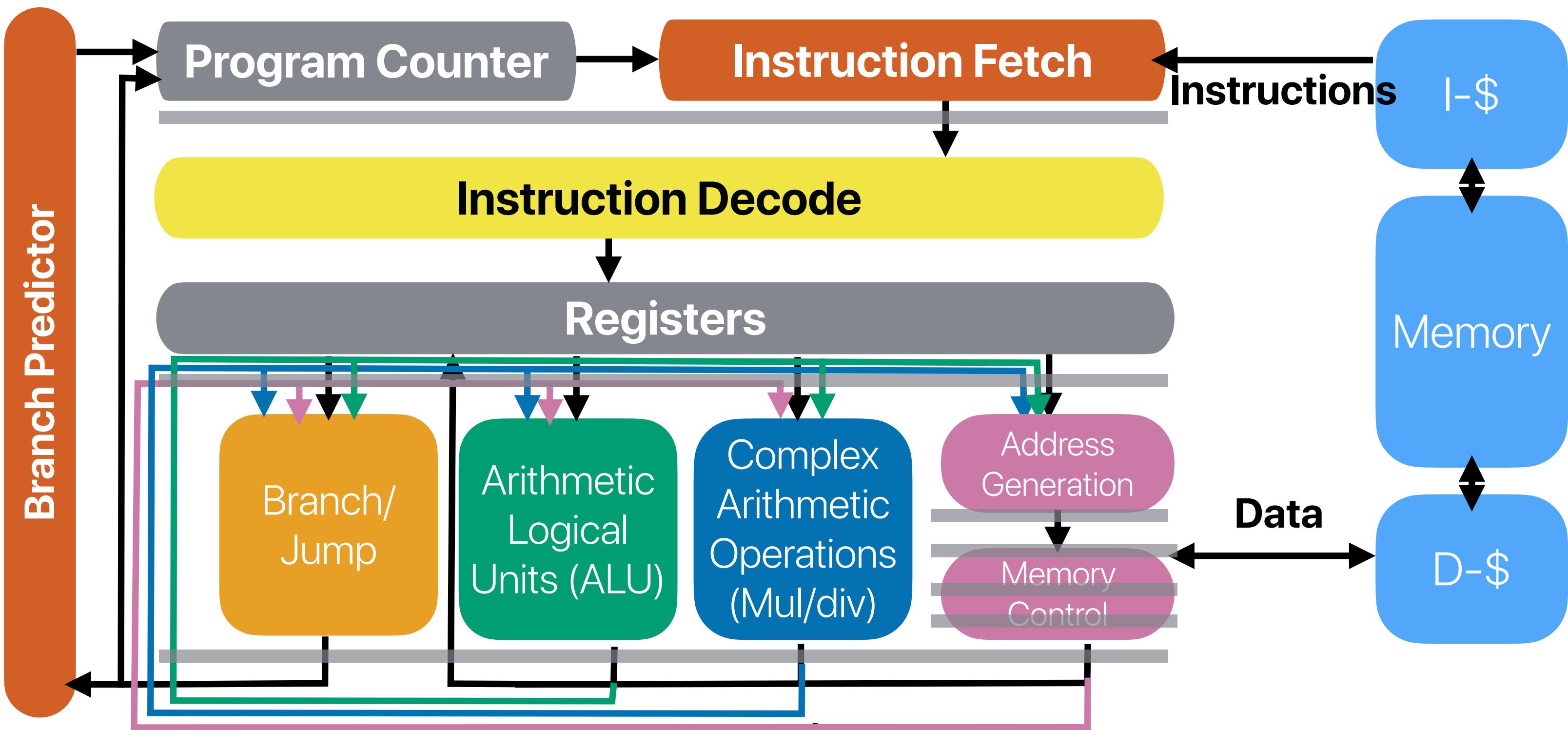


# **Modern Processor Design (IV): Try everything**

Hung-Wei Tseng

# Recap: Data “forwarding”



# Takeaways: data hazards

- More data dependencies, more likelihood of data hazards
- Stalls and data forwarding can both address data hazards to generate correct code execution results — but not very efficient
- Compiler optimizations can help, but to a limited extent



# How many of data hazards w/ Data Forwarding?

- How many pairs of back-to-back data dependences in the following x86 instructions will result in stalls even with data forwarding and both memory operations & xorl take 5 cycles?

# How many of data hazards w/ Data Forwarding?

- How many pairs of back-to-back data dependences in the following x86 instructions will result in stalls even with data forwarding and both memory operations & xorl take 5 cycles?

