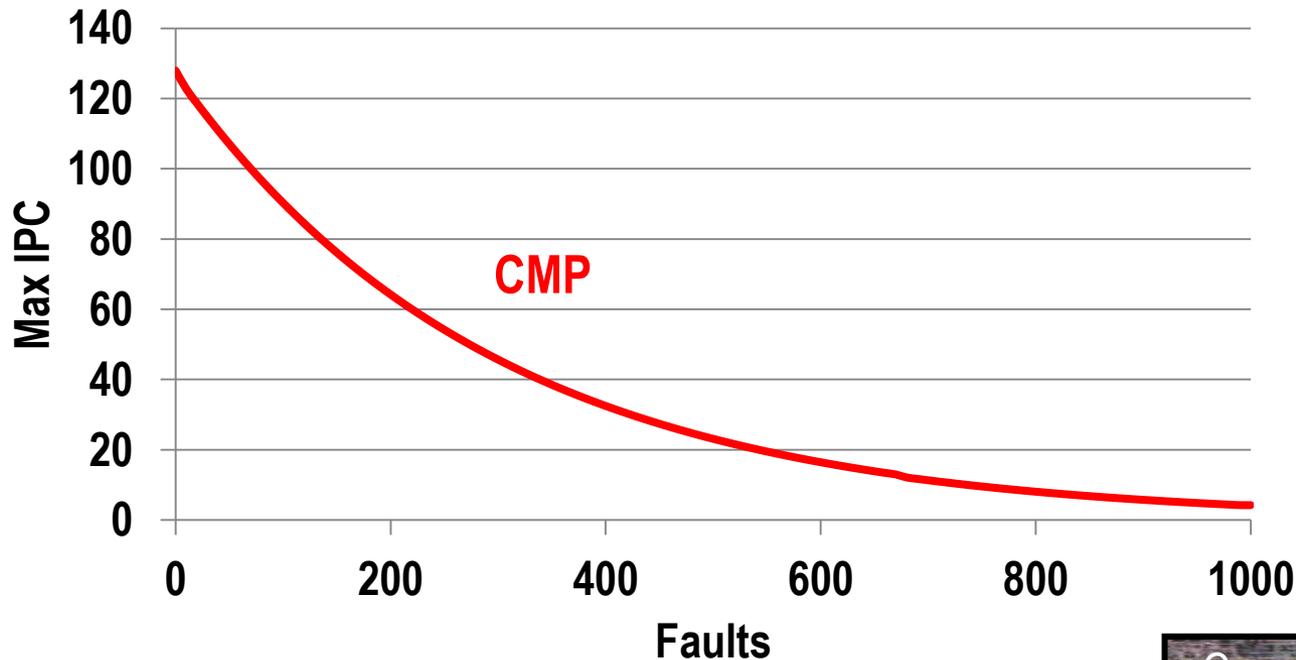
# Impact of Faults on Traditional CMPs



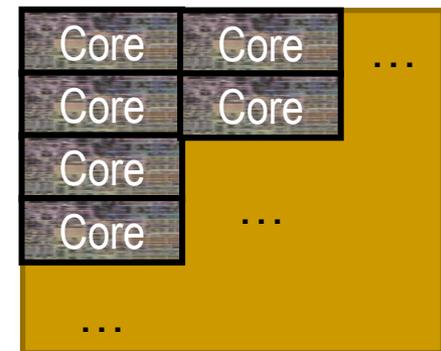
# Impact of Faults on Traditional CMPs



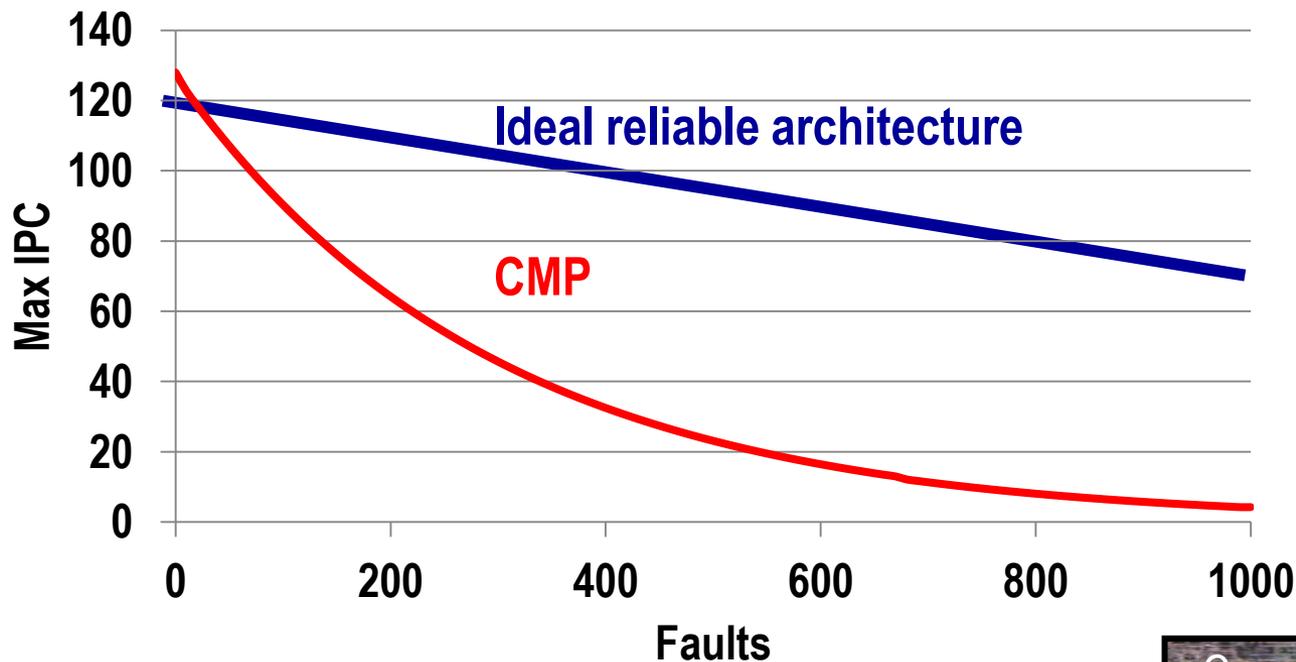
# Faults Effects on CMP Throughput



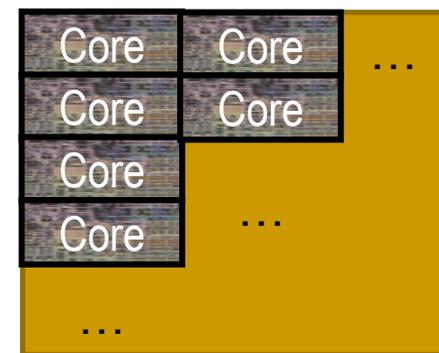
- CMP system w/ 2 billion transistors fitting 128 cores – no caches
  - 15M transistors/core, similar to Intel Atom



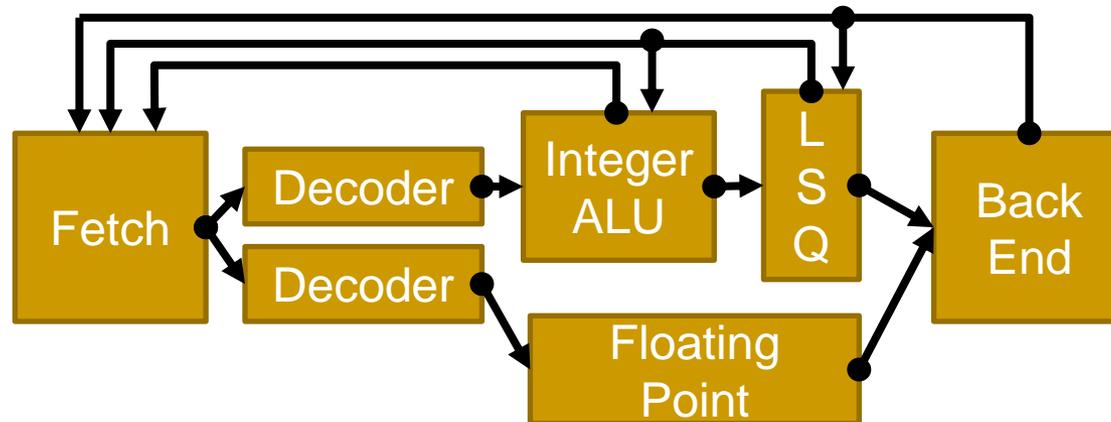
# Faults Effects on CMP Throughput



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



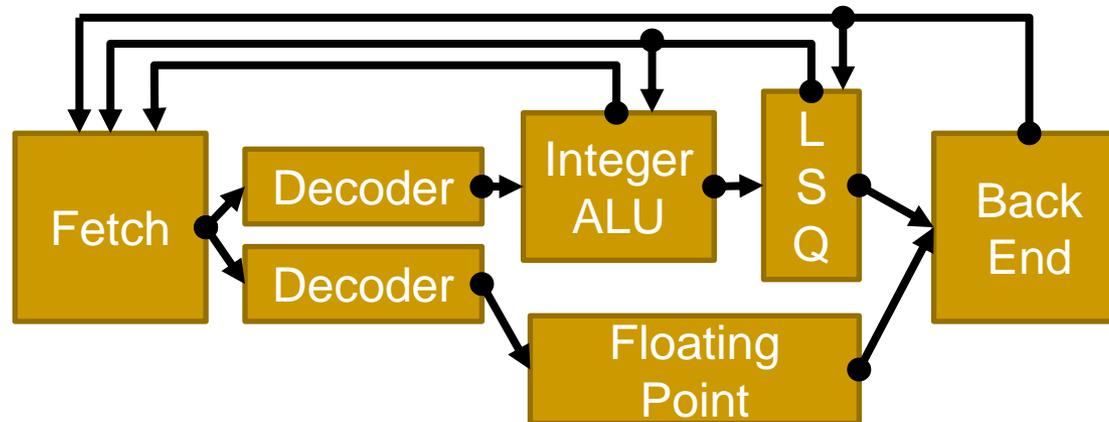
# Limitations of Current $\mu$ -Architectures



# Limitations of Current $\mu$ -Architectures



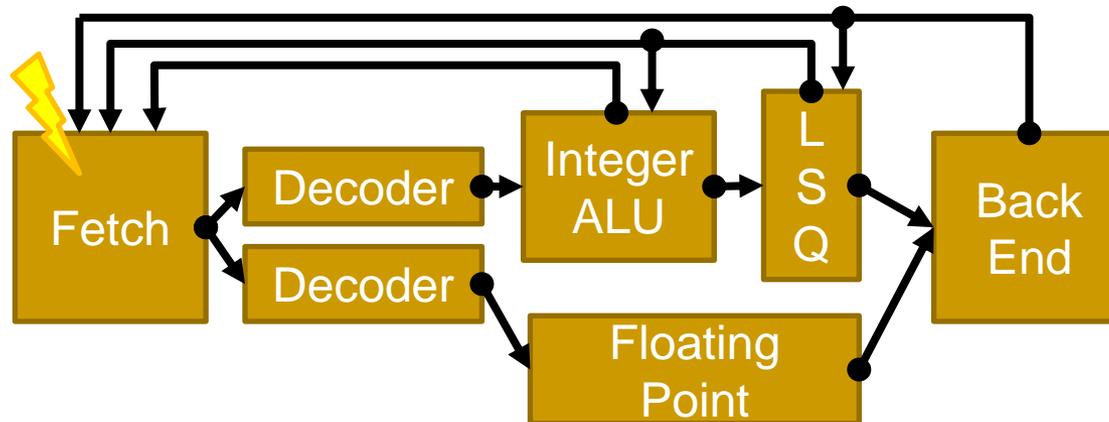
Single point of failures



# Limitations of Current $\mu$ -Architectures



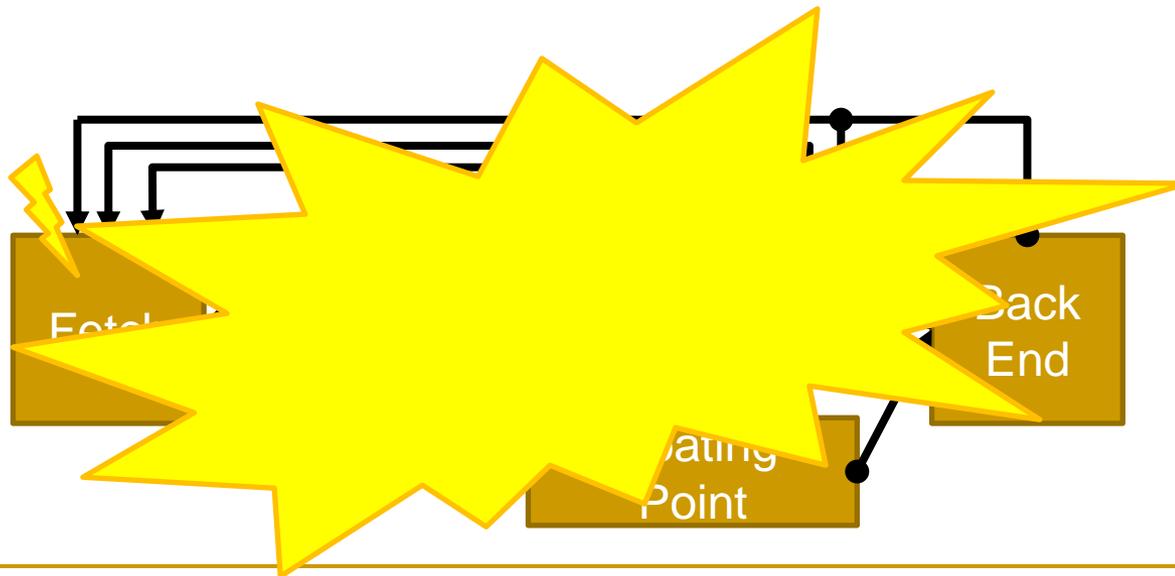
Single point of failures



# Limitations of Current $\mu$ -Architectures



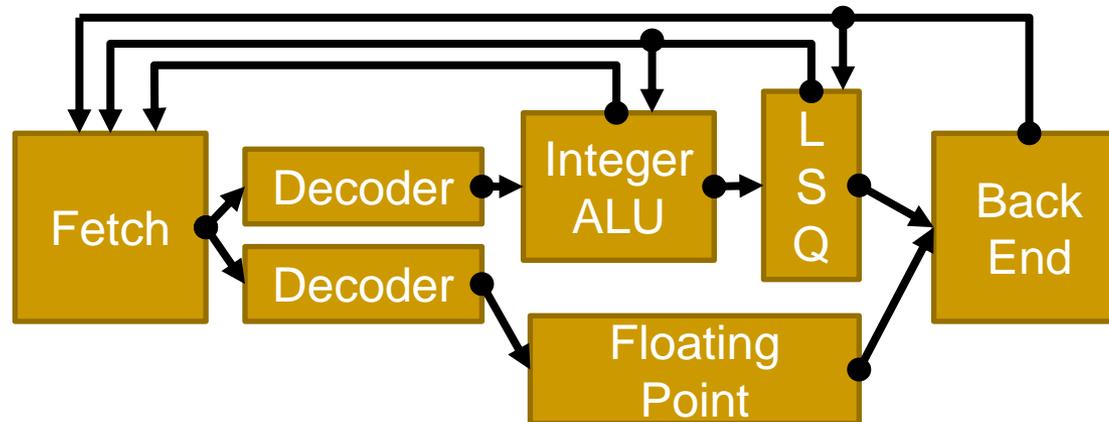
Single point of failures



# Limitations of Current $\mu$ -Architectures



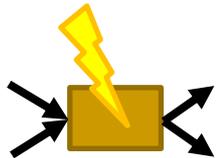
Single point of failures



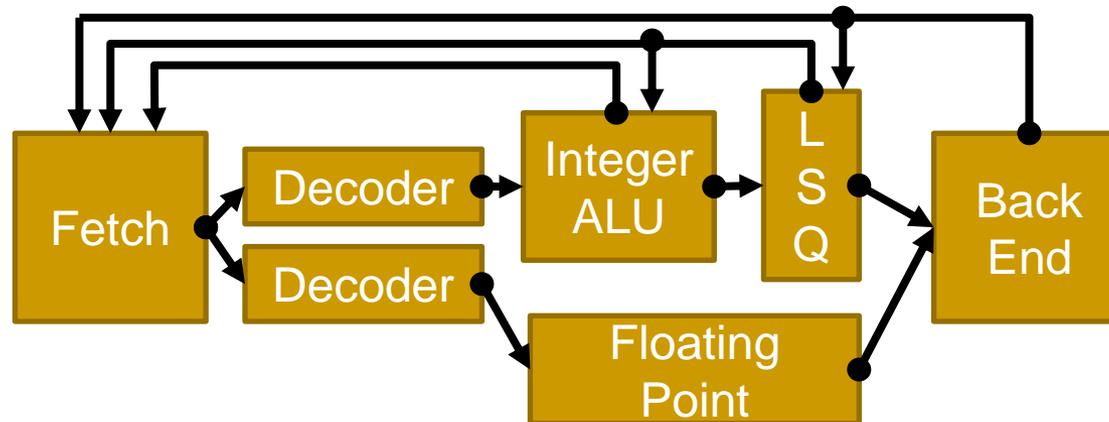
# Limitations of Current $\mu$ -Architectures



Single point of failures



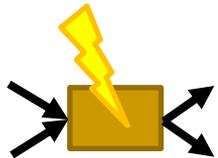
Rigidly connected pipeline stages



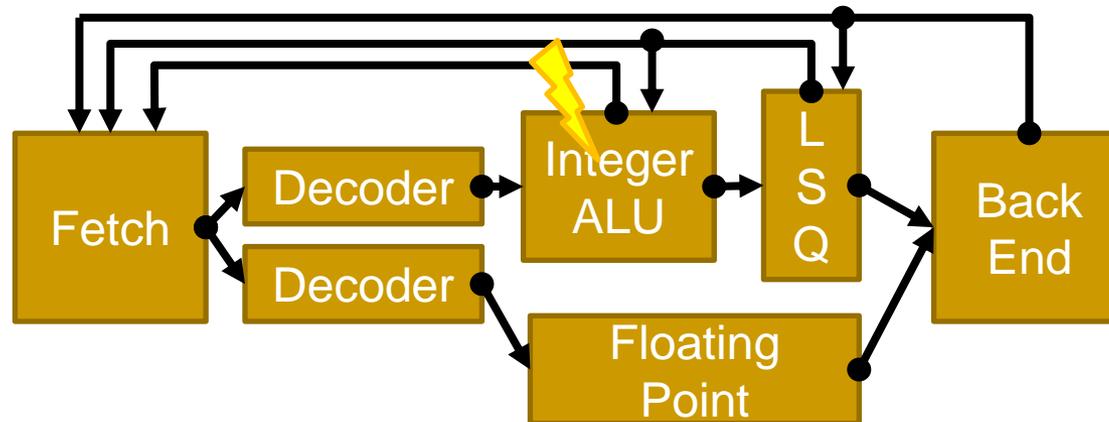
# Limitations of Current $\mu$ -Architectures



Single point of failures



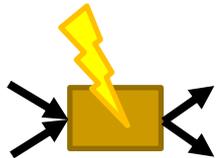
Rigidly connected pipeline stages



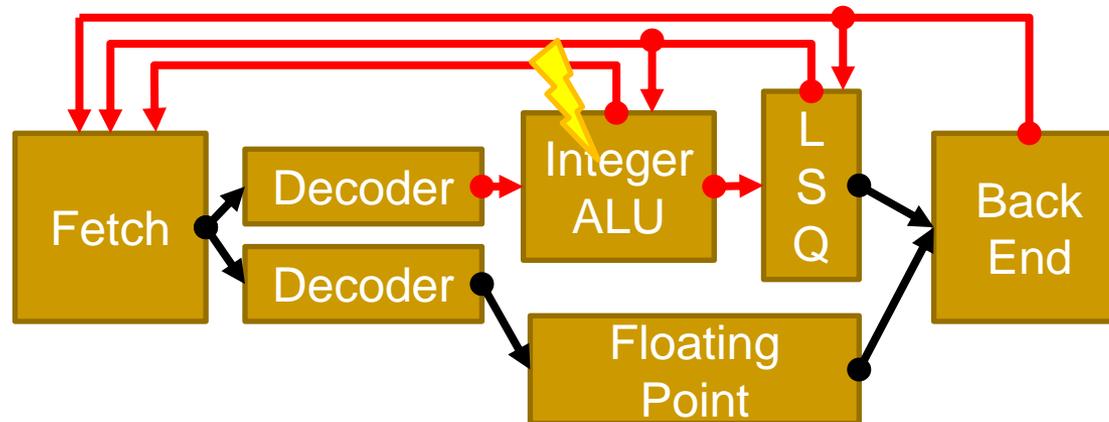
# Limitations of Current $\mu$ -Architectures



Single point of failures



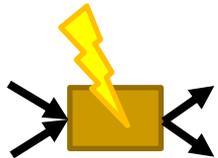
Rigidly connected pipeline stages



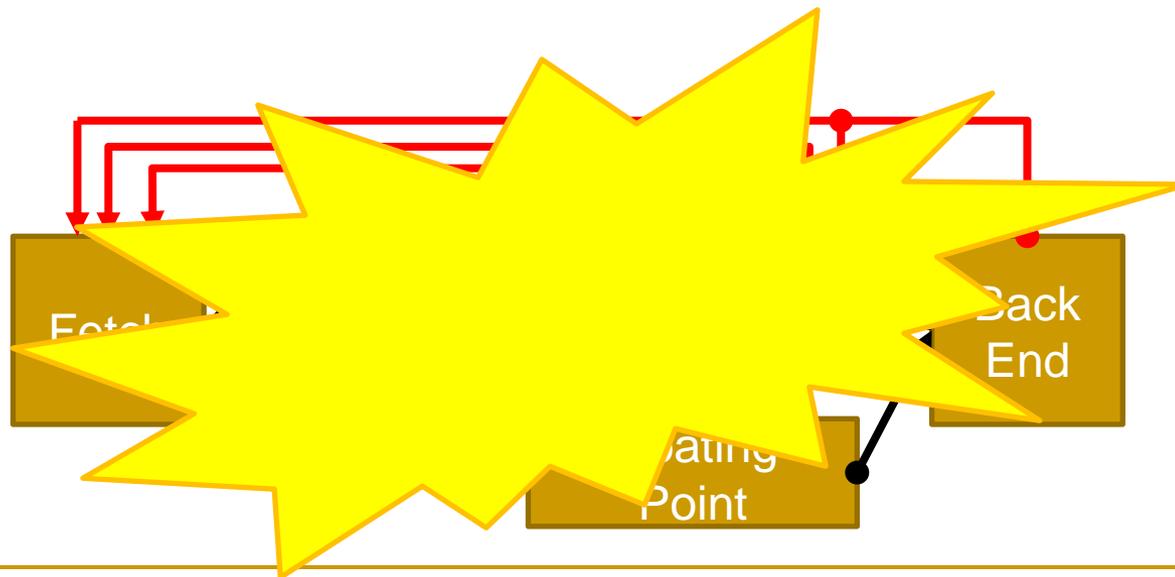
# Limitations of Current $\mu$ -Architectures



Single point of failures



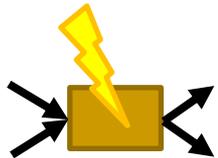
Rigidly connected pipeline stages



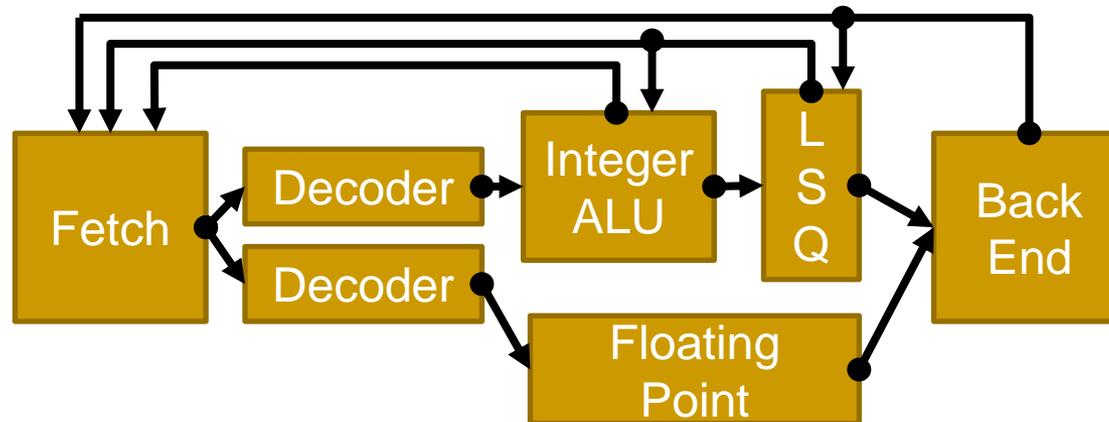
# Limitations of Current $\mu$ -Architectures



Single point of failures



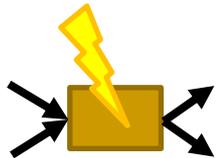
Rigidly connected pipeline stages



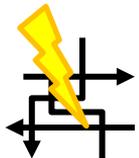
# Limitations of Current $\mu$ -Architectures



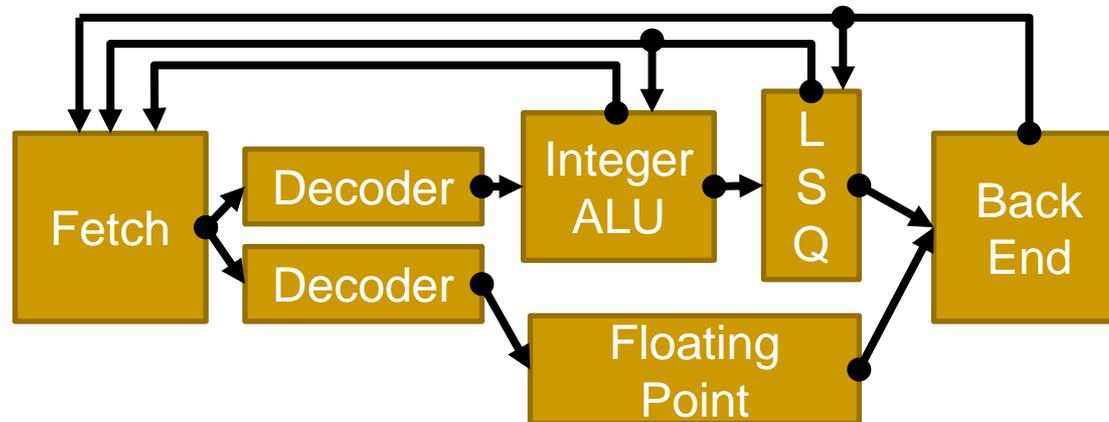
Single point of failures



Rigidly connected pipeline stages



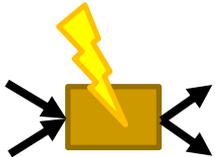
Centralized control logic



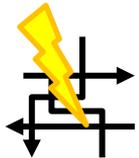
# Limitations of Current $\mu$ -Architectures



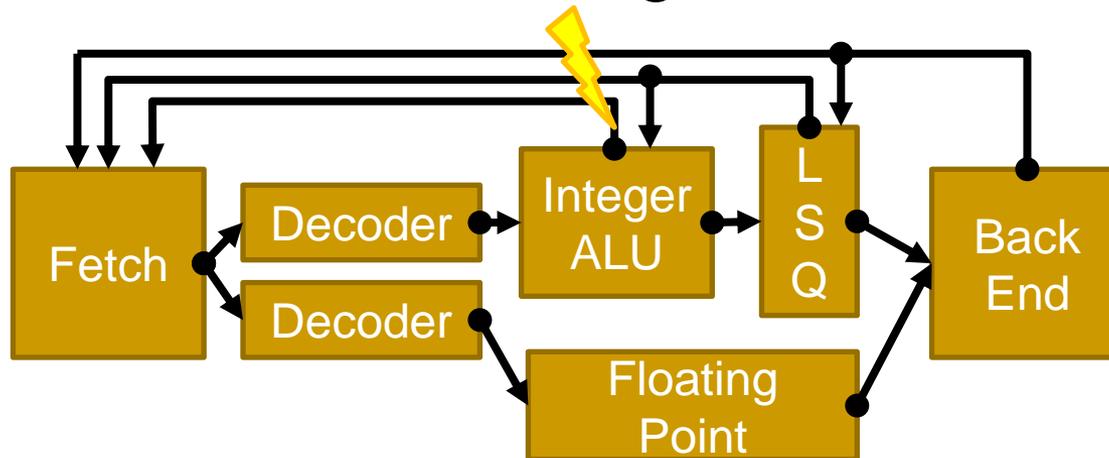
Single point of failures



Rigidly connected pipeline stages



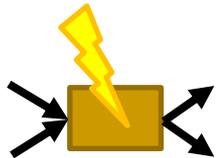
Centralized control logic



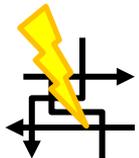
# Limitations of Current $\mu$ -Architectures



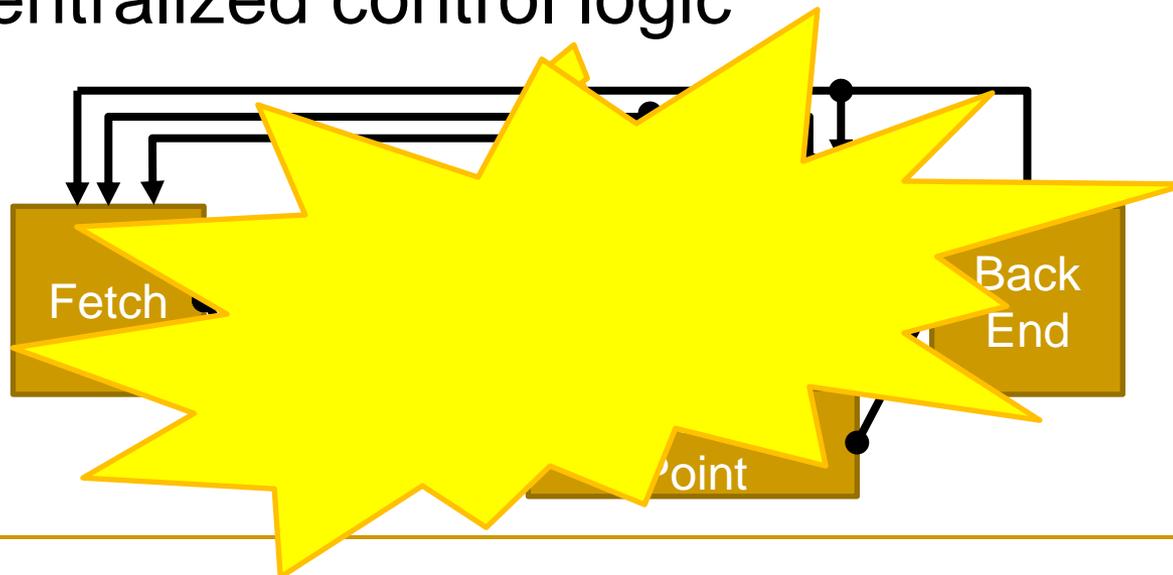
Single point of failures



Rigidly connected pipeline stages



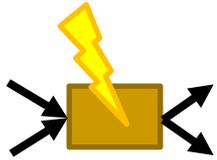
Centralized control logic



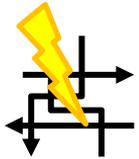
# Can We Overcome These Limitations?



Single point of failures



Rigidly connected pipeline stages



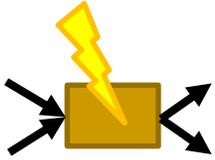
Centralized control logic

# Can We Overcome These Limitations?

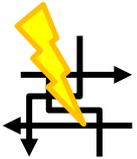


Single point of failures

✓ Arrays of redundant hardware units



Rigidly connected pipeline stages



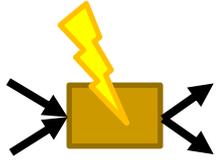
Centralized control logic

# Can We Overcome These Limitations?



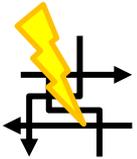
Single point of failures

✓ Arrays of redundant hardware units



Rigidly connected pipeline stages

✓ Loosely connected hardware modules



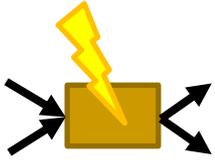
Centralized control logic

# Can We Overcome These Limitations?



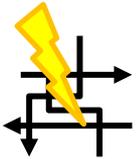
Single point of failures

- ✓ Arrays of redundant hardware units



Rigidly connected pipeline stages

- ✓ Loosely connected hardware modules



Centralized control logic

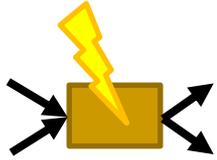
- ✓ Decentralized and redundant controls

# Can We Overcome These Limitations?



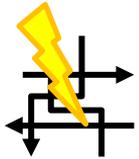
Single point of failures

✓ Arrays of redundant hardware units



Rigidly connected pipeline stages

✓ Loosely connected hardware modules



Centralized control logic

✓ Decentralized and redundant controls

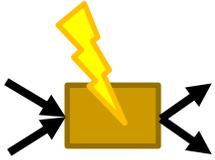
Service-oriented  $\mu$ -architecture  
to tackle all three issues:

# Can We Overcome These Limitations?



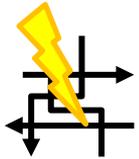
Single point of failures

✓ Arrays of redundant hardware units



Rigidly connected pipeline stages

✓ Loosely connected hardware modules



Centralized control logic

✓ Decentralized and redundant controls

Service-oriented  $\mu$ -architecture

to tackle all three issues:

**Virtual Pipelines for Enhanced Reliability**

---

# Service-Oriented $\mu$ -Architecture



---

# Service-Oriented $\mu$ -Architecture



**Renew Driving License:**

# Service-Oriented $\mu$ -Architecture

## Renew Driving License:

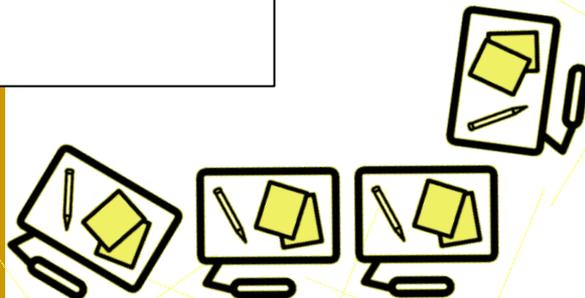
- 1. Check in



# Service-Oriented $\mu$ -Architecture

## Renew Driving License:

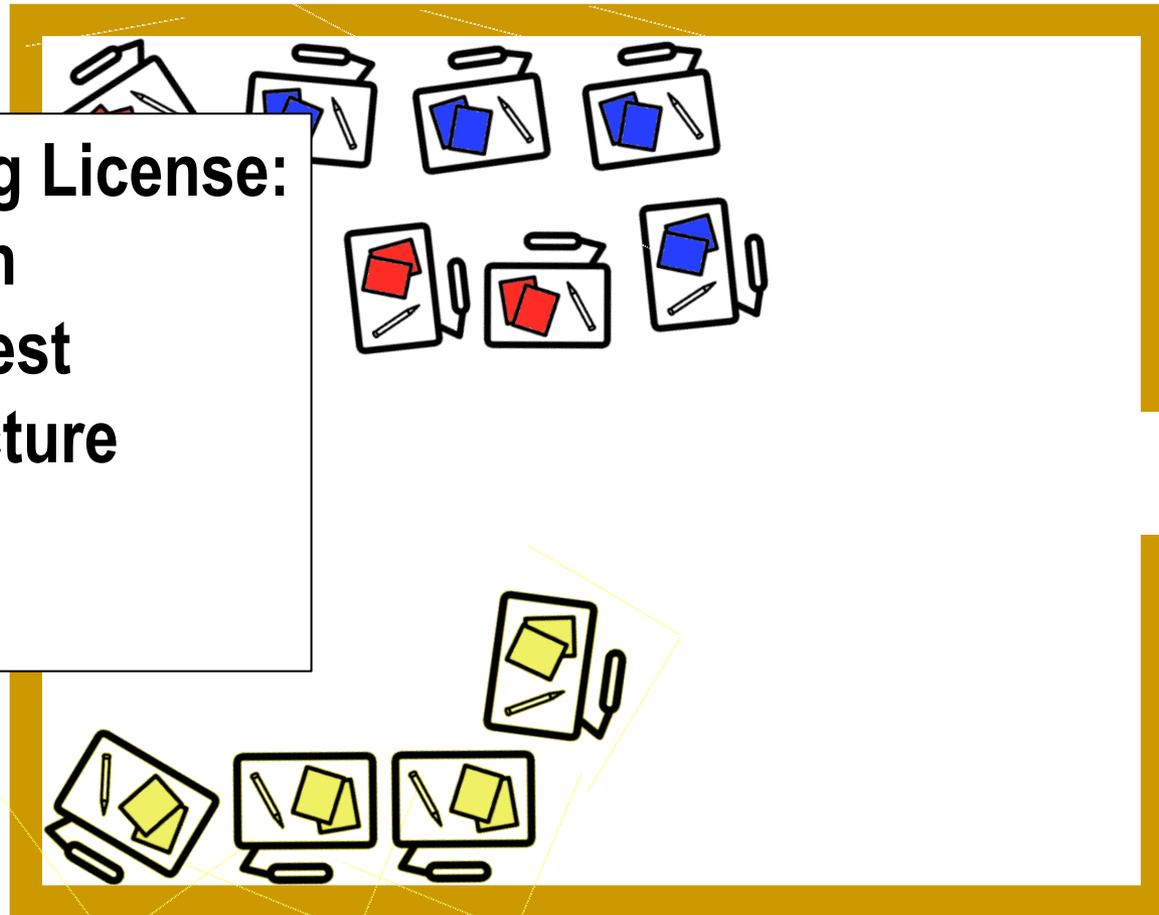
- 1. Check in
- 2. Vision test



# Service-Oriented $\mu$ -Architecture

## Renew Driving License:

- 1. Check in
- 2. Vision test
- 3. Take picture



# Service-Oriented $\mu$ -Architecture

## Renew Driving License:

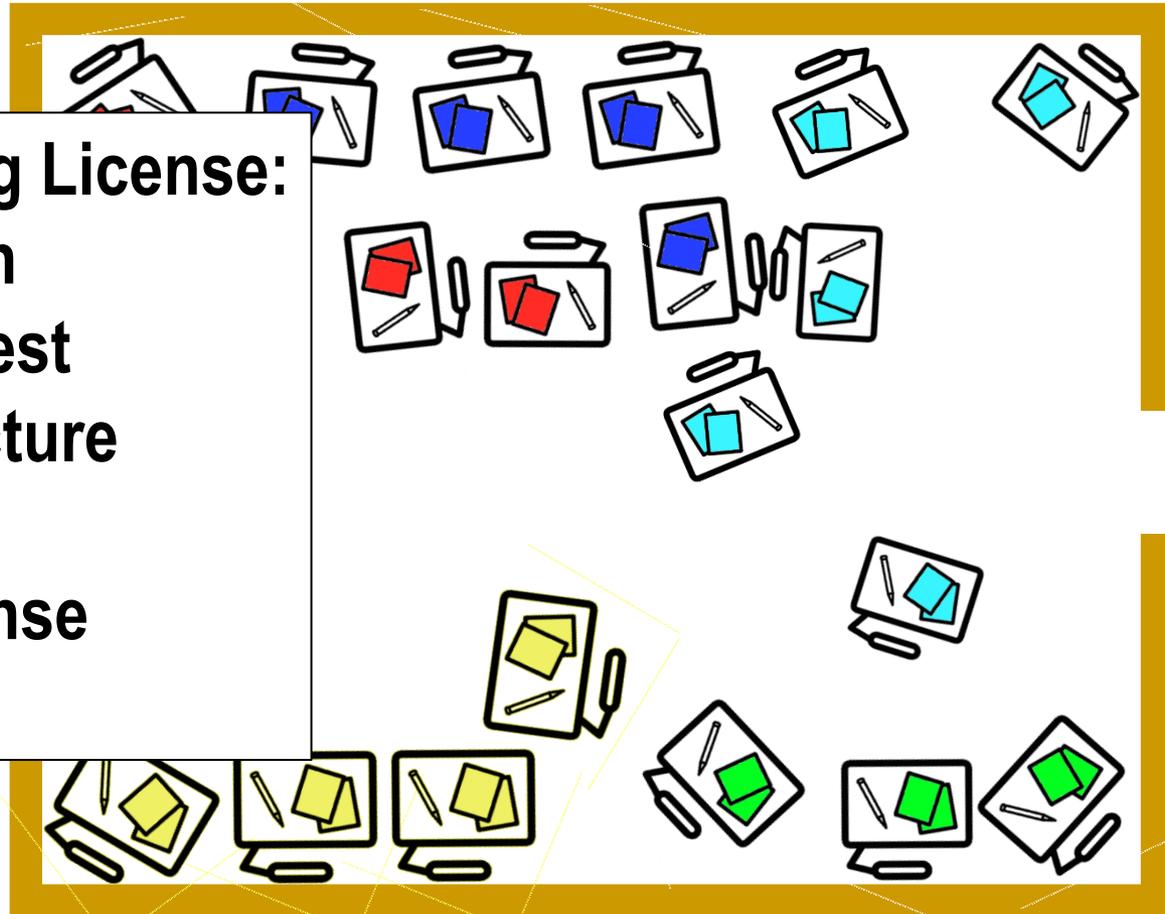
- 1. Check in
- 2. Vision test
- 3. Take picture
- 4. Pay fee