- ① movl (%rdi), %eax
- ② xorl (%rsi), %eax
- ③ movl %eax, (%rdi)
- ④ xorl (%rsi), %eax
- ⑤ movl %eax, (%rsi)
- ⑥ xorl %eax, (%rdi)

$*a \wedge= *b;$   
 $*b \wedge= *a;$   
 $*a \wedge= *b;$   
**D. 3**  
E. 4

	IF	ID	ALU/BR/AG	M1	M2	M3	M4/XORL	WB/Retire
1	(1)							
2	(2)	(1)						
3	(3)	(2)	(1)					
4	(3)	(2)		(1)				
5	(3)	(2)			(1)			
6	(3)	(2)				(1)		
7	(3)	(2)					(1)	
8	(4)	(3)	(2)					(1)
9	(4)	(3)		(2)				
10	(4)	(3)			(2)			
11	(4)	(3)				(2)		
12	(4)	(3)					(2)	
13	(5)	(4)	(3)					
14	(6)	(5)	(4)	(3)				
15	(6)	(5)		(4)	(3)			
16	(6)	(5)			(4)	(3)		
17	(6)	(5)				(4)		
18	(6)	(5)					(3)	
19	(6)	(5)					(4)	(3)
20	(6)		(5)				(4)	(4)
21			(6)	(5)				
22				(6)	(5)			
23					(6)	(5)		
24						(6)	(5)	
25							(6)	(5)
26								(6)

**19 cycles for 6 instructions  
CPI = 3.17!**

A close-up shot of Nick Wilde's face from the Disney Pixar movie Zootopia. He is looking directly at the camera with a serious expression. He has his signature spiky hair and is wearing a green shirt and tie.

Where do you think is always  
inefficient and why is it inefficient?



NEWS

## DMV Work Among Consultant McKinsey's State Contracts

From education and the future of work to high-speed rail, Gov. Gavin Newsom has turned at least four times to global consulting firm McKinsey & Co. in the past year. That includes bringing the company on-board to assist in the reinvention and modernization of the California Department of Motor Vehicles.

January 06, 2020 • Tribune News Service



# Takeaways: data hazards

- More data dependencies, more likelihood of data hazards
- Stalls and data forwarding can both address data hazards to generate correct code execution results — but not very efficient

# Let's extend the example a bit...

```

for(i = 0; i < count; i++) {
    int64_t temp = a[i];
    a[i] = b[i];
    b[i] = temp;
}
.L9:
①  movq    (%rdi,%rax), %rsi
②  movq    (%rcx,%rax), %r8
③  movq    %r8, (%rdi,%rax)
④  movq    %rsi, (%rcx,%rax)
⑤  addq    $8, %rax
⑥  cmpq    %r9, %rax
⑦  jne     .L9
⑧  movq    (%rdi,%rax), %rsi
⑨  movq    (%rcx,%rax), %r8
⑩  movq    %r8, (%rdi,%rax)
⑪  movq    %rsi, (%rcx,%rax)
⑫  addq    $8, %rax
⑬  cmpq    %r9, %rax
⑭  jne     .L9

```

	IF	ID	ALU/BR/AG	M1	M2	M3	M4/XORL	WB/Retire
1	(1)							
2	(2)	(1)						
3	(3)	(2)	(1)					
4	(4)	(3)	(2)	(1)				
5	(4)	(3)		(2)	(1)			
6	(4)	(3)			(2)	(1)		
7	(4)	(3)				(2)	(1)	
8	(4)	(3)					(2)	(1)
9	(5)	(4)	(3)					(2)
10	(6)	(5)	(4)	(3)				(2)
11	(7)	(6)	(5)	(4)	(3)			
12	(8)	(7)	(6)		(4)	(3)		
13	(9)	(8)	(7)			(4)	(3)	
14	(10)	(9)	(8)				(4)	(3)
15	(11)	(10)	(9)	(8)				(4)
16	(11)	(10)		(9)	(8)			
17	(11)	(10)			(9)	(8)		(5)
18	(11)	(10)				(9)	(8)	
19	(11)	(10)					(9)	(8)
20	(12)	(11)	(10)					(9)
21	(13)	(12)	(11)					
22	(14)	(13)	(12)		(11)	(10)		
23		(14)	(13)		(12)	(11)	(10)	
24			(14)		(13)	(12)	(11)	(10)

11 cycles for 7 instructions  
CPI = 1.57



# The effect of code optimization

- By reordering which pair of the following instruction stream can we reduce stalls without affecting the correctness of the code?

① movq (%rdi,%rax), %rsi  
② movq (%rcx,%rax), %r8  
③ movq %r8, (%rdi,%rax)  
④ movq %rsi, (%rcx,%rax)  
⑤ addq \$8, %rax  
⑥ cmpq %r9, %rax  
⑦ jne .L9

- A. (1) & (2)
- B. (2) & (3)
- C. (3) & (5)
- D. (4) & (6)
- E. No ordering can help reduce the stalls

# The effect of code optimization

- By reordering which pair of the following instruction stream can we reduce stalls without affecting the correctness of the code?

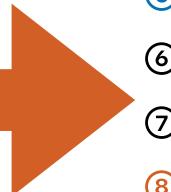
①	movq	(%rdi,%rax), %rsi
②	movq	(%rcx,%rax), %r8
③	movq	%r8, (%rdi,%rax)
④	movq	%rsi, (%rcx,%rax)
⑤	addq	\$8, %rax
⑥	cmpq	%r9, %rax
⑦	jne	.L9

- A. (1) & (2)
- B. (2) & (3)
- C. (3) & (5)
- D. (4) & (6)
- E. No ordering can help reduce the stalls

# Compiler optimization

```
for(i = 0; i < count; i++) {
    int64_t temp = a[i];
    a[i] = b[i];
    b[i] = temp;
}
```

```
.L9:
①  movq (%rdi,%rax), %rsi
②  movq (%rcx,%rax), %r8
③  addq %r8, (%rdi,%rax)
④  movq %rsi, (%rcx,%rax)
⑤  movq $8, %rax
⑥  cmpq %r9, %rax
⑦  jne .L9
⑧  movq (%rdi,%rax), %rsi
⑨  movq (%rcx,%rax), %r8
⑩  addq %r8, (%rdi,%rax)
⑪  movq %rsi, (%rcx,%rax)
⑫  movq $8, %rax
⑬  cmpq %r9, %rax
⑭  jne .L9
```



	IF	ID	ALU/BR/AG	M1	M2	M3	M4/XORL	WB/Retire
1	(1)							
2	(2)	(1)						
3	(3)	(2)	(1)					
4	(4)	(3)	(2)	(1)				
5	(5)	(4)	(3)	(2)	(1)			
6	(5)	(4)		(2)	(1)			
7	(5)	(4)			(2)	(1)		
8	(6)	(5)	(4)			(2)		(1)
9	(7)	(6)	(5)	(4)				(2)
10	(8)	(7)	(6)	(5)	(4)			(3)
11	(9)	(8)	(7)	(5)	(4)	(5)	(4)	
12	(10)	(9)	(8)		(5)	(4)		
13	(11)	(10)	(9)	(8)		(5)		(4)
14	(11)	(10)	(10)	(9)	(8)			(5)
15	(11)	(10)		(9)	(8)		(9)	(6)
16	(11)	(10)			(9)	(8)		(7)
17	(12)	(11)				(9)	(8)	(8)
18	(12)	(11)						(9)
19	(13)	(12)						(10)
20	(14)	(13)						
21		(14)						
22								
23								
24								
25								

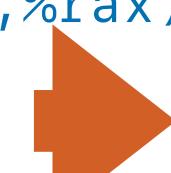
9 cycles for 7 instructions  
CPI = 1.29

# Missing opportunities: if we know the loop always run even times?

```
for(i = 0; i < count; i++) {
    int64_t temp = a[i];
    a[i] = b[i];
    b[i] = temp;
}
```

Opportunities of hiding data hazards  
through out-of-order execution!

```
:
movq (%rcx,%rax), %r8 .L9:
movq (%rdi,%rax), %rsi ① movq (%rcx,%rax), %r8
addq $8, %rax ② movq (%rdi,%rax), %rsi
movq %r8, -8(%rdi,%rax) ③ addq $8, %rax
movq %rsi, -8(%rcx,%rax) ④ movq %r8, -8(%rdi,%rax)
cmpq %r9, %rax ⑤ movq %rsi, -8(%rcx,%rax)
jne .L9
:
movq (%rcx,%rax), %r8
movq (%rdi,%rax), %rsi
addq $8, %rax
movq %r8, -8(%rdi,%rax)
movq %rsi, -8(%rcx,%rax)
cmpq %r9, %rax
jne .L9
:
    ⑥ movq (%rcx,%rax), %r8
    ⑦ movq (%rdi,%rax), %rsi
    ⑧ cmpq %r9, %rax
    ⑨ jne .L9
    ⑩ addq $8, %rax
    ⑪ movq %r8, -8(%rdi,%rax)
    ⑫ movq %rsi, -8(%rcx,%rax)
    ⑬ cmpq %r9, %rax
    ⑭ jne .L9
```