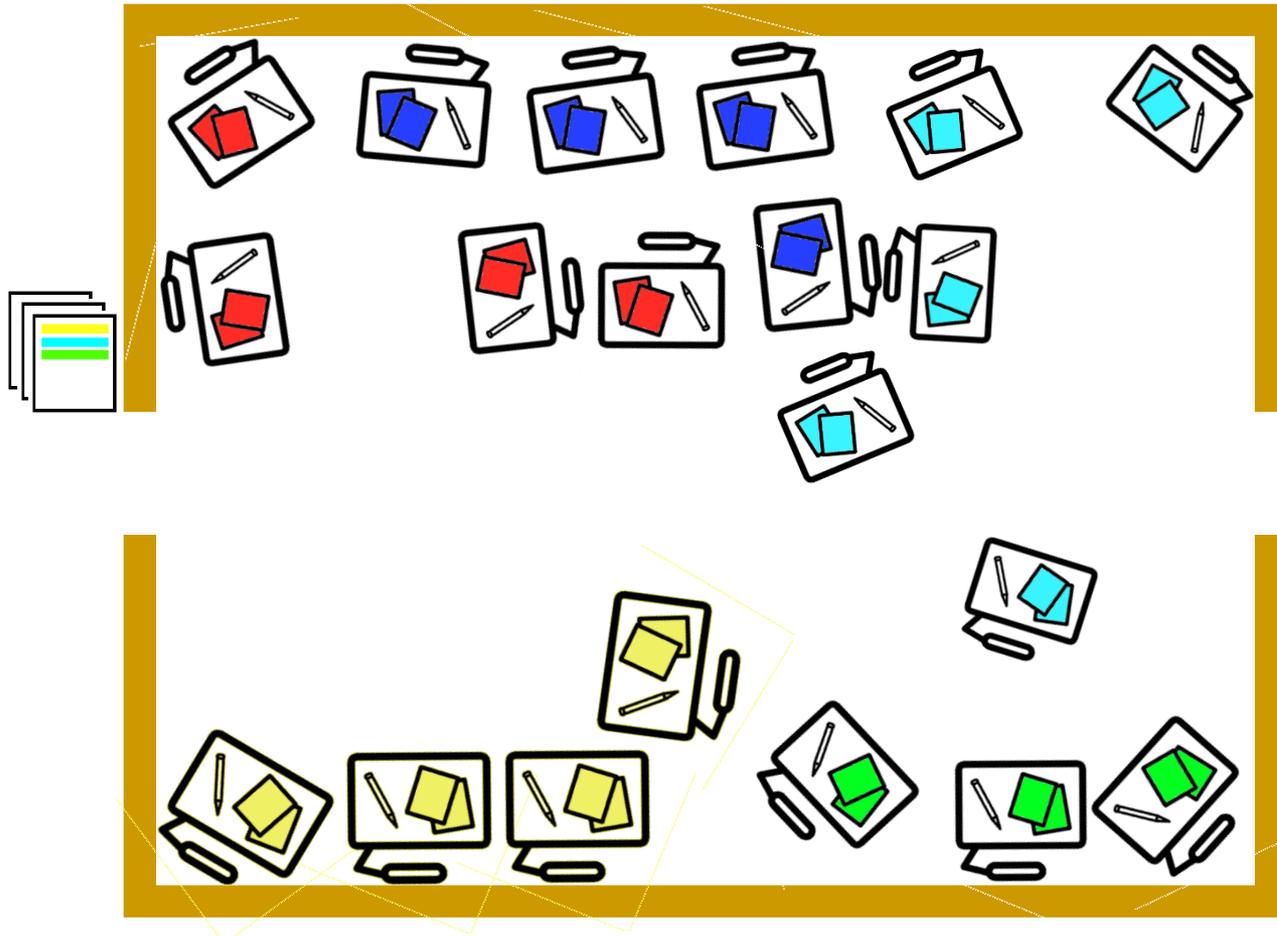
# Service-Oriented $\mu$ -Architecture

## Renew Driving License:

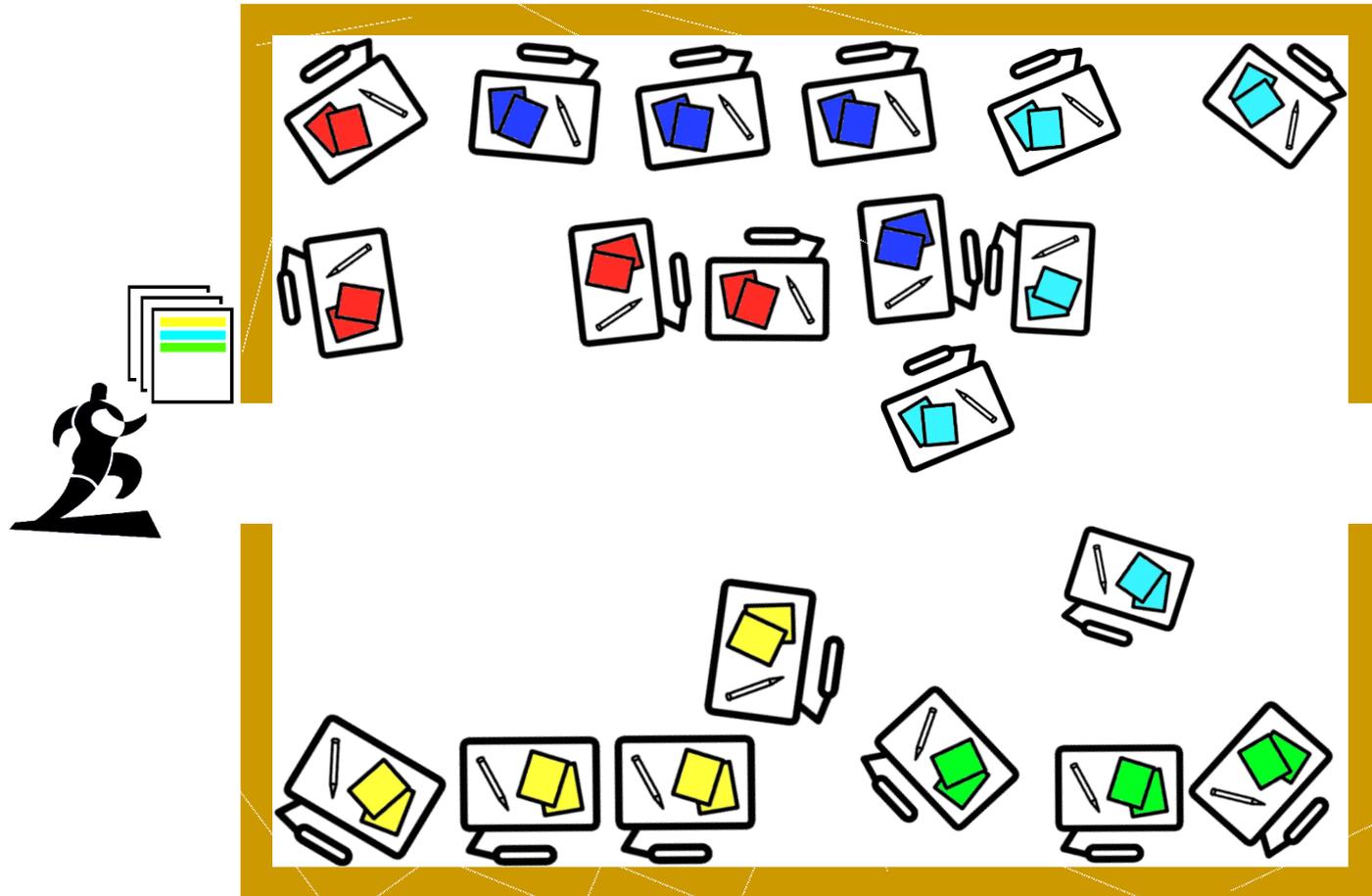
-  1. Check in
-  2. Vision test
-  3. Take picture
-  4. Pay fee
-  5. Get license