	IF	ID	ALU/BR/AG	M1	M2	M3	M4/XORL	WB/Retire
1	(1)							
2	(2)	(1)						
3	(3)	(2)	(1)					
4	(4)	(3)	(2)	(1)				
5	(5)	(4)	(3)	(2)	(1)			
6	(5)	(4)		(2)	(1)			
7	(6)	(5)			(2)	(1)		
8	(6)	(5)		(4)				
9	(7)	(6)	(5)	(4)				
10	(8)	(7)	(6)	(5)	(4)			
11	(9)	(8)	(7)	(6)	(5)	(4)		
12	(10)	(9)	(8)	(7)	(6)	(5)	(4)	
13	(11)	(10)	(9)	(8)	(7)	(6)	(5)	
14	(12)	(11)	(10)					
15	(13)	(12)	(11)	(10)				
16	(14)	(13)	(12)	(11)	(10)			
17		(14)	(13)	(12)	(11)	(10)		
18			(14)	(13)	(12)	(11)	(10)	
19				(14)	(13)	(12)	(11)	
20					(14)	(13)	(12)	
21						(14)	(13)	
22							(14)	
23								(14)

7 cycles for 7 instructions  
CPI = 1

# Missing opportunities: if we know the loop always run even times?

```
for(i = 0; i < count; i++) {
    int64_t temp = a[i];
    a[i] = b[i];
    b[i] = temp;
}
```

Compiler can only do this when it's 100% for sure count is always an even number! —loop unrolling

Compilers are limited by the number of registers available

to the software!

7 cycles for 7 instructions

CPI = 1

	IF	ID	ALU/BR/AG	M1	M2	M3	M4/XORL	WB/Retire
1	(1)							
2	(2)	(1)						
3	(3)	(2)		(1)				
4	(4)	(3)		(2)	(1)			
5	(5)	(4)		(3)	(2)	(1)		
6	(5)	(4)			(2)	(1)		
7	(5)						(1)	
8	(6)	(5)		(4)			(2)	(1)
9				(5)	(4)	(3)	(2)	(1)
10				(6)	(5)	(4)	(3)	(2)
11	(9)	(8)		(7)	(6)	(5)	(4)	(3)
12				(8)	(7)	(6)	(5)	(4)
13	(11)	(10)		(9)				
14	(12)	(11)		(10)				
15	(13)	(12)		(11)	(10)			
16	(14)	(13)		(12)	(11)	(10)		
17				(13)	(12)	(11)	(10)	(9)
18				(14)	(13)	(12)	(11)	(10)
19					(14)	(13)	(12)	(11)
20						(14)	(13)	(12)
21							(14)	(13)
22							(14)	(13)
23								(14)

# Limitations of Compiler Optimizations

- If the hardware (e.g., pipeline changes), the same compiler optimization may not be that helpful
- The compiler can only optimize on static instructions, but cannot optimize dynamic instruction
  - Compiler cannot predict branches
  - Compiler does not know if cache has the data/instructions

# Takeaways: data hazards

- More data dependencies, more likelihood of data hazards
- Stalls and data forwarding can both address data hazards to generate correct code execution results — but not very efficient
- Compiler optimizations can help, but to a limited extent

# Missing opportunities

```
for(i = 0; i < count; i++) {  
    int64_t temp = a[i];  
    a[i] = b[i];  
    b[i] = temp;  
}
```

Processor can predict what  
should happen and unroll the

loop "dynamically"

```
:  
    movq    (%rcx,%rax), %r8  
    movq    (%rdi,%rax), %rsi  
    addq    $8, %rax  
    movq    %r8, -8(%rdi,%rax)  
    movq    %rsi, -8(%rcx,%rax)  
    cmpq    %r9, %rax  
    jne     .L9  
    movq    (%rcx,%rax), %r8  
    movq    (%rdi,%rax), %rsi  
    cmpq    %r9, %rax  
    jne     .L9  
    addq    $8, %rax  
    movq    %r8, -8(%rdi,%rax)  
    movq    %rsi, -8(%rcx,%rax)  
    cmpq    %r9, %rax  
    jne     .L9
```



① movq (%rcx,%rax), %r8  
② movq (%rdi,%rax), %rsi  
③ addq \$8, %rax  
④ movq %r8, -8(%rdi,%rax)  
⑤ movq %rsi, -8(%rcx,%rax)  
⑥ movq (%rcx,%rax), %r8  
⑦ movq (%rdi,%rax), %rsi  
⑧ cmpq %r9, %rax  
⑨ jne .L9  
⑩ addq \$8, %rax  
⑪ movq %r8, -8(%rdi,%rax)  
⑫ movq %rsi, -8(%rcx,%rax)  
⑬ cmpq %r9, %rax  
⑭ jne .L9

	IF	ID	ALU/BR/AG	M1	M2	M3	M4/XORL	WB/Retire
1	(1)							
2		(2)						
3	(3)	(2)						
4	(4)	(3)						
5		(4)						
6	(5)	(4)						
7	(5)	(4)						
8		(5)						
9								
10	(8)	(7)						
11	(9)	(8)						
12	(10)	(9)						
13	(11)	(10)						
14	(12)	(11)						
15	(13)	(12)						
16	(14)	(13)						
17		(14)						
18								
19								
20								
21								
22								
23								

7 cycles for 7  
instructions

CPI = 1

# **Dynamic instruction scheduling/ Out-of-order (OoO) execution**

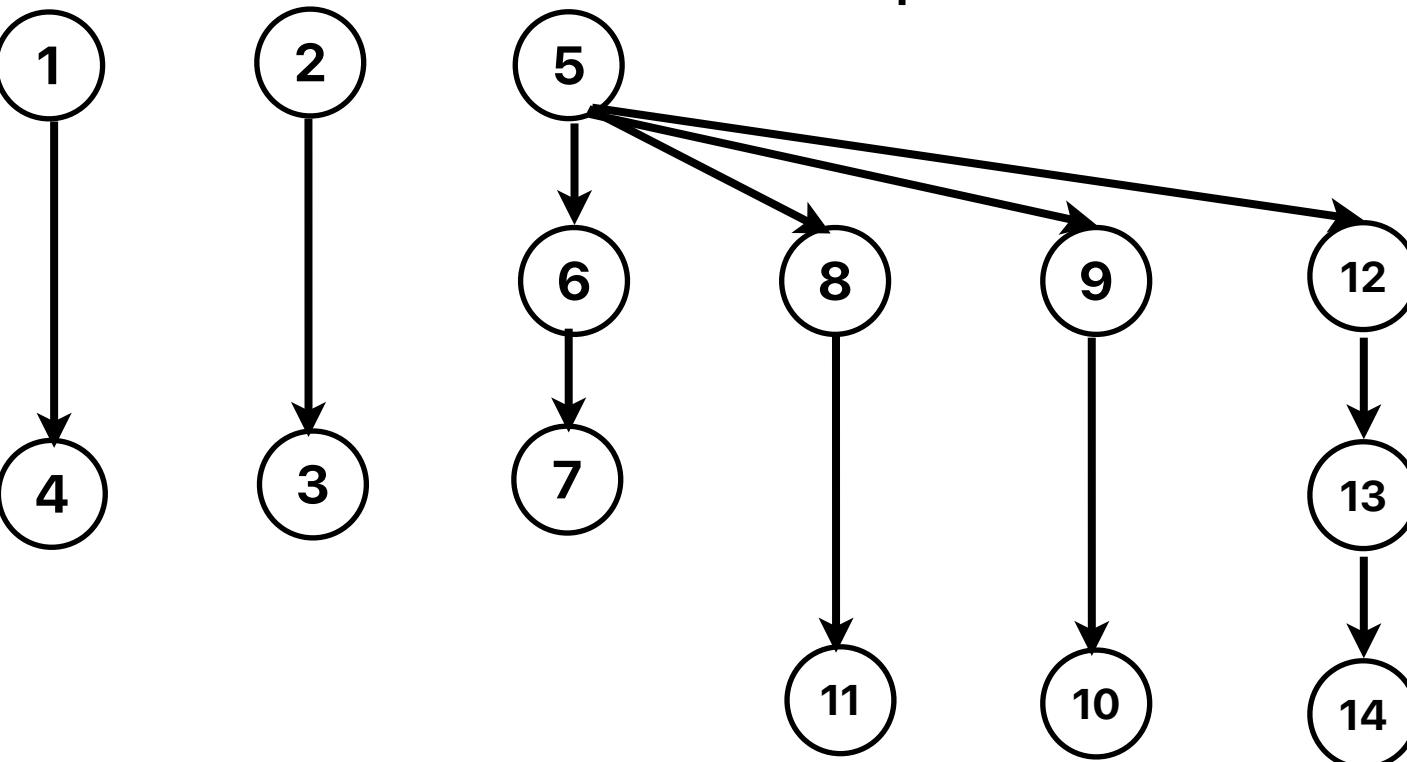
# What do you need to execution an instruction?

- Whenever the instruction is decoded — put decoded instruction somewhere
- Whenever the inputs are ready — **all data dependencies are resolved**
- Whenever the target functional unit is available

# Scheduling instructions: based on data dependencies

- Draw the data dependency graph, put an arrow if an instruction depends on the other.