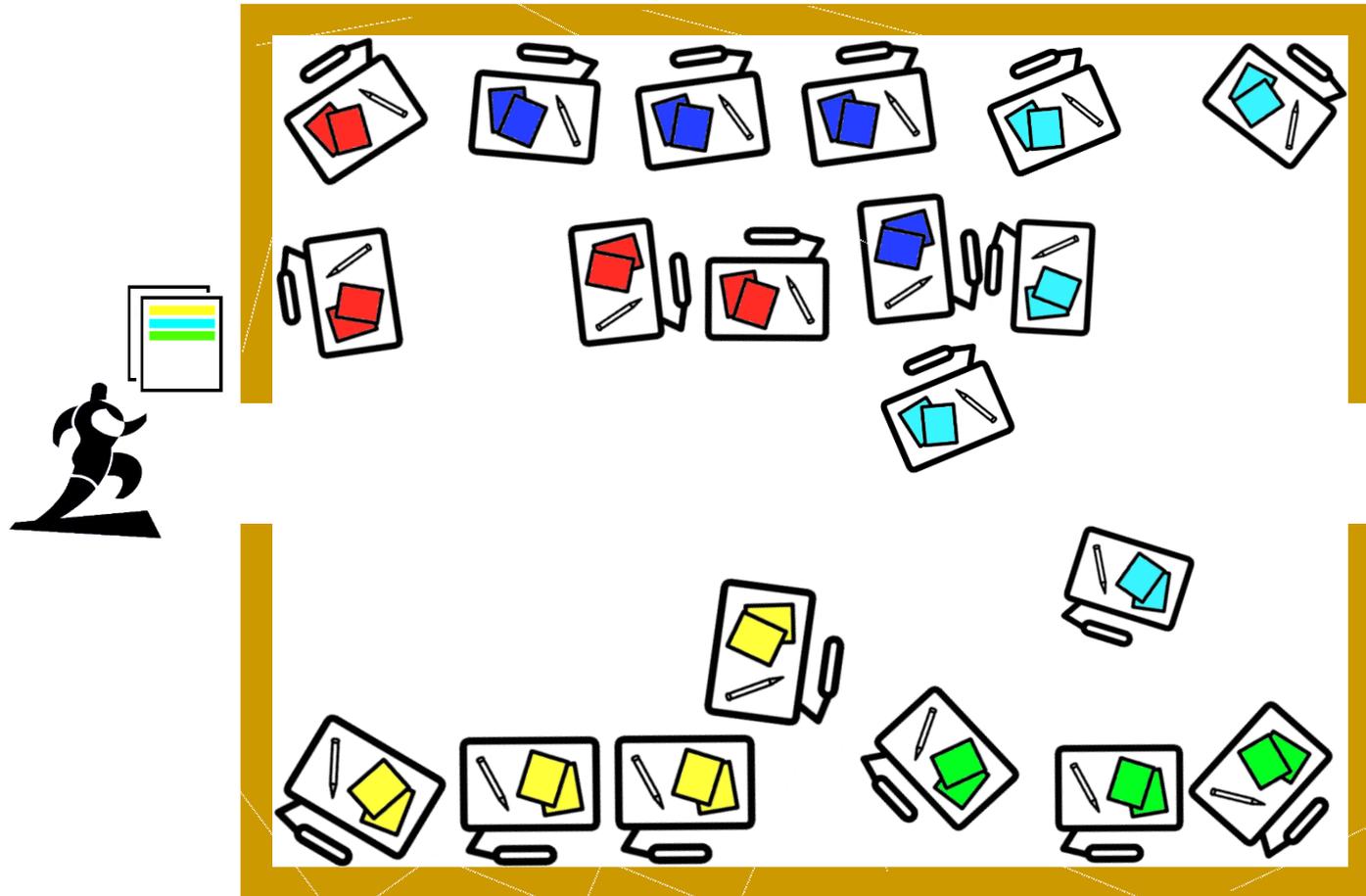
# Service-Oriented $\mu$ -Architecture



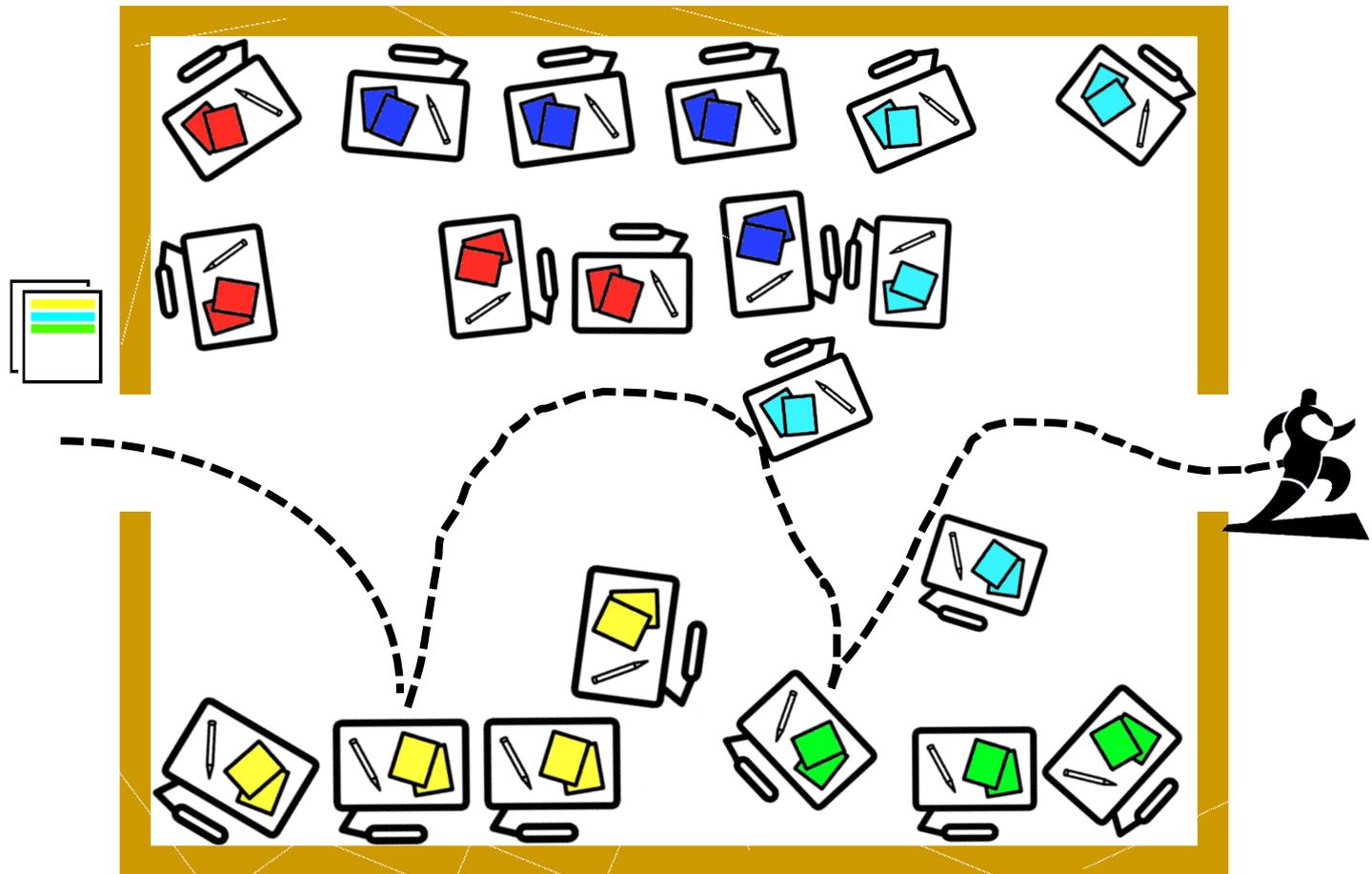
# Service-Oriented $\mu$ -Architecture



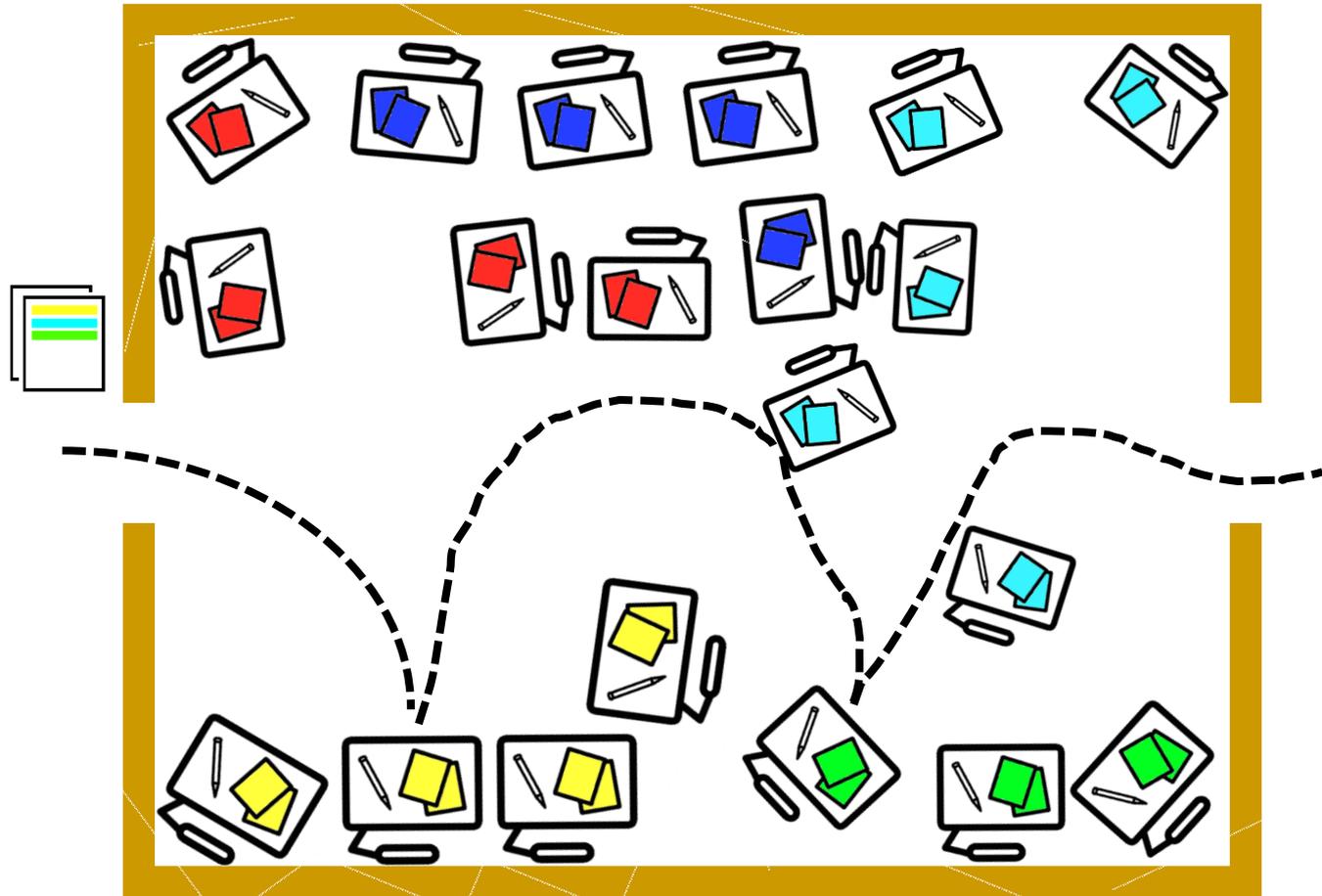
# Service-Oriented $\mu$ -Architecture



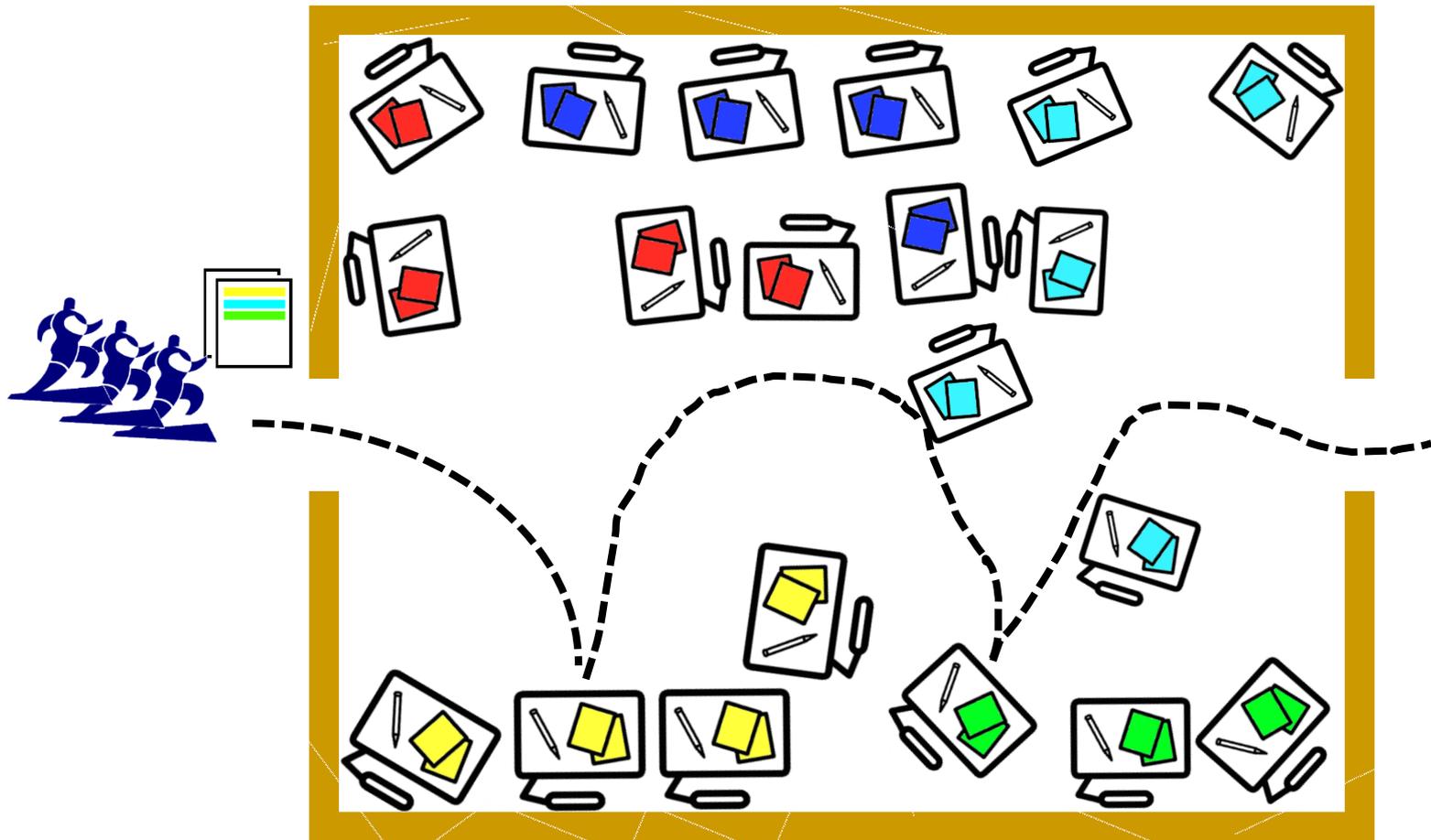
# Service-Oriented $\mu$ -Architecture



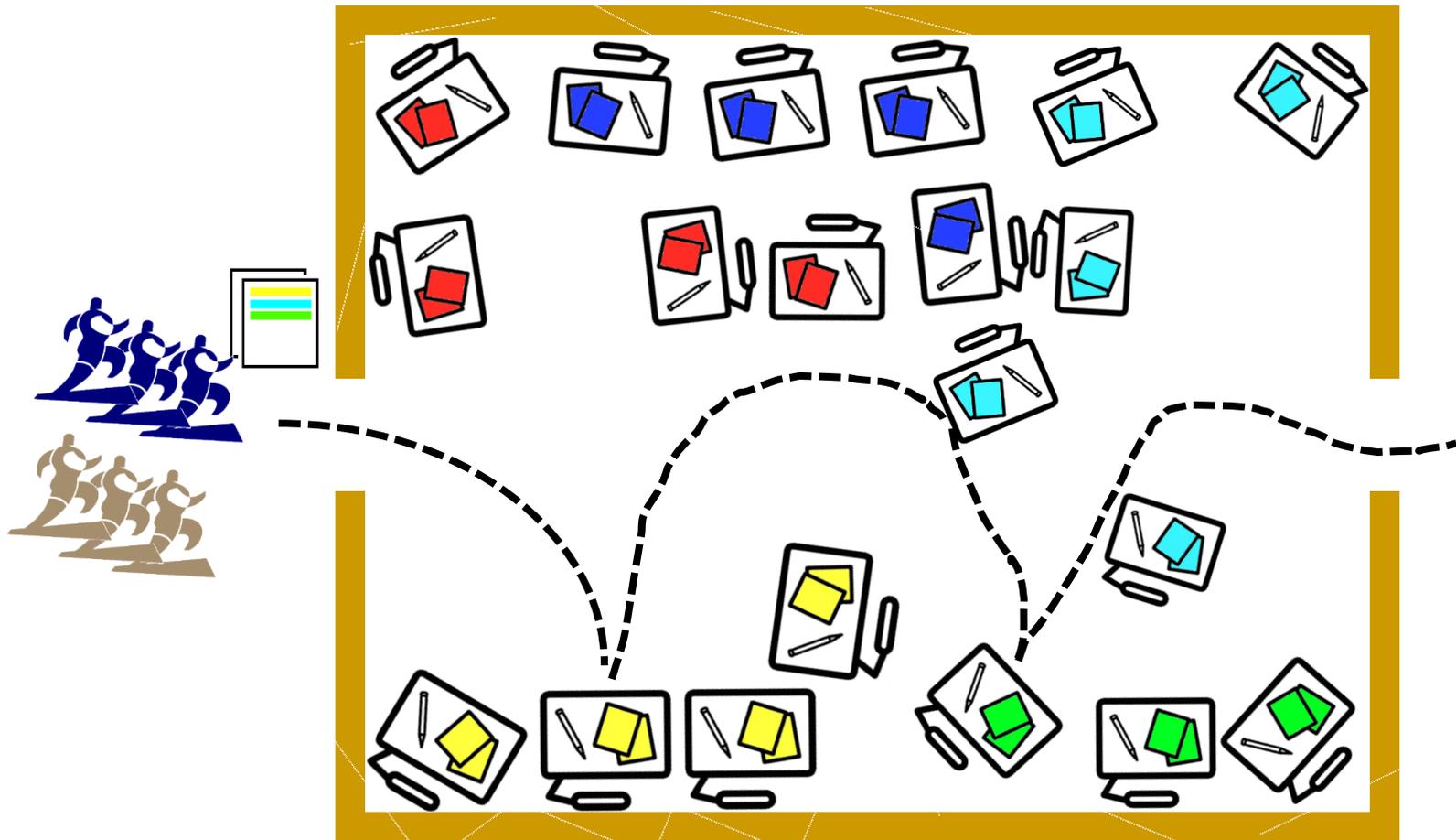
# Service-Oriented $\mu$ -Architecture



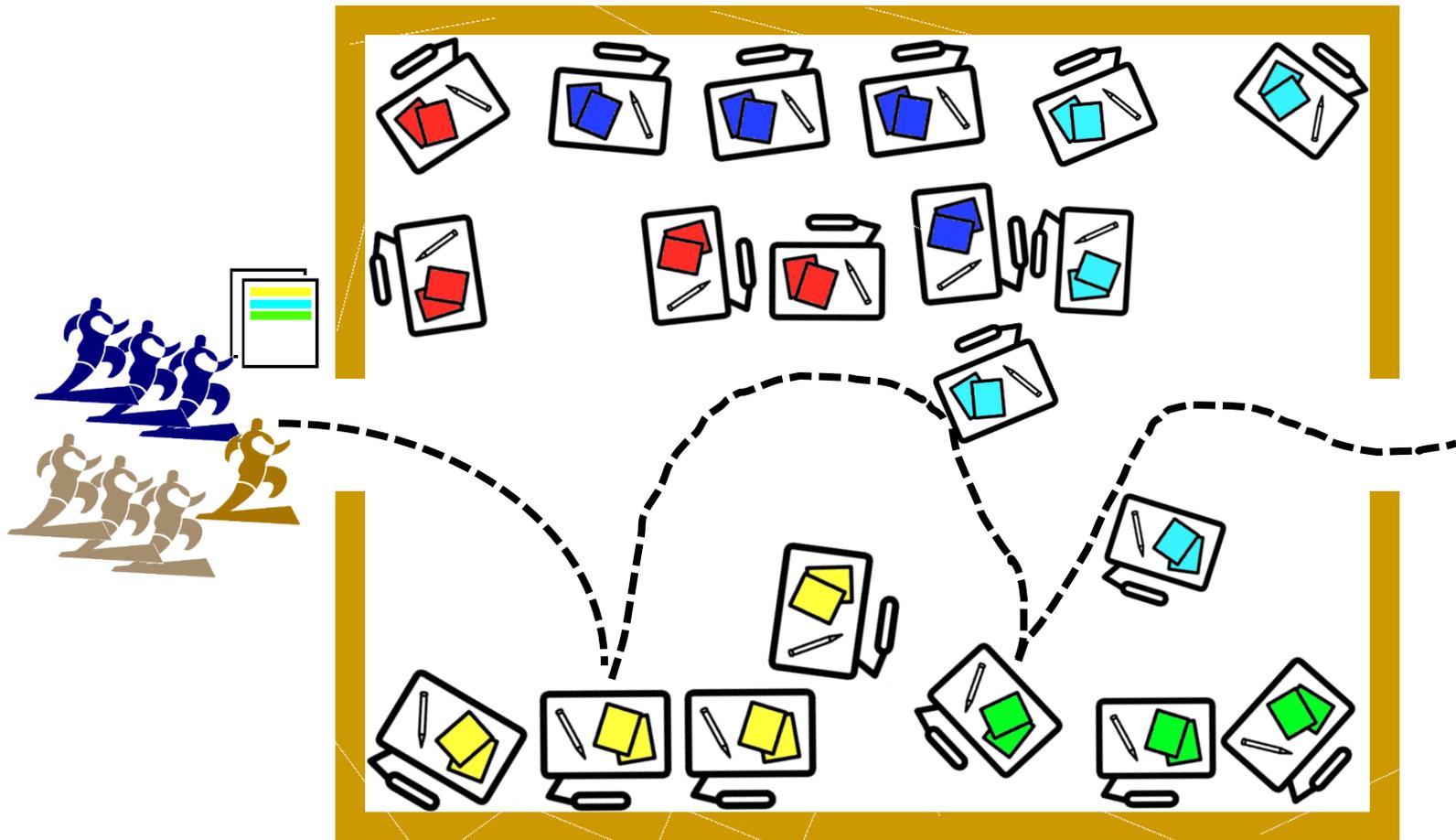
# Service-Oriented $\mu$ -Architecture



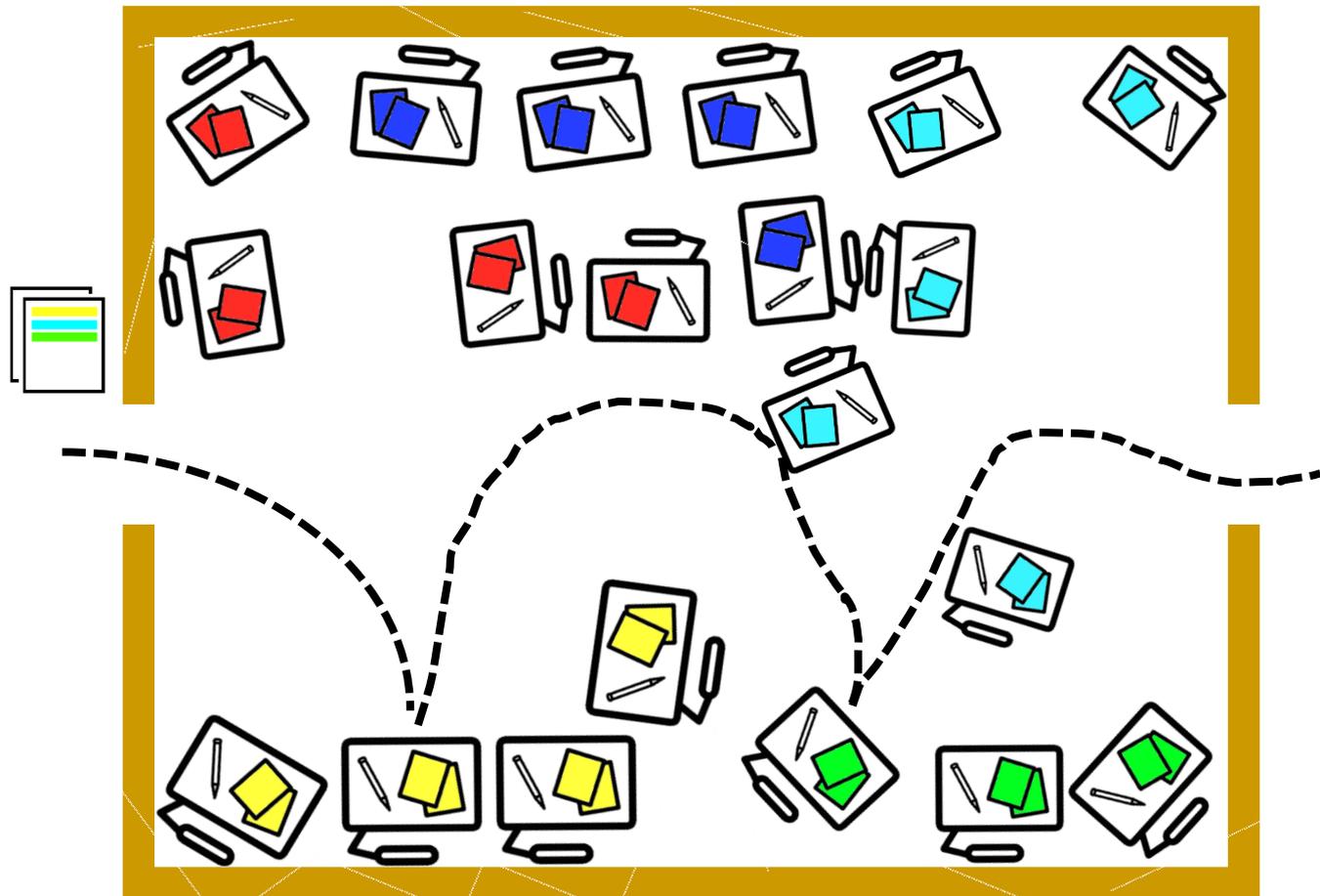
# Service-Oriented $\mu$ -Architecture



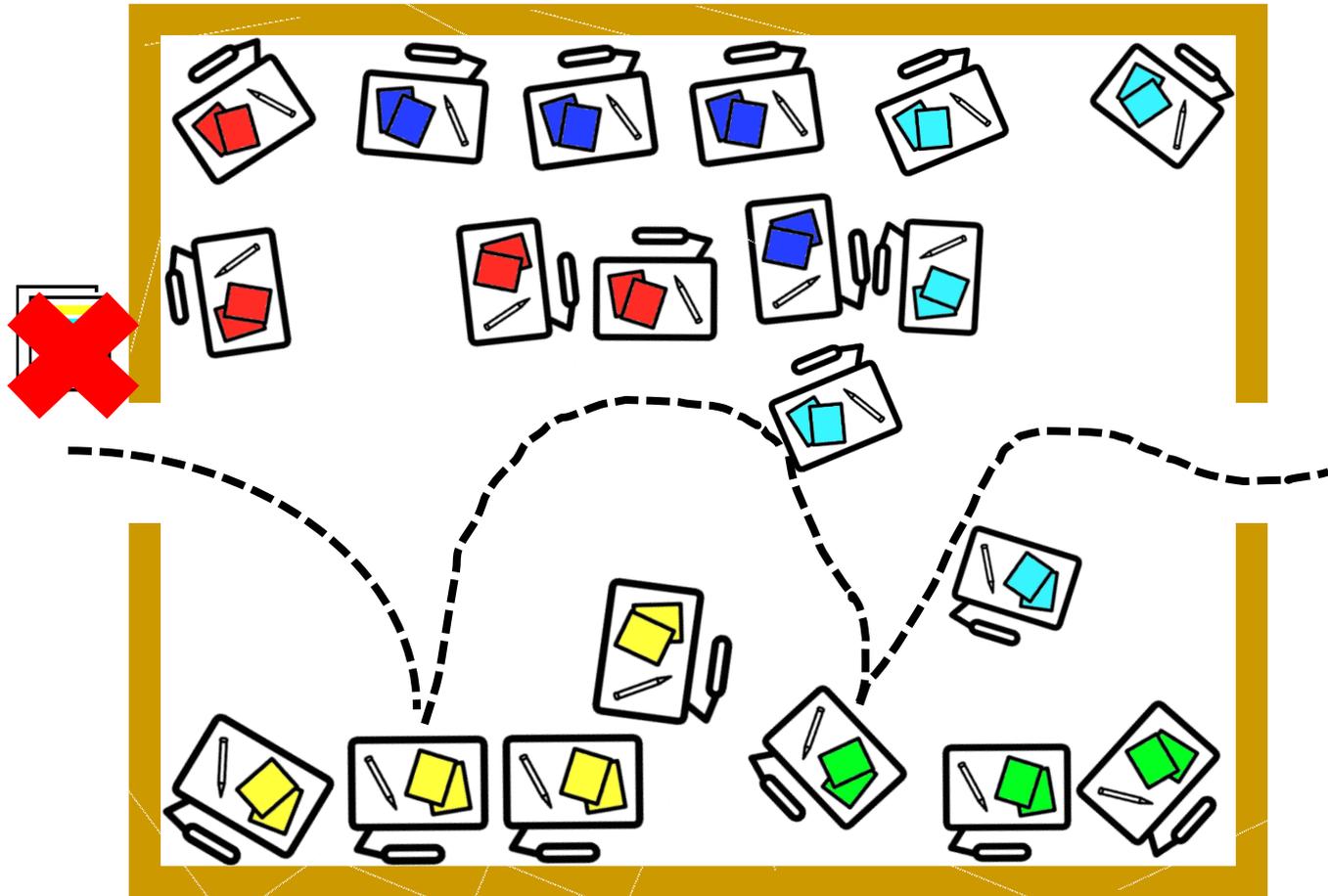
# Service-Oriented $\mu$ -Architecture



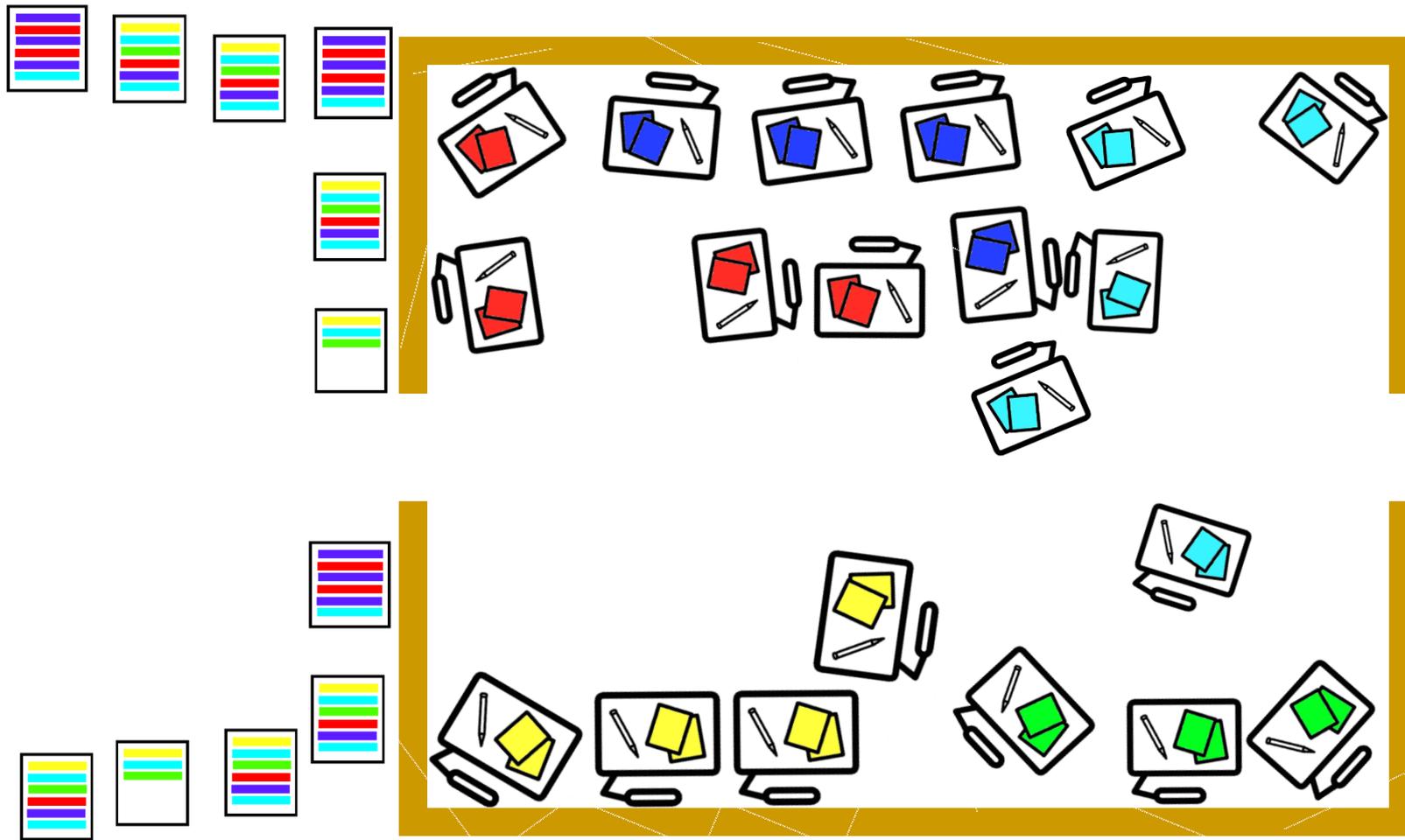
# Service-Oriented $\mu$ -Architecture



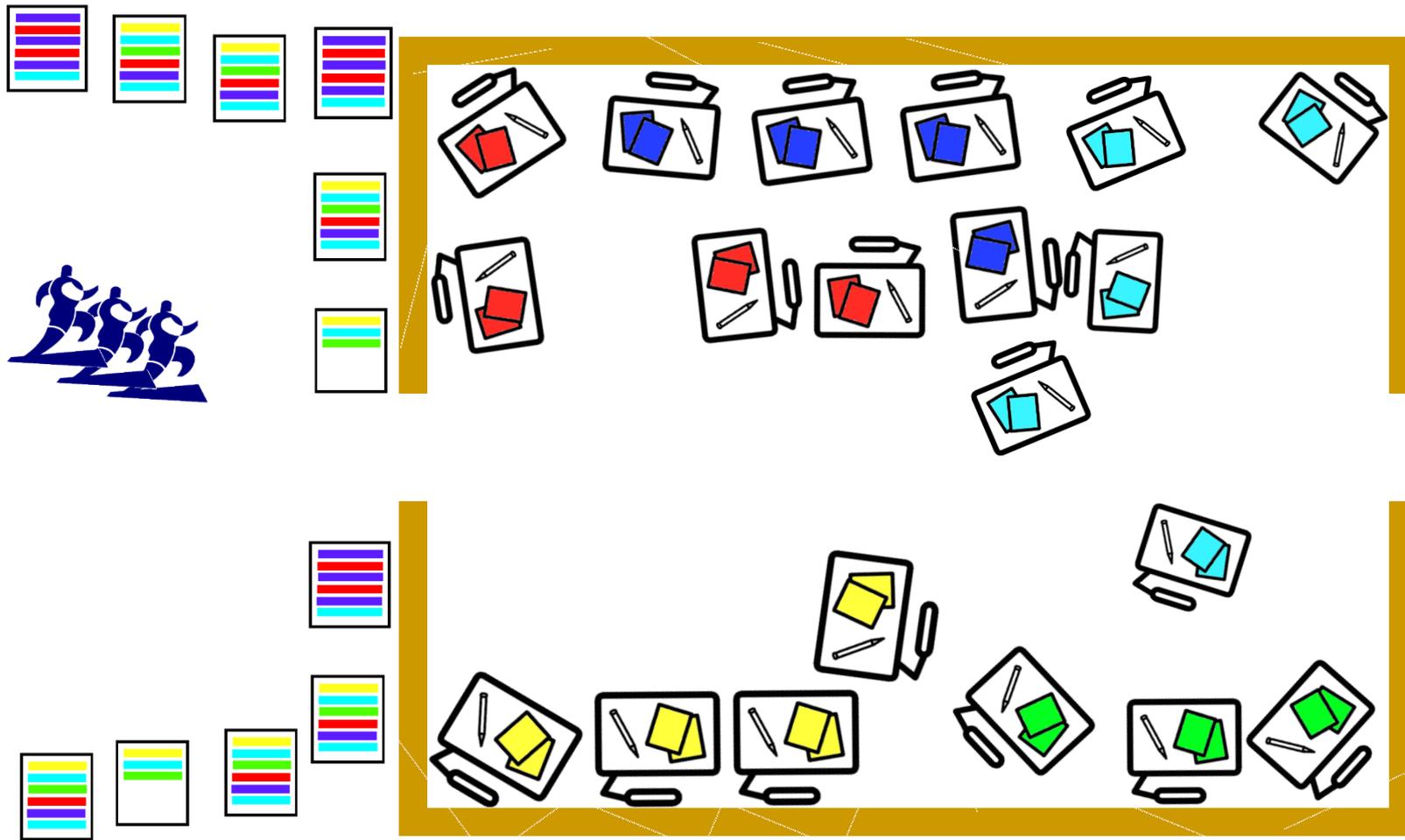
# Service-Oriented $\mu$ -Architecture



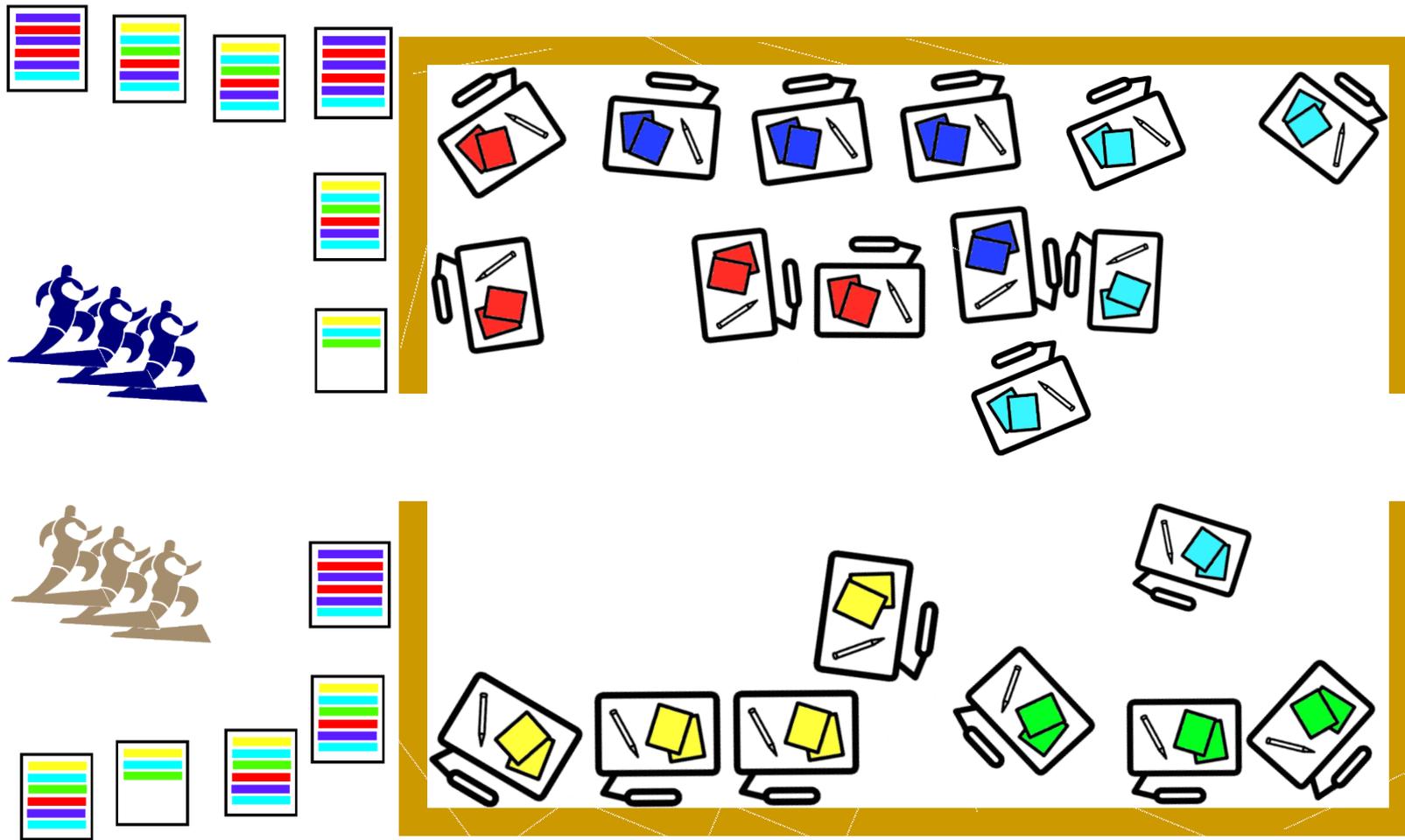
# Service-Oriented $\mu$ -Architecture



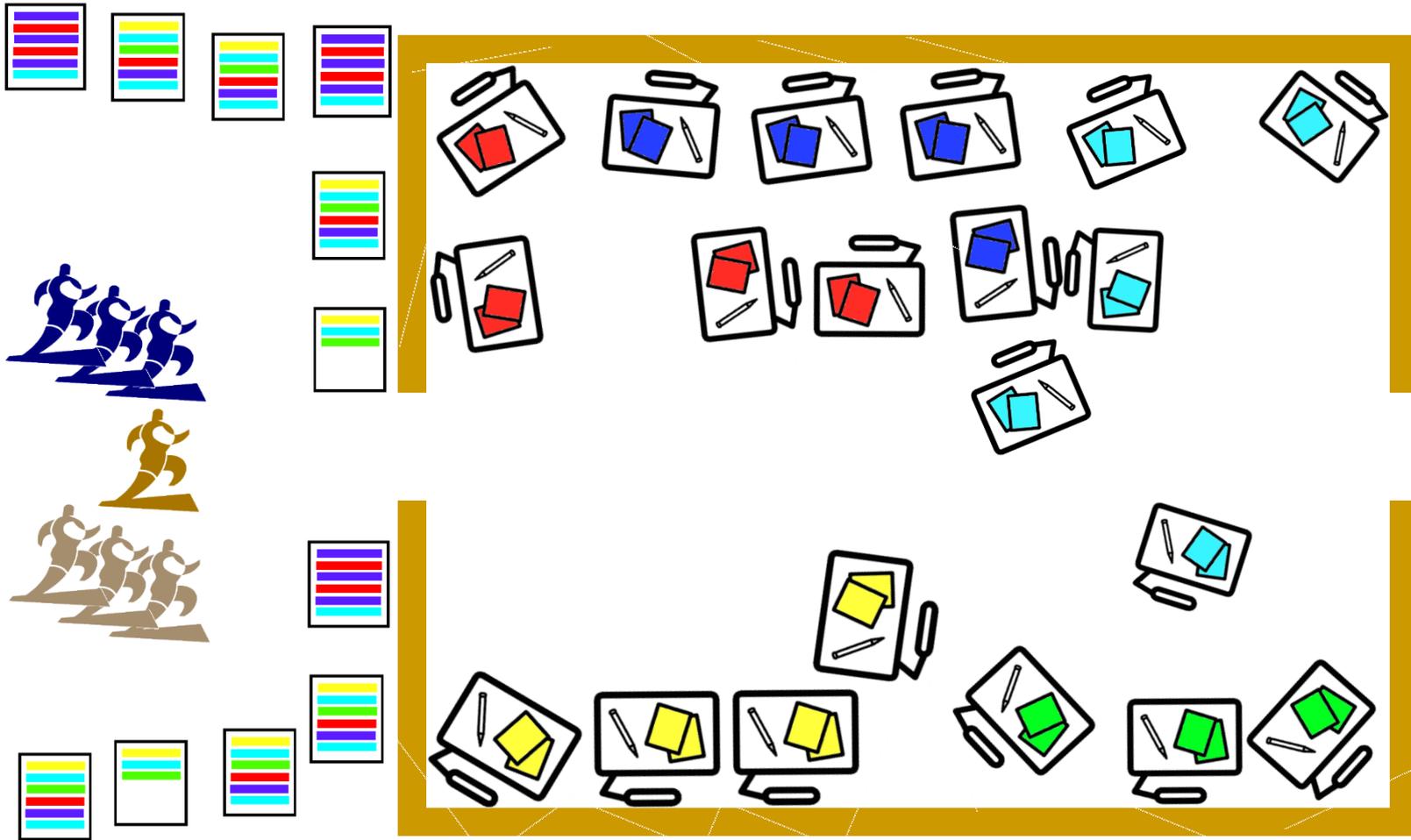
# Service-Oriented $\mu$ -Architecture



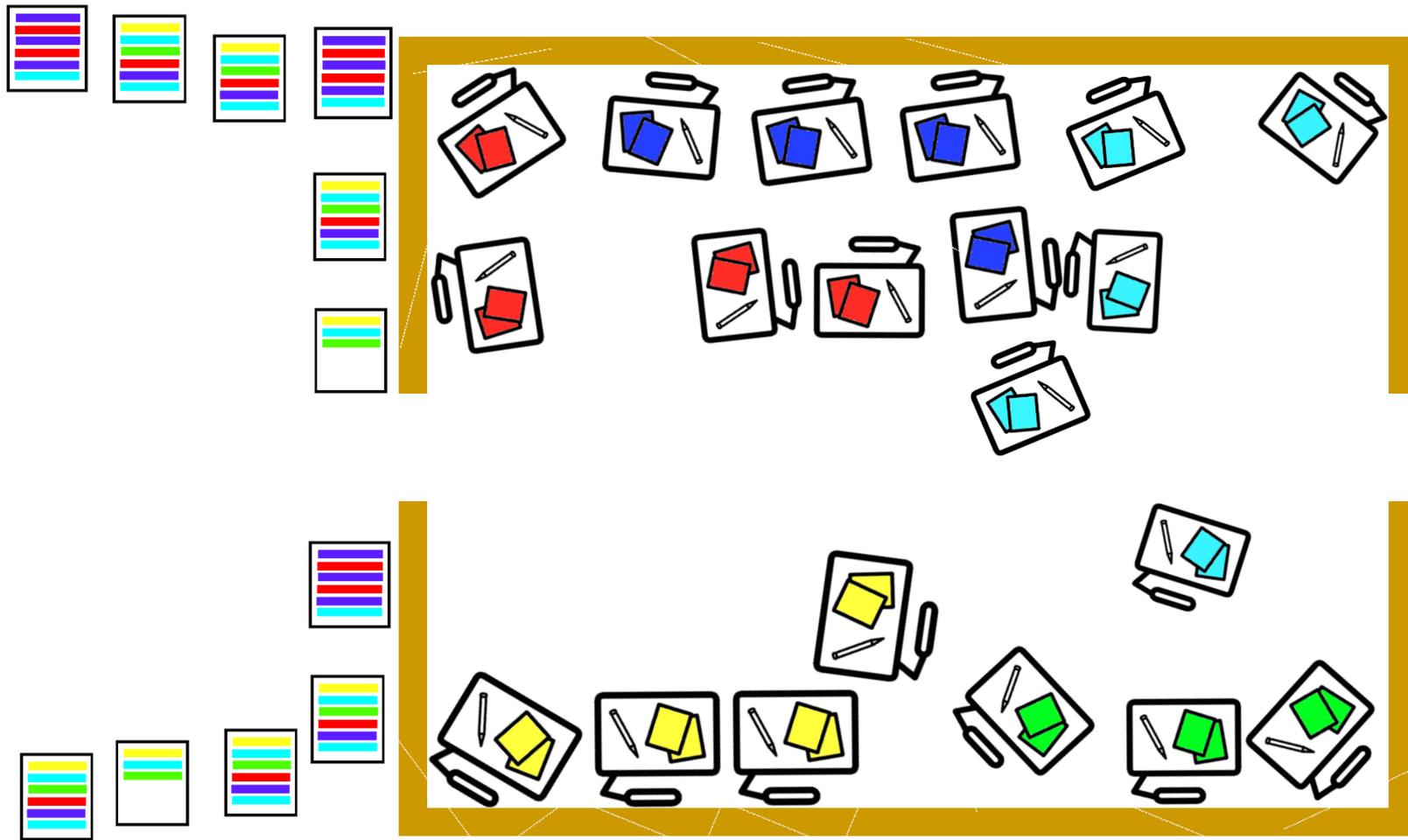
# Service-Oriented $\mu$ -Architecture



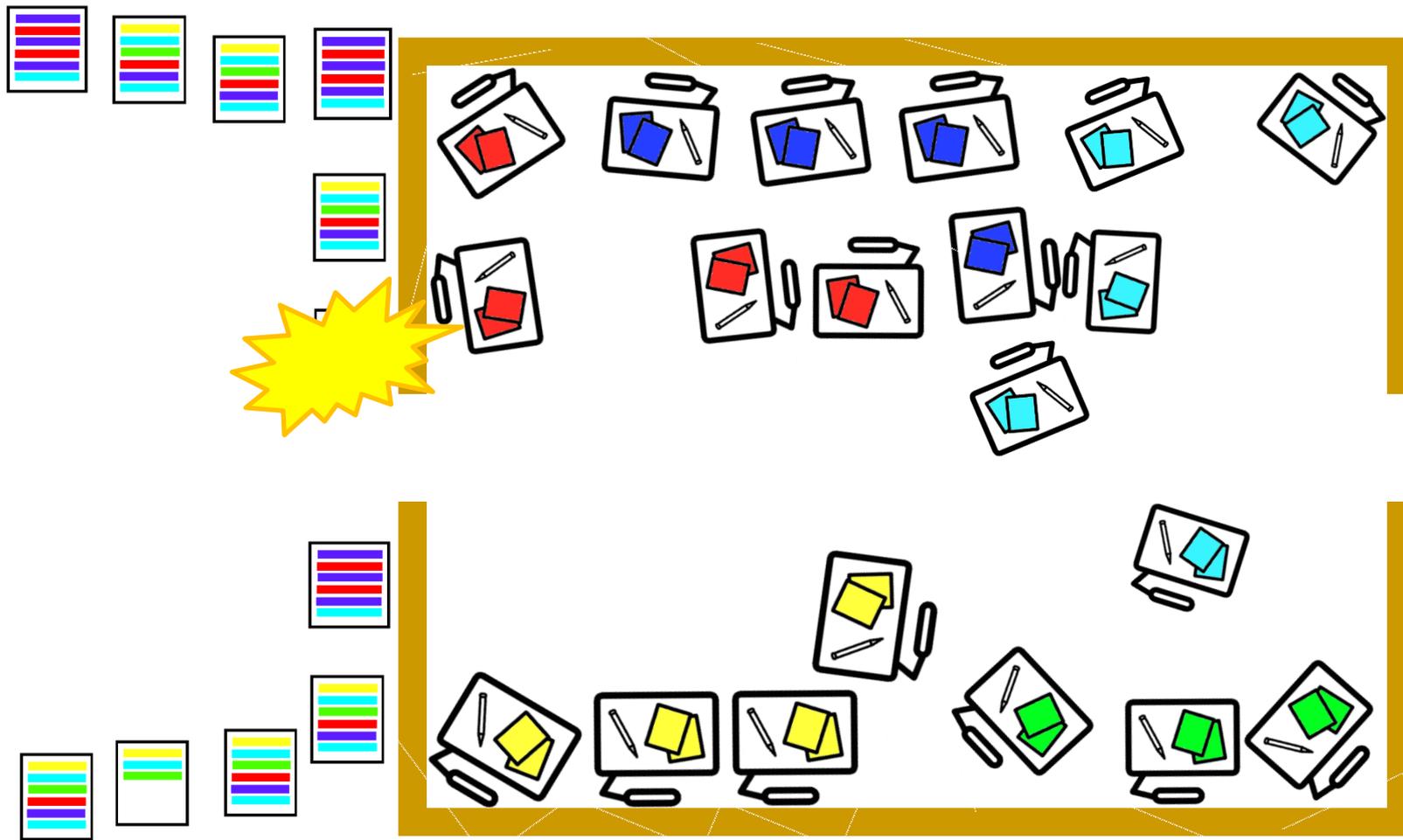
# Service-Oriented $\mu$ -Architecture



# Service-Oriented $\mu$ -Architecture



# Service-Oriented $\mu$ -Architecture

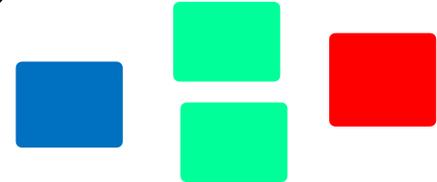


---

# Viper - Overview

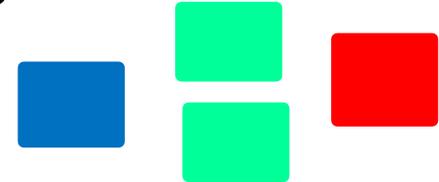
# Viper - Overview

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



# Viper - Overview

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



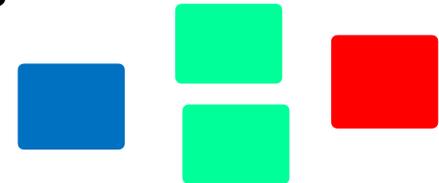
- **Bundles** are instruction sequences terminating with a control instruction (JMP)



```
4013c3:add %a1, [%ebx]
4013c5:div cl
4013c8: jmp 40140a
```

# Viper - Overview

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

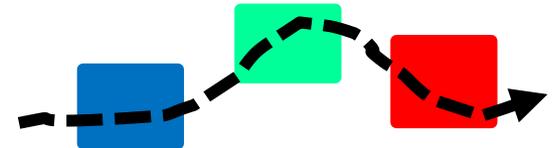


- **Bundles** are instruction sequences terminating with a control instruction (JMP)



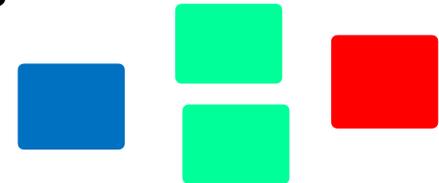
```
4013c3:add %a1, [%ebx]
4013c5:div cl
4013c8: jmp 40140a
```

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



# Viper - Overview

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

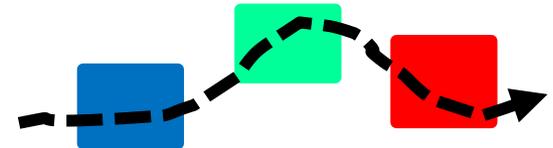


- **Bundles** are instruction sequences terminating with a control instruction (JMP)



```
4013c3:add %a1, [%ebx]
4013c5:div cl
4013c8: jmp 40140a
```

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

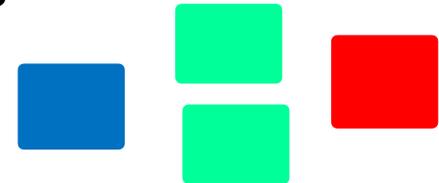


- **Bundle Scheduling Unit** allows instructions to use and be scheduled on the available HW units



# Viper - Overview

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

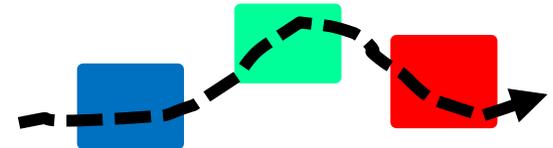


- **Bundles** are instruction sequences terminating with a control instruction (JMP)



```
4013c3:add %a1, [%ebx]
4013c5:div cl
4013c8: jmp 40140a
```

- 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

---

# Viper Hardware Organization

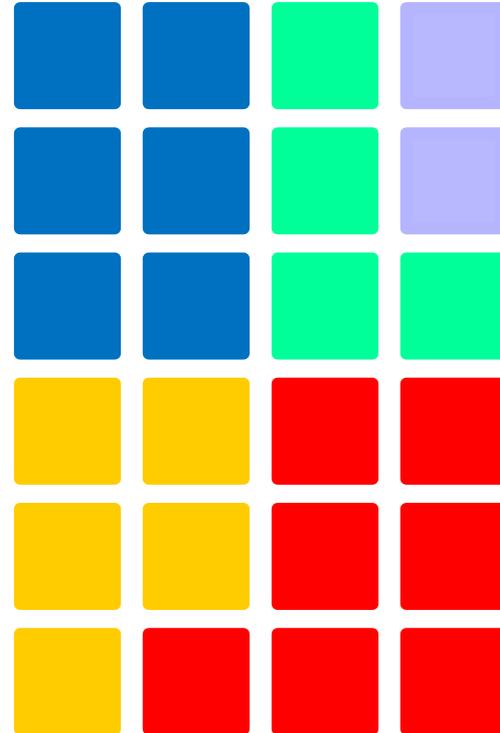
—

—

# Viper Hardware Organization



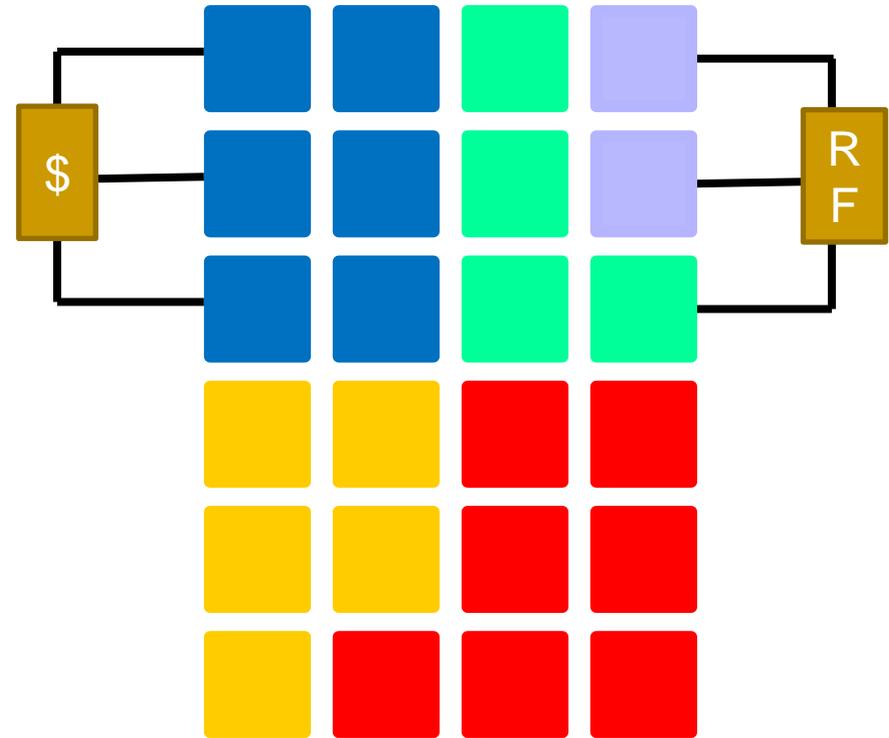
**Sea of redundant  
HW modules**



# Viper Hardware Organization



Sea of redundant  
HW modules

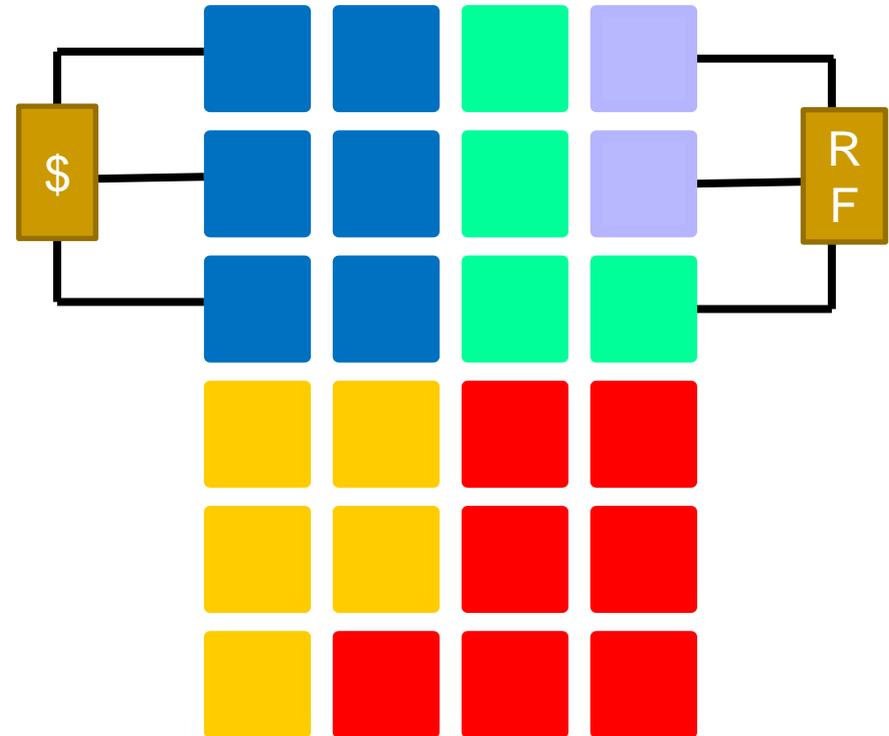


# Viper Hardware Organization



Sea of redundant  
HW modules

~~Single point of failure~~

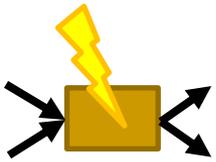


# Viper Hardware Organization

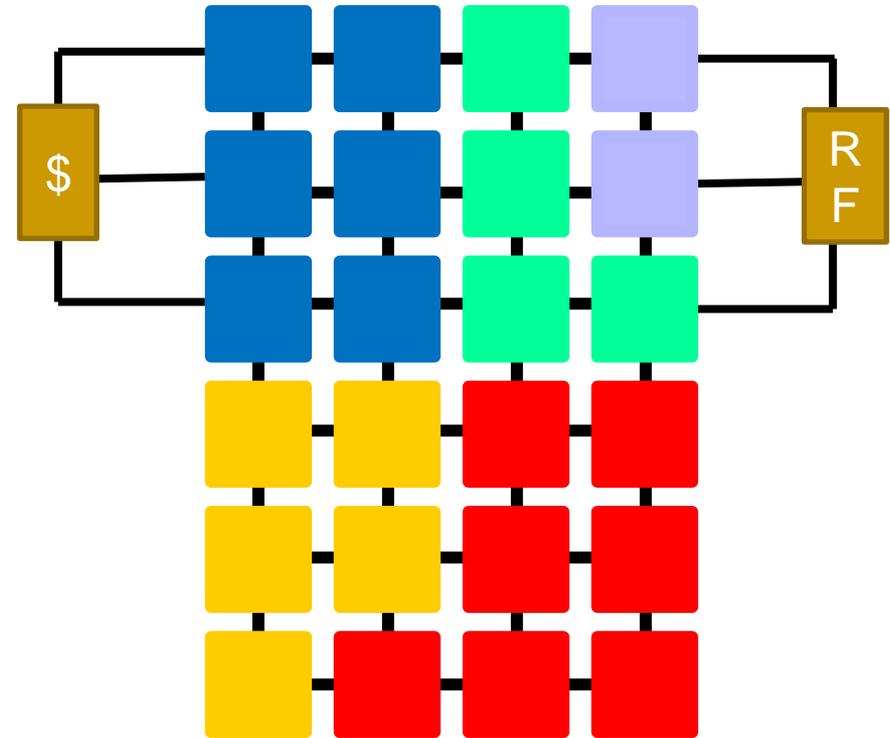


Sea of redundant  
HW modules

~~Single point of failure~~



Homogenous module  
interconnect

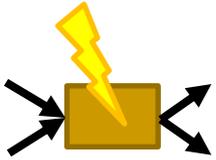


# Viper Hardware Organization



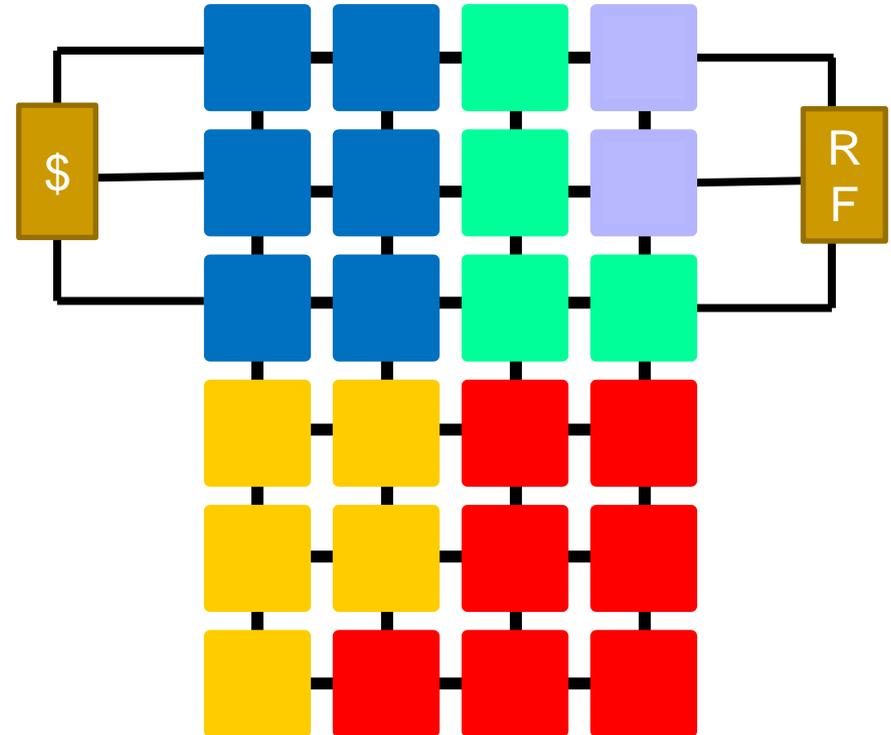
Sea of redundant  
HW modules

~~Single point of failure~~



Homogenous module  
interconnect

~~Rigidly connected  
hardware modules~~



# Viper Hardware Organization



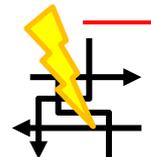
Sea of redundant HW modules

~~Single point of failure~~

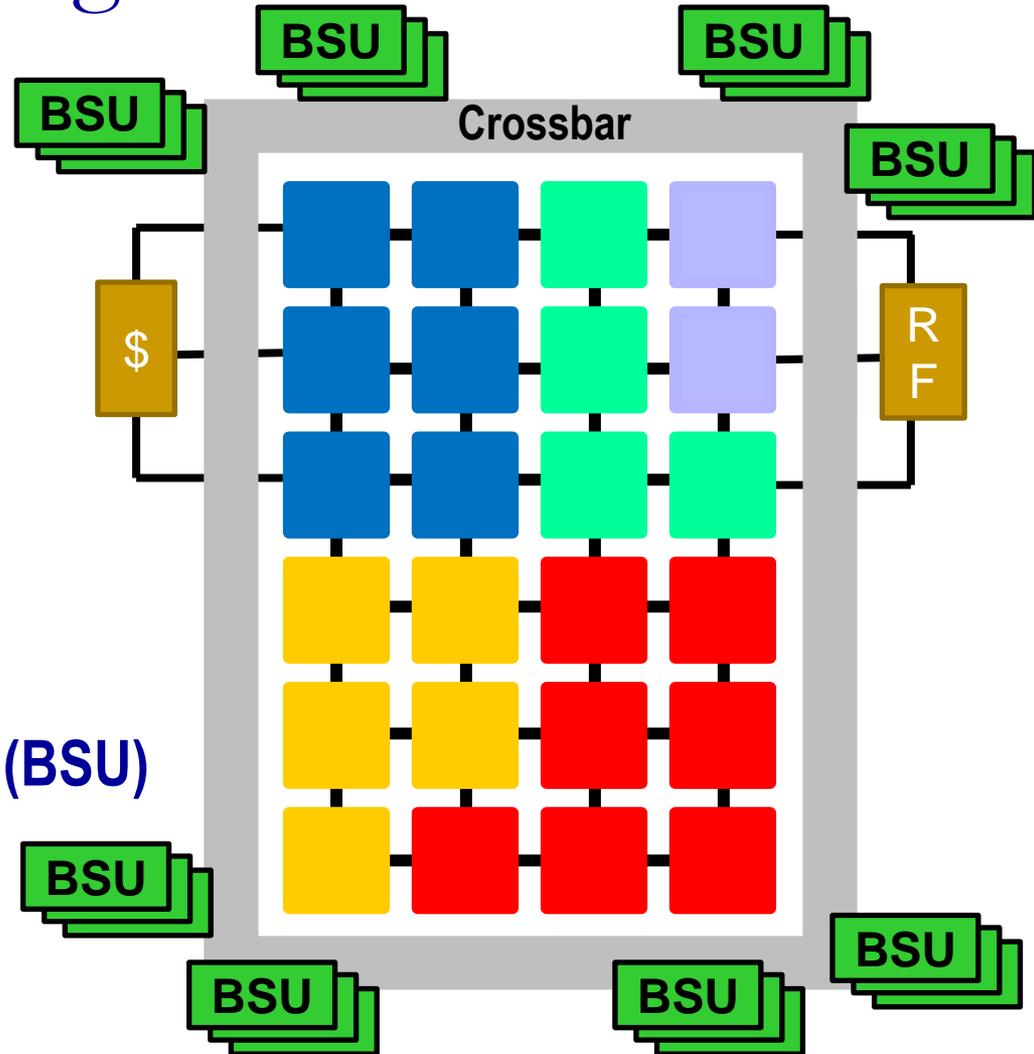


Homogenous module interconnect

~~Rigidly connected hardware modules~~



Bundle Scheduling Units (BSU) to schedule HW modules



# Viper Hardware Organization



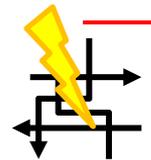
Sea of redundant HW modules

~~Single point of failure~~



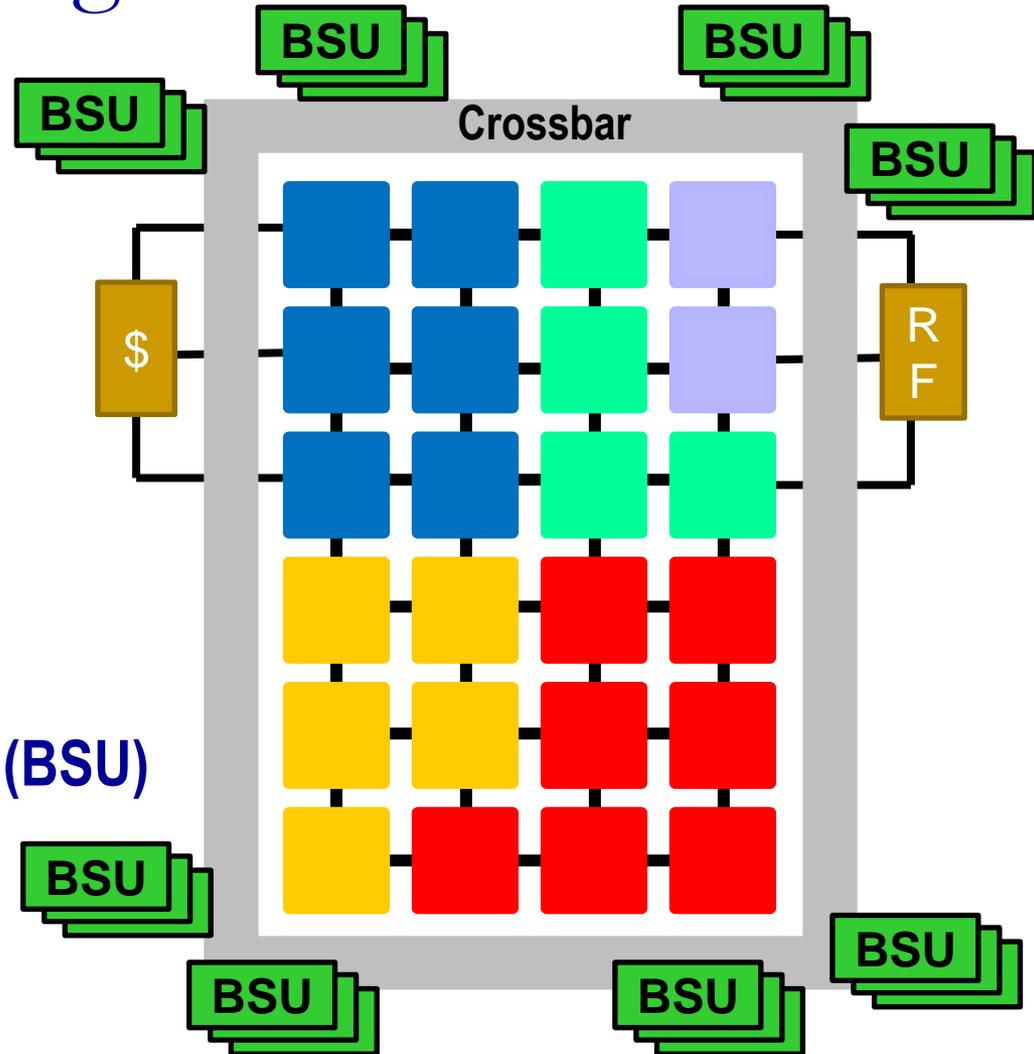
Homogenous module interconnect

~~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

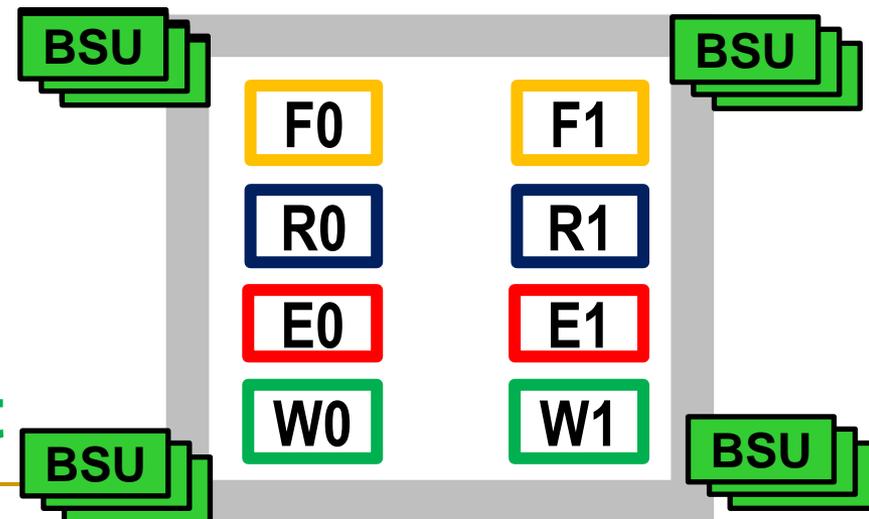
1. *Building Virtual Pipelines*
2. *Inter-Cluster Dependencies*
3. *Mispredictions*
4. *Precise Exceptions*

Fetch instruction

Register access

Execute

Write back/Commit



# 1. Building Virtual Pipelines

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

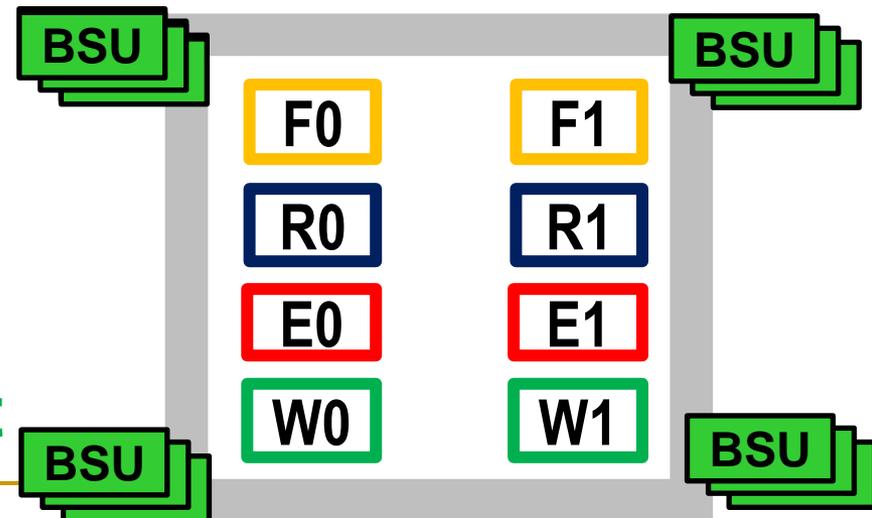
```
4013c3: add %a1, [%ebx]
```

Fetch instruction

Register access

Execute

Write back/Commit



# 1. Building Virtual Pipelines

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

```
4013c3: add %a1, [%ebx]
```

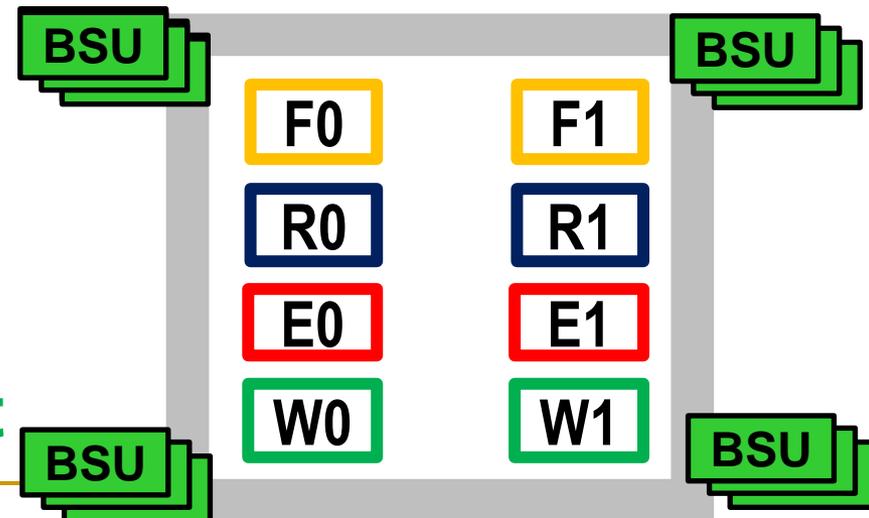
```
4013c8: or %a1, %b1
```

Fetch instruction

Register access

Execute

Write back/Commit



# 1. Building Virtual Pipelines

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

*BSU – Bundle Scheduling Unit*

```
4013c3: add %a1, [%ebx]
```

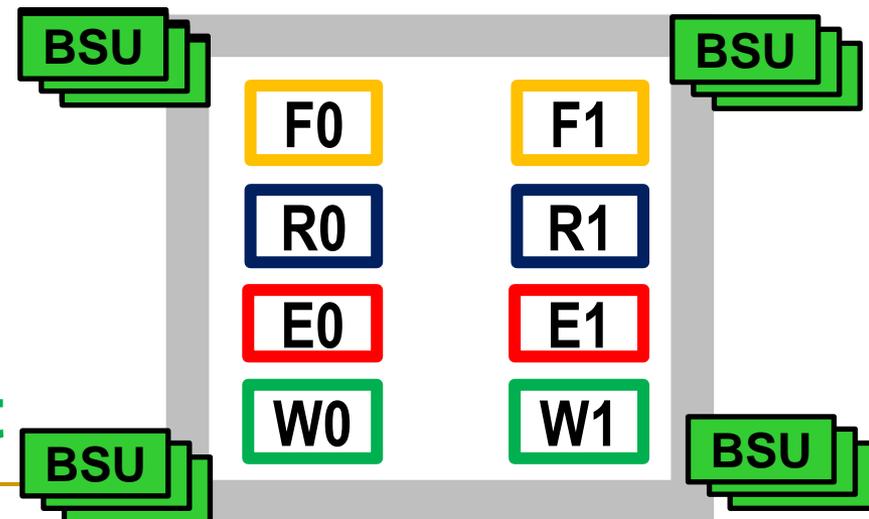
```
4013c8: or %a1, %b1
```

Fetch instruction

Register access

Execute

Write back/Commit



# 1. Building Virtual Pipelines

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

BSU – Bundle Scheduling Unit

4013c3: add %a1, [%ebx]

4013c8: or %a1, %b1

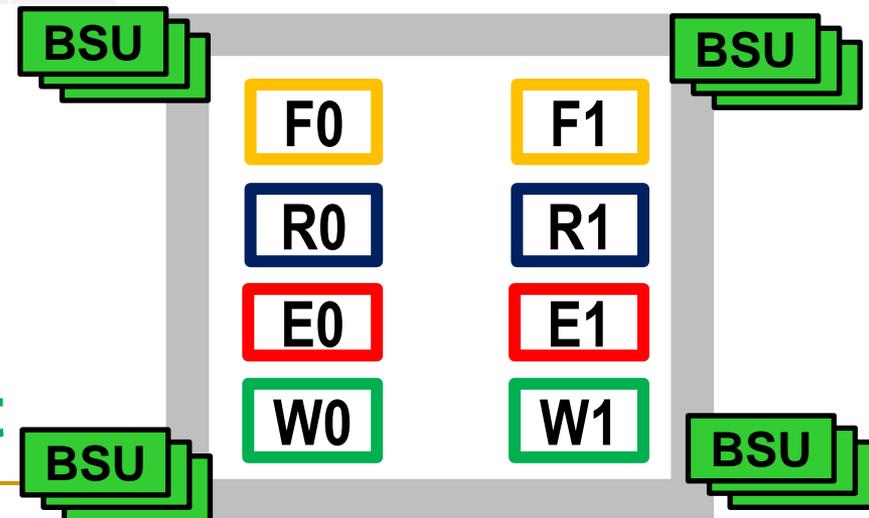
BSU ID	PC	NPC	Fetch	Reg	Exec	WB
1	4013c3					
2						
3						

Fetch instruction

Register access

Execute

Write back/Commit



# 1. Building Virtual Pipelines

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

4013c3: add %a1, [%ebx]

4013c8: or %a1, %b1

BSU – Bundle Scheduling Unit

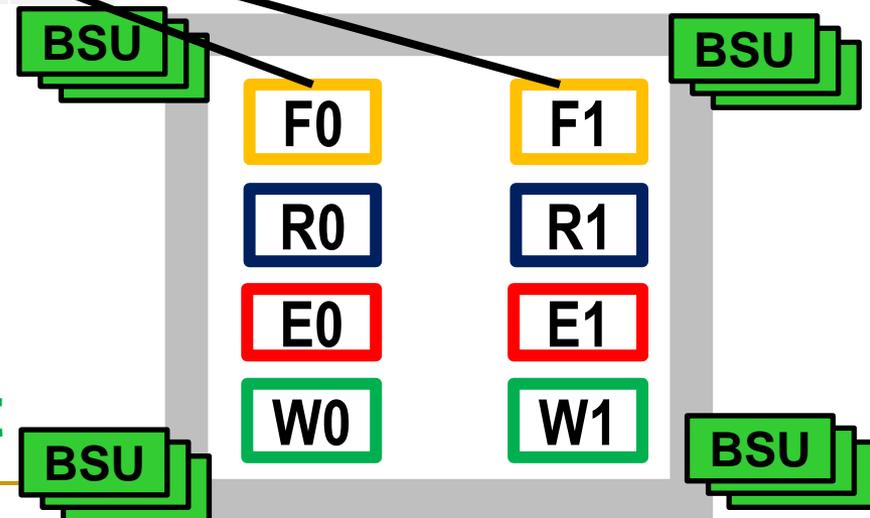
BSU ID	PC	NPC	Fetch	Reg	Exec	WB
1	4013c3					
2						
3						

Fetch instruction

Register access

Execute

Write back/Commit



# 1. Building Virtual Pipelines

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

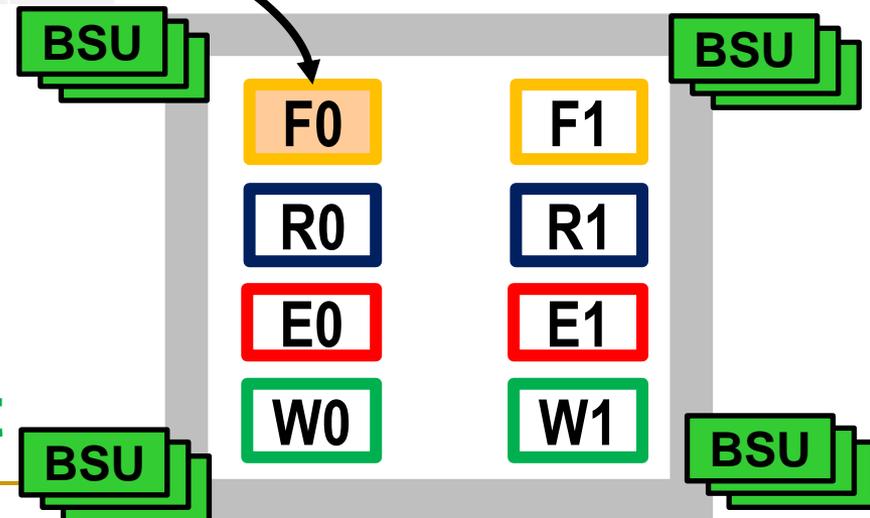
BSU – Bundle Scheduling Unit

4013c3: add %a1, [%ebx]

4013c8: or %a1, %b1

BSU ID	PC	NPC	Fetch	Reg	Exec	WB
1	4013c3		F0			
2						
3						

Fetch instruction  
Register access  
Execute  
Write back/Commit



# 1. Building Virtual Pipelines

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

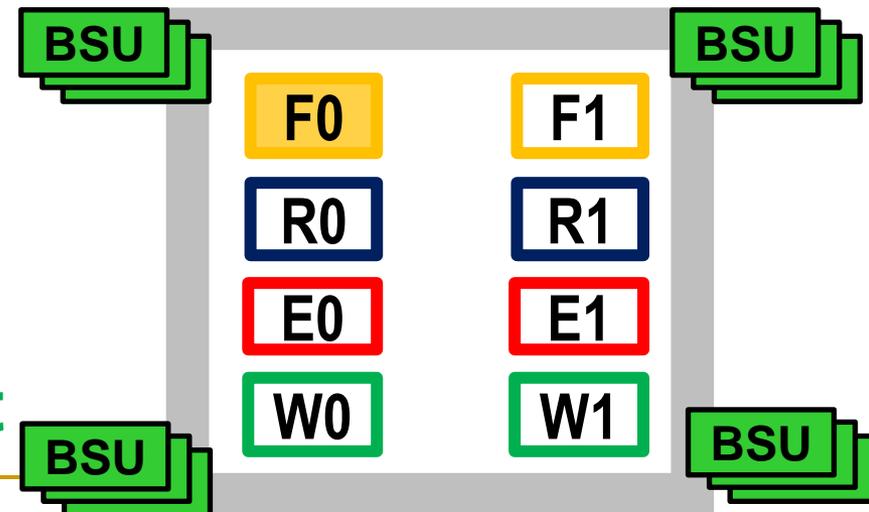
BSU – Bundle Scheduling Unit

```
4013c3: add %a1, [%ebx]
```

```
4013c8: or %a1, %b1
```

BSU ID	PC	NPC	Fetch	Reg	Exec	WB
1	4013c3		F0			
2						
3						

Fetch instruction  
Register access  
Execute  
Write back/Commit



# 1. Building Virtual Pipelines

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

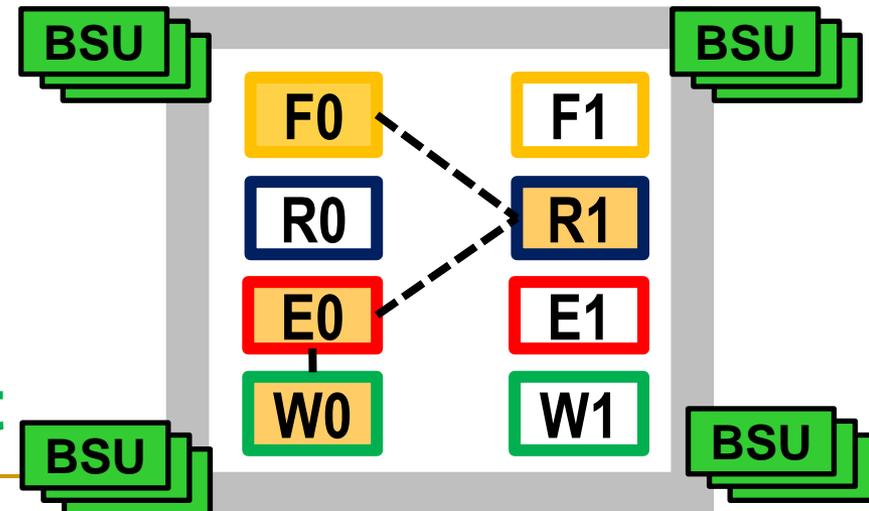
BSU – Bundle Scheduling Unit

```
4013c3: add %a1, [%ebx]
```

```
4013c8: or %a1, %b1
```

BSU ID	PC	NPC	Fetch	Reg	Exec	WB
1	4013c3		F0	R1	E0	W0
2						
3						

Fetch instruction  
Register access  
Execute  
Write back/Commit



# 1. Building Virtual Pipelines

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

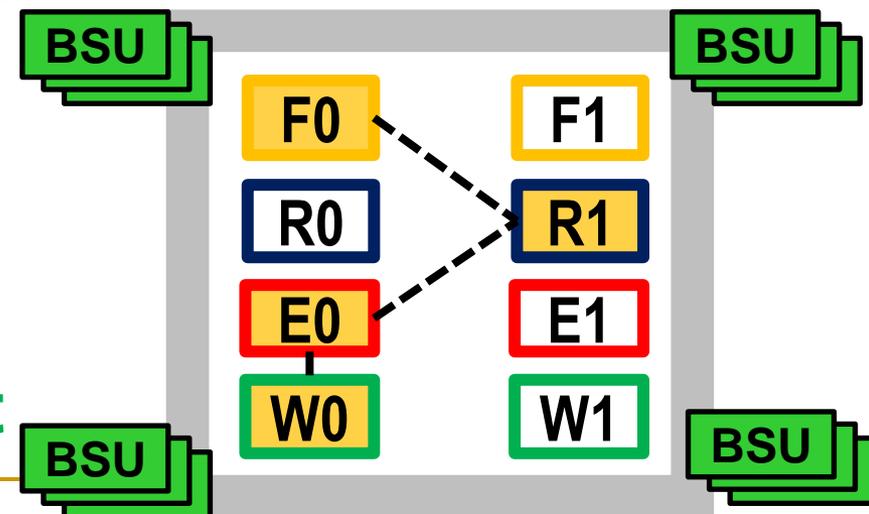
BSU – Bundle Scheduling Unit

```
4013c3: add %a1, [%ebx]
```

```
4013c8: or %a1, %b1
```

BSU ID	PC	NPC	Fetch	Reg	Exec	WB
1	4013c3		F0	R1	E0	W0
2						
3						

Fetch instruction  
Register access  
Execute  
Write back/Commit



# 1. Building Virtual Pipelines

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

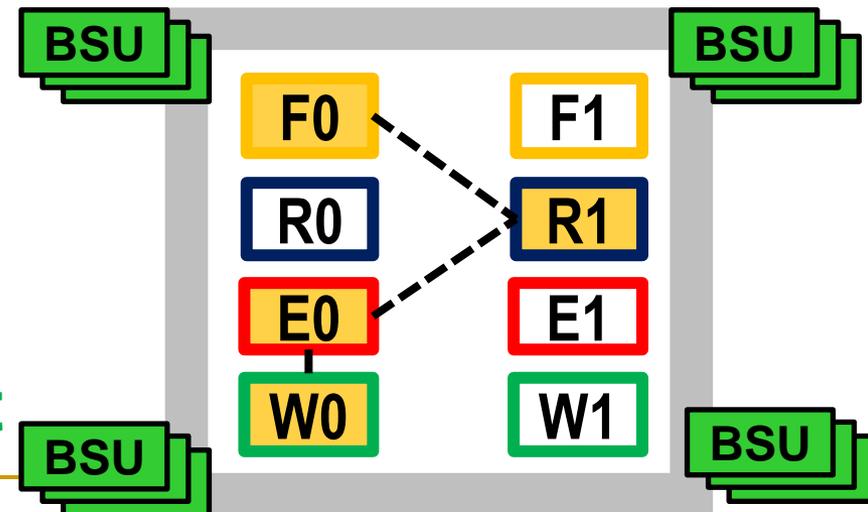
*BSU – Bundle Scheduling Unit*

4013c3: add %a1, [%ebx]

4013c8: or %a1, %b1

BSU ID	PC	NPC	Fetch	Reg	Exec	WB
1	4013c3	4013c8	F0	R1	E0	W0
2						
3						

Fetch instruction  
Register access  
Execute  
Write back/Commit



# 1. Building Virtual Pipelines

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

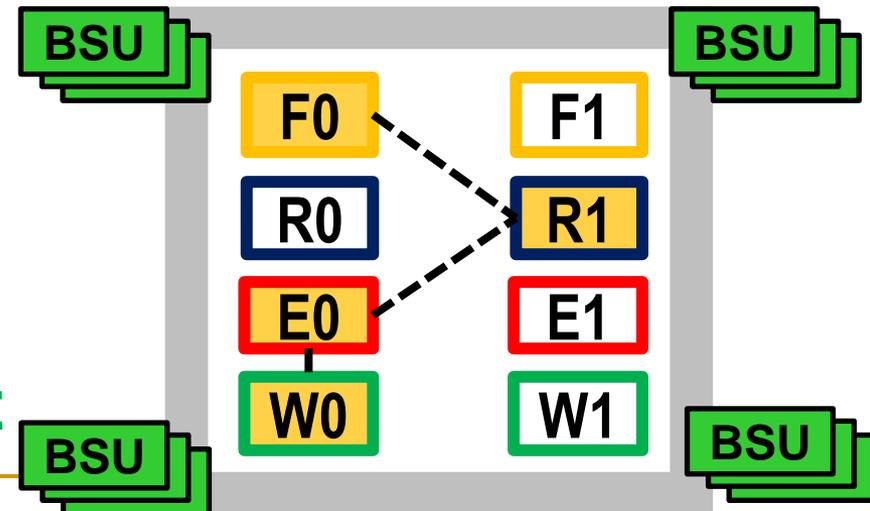
BSU – Bundle Scheduling Unit

4013c3: add %a1, [%ebx]

4013c8: or %a1, %b1

BSU ID	PC	NPC	Fetch	Reg	Exec	WB
1	4013c3	4013c8	F0	R1	E0	W0
2						
3						

Fetch instruction  
Register access  
Execute  
Write back/Commit



# 1. Building Virtual Pipelines

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

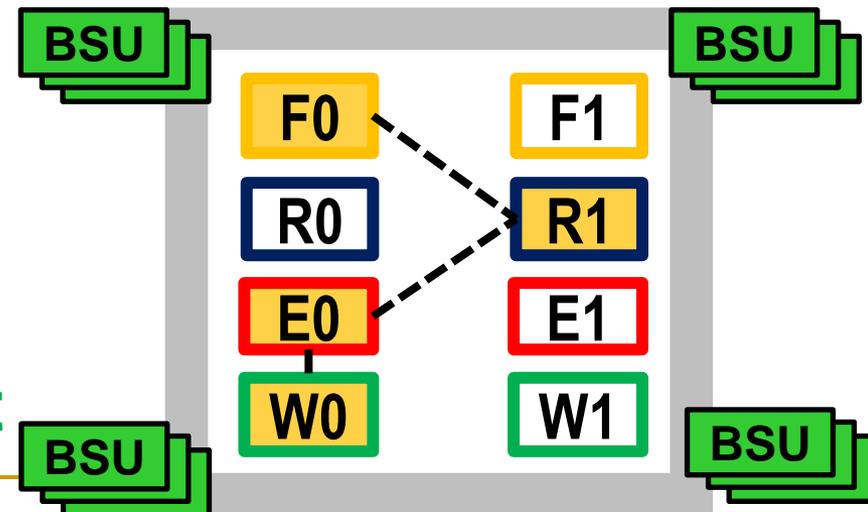
BSU – Bundle Scheduling Unit

4013c3: add %a1, [%ebx]

4013c8: or %a1, %b1

BSU ID	PC	NPC	Fetch	Reg	Exec	WB
1	4013c3	4013c8	F0	R1	E0	W0
2	4013c8					
3						

Fetch instruction  
 Register access  
 Execute  
 Write back/Commit



# 1. Building Virtual Pipelines

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

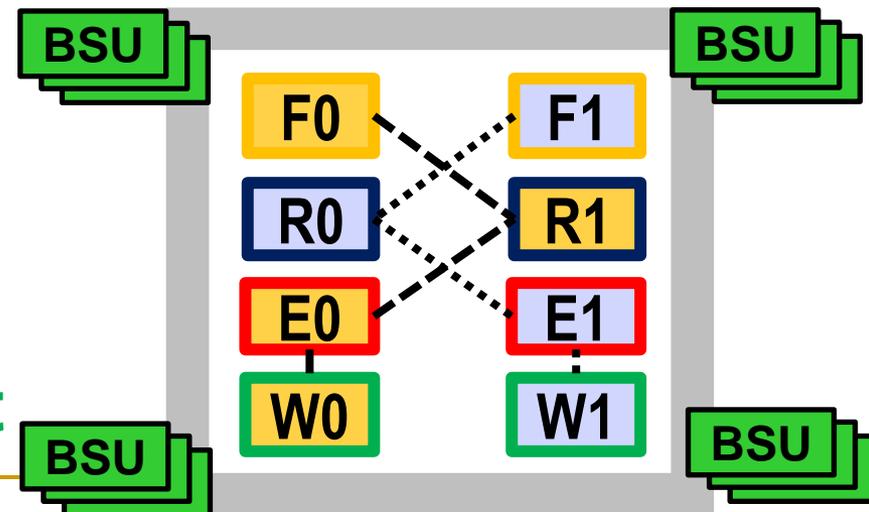
4013c3: add %a1, [%ebx]

4013c8: or %a1, %b1

BSU – Bundle Scheduling Unit

BSU ID	PC	NPC	Fetch	Reg	Exec	WB
1	4013c3	4013c8	F0	R1	E0	W0
2	4013c8		F1	R0	E1	W1
3						

Fetch instruction  
 Register access  
 Execute  
 Write back/Commit



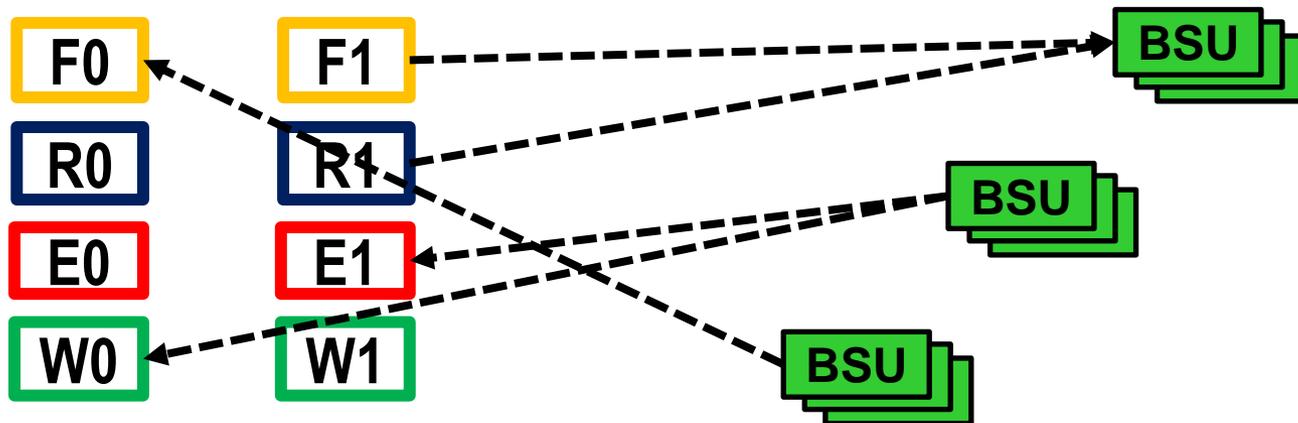
# Viper's Distributed Control Logic

- 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



# 2. Inter-Cluster Dependencies

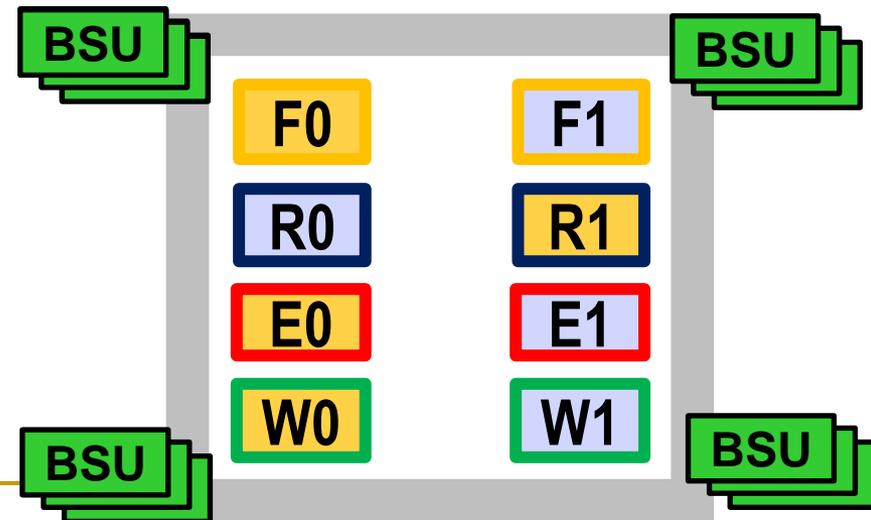
- 1. Building Virtual Pipelines
- 2. Inter-Cluster Dependencies**
- 3. Mispredictions
- 4. Precise Exceptions

*Clusters might need operands generated by others*

BSU ID	PC	NPC	Fetch	Reg	Exec	WB
1	4013c3	4013c8	F0	R1	E0	W0
2	4013c8		F1	R0	E1	W1
3						

4013c3: add %a1, [%ebx]

4013c8: or %a1, %b1

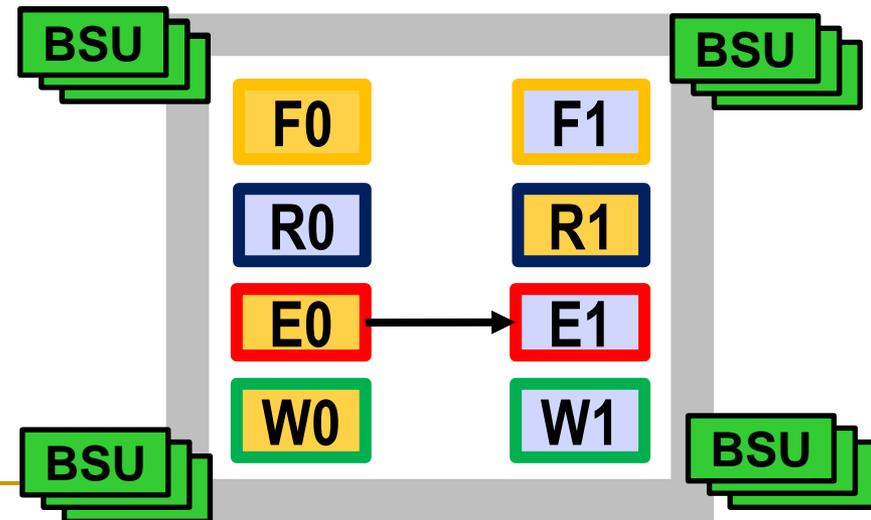
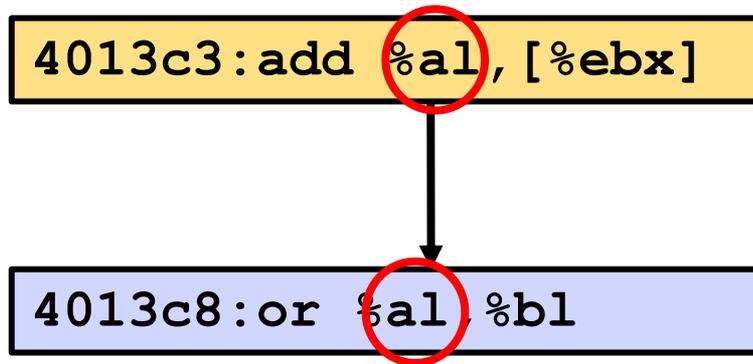


# 2. Inter-Cluster Dependencies

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

*Clusters might need operands generated by others*

BSU ID	PC	NPC	Fetch	Reg	Exec	WB
1	4013c3	4013c8	F0	R1	E0	W0
2	4013c8		F1	R0	E1	W1
3						

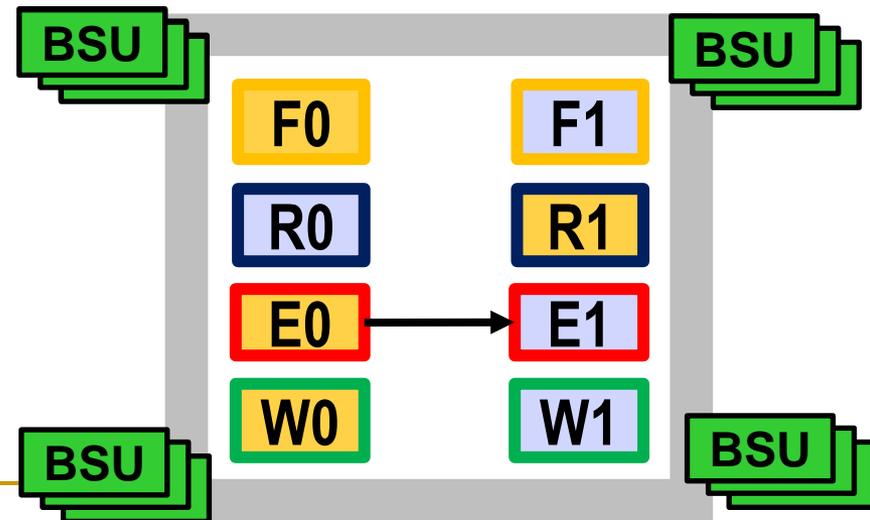
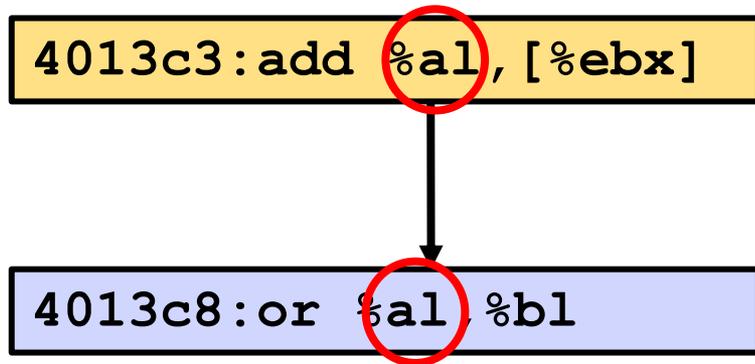


# 2. Inter-Cluster Dependencies

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

Clusters might need operands generated by others

BSU ID	PC	NPC	Fetch	Reg	Exec	WB	Input Tags				Output Tags			
							RA	RB	RC	RD	RA	RB	RC	RD
1	4013c3	4013c8	F0	R1	E0	W0								
2	4013c8		F1	R0	E1	W1								
3														

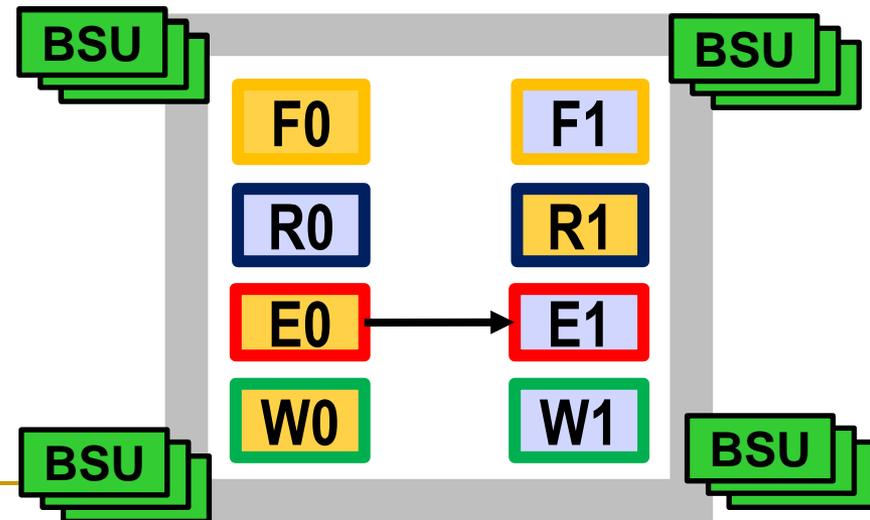
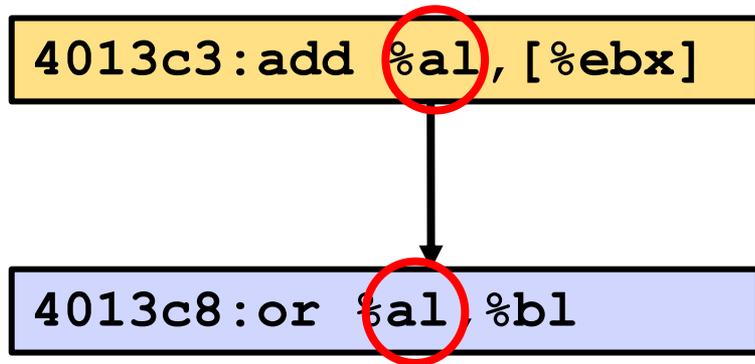


# 2. Inter-Cluster Dependencies

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

Clusters might need operands generated by others

BSU ID	PC	NPC	Fetch	Reg	Exec	WB	Input Tags				Output Tags				
							RA	RB	RC	RD	RA	RB	RC	RD	
1	4013c3	4013c8	F0	R1	E0	W0									
2	4013c8		F1	R0	E1	Additional storage required (768 bits/BSU for x86)									
3															



# 2. Inter-Cluster Dependencies

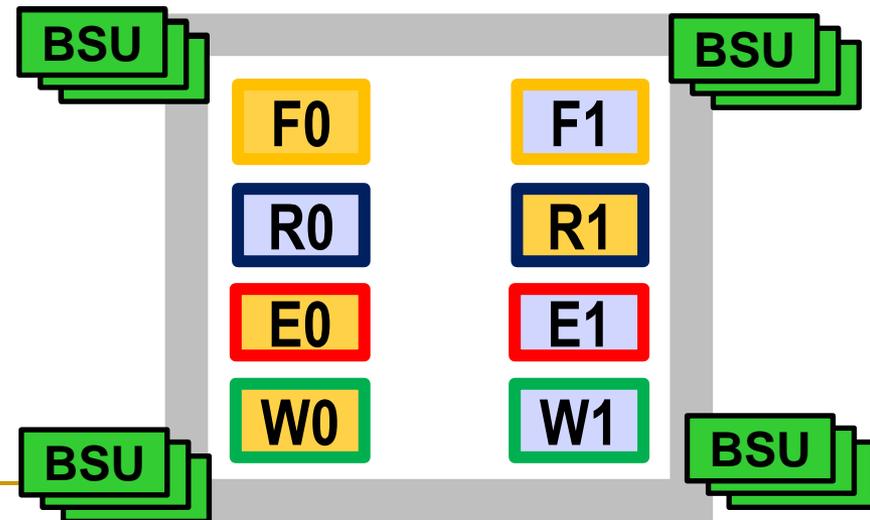
- 1. Building Virtual Pipelines
- 2. Inter-Cluster Dependencies**
- 3. Mispredictions
- 4. Precise Exceptions

Clusters might need operands generated by others

BSU ID	PC	NPC	Fetch	Reg	Exec	WB	Input Tags				Output Tags			
							RA	RB	RC	RD	RA	RB	RC	RD
1	4013c3	4013c8	F0	R1	E0	W0	1	5	6	10	1	5	6	10
2	4013c8		F1	R0	E1	W1								
3														

```
4013c3: add %a1, [%ebx]
```

```
4013c8: or %a1, %b1
```



# 2. Inter-Cluster Dependencies

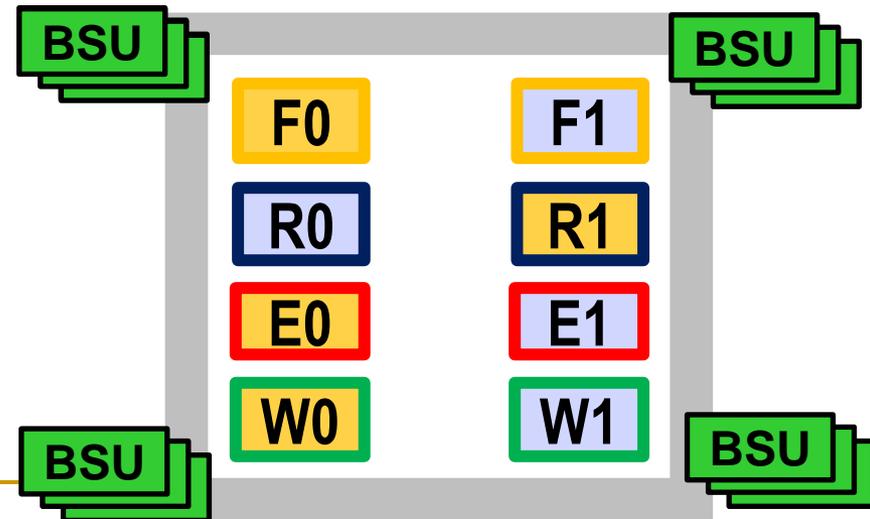
- 1. Building Virtual Pipelines
- 2. Inter-Cluster Dependencies**
- 3. Mispredictions
- 4. Precise Exceptions

Clusters might need operands generated by others

BSU ID	PC	NPC	Fetch	Reg	Exec	WB	Input Tags				Output Tags			
							RA	RB	RC	RD	RA	RB	RC	RD
1	4013c3	4013c8	F0	R1	E0	W0	1	5	6	10	1	5	6	10
2	4013c8		F1	R0	E1	W1								
3														

```
4013c3: add %a1, [%ebx]
```

```
4013c8: or %a1, %b1
```