①	movq	(%rdi,%rax), %rsi
②	movq	(%rcx,%rax), %r8
③	movq	%r8, (%rdi,%rax)
④	movq	%rsi, (%rcx,%rax)
⑤	addq	\$8, %rax
⑥	cmpq	%r9, %rax
⑦	jne	.L9
⑧	movq	(%rdi,%rax), %rsi
⑨	movq	(%rcx,%rax), %r8
⑩	movq	%r8, (%rdi,%rax)
⑪	movq	%rsi, (%rcx,%rax)
⑫	addq	\$8, %rax
⑬	cmpq	%r9, %rax
⑭	jne	.L9

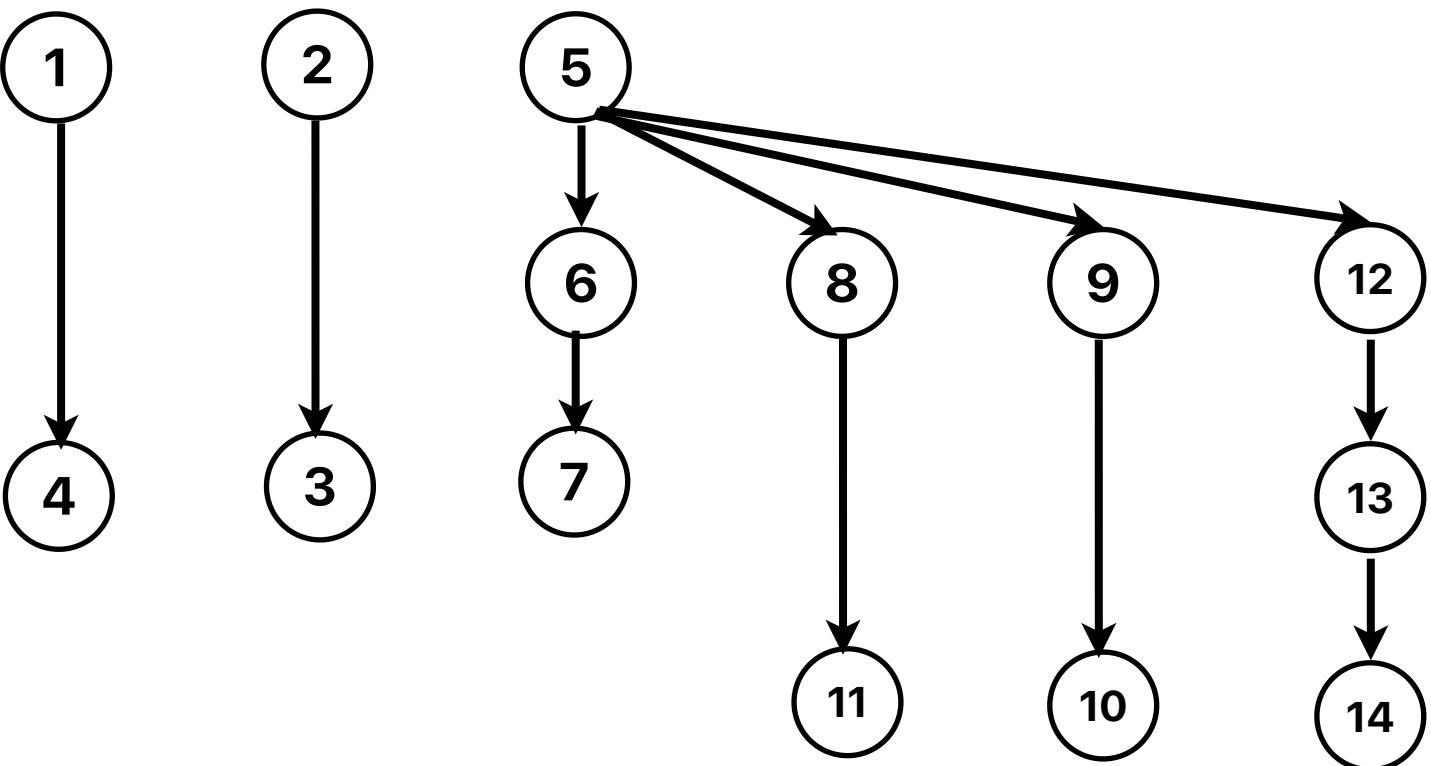


- **In theory**, instructions without dependencies can be executed in parallel or out-of-order
- Instructions with dependencies (on the same path) can never be reordered