# 2. Inter-Cluster Dependencies

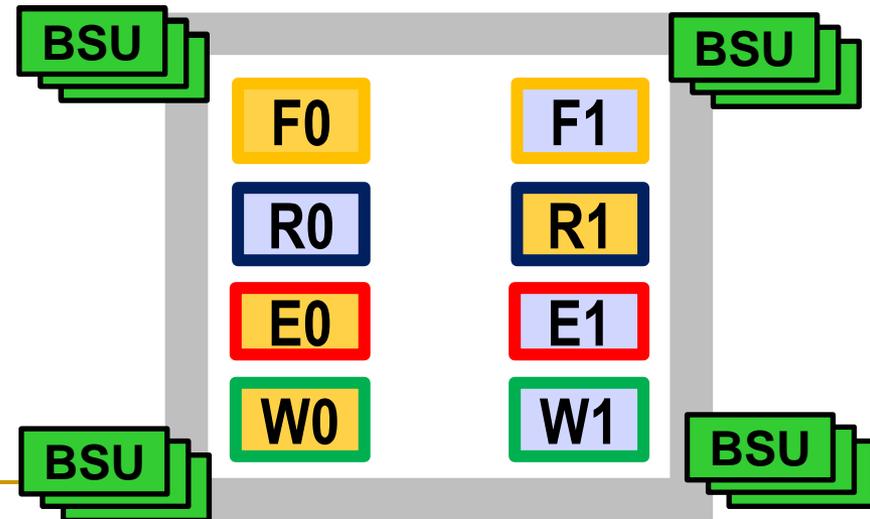
- 1. Building Virtual Pipelines
- 2. **Inter-Cluster Dependencies**
- 3. Mispredictions
- 4. Precise Exceptions

Clusters might need operands generated by others

BSU ID	PC	NPC	Fetch	Reg	Exec	WB	Input Tags				Output Tags			
							RA	RB	RC	RD	RA	RB	RC	RD
1	4013c3	4013c8	F0	R1	E0	W0	1	5	6	10	13	5	6	10
2	4013c8		F1	R0	E1	W1								
3														

```
4013c3: add %a1, [%ebx]
```

```
4013c8: or %a1, %b1
```



# 2. Inter-Cluster Dependencies

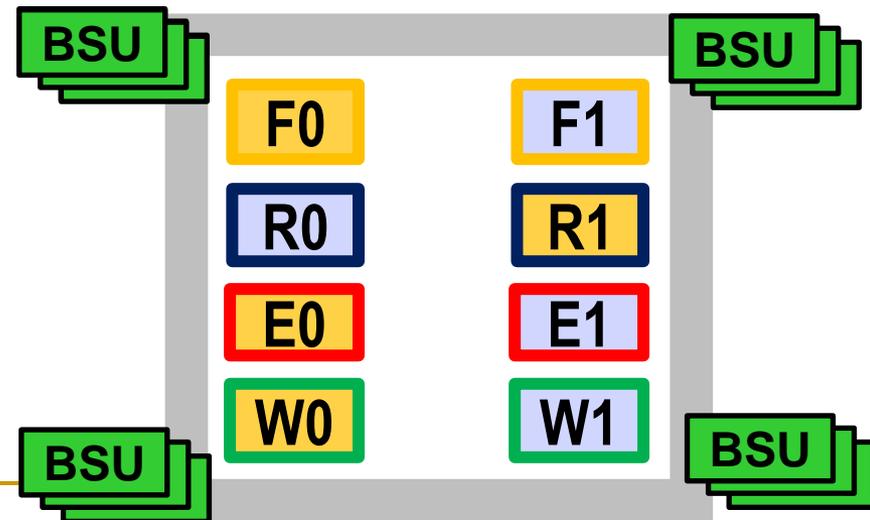
- 1. Building Virtual Pipelines
- 2. Inter-Cluster Dependencies**
- 3. Mispredictions
- 4. Precise Exceptions

*Clusters might need operands generated by others*

BSU ID	PC	NPC	Fetch	Reg	Exec	WB	Input Tags				Output Tags			
							RA	RB	RC	RD	RA	RB	RC	RD
1	4013c3	4013c8	F0	R1	E0	W0	1	5	6	10	13	5	6	10
2	4013c8		F1	R0	E1	W1								
3														

```
4013c3: add %a1, [%ebx]
```

```
4013c8: or %a1, %b1
```



# 2. Inter-Cluster Dependencies

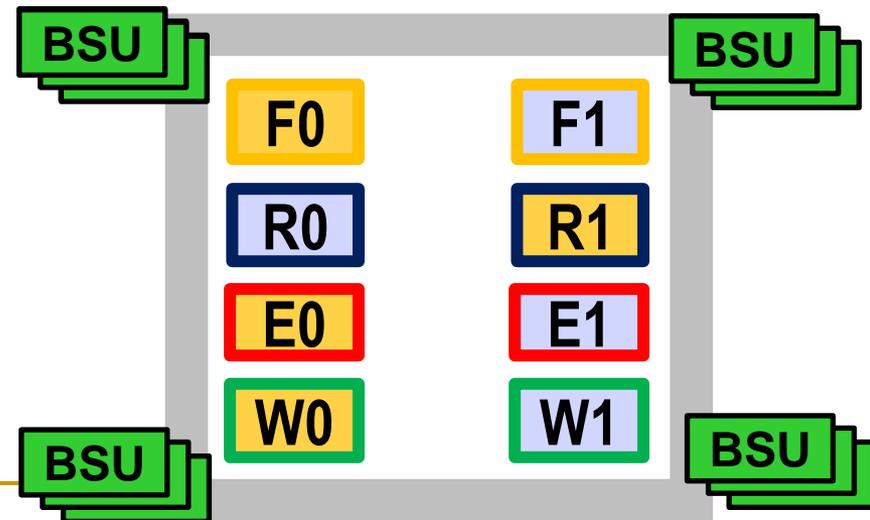
- 1. Building Virtual Pipelines
- 2. **Inter-Cluster Dependencies**
- 3. Mispredictions
- 4. Precise Exceptions

*Clusters might need operands generated by others*

BSU ID	PC	NPC	Fetch	Reg	Exec	WB	Input Tags				Output Tags			
							RA	RB	RC	RD	RA	RB	RC	RD
1	4013c3	4013c8	F0	R1	E0	W0	1	5	6	10	13	5	6	10
2	4013c8		F1	R0	E1	W1								
3														

```
4013c3: add %a1, [%ebx]
```

```
4013c8: or %a1, %b1
```



# 2. Inter-Cluster Dependencies

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

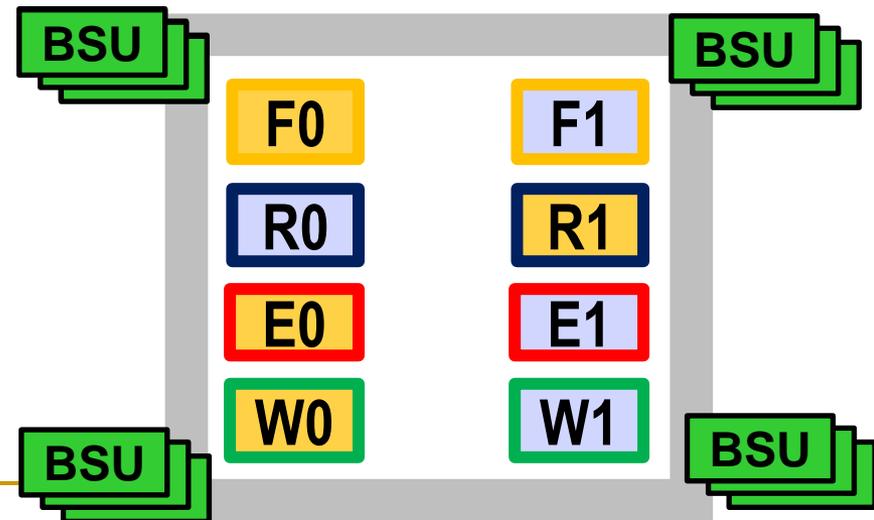
Clusters might need operands generated by others

BSU ID	PC	NPC	Fetch	Reg	Exec	WB	Input Tags				Output Tags			
							RA	RB	RC	RD	RA	RB	RC	RD
1	4013c3	4013c8	F0	R1	E0	W0	1	5	6	10	13	5	6	10
2	4013c8		F1	R0	E1	W1								
3														

Tag creation is *serialized*

```
4013c3: add %a1, [%ebx]
```

```
4013c8: or %a1, %b1
```



# 2. Inter-Cluster Dependencies

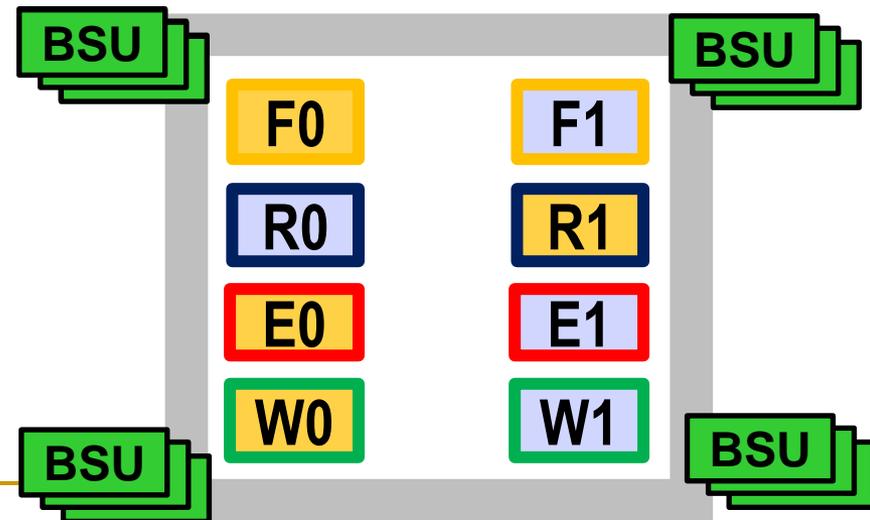
- 1. Building Virtual Pipelines
- 2. **Inter-Cluster Dependencies**
- 3. Mispredictions
- 4. Precise Exceptions

*Clusters might need operands generated by others*

BSU ID	PC	NPC	Fetch	Reg	Exec	WB	Input Tags				Output Tags			
							RA	RB	RC	RD	RA	RB	RC	RD
1	4013c3	4013c8	F0	R1	E0	W0	1	5	6	10	13	5	6	10
2	4013c8		F1	R0	E1	W1								
3														

```
4013c3: add %a1, [%ebx]
```

```
4013c8: or %a1, %b1
```



# 2. Inter-Cluster Dependencies

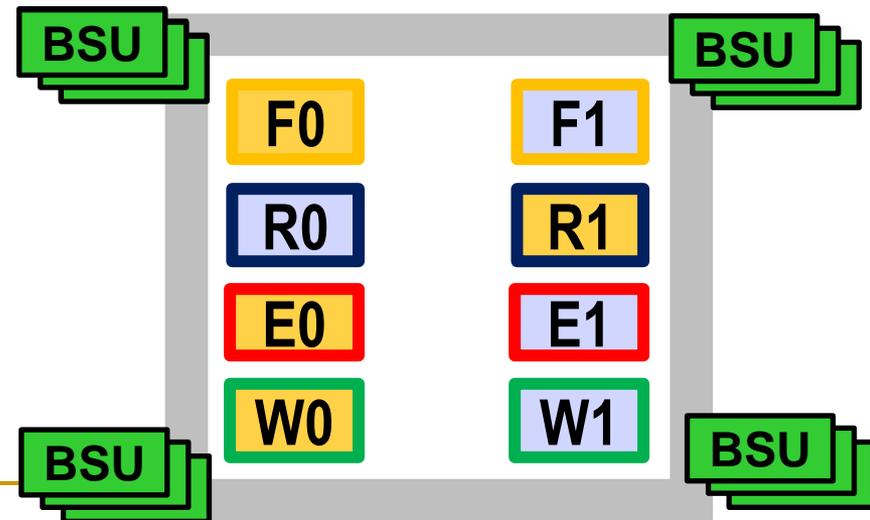
- 1. Building Virtual Pipelines
- 2. Inter-Cluster Dependencies**
- 3. Mispredictions
- 4. Precise Exceptions

*Clusters might need operands generated by others*

BSU ID	PC	NPC	Fetch	Reg	Exec	WB	Input Tags				Output Tags			
							RA	RB	RC	RD	RA	RB	RC	RD
1	4013c3	4013c8	F0	R1	E0	W0	1	5	6	10	13	5	6	10
2	4013c8		F1	R0	E1	W1	13	5	6	10				
3														

```
4013c3: add %a1, [%ebx]
```

```
4013c8: or %a1, %b1
```



# 2. Inter-Cluster Dependencies

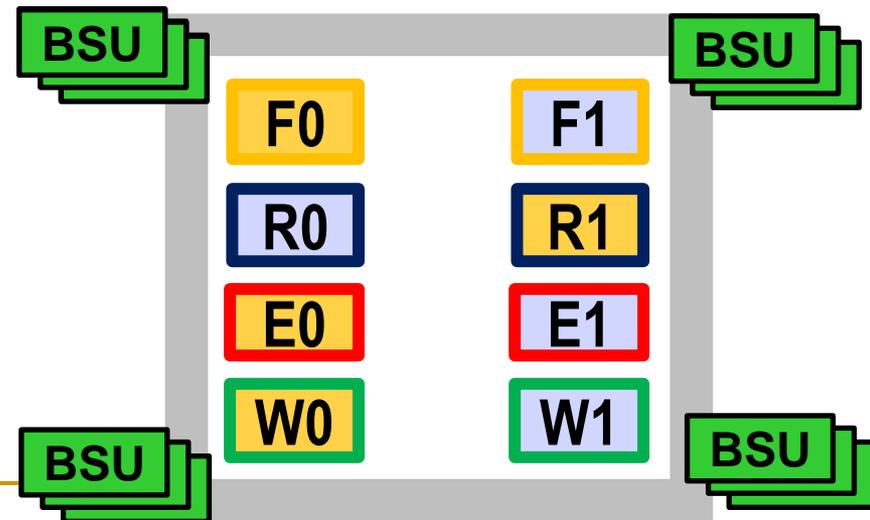
- 1. Building Virtual Pipelines
- 2. **Inter-Cluster Dependencies**
- 3. Mispredictions
- 4. Precise Exceptions

Clusters might need operands generated by others

BSU ID	PC	NPC	Fetch	Reg	Exec	WB	Input Tags				Output Tags			
							RA	RB	RC	RD	RA	RB	RC	RD
1	4013c3	4013c8	F0	R1	E0	W0	1	5	6	10	13	5	6	10
2	4013c8		F1	R0	E1	W1	13	5	6	10				
3														

```
4013c3: add %a1, [%ebx]
```

```
4013c8: or %a1, %b1
```



# 2. Inter-Cluster Dependencies

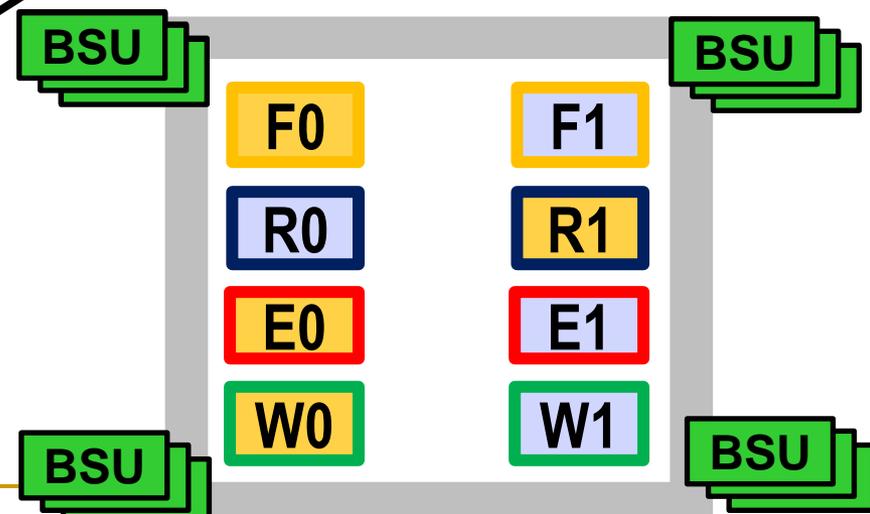
- 1. Building Virtual Pipelines
- 2. **Inter-Cluster Dependencies**
- 3. Mispredictions
- 4. Precise Exceptions

*Clusters might need operands generated by others*

BSU ID	PC	NPC	Fetch	Reg	Exec	WB	Input Tags				Output Tags			
							RA	RB	RC	RD	RA	RB	RC	RD
1	4013c3	4013c8	F0	R1	E0	W0	1	5	6	10	13	5	6	10
2	4013c8		F1	R0	E1	W1	13	5	6	10				
3														

```
4013c3:add %a1, [%ebx]
```

```
4013c8:or %a1,%b1
```



# 2. Inter-Cluster Dependencies

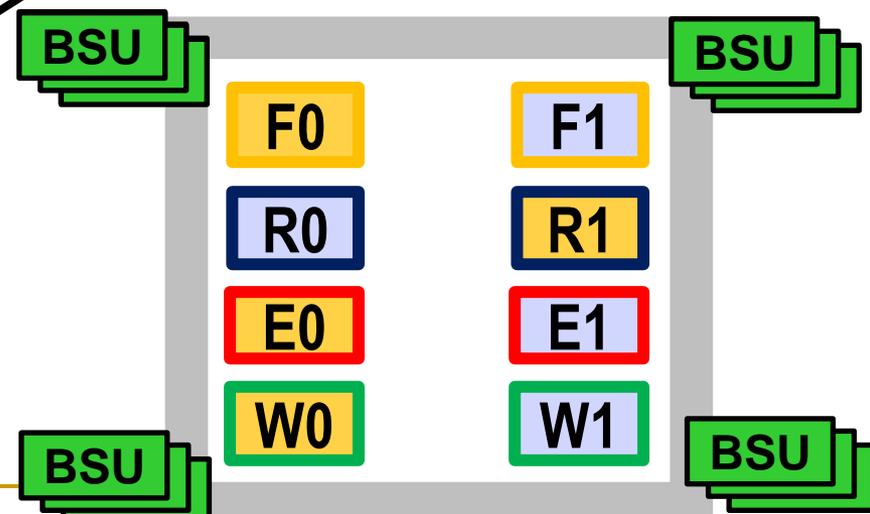
- 1. Building Virtual Pipelines
- 2. **Inter-Cluster Dependencies**
- 3. Mispredictions
- 4. Precise Exceptions

Clusters might need operands generated by others

BSU ID	PC	NPC	Fetch	Reg	Exec	WB	Input Tags				Output Tags			
							RA	RB	RC	RD	RA	RB	RC	RD
1	4013c3	4013c8	F0	R1	E0	W0	1	5	6	10	13	5	6	10
2	4013c8		F1	R0	E1	W1	13	5	6	10				
3														

```
4013c3:add %a1, [%ebx]
```

```
4013c8:or %a1, %b1
```



# 2. Inter-Cluster Dependencies

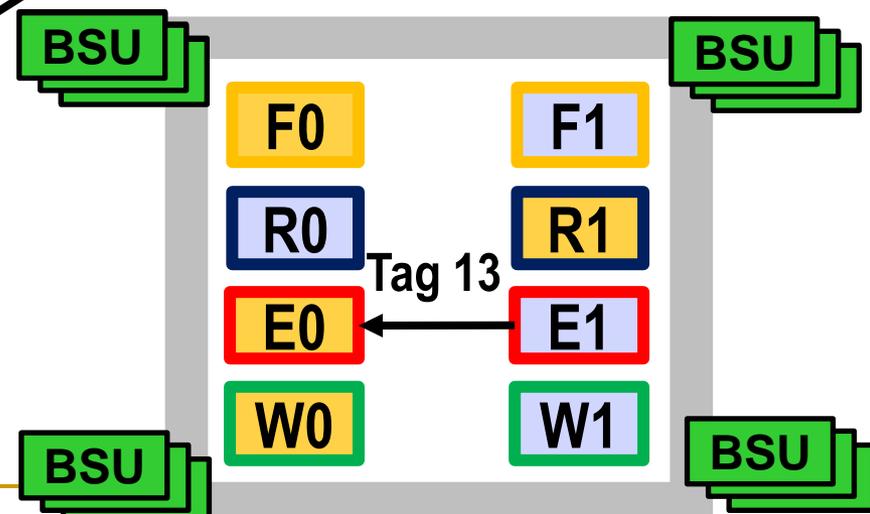
- 1. Building Virtual Pipelines
- 2. **Inter-Cluster Dependencies**
- 3. Mispredictions
- 4. Precise Exceptions

Clusters might need operands generated by others

BSU ID	PC	NPC	Fetch	Reg	Exec	WB	Input Tags				Output Tags			
							RA	RB	RC	RD	RA	RB	RC	RD
1	4013c3	4013c8	F0	R1	E0	W0	1	5	6	10	13	5	6	10
2	4013c8		F1	R0	E1	W1	13	5	6	10				
3														

```
4013c3: add %a1, [%ebx]
```

```
4013c8: or %a1, %b1
```



# 2. Inter-Cluster Dependencies

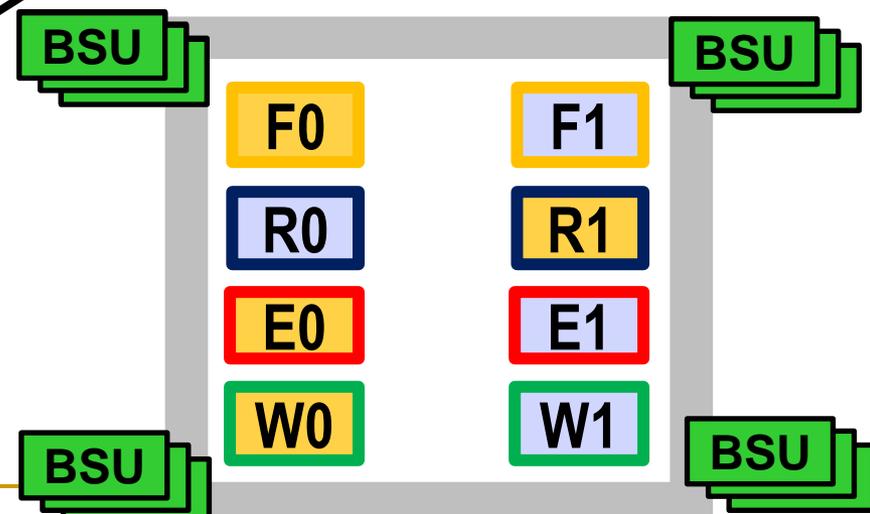
- 1. Building Virtual Pipelines
- 2. **Inter-Cluster Dependencies**
- 3. Mispredictions
- 4. Precise Exceptions

Clusters might need operands generated by others

BSU ID	PC	NPC	Fetch	Reg	Exec	WB	Input Tags				Output Tags			
							RA	RB	RC	RD	RA	RB	RC	RD
1	4013c3	4013c8	F0	R1	E0	W0	1	5	6	10	13	5	6	10
2	4013c8		F1	R0	E1	W1	13	5	6	10				
3														

```
4013c3:add %a1, [%ebx]
```

```
4013c8:or %a1, %b1
```



# 2. Inter-Cluster Dependencies

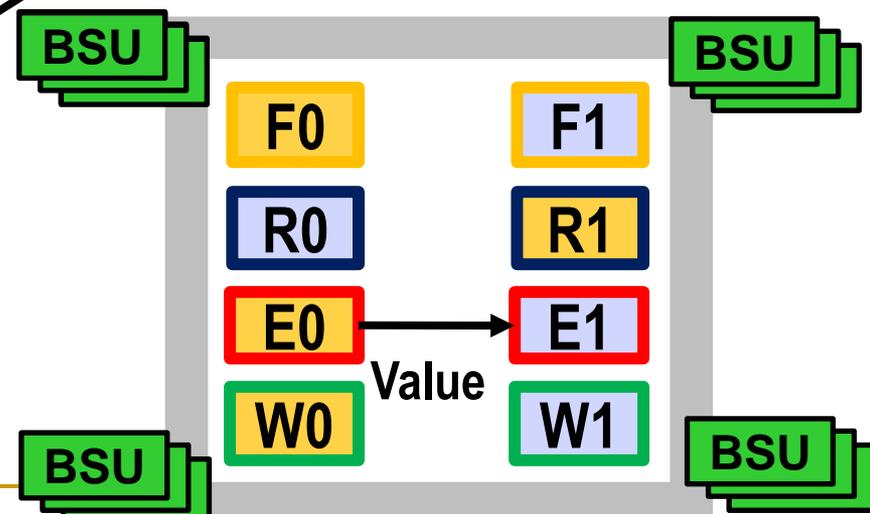
- 1. Building Virtual Pipelines
- 2. **Inter-Cluster Dependencies**
- 3. Mispredictions
- 4. Precise Exceptions

Clusters might need operands generated by others

BSU ID	PC	NPC	Fetch	Reg	Exec	WB	Input Tags				Output Tags			
							RA	RB	RC	RD	RA	RB	RC	RD
1	4013c3	4013c8	F0	R1	E0	W0	1	5	6	10	13	5	6	10
2	4013c8		F1	R0	E1	W1	13	5	6	10				
3														

```
4013c3:add %a1, [%ebx]
```

```
4013c8:or %a1, %b1
```

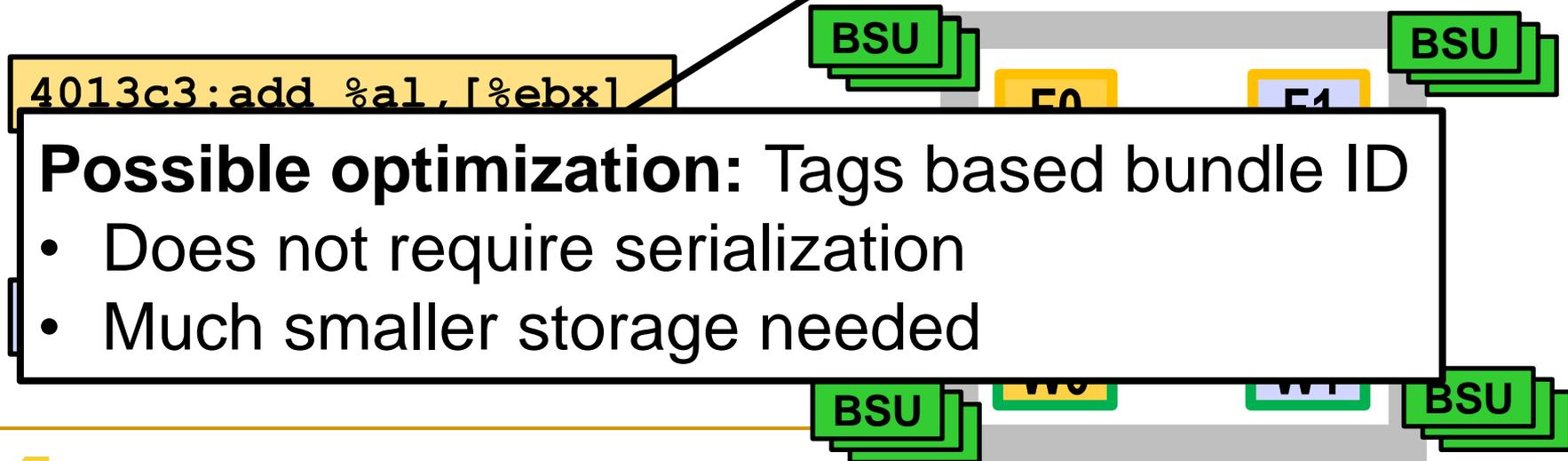


## 2. Inter-Cluster Dependencies

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

*Clusters might need operands generated by others*

BSU ID	PC	NPC	Fetch	Reg	Exec	WB	Input Tags				Output Tags			
							RA	RB	RC	RD	RA	RB	RC	RD
1	4013c3	4013c8	F0	R1	E0	W0	1	5	6	10	13	5	6	10
2	4013c8		F1	R0	E1	W1	13	5	6	10				
3														

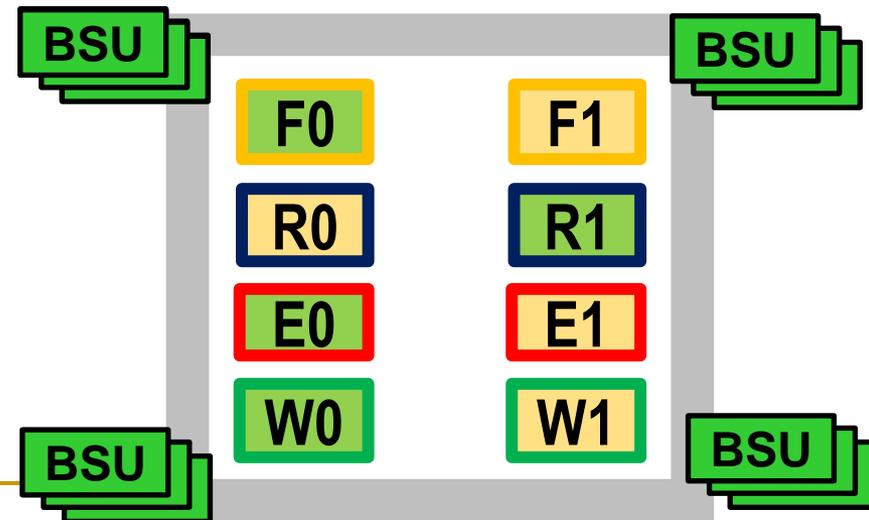


# 3. Handling Mispredictions

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

```
4013c0: jmp 4013eb  
4013c3: add %a1, [%ebx]  
4013c8: or %a1, %b1
```

BSU ID	Next	PC	NPC	Fetch	Reg	Exec	WB
0	1	4013c0	4013c3	F0	R1	E0	W0
1	2	4013c3	4013c8	F1	R0	E1	W1
2	-	4013c8					



# 3. Handling Mispredictions

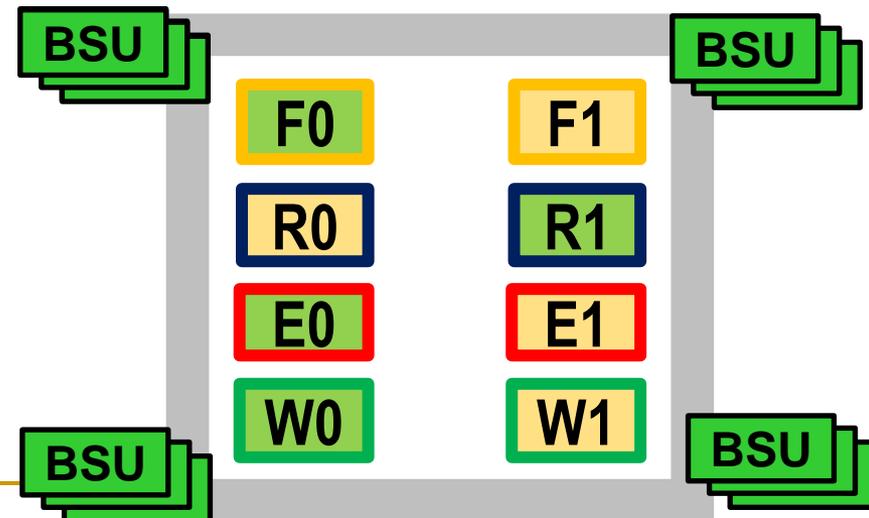
1. Building Virtual Pipelines
2. Inter-Cluster Dependencies
3. Mispredictions
4. Precise Exceptions

4013c0: jmp 4013eb

4013c3: add %a1, [%ebx]

4013c8: or %a1, %b1

BSU ID	Next	PC	NPC	Fetch	Reg	Exec	WB
0	1	4013c0	4013c3	F0	R1	E0	W0
1	2	4013c3	4013c8	F1	R0	E1	W1
2	-	4013c8					

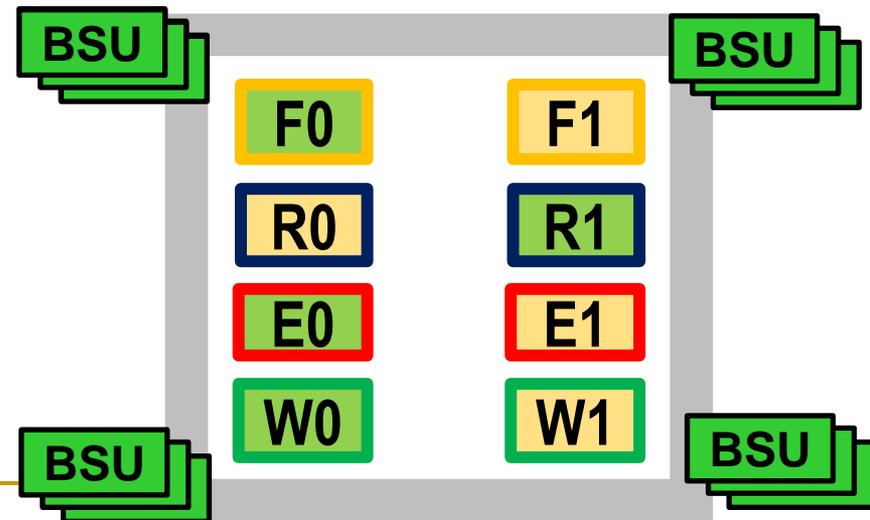


# 3. Handling Mispredictions

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

```
4013c0: jmp 4013eb  
4013c3: add %a1, [%ebx]  
4013c8: or %a1, %b1
```

BSU ID	Next	PC	NPC	Fetch	Reg	Exec	WB
0	1	4013c0	4013c3	F0	R1	E0	W0
1	2	4013c3	4013c8	F1	R0	E1	W1
2	-	4013c8					

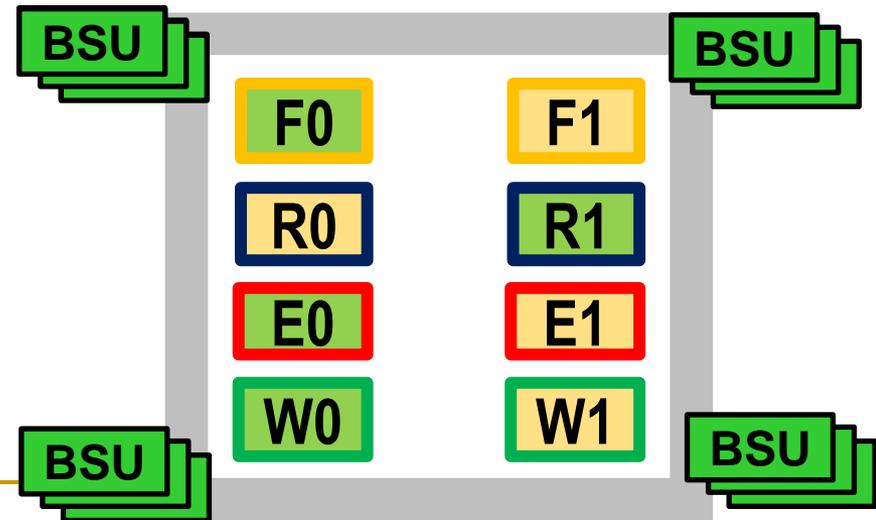


# 3. Handling Mispredictions

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

```
4013c0: jmp 4013eb
4013c3: add %a1, [%ebx]
4013c8: or %a1, %b1
```

BSU ID	Next	PC	NPC	Fetch	Reg	Exec	WB
0	1	4013c0	4013c3	F0	R1	E0	W0
1	2	4013c3	4013c8	F1	R0	E1	W1
2	-	4013c8					



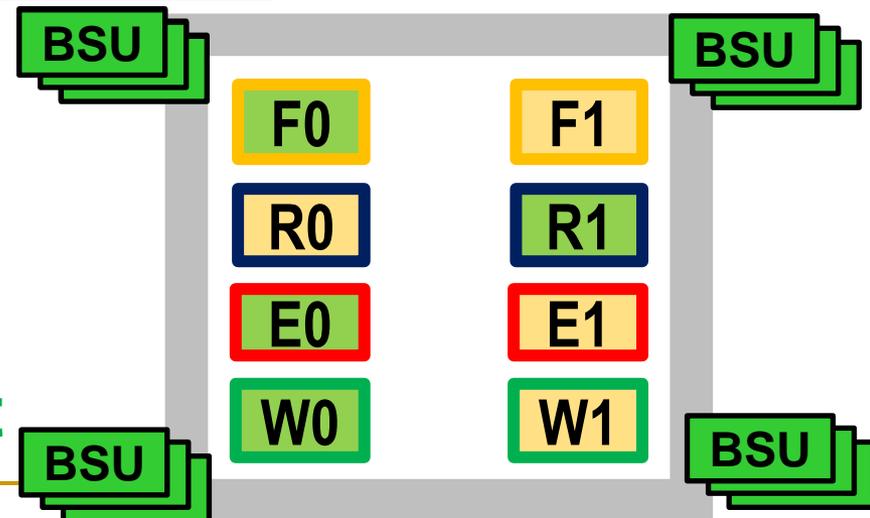
# 3. Handling Mispredictions

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

```
4013c0: jmp 4013eb
4013c3: add %a1, [%ebx]
4013c8: or %a1, %b1
```

BSU ID	Next	PC	NPC	Fetch	Reg	Exec	WB
0	1	4013c0	4013c3	F0	R1	E0	W0
1	2	4013c3	4013c8	F1	R0	E1	W1
2	-	4013c8					

Fetch instruction  
 Register access  
 Execute  
 Write back/Commit



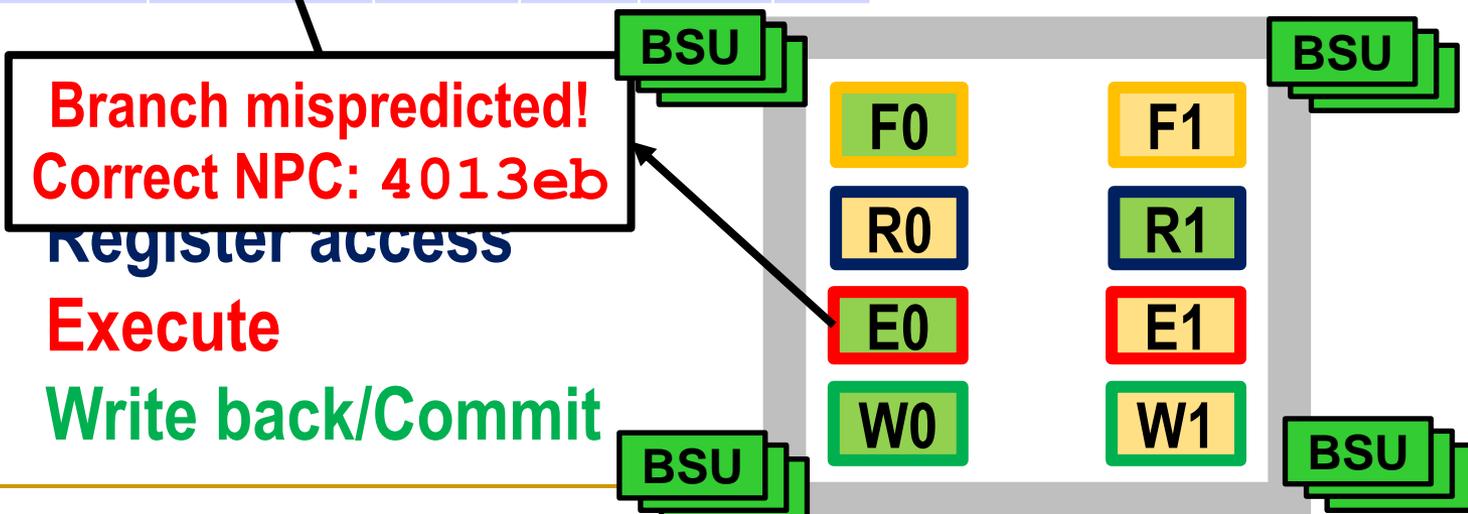
# 3. Handling Mispredictions

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

```

4013c0: jmp 4013eb
4013c3: add %a1, [%ebx]
4013c8: or %a1, %b1
    
```

BSU ID	Next	PC	NPC	Fetch	Reg	Exec	WB
0	1	4013c0	4013c3	F0	R1	E0	W0
1	2	4013c3	4013c8	F1	R0	E1	W1
2	-	4013c8					



**Branch mispredicted!**  
**Correct NPC: 4013eb**  
 Register access  
 Execute  
 Write back/Commit

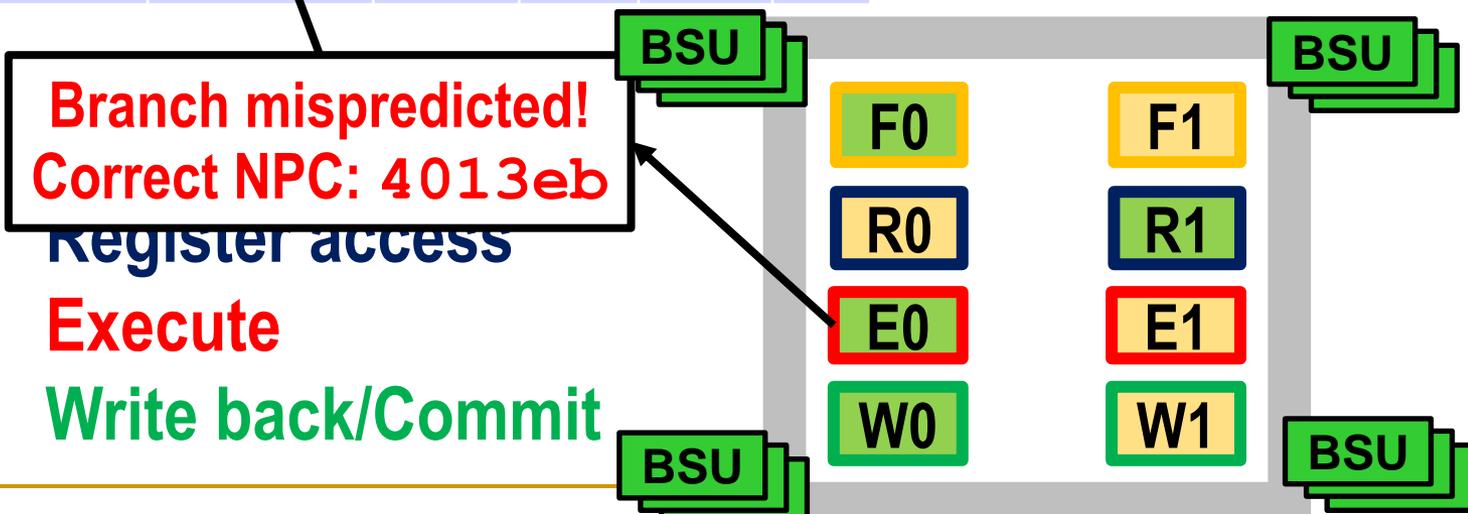
# 3. Handling Mispredictions

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

```

4013c0: jmp 4013eb
4013c3: add %a1, [%ebx]
4013c8: or %a1, %b1
    
```

BSU ID	Next	PC	NPC	Fetch	Reg	Exec	WB
0	1	4013c0	4013eb	F0	R1	E0	W0
1	2	4013c3	4013c8	F1	R0	E1	W1
2	-	4013c8					

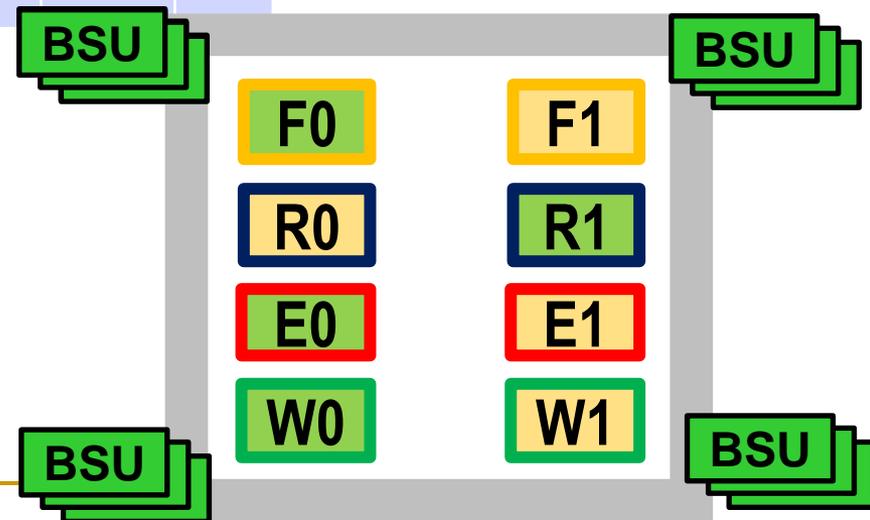


# 3. Handling Mispredictions

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

```
4013c0: jmp 4013eb  
4013c3: add %a1, [%ebx]  
4013c8: or %a1, %b1
```

BSU ID	Next	PC	NPC	Fetch	Reg	Exec	WB
0	1	4013c0	4013eb	F0	R1	E0	W0
1	2	4013c3	4013c8	F1	R0	E1	W1
2	-	4013c8					

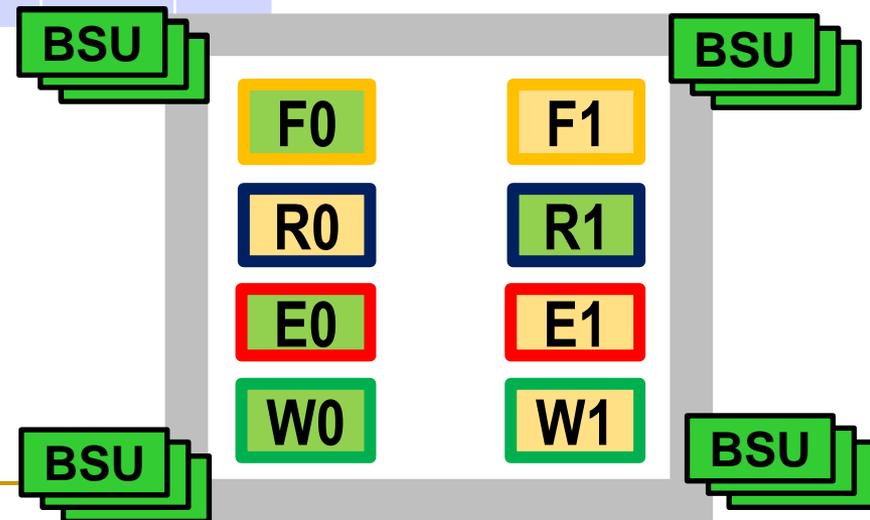


# 3. Handling Mispredictions

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

```
4013c0: jmp 4013eb
4013c3: add %a1, [%ebx]
4013c8: or %a1, %b1
```

BSU ID	Next	PC	NPC	Fetch	Reg	Exec	WB
0	1	4013c0	4013eb	F0	R1	E0	W0
1	2	4013c3	4013c8	F1	R0	E1	W1
2	-	4013c8					

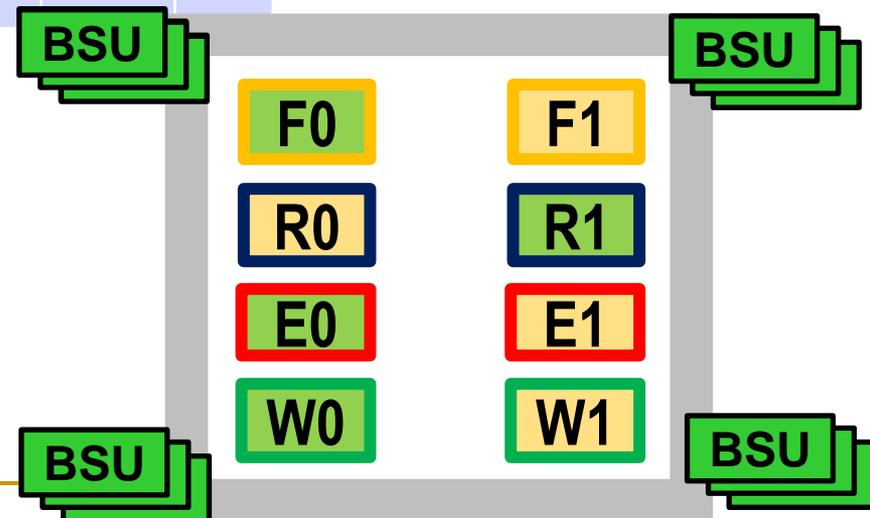


# 3. Handling Mispredictions

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

```
4013c0: jmp 4013eb
4013c3: add %a1, [%ebx]
4013c8: or %a1, %b1
```

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

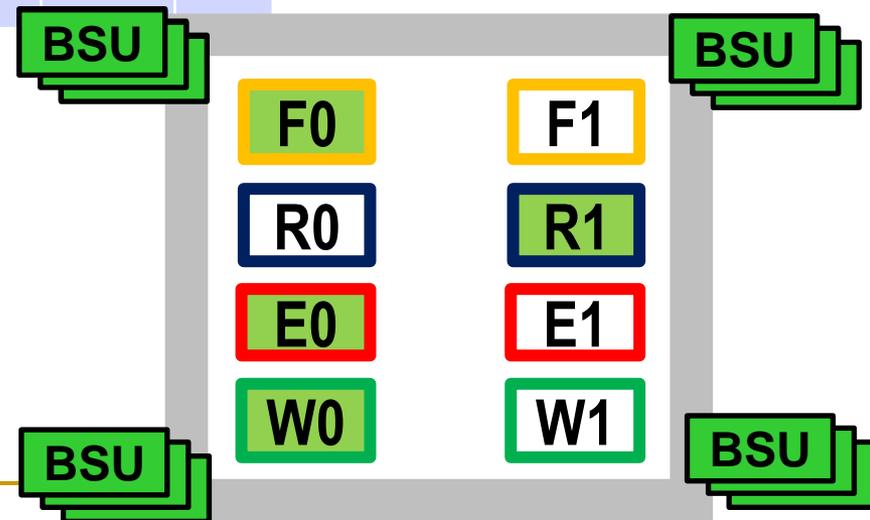


# 3. Handling Mispredictions

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

```
4013c0: jmp 4013eb
4013c3: add %a1, [%ebx]
4013c8: or %a1, %b1
```

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



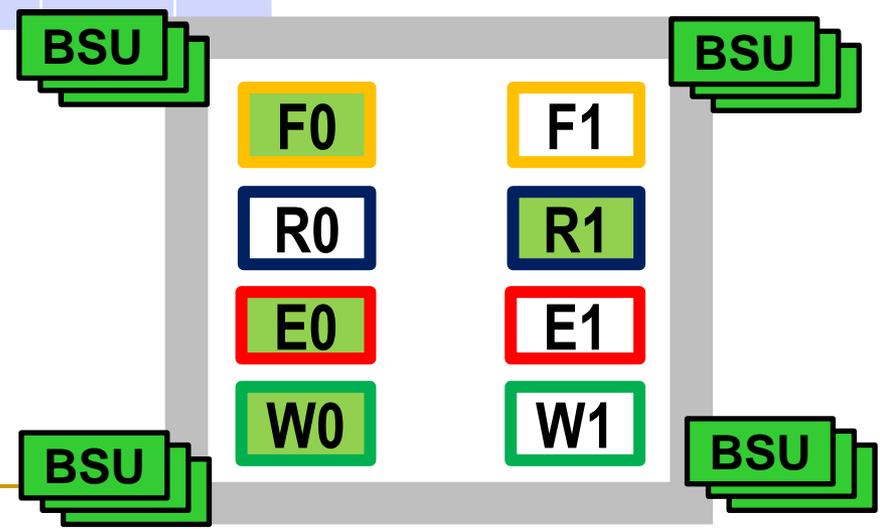
# 3. Handling Mispredictions

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

```

4013c0: jmp  4013eb
4013c3: add  %a1, [%ebx]
4013c8: or  %a1, %b1
    
```

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

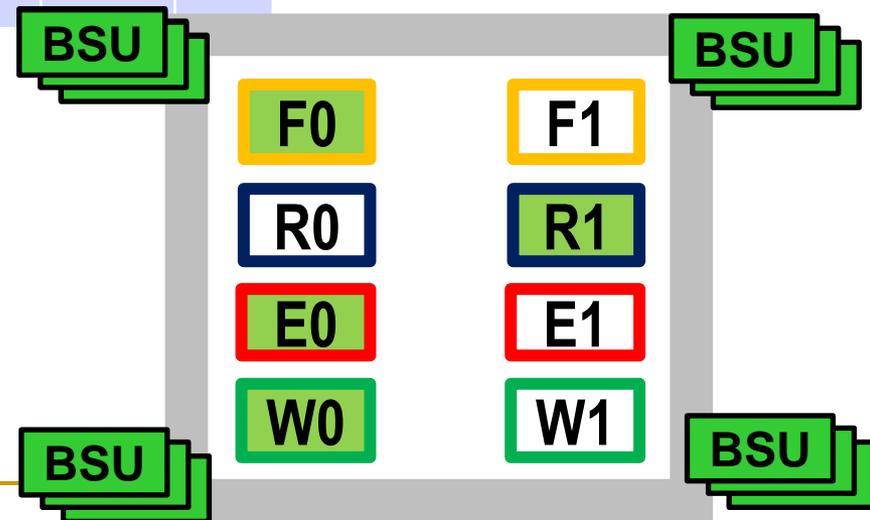


# 3. Handling Mispredictions

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

```
4013c0: jmp 4013eb
4013c3: add %a1, [%ebx]
4013c8: or %a1, %b1
```

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

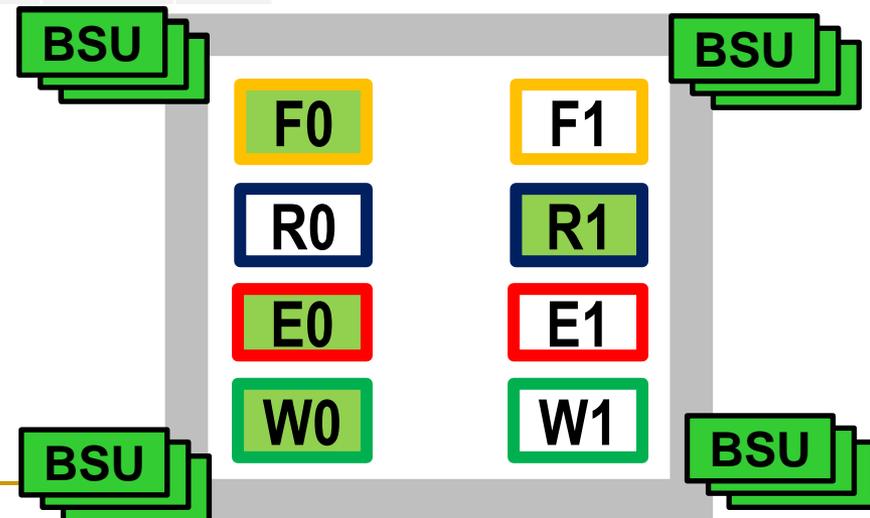


# 4. Precise Exceptions

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

```
4013c3: add %a1, [%ebx]  
4013c5: div cl  
4013c8: jmp 40140a
```

BSU ID	Next	PC	NPC	Fetch	Reg	Exec	WB
0	1	4013c3		F0	R1	E0	W0
1							
2							

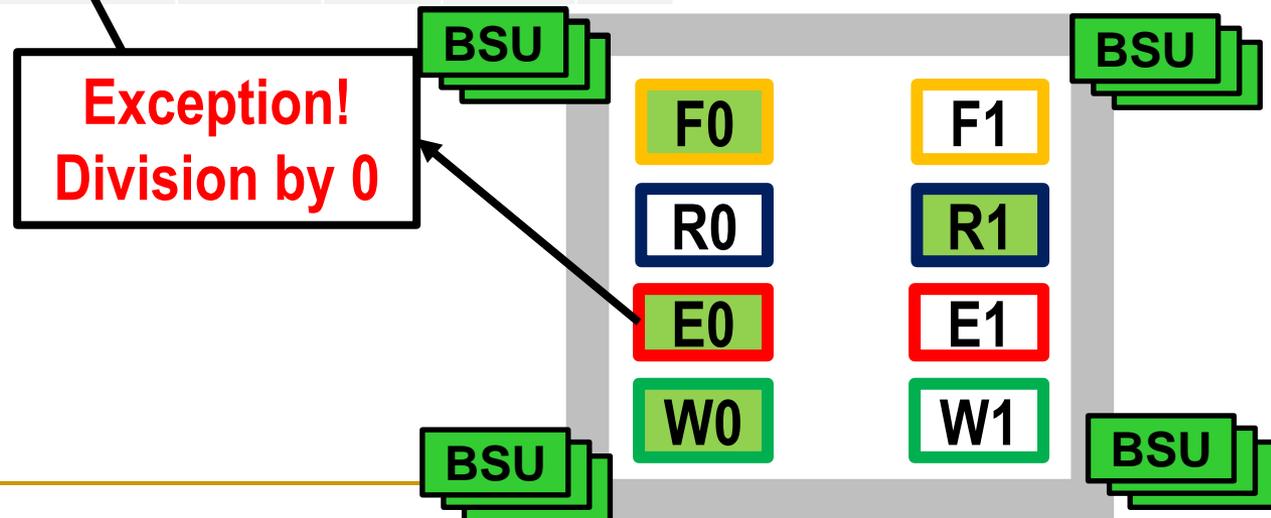


# 4. Precise Exceptions

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

```
4013c3: add %a1, [%ebx]
4013c5: div cl
4013c8: jmp 40140a
```

BSU ID	Next	PC	NPC	Fetch	Reg	Exec	WB
0	1	4013c3		F0	R1	E0	W0
1							
2							



# 4. Precise Exceptions

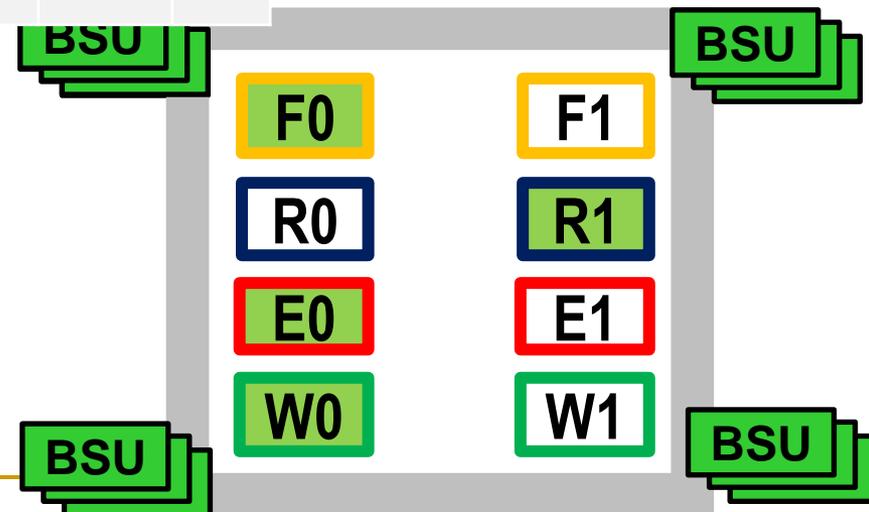
1. Building Virtual Pipelines
2. Inter-Cluster Dependencies
3. Mispredictions
4. Precise Exceptions

4013c3: add %a1, [%ebx]

4013c5: **div cl**

4013c8: jmp 40140a

BSU ID	Next	PC	NPC	Fetch	Reg	Exec	WB
0	1	4013c3		F0	R1	E0	W0
1							
2							



# 4. Precise Exceptions

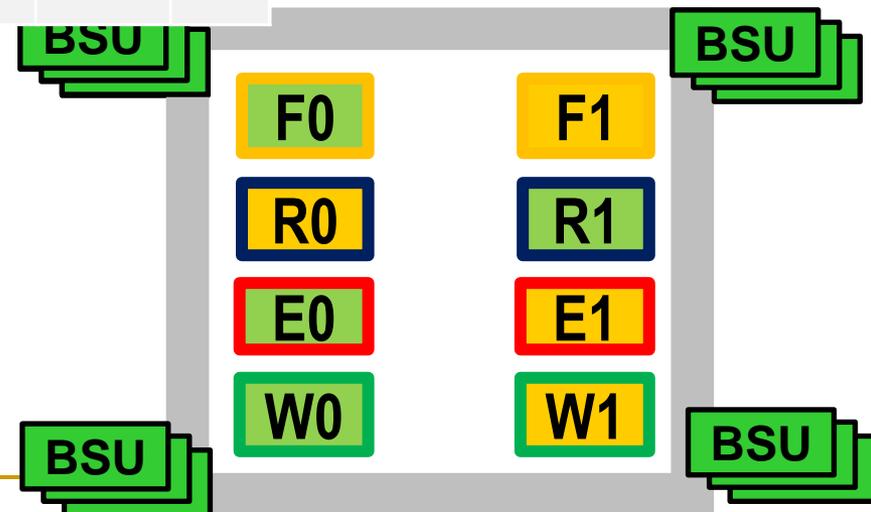
1. Building Virtual Pipelines
2. Inter-Cluster Dependencies
3. Mispredictions
4. Precise Exceptions

4013c3: add %a1, [%ebx]

4013c5: **div cl**

4013c8: jmp 40140a

BSU ID	Next	PC	NPC	Fetch	Reg	Exec	WB
0	1	4013c3	4013c5	F0	R1	E0	W0
1		4013c5		F1	R0	E1	W1
2							



# 4. Precise Exceptions

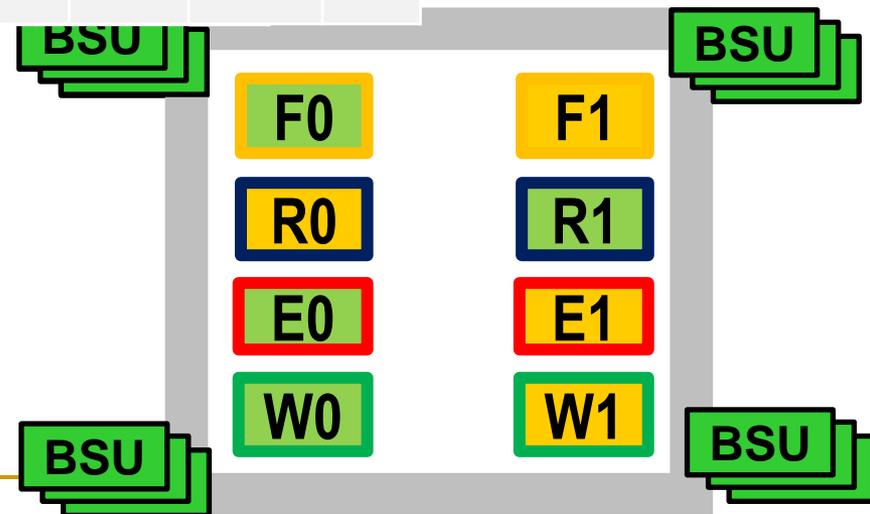
- 1. Building Virtual Pipelines
- 2. Inter-Cluster Dependencies
- 3. Mispredictions
- 4. Precise Exceptions

4013c3: add %a1, [%ebx]

4013c5: **div cl**

4013c8: jmp 40140a

BSU ID	Next	PC	NPC	Fetch	Reg	Exec	WB
0	1	4013c3	4013c5	F0	R1	E0	W0
1		4013c5	<b>Exc. Handler</b>	F1	R0	E1	W1
2							



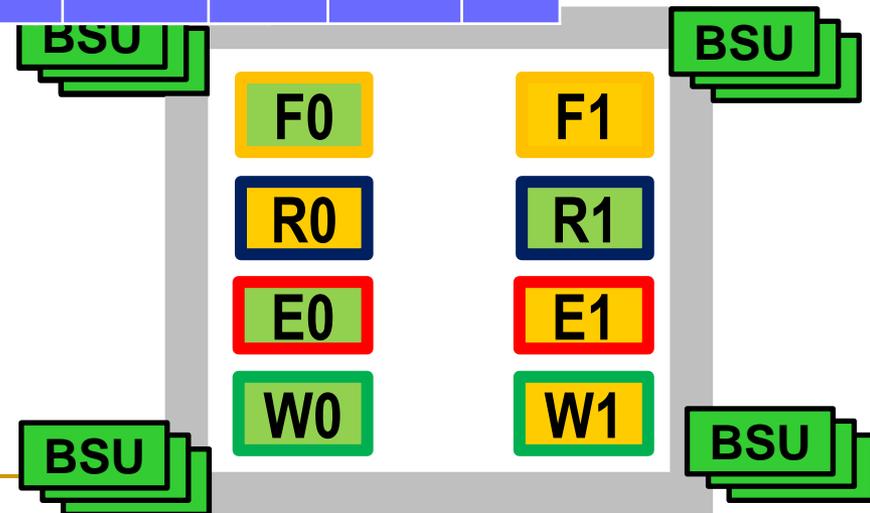
# 4. Precise Exceptions

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

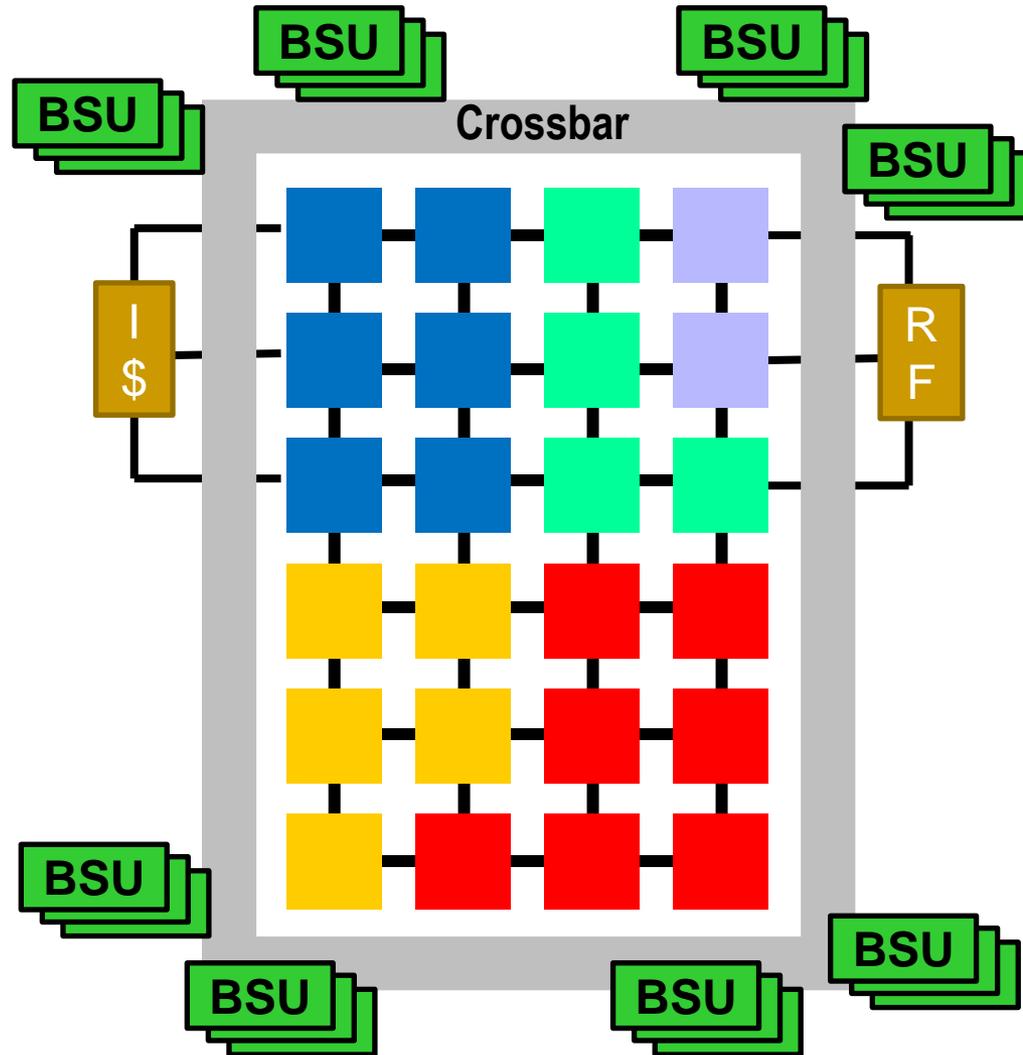
```

4013c3: add %a1, [%ebx]
4013c5: div cl
4013c8: jmp 40140a
    
```

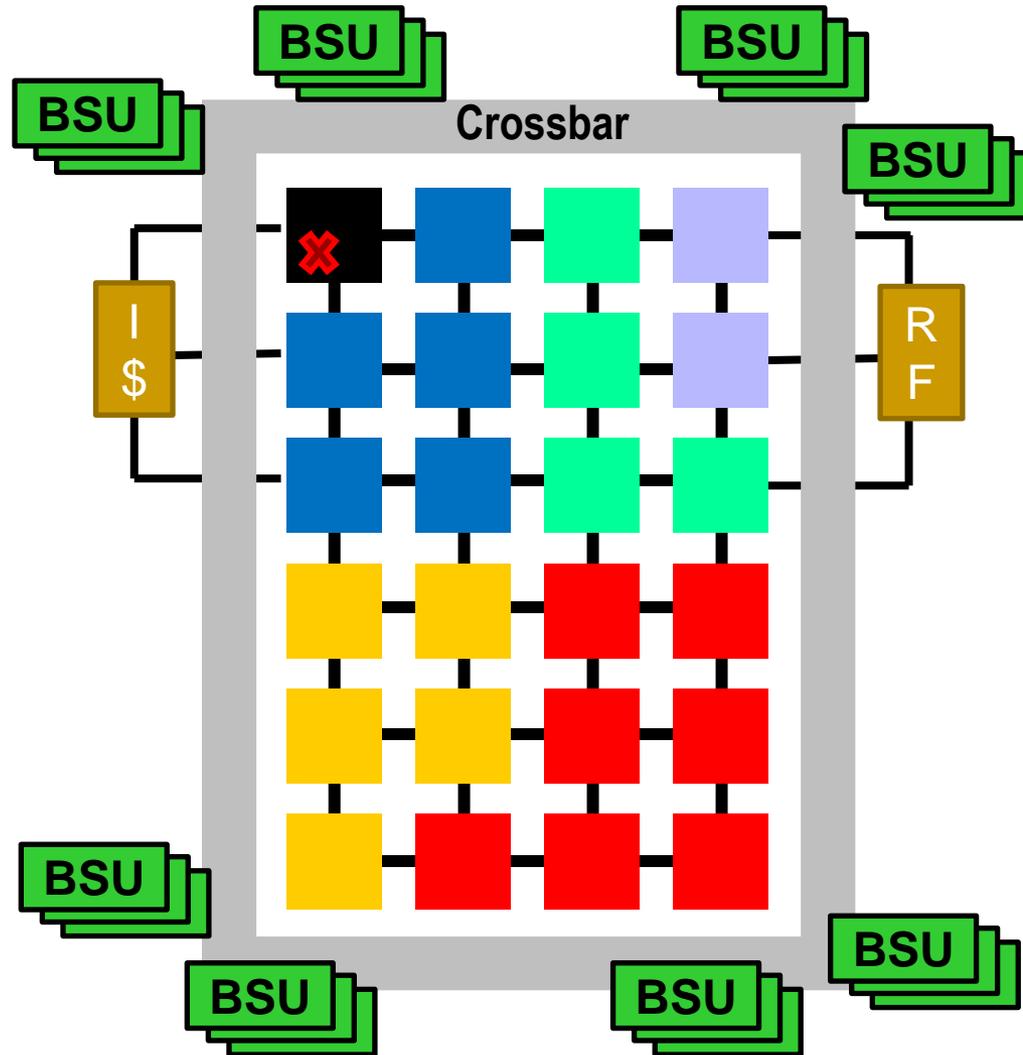
BSU ID	Next	PC	NPC	Fetch	Reg	Exec	WB
0	1	4013c3	4013c5	F0	R1	E0	W0
1	2	4013c5	Exc. Handler	F1	R0	E1	W1
2		Exc. Handler	4013c8				