# If we can predict the future ...

- Consider the following dynamic instructions:

```
① movq    (%rdi,%rax), %rsi  
② movq    (%rcx,%rax), %r8  
③ movq    %r8, (%rdi,%rax)  
④ movq    %rsi, (%rcx,%rax)  
⑤ addq    $8, %rax  
⑥ cmpq    %r9, %rax  
⑦ jne     .L9  
⑧ movq    (%rdi,%rax), %rsi  
⑨ movq    (%rcx,%rax), %r8  
⑩ movq    %r8, (%rdi,%rax)  
⑪ movq    %rsi, (%rcx,%rax)  
⑫ addq    $8, %rax  
⑬ cmpq    %r9, %rax  
⑭ jne     .L9
```



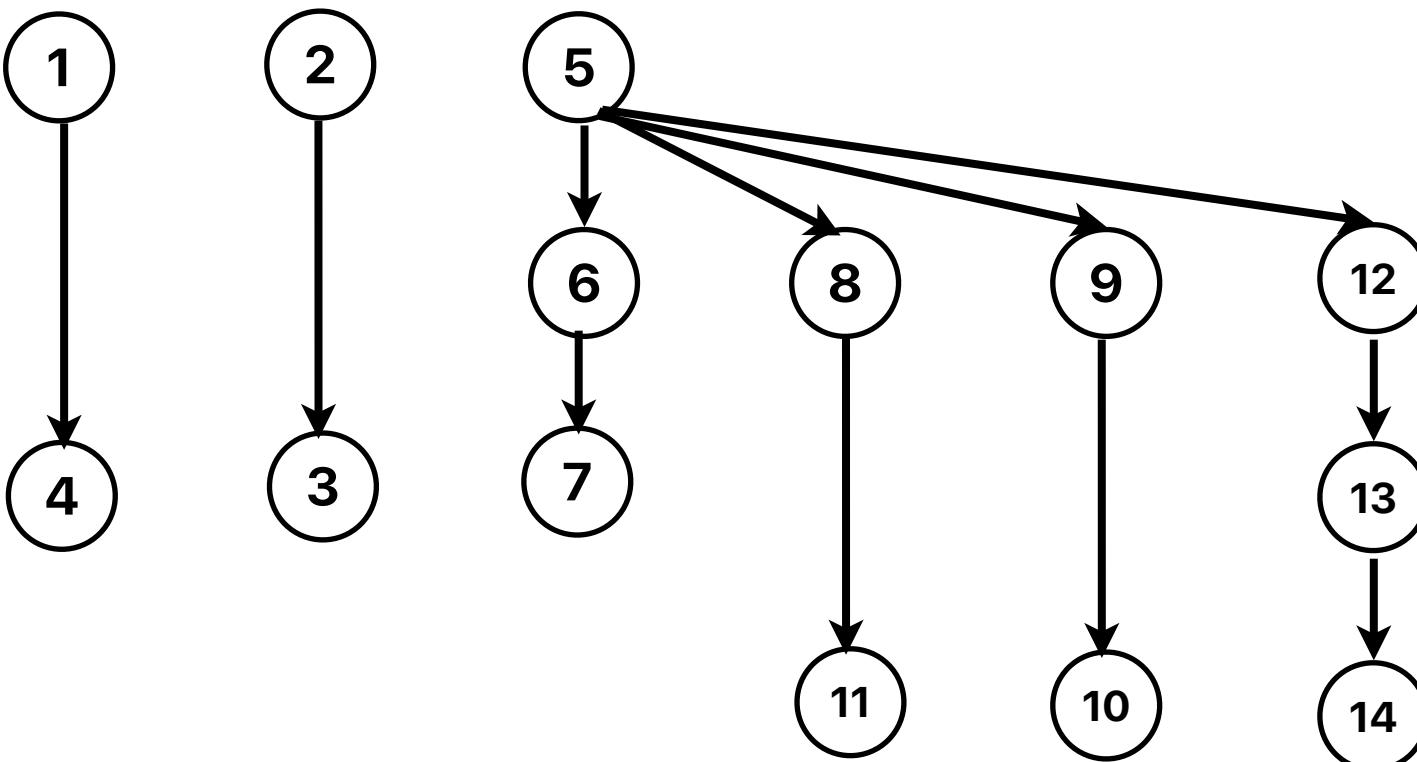
Which of the following pair can we reorder without affecting the correctness if the **branch prediction is perfect**?