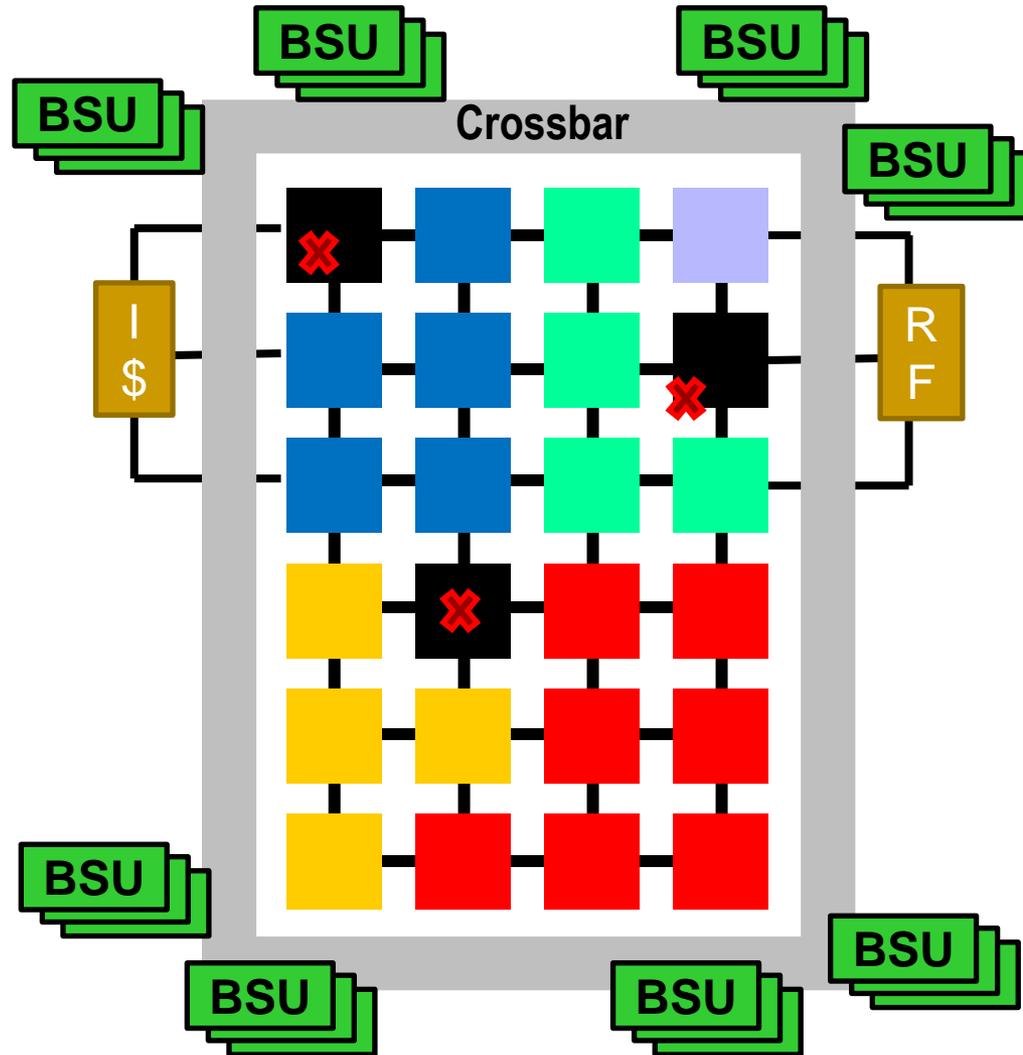
# Impact of Faults in Viper



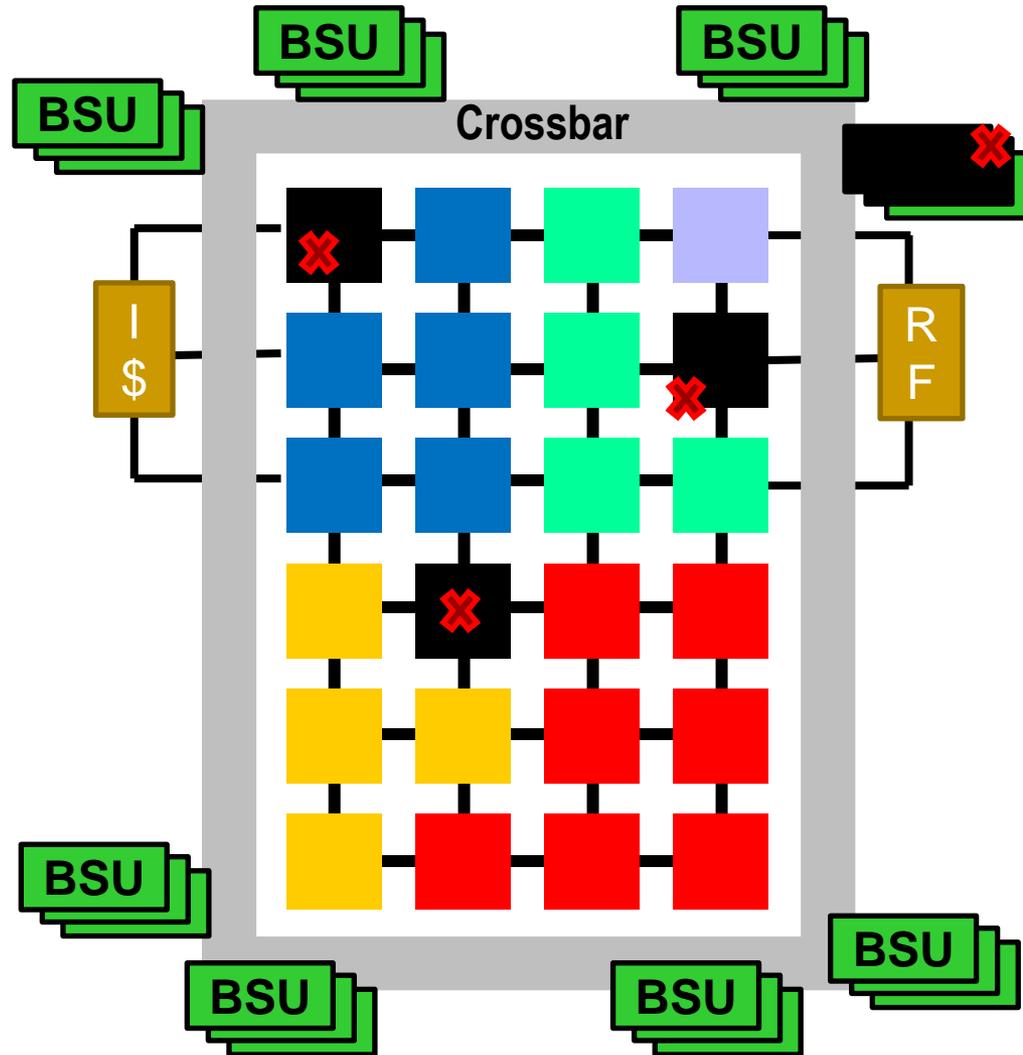
# Impact of Faults in Viper



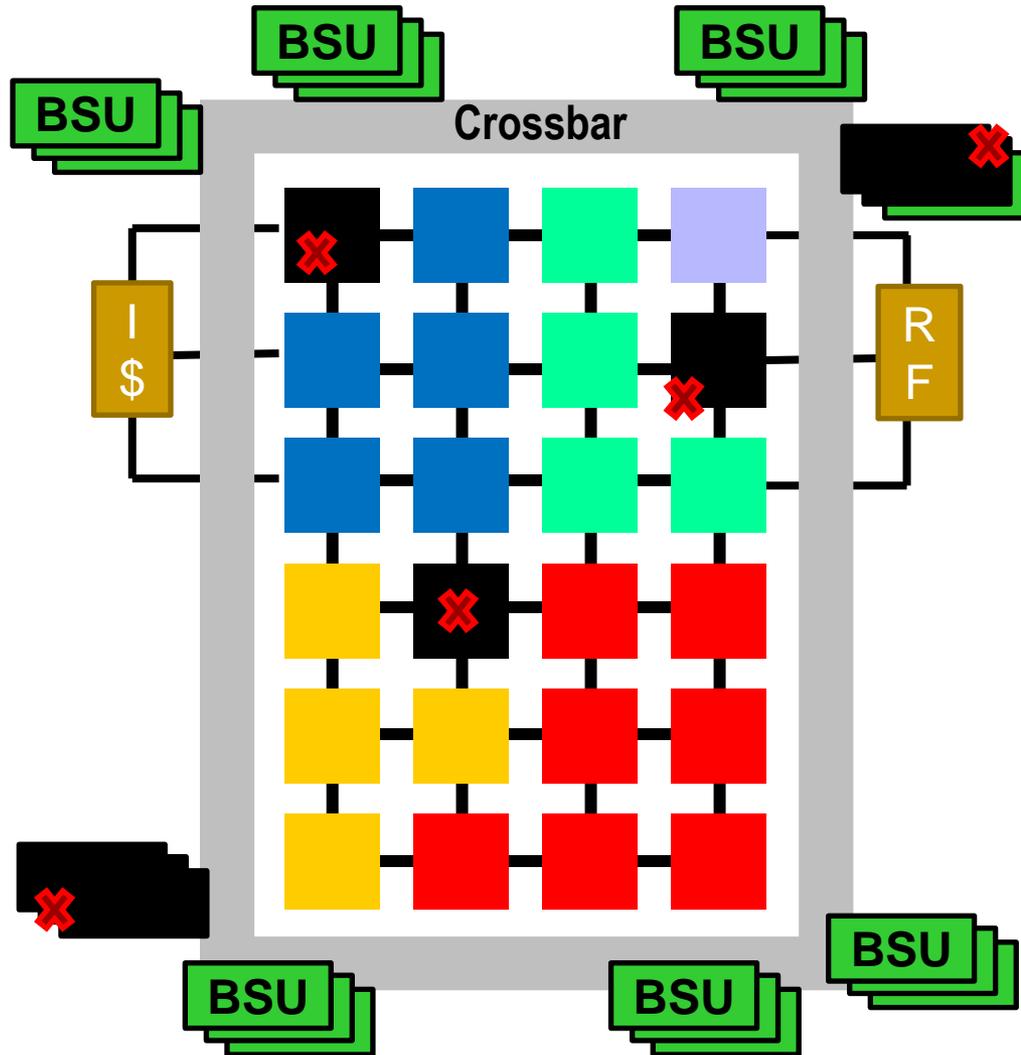
# Impact of Faults in Viper



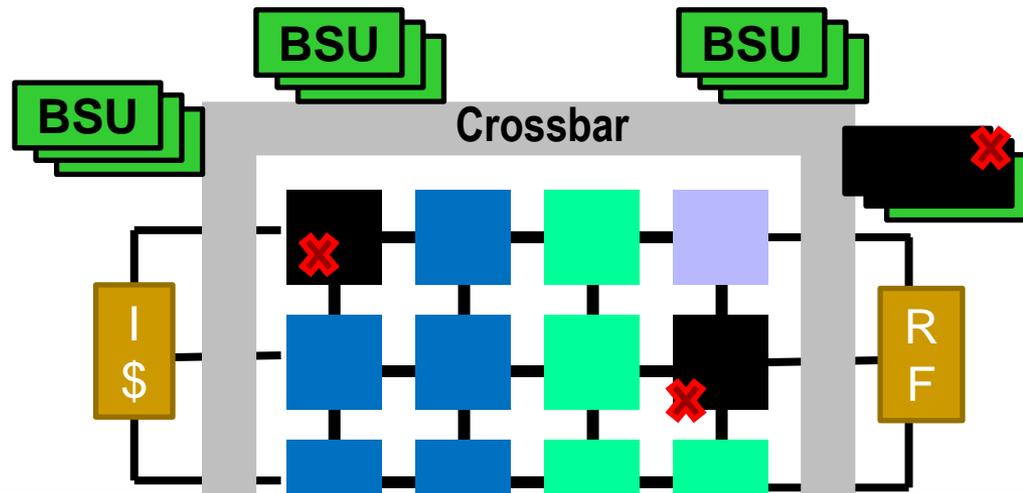
# Impact of Faults in Viper



# Impact of Faults in Viper



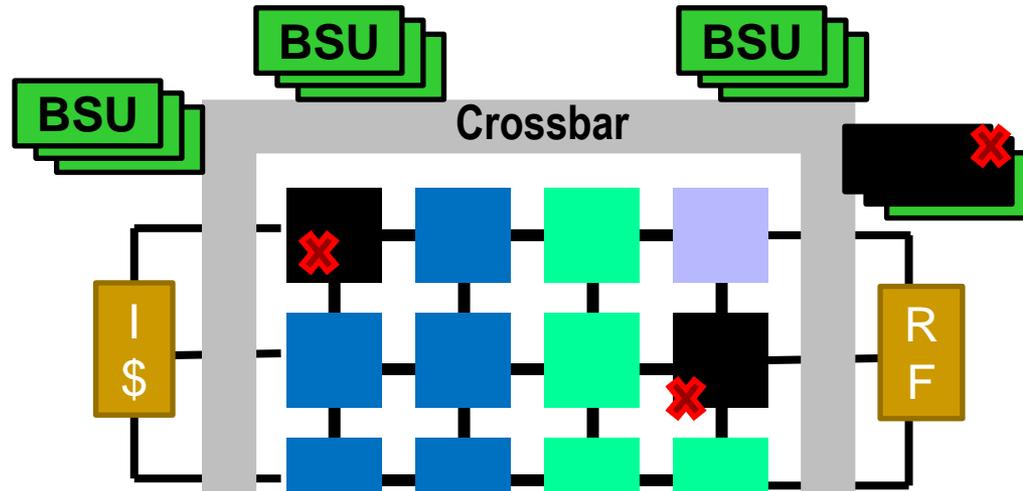
# Impact of Faults in Viper



## Handling runtime failures:

1. Periodic full system checkpoint

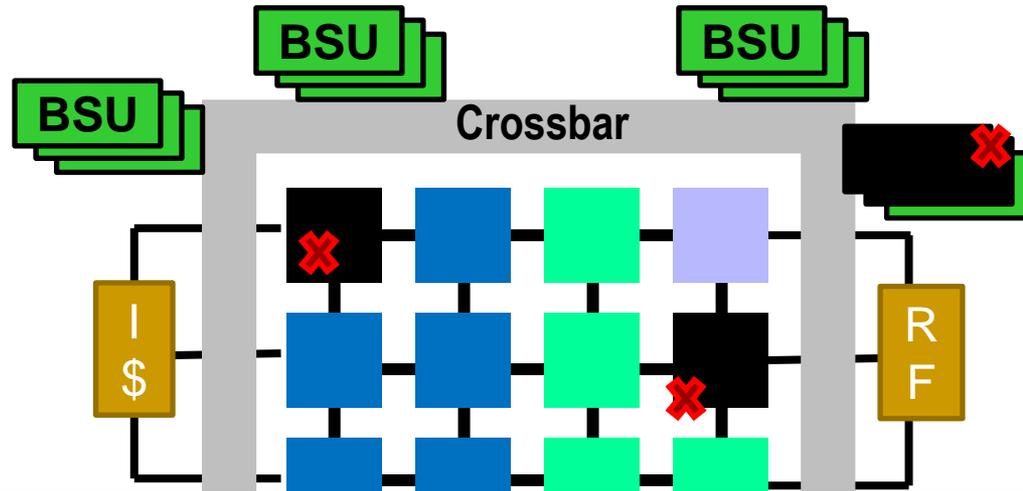
# Impact of Faults in Viper



## Handling runtime failures:

1. Periodic full system checkpoint
2. Detected through hardware self-tests or SW symptoms

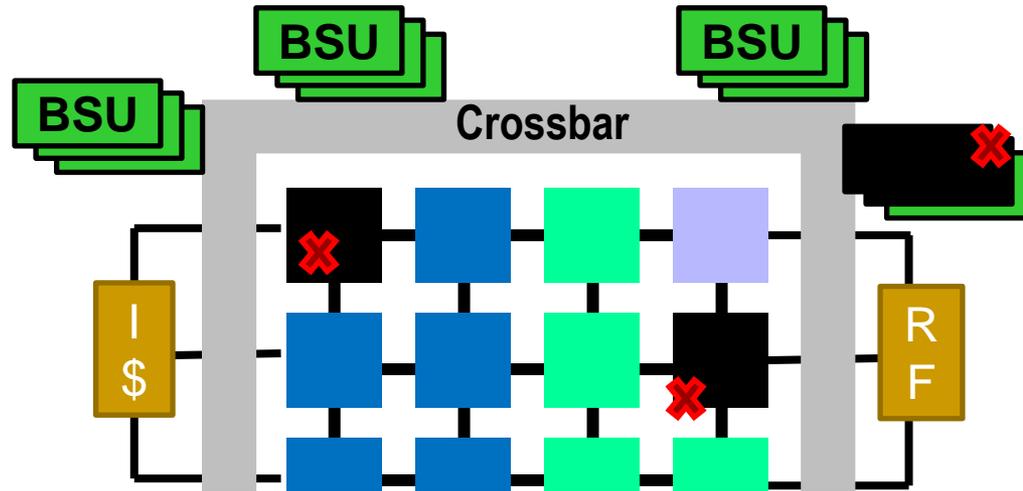
# Impact of Faults in Viper



## Handling runtime failures:

1. Periodic full system checkpoint
2. Detected through hardware self-tests or SW symptoms
3. If a fault is detected:

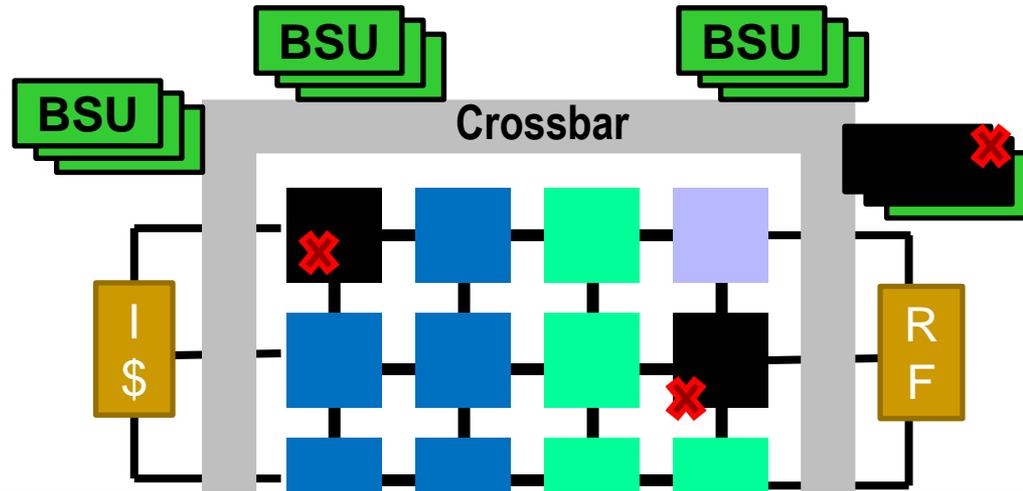
# Impact of Faults in Viper



## 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

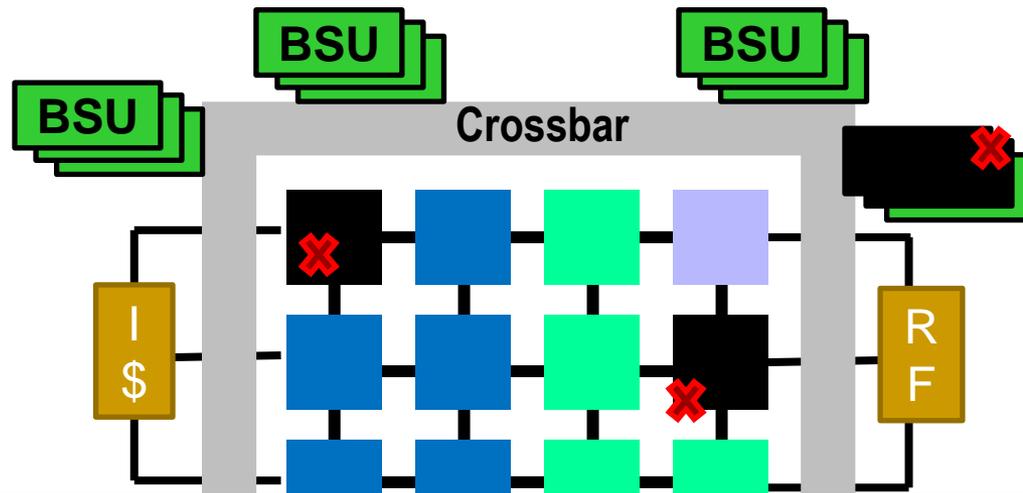
# Impact of Faults in Viper



## 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



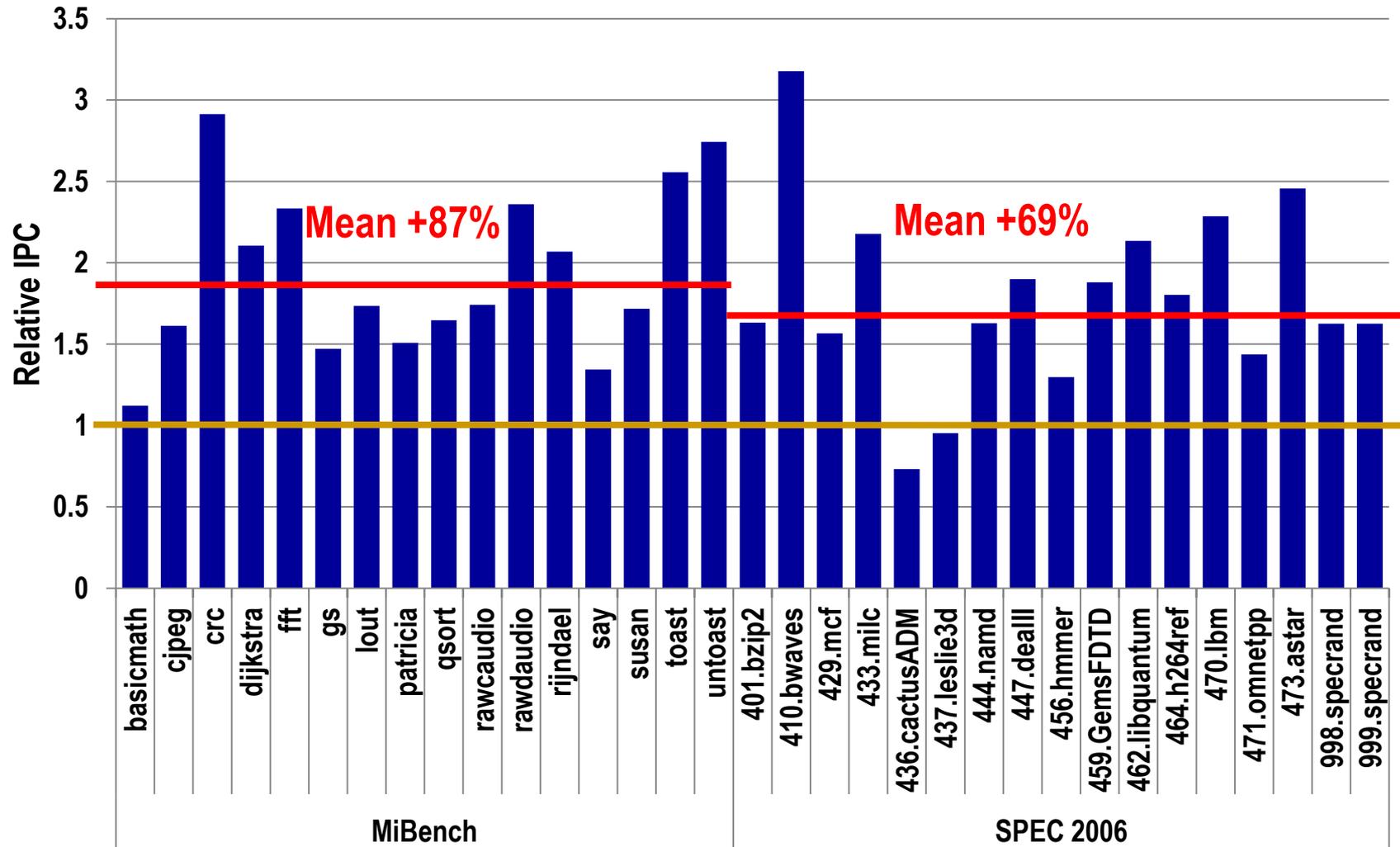
## Handling runtime failures:

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

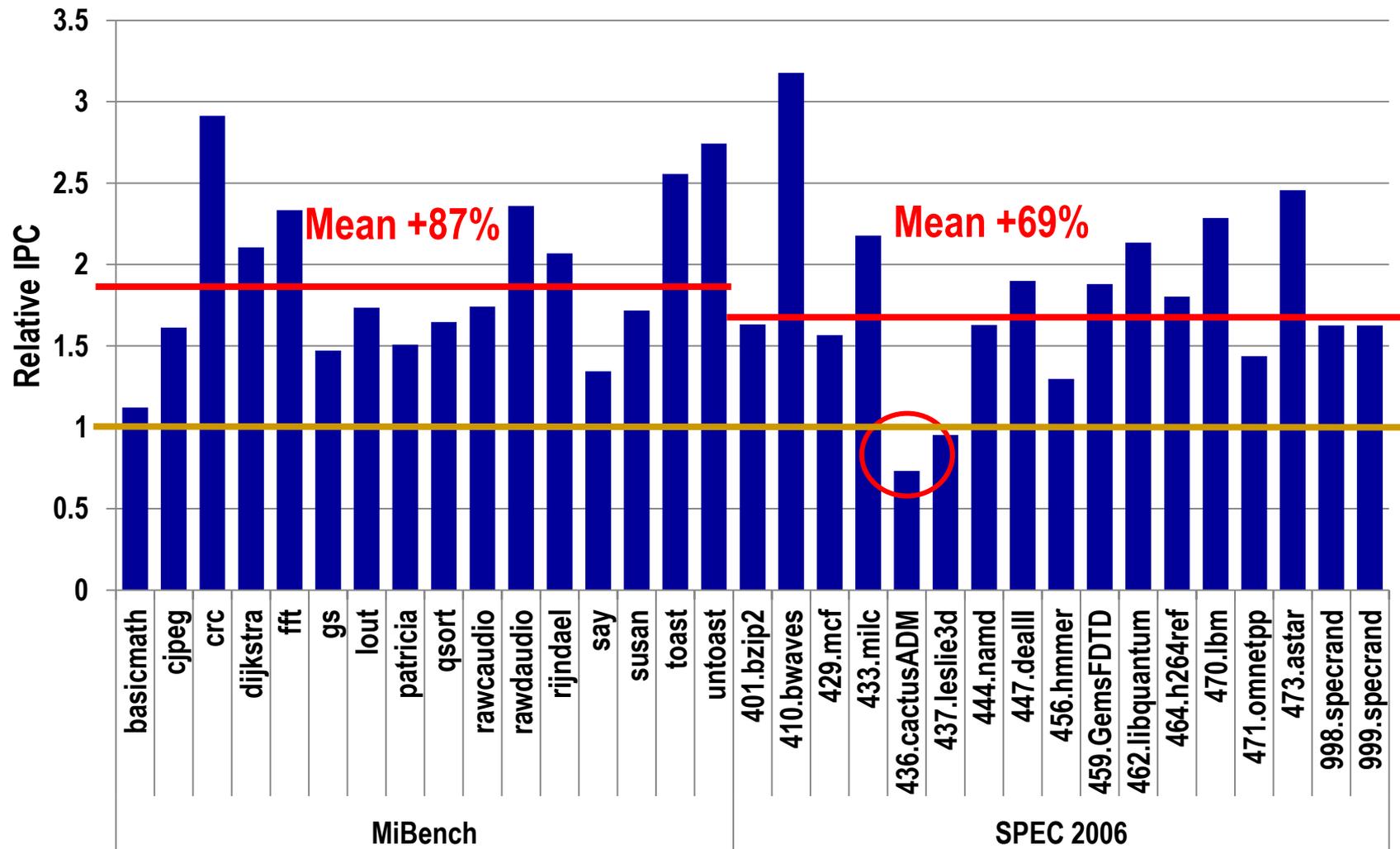
# Experimental Setup

- 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

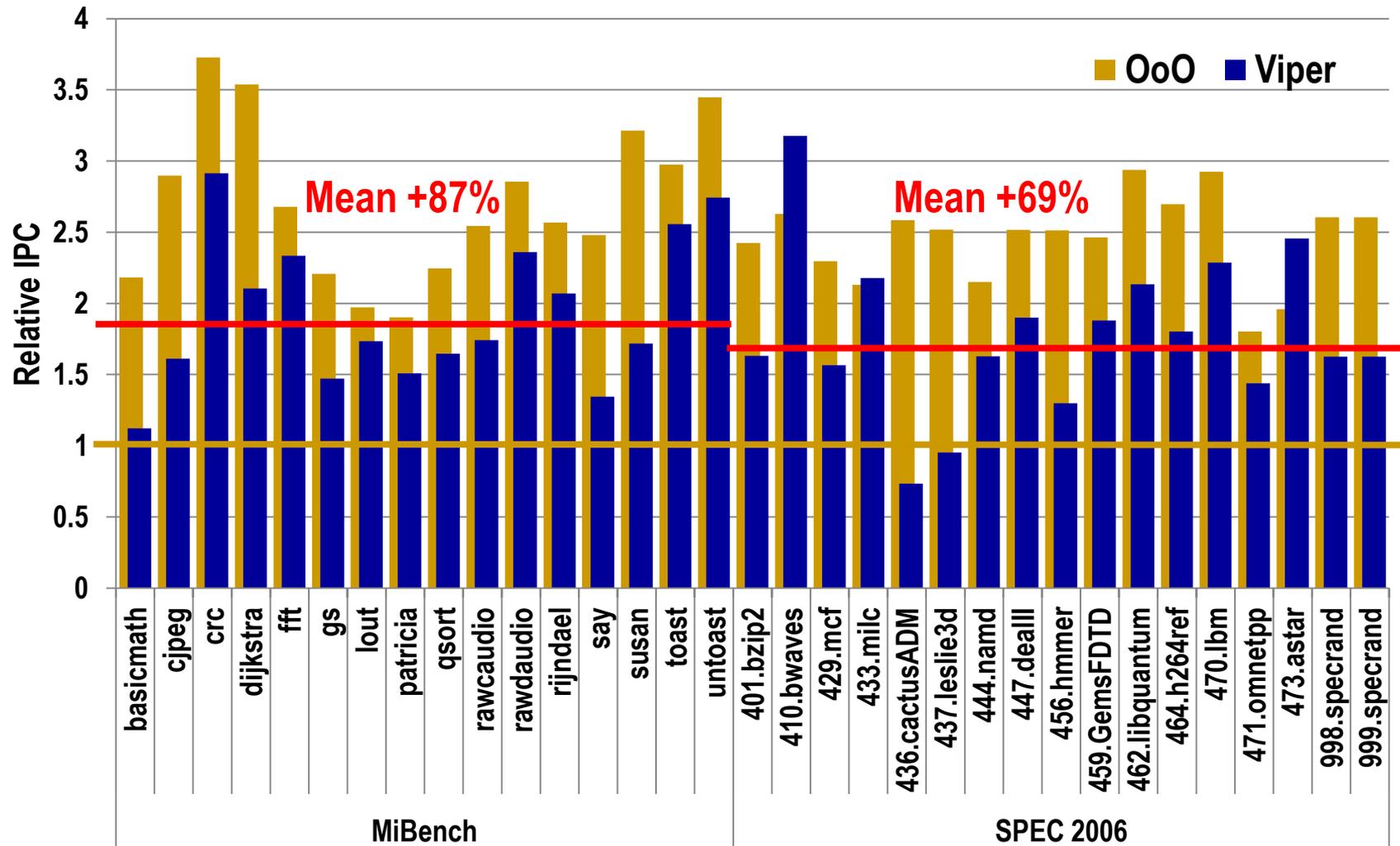
# Viper Enables Reliable OoO Execution



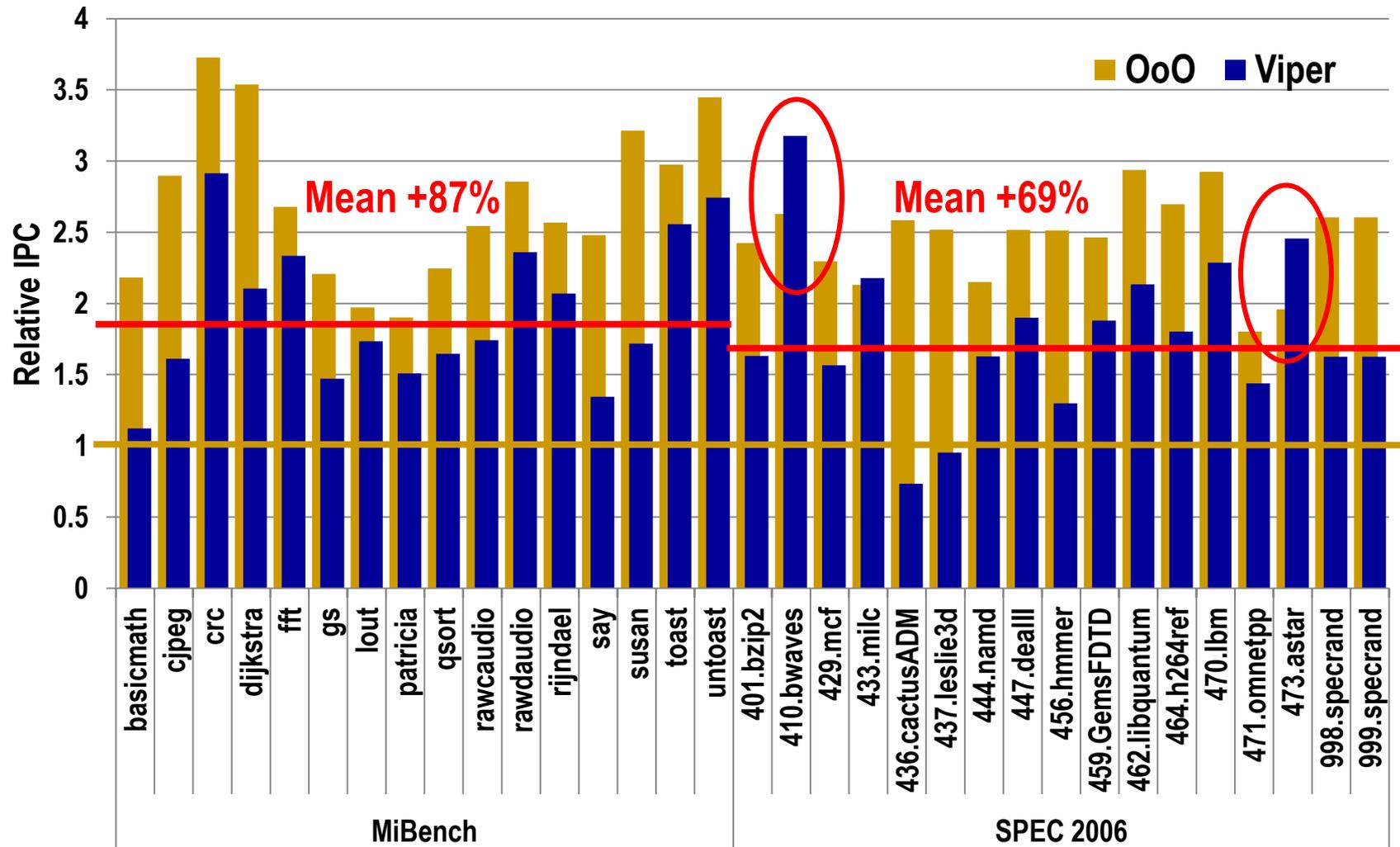
# Viper Enables Reliable OoO Execution



# Viper is Competitive vs Unprotected OoO



# Viper is Competitive vs Unprotected OoO



# Comparison With Other Solutions

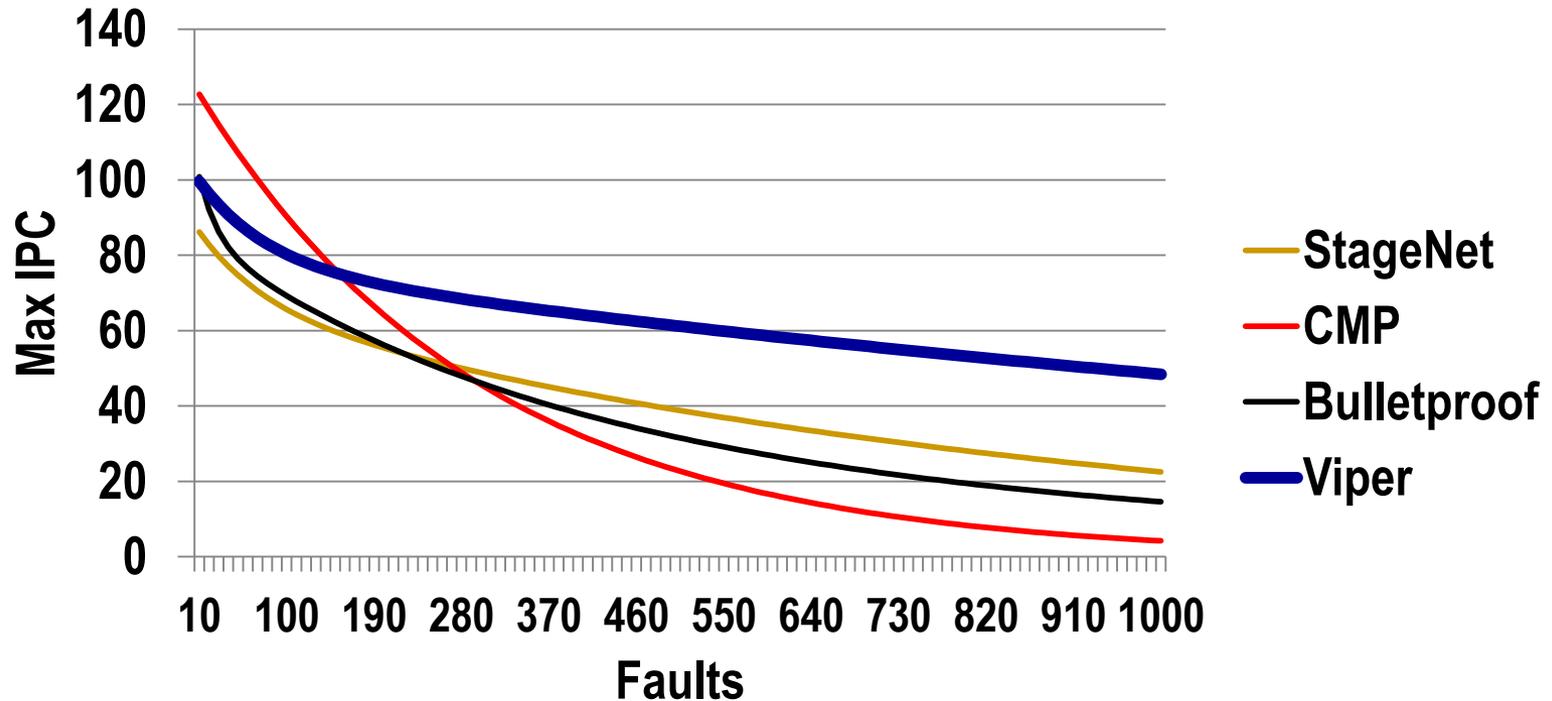
	Cardio [Pellegrini10]	Core Salvaging [Powell09]	Bulletproof [Shyam06]	StageNet [Gupta08]	Viper
<i>Reconfig. Granularity</i>	Core	Functional units	Functional units	Pipeline stages	Arbitrary
<i>Control logic</i>	Central	Central	Central	Central	Decoupled
<i>Interconnect</i>	NoC	Specialized	Specialized	Stage-specific	Loose
<i>Slowdown</i>	~3%	~5%	~18%	~20%	~24%

Low fault rate

High fault rate



# Performance degradation



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

---

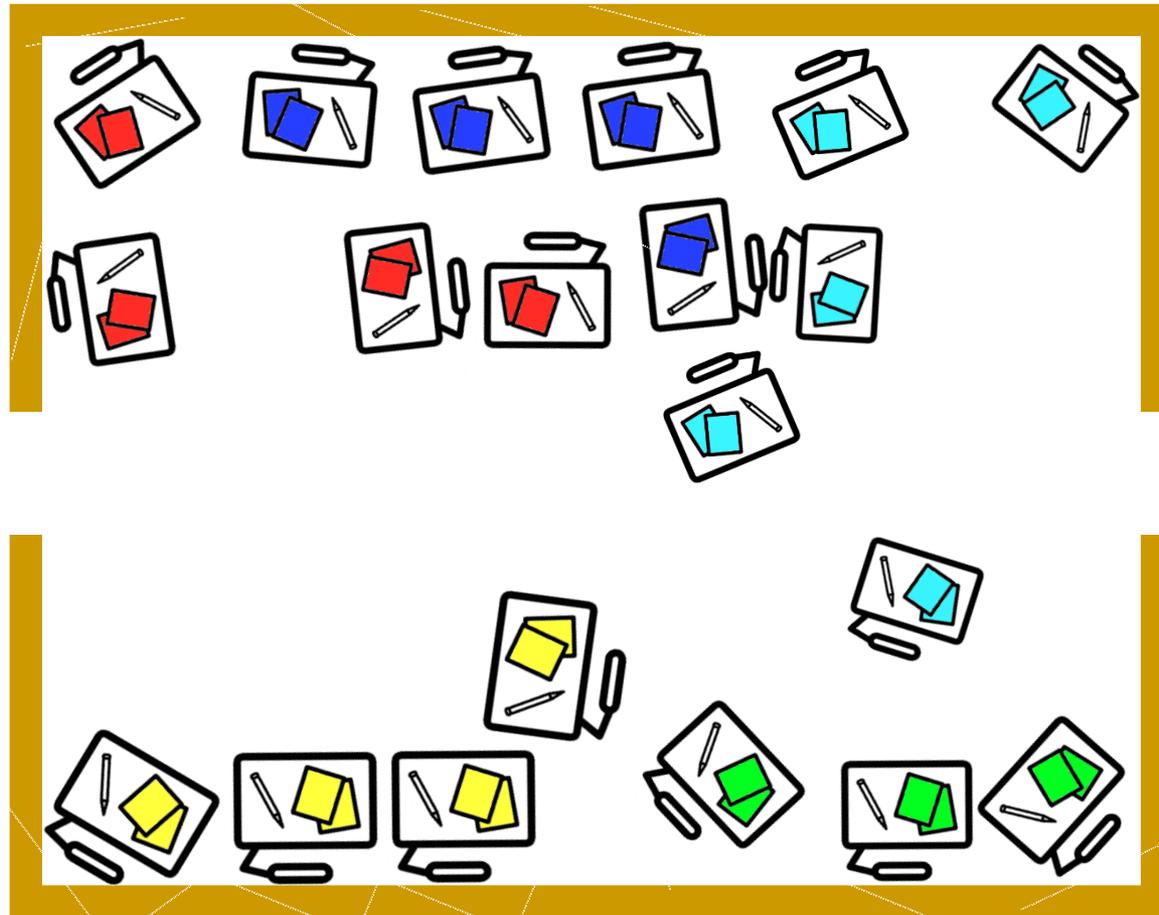
# Conclusions

- Viper: scalable, service-oriented  $\mu$ -architecture
- Hardware reconfiguration granularity as a **design choice**
- Much more graceful performance degradation
  - Can exploit available hardware to improve performance



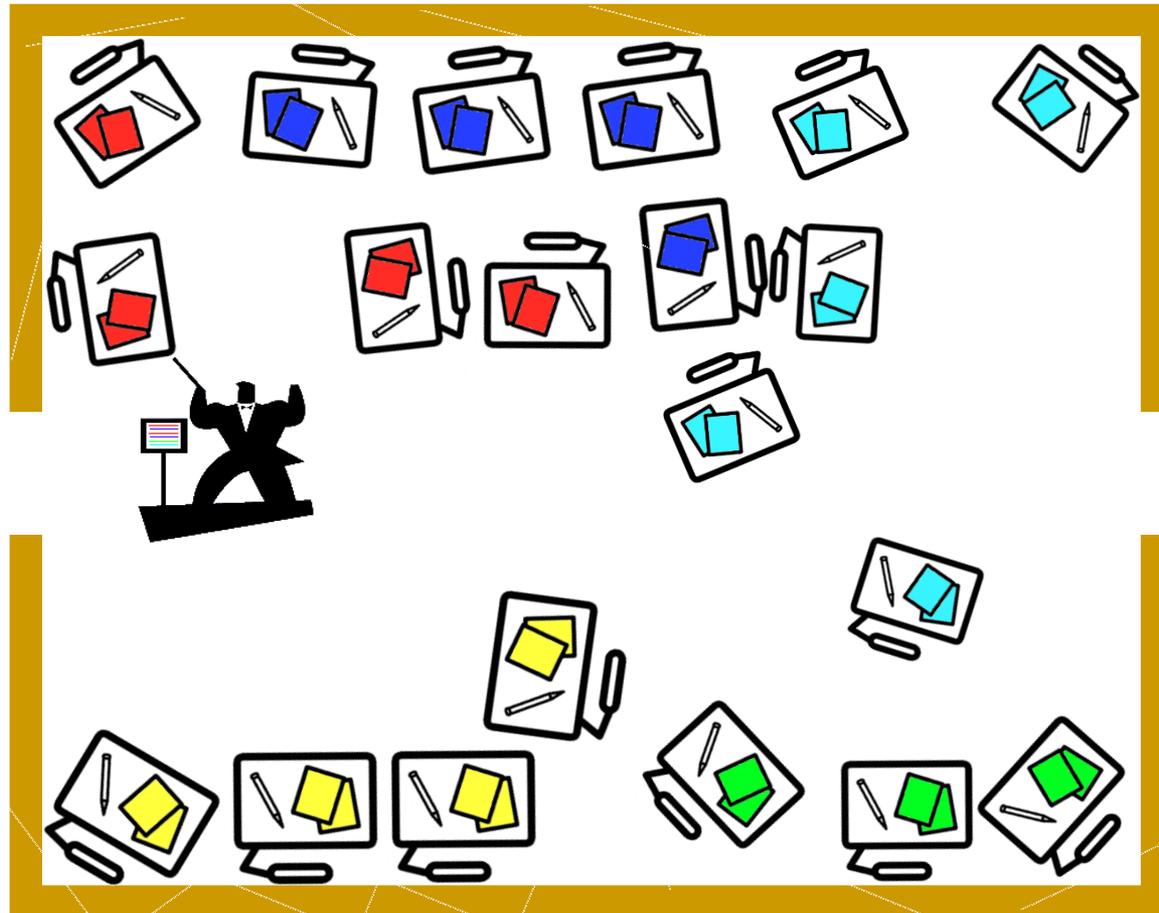
# Service-Oriented $\mu$ -Architecture

MicroProcessor



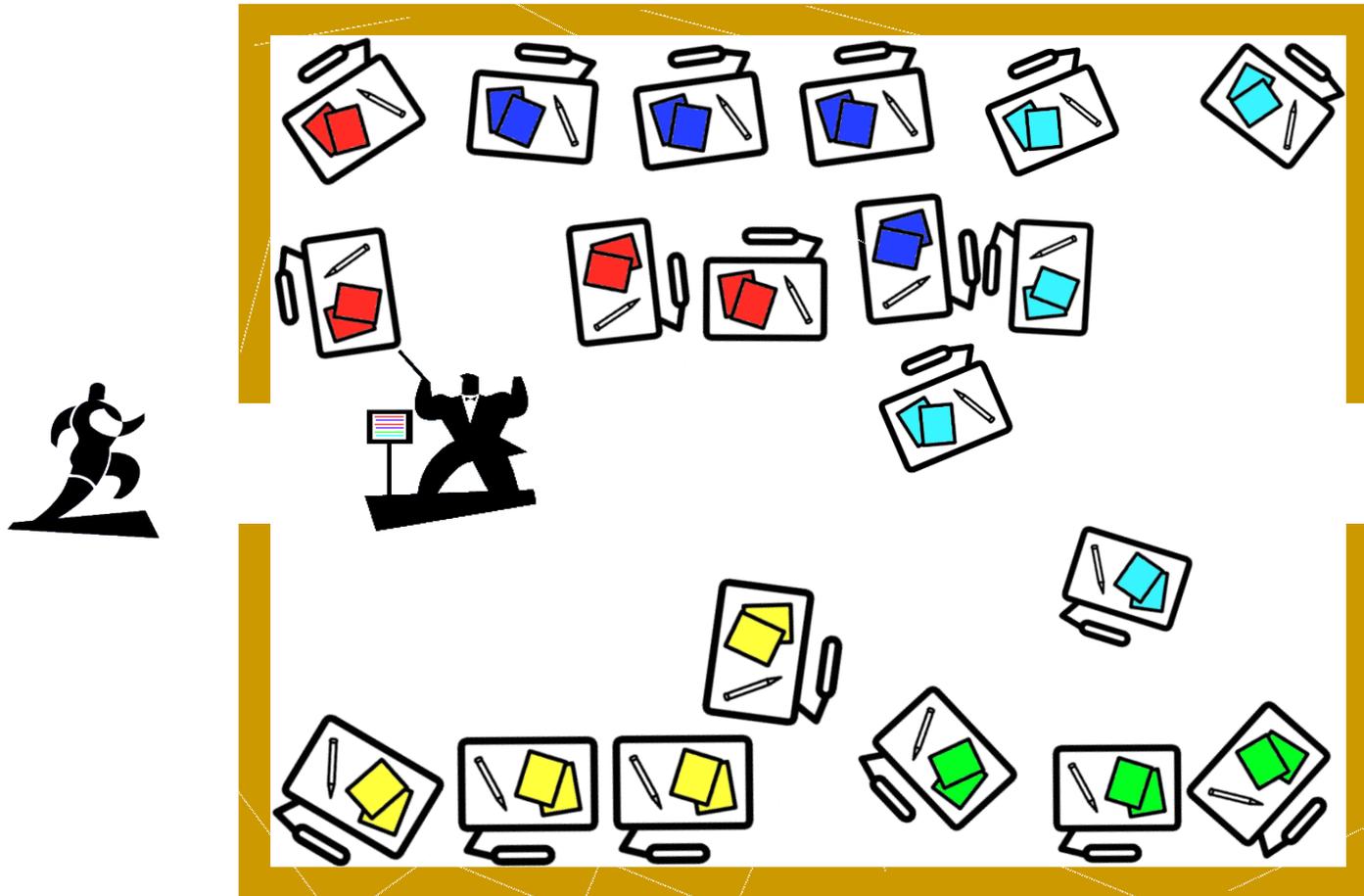
# Service-Oriented $\mu$ -Architecture

MicroProcessor



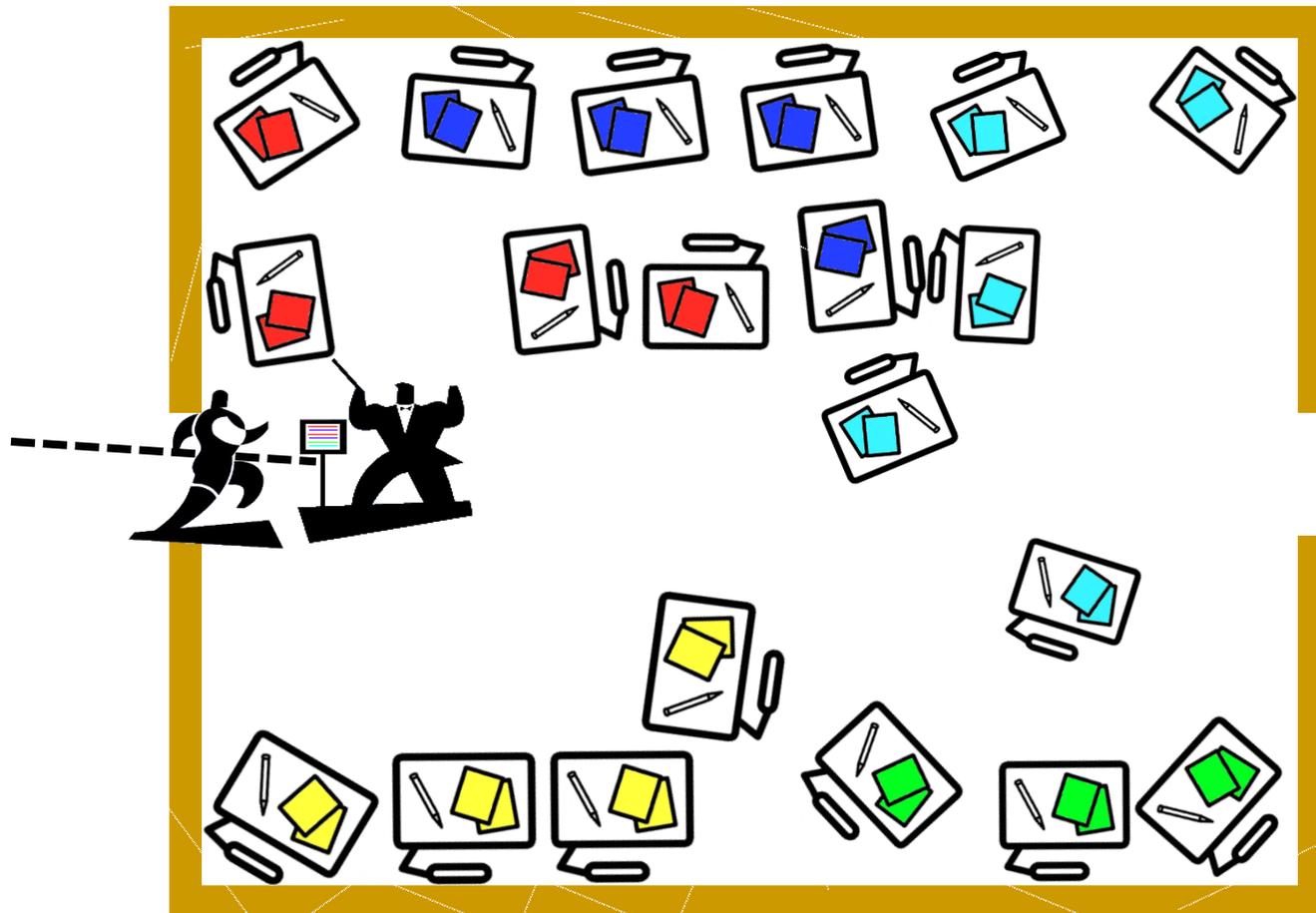
# Service-Oriented $\mu$ -Architecture

MicroProcessor



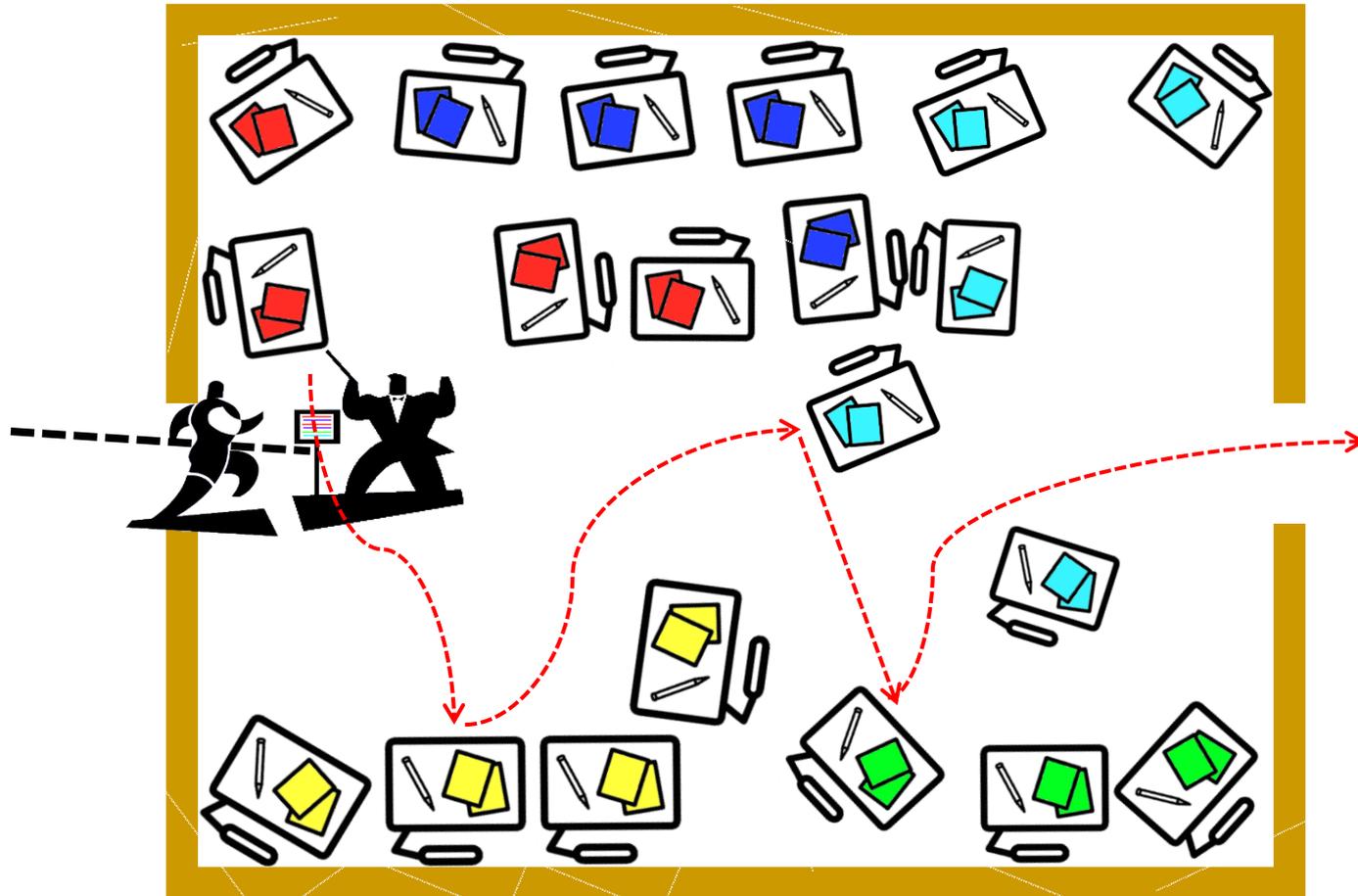
# Service-Oriented $\mu$ -Architecture

MicroProcessor



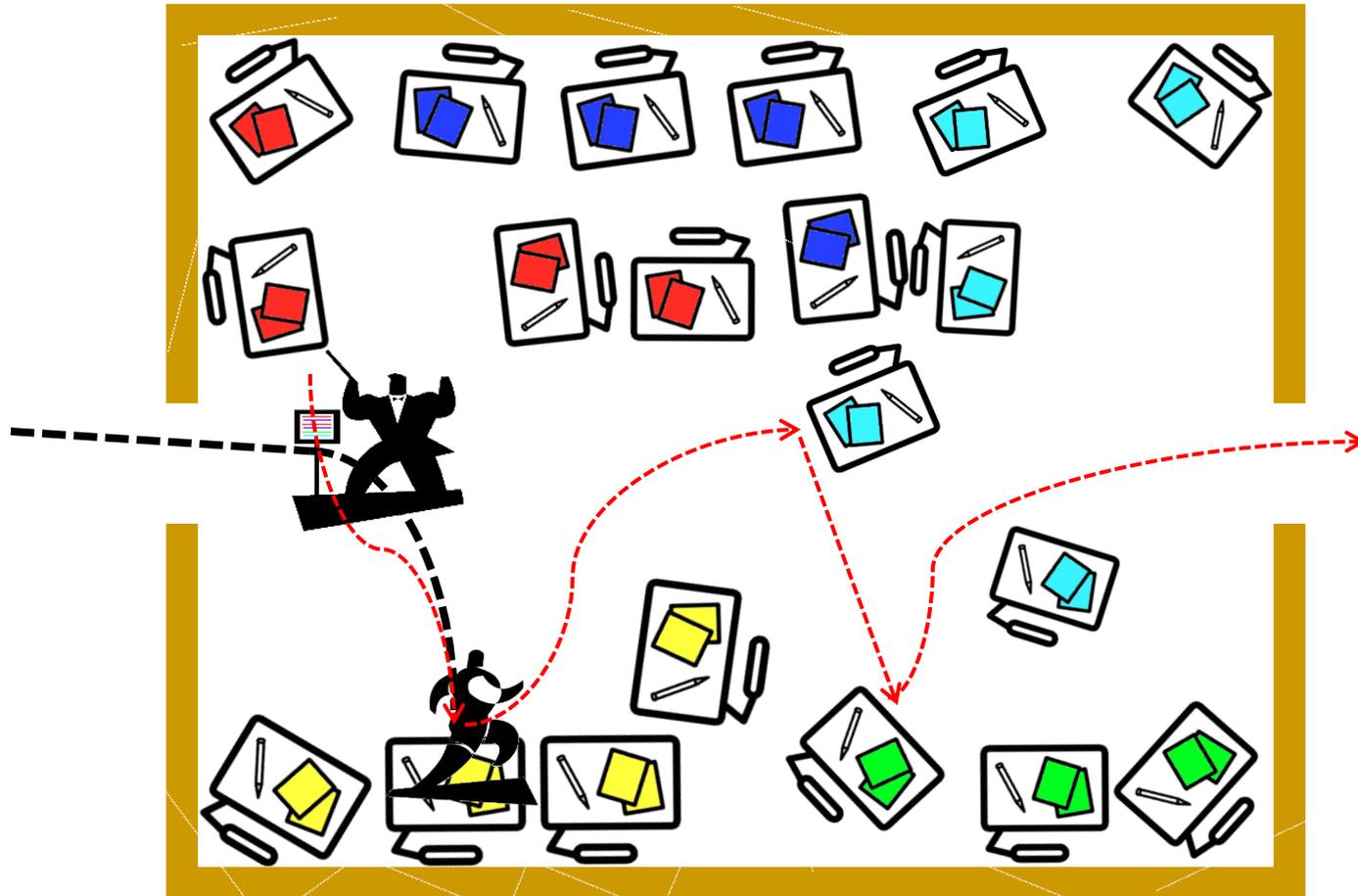
# Service-Oriented $\mu$ -Architecture

MicroProcessor



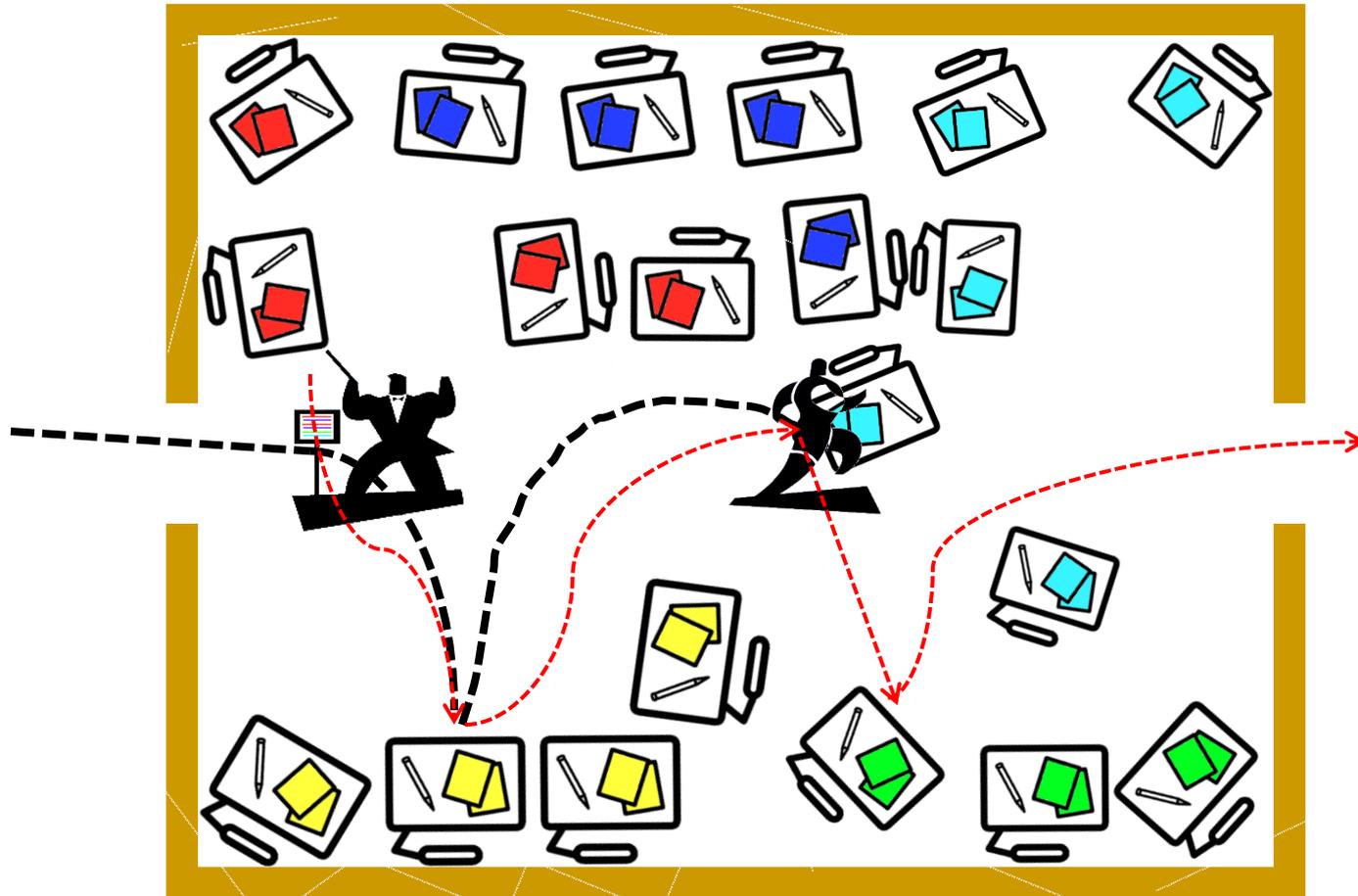
# Service-Oriented $\mu$ -Architecture

MicroProcessor



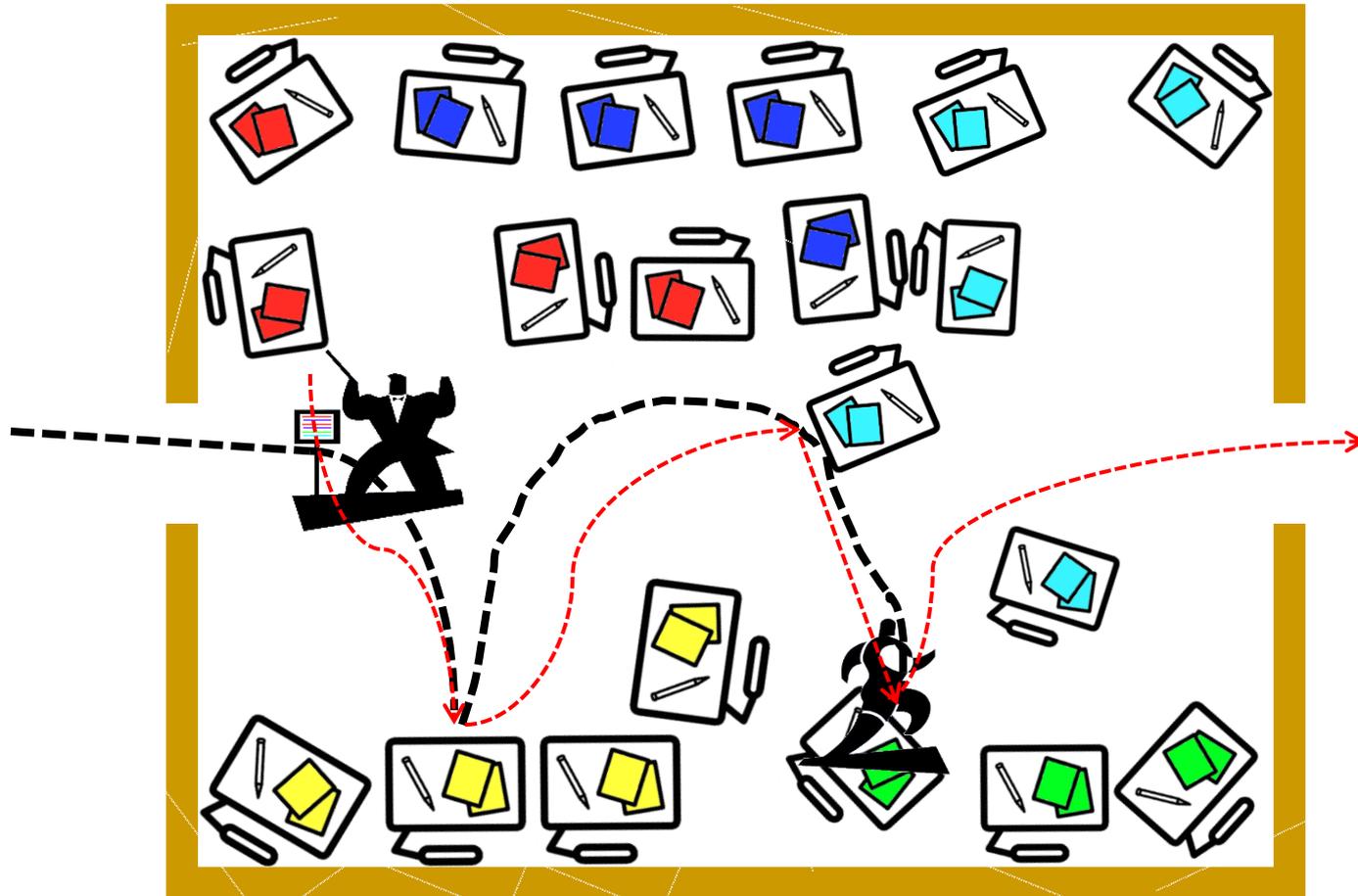
# Service-Oriented $\mu$ -Architecture

MicroProcessor



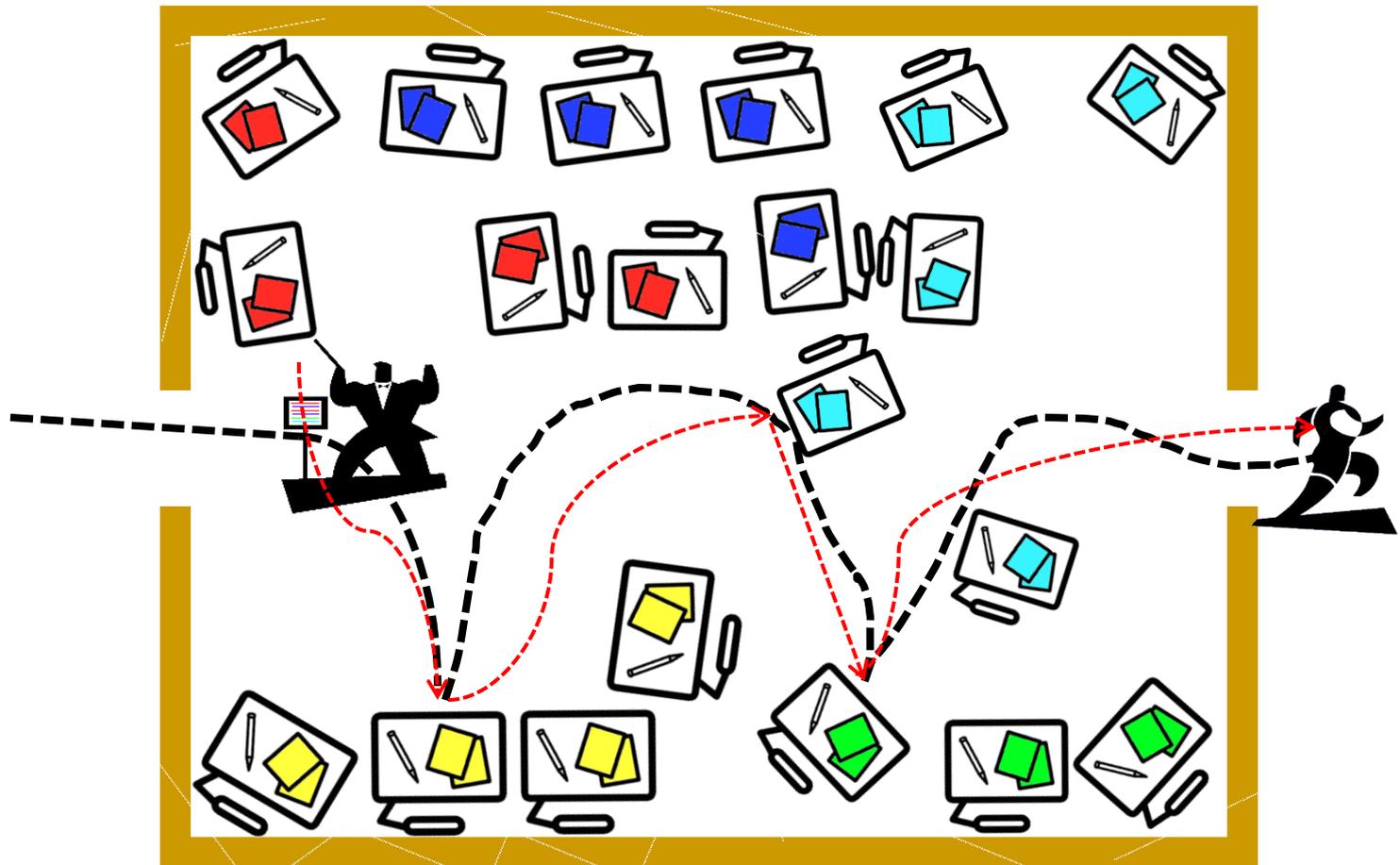
# Service-Oriented $\mu$ -Architecture

MicroProcessor



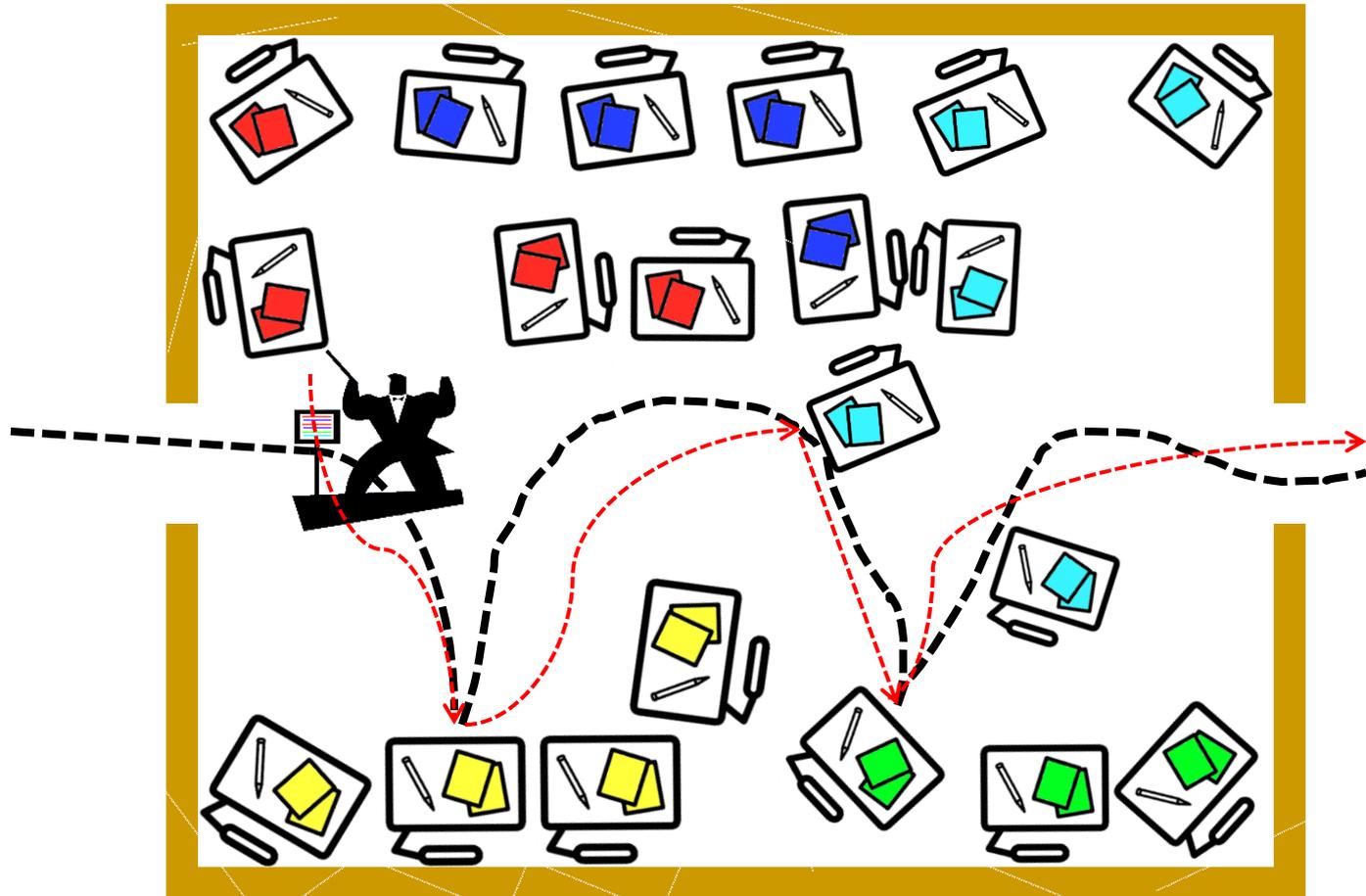
# Service-Oriented $\mu$ -Architecture

MicroProcessor



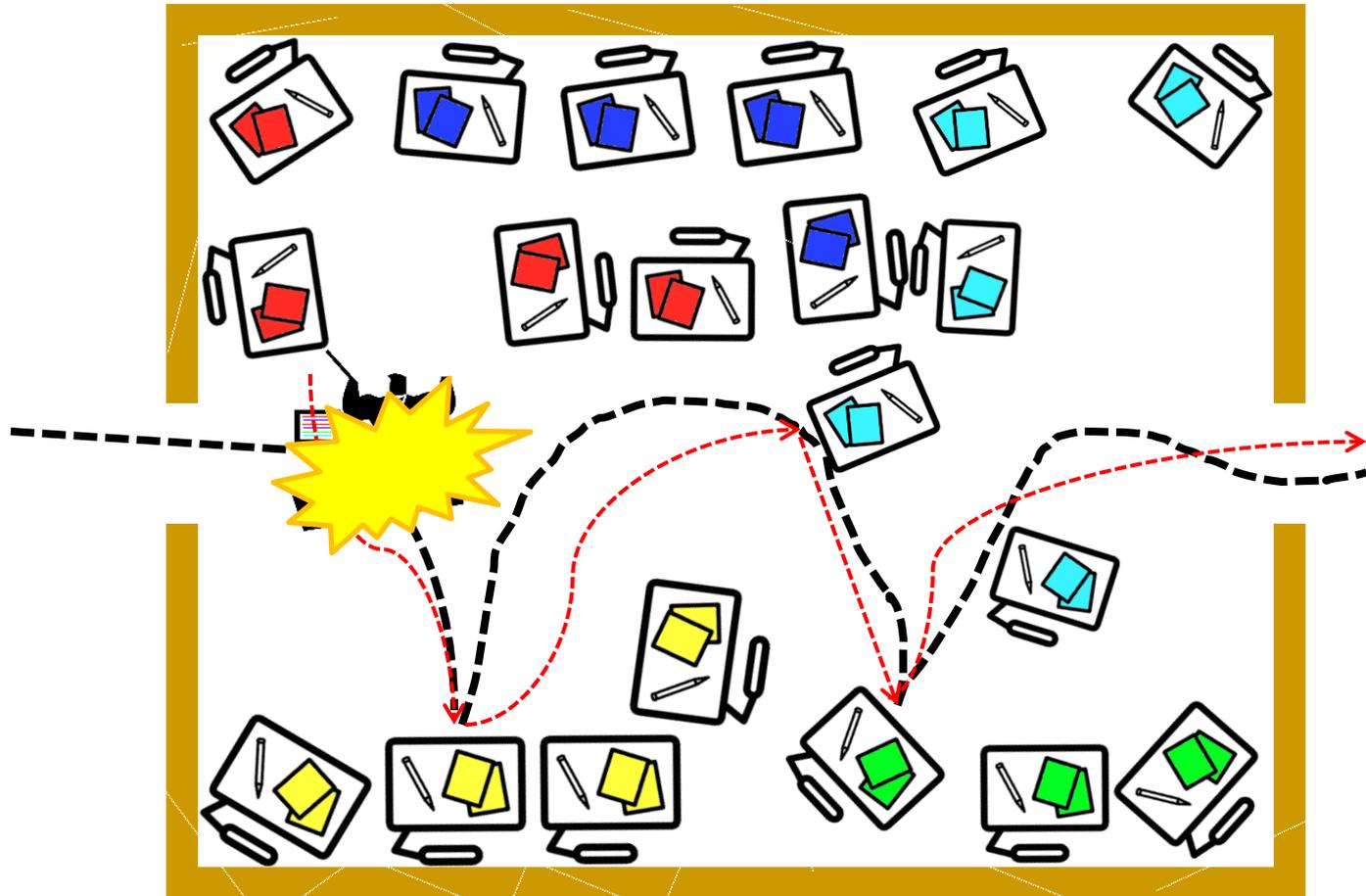
# Service-Oriented $\mu$ -Architecture

MicroProcessor

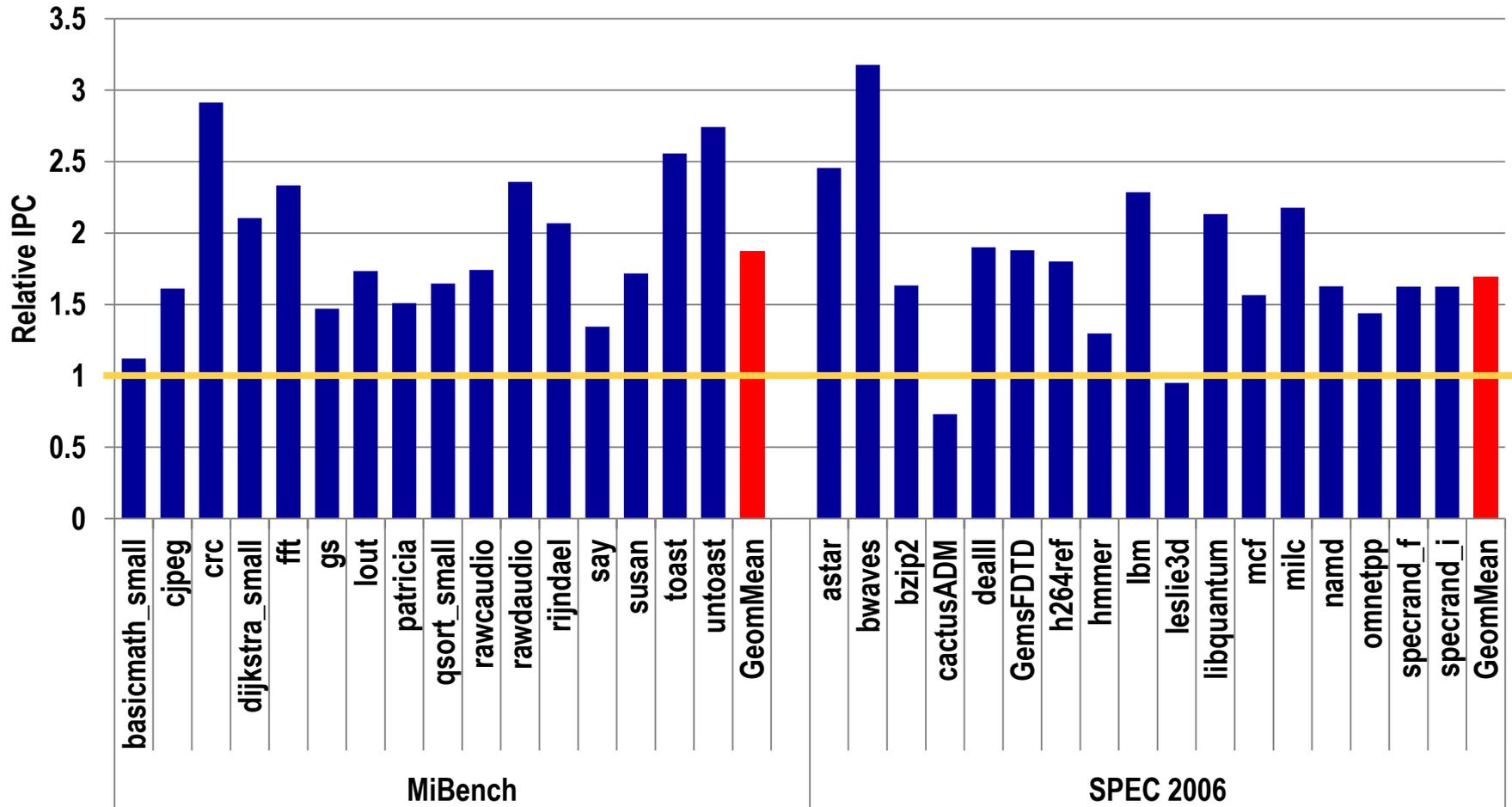


# Service-Oriented $\mu$ -Architecture

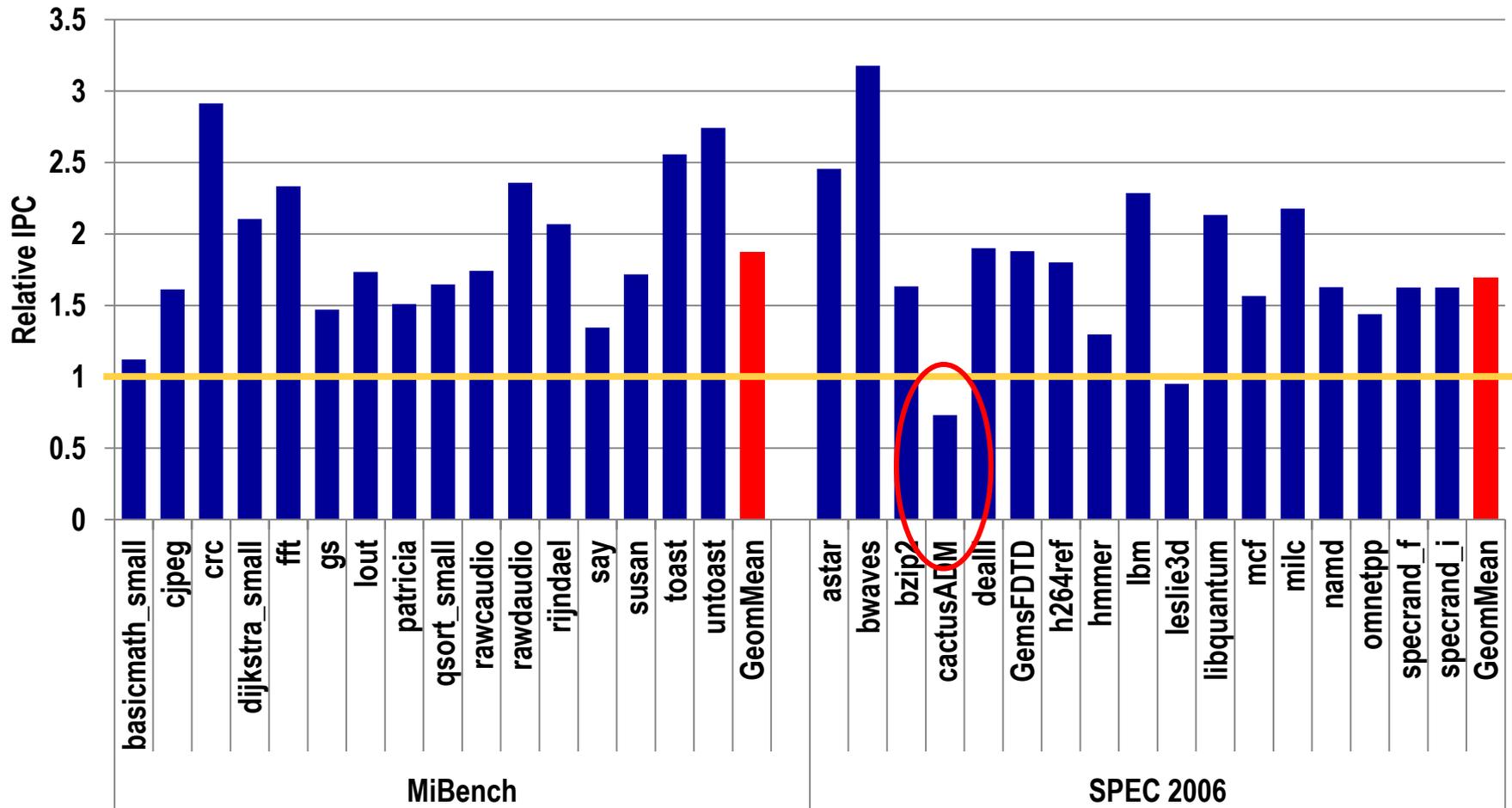
MicroProcessor



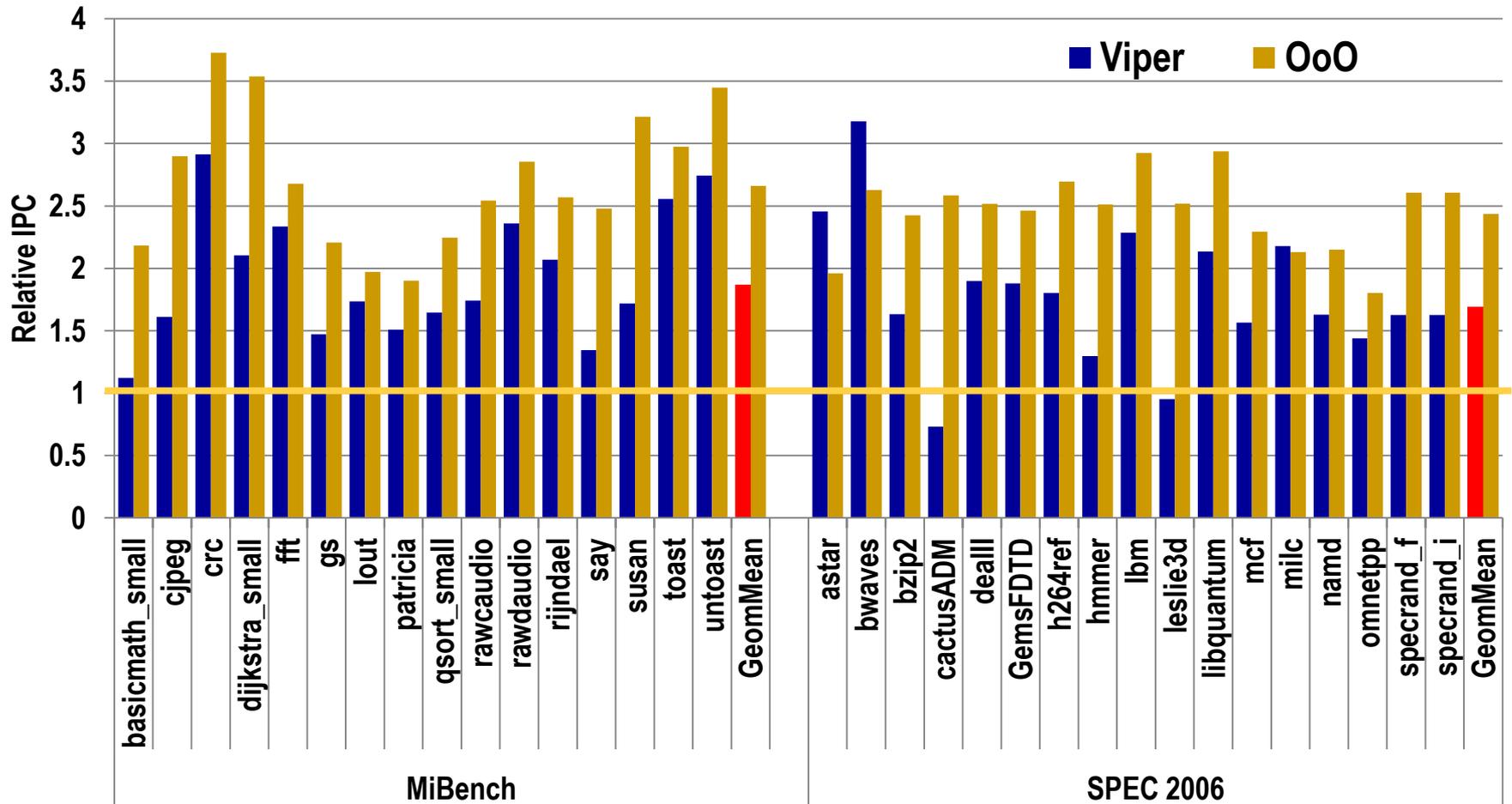
# Viper Enables Reliable OoO Execution



# Viper Enables Reliable OoO Execution



# Viper is Competitive vs Unprotected OoO



# Viper is Competitive vs Unprotected OoO