- A. (1) and (2)
- B. (2) and (5)
- C. (4) and (5)
- D. (3) and (8)
- E. (6) and (12)

# If we can predict the future ...

- Consider the following dynamic instructions:

```
① movq    (%rdi,%rax), %rsi  
② movq    (%rcx,%rax), %r8  
③ movq    %r8, (%rdi,%rax)  
④ movq    %rsi, (%rcx,%rax)  
⑤ addq    $8, %rax  
⑥ cmpq    %r9, %rax  
⑦ jne     .L9  
⑧ movq    (%rdi,%rax), %rsi  
⑨ movq    (%rcx,%rax), %r8  
⑩ movq    %r8, (%rdi,%rax)  
⑪ movq    %rsi, (%rcx,%rax)  
⑫ addq    $8, %rax  
⑬ cmpq    %r9, %rax  
⑭ jne     .L9
```



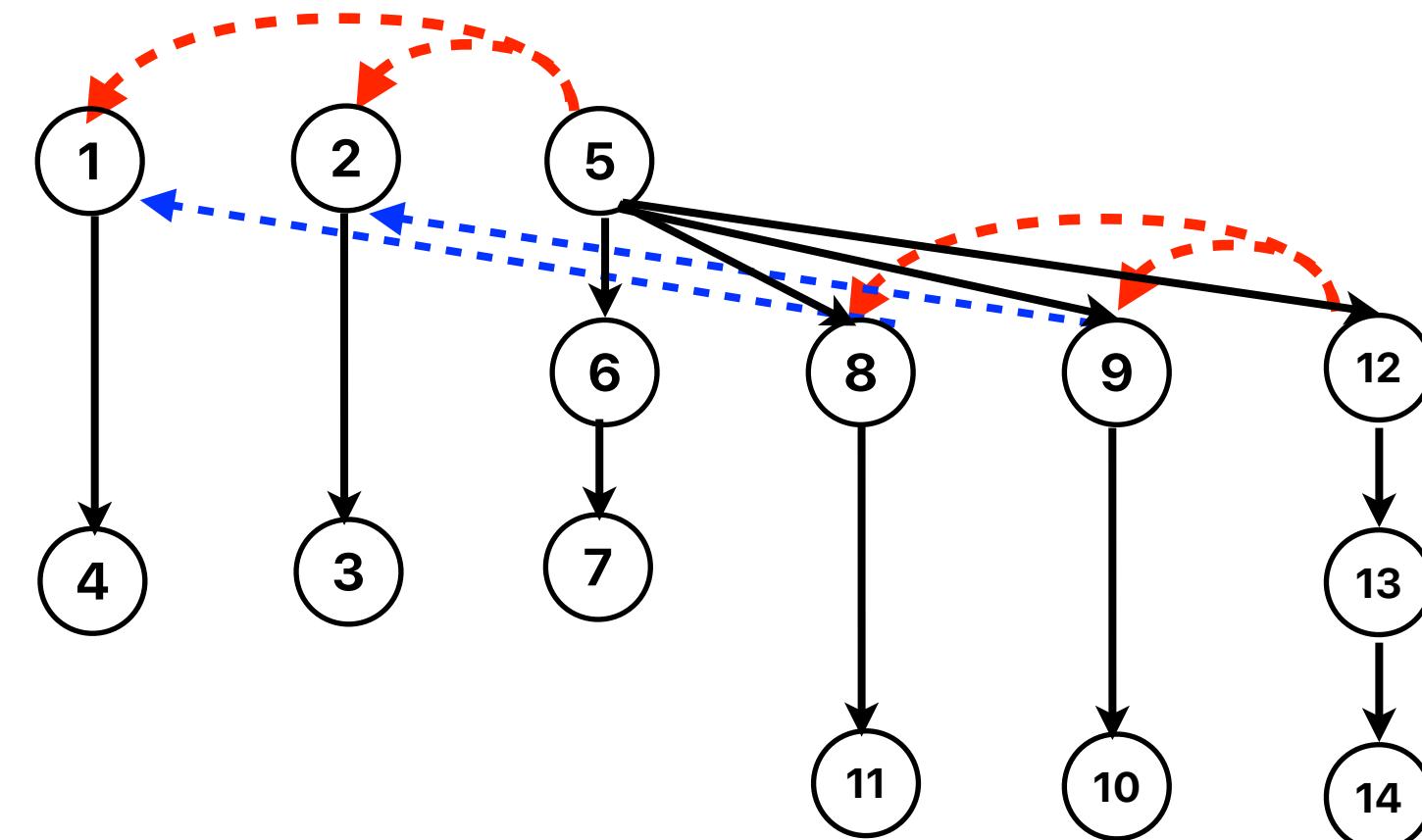
Which of the following pair can we reorder without affecting the correctness if the **branch prediction is perfect**?

- A. (1) and (2)
- B. (2) and (5) **WAR (Write After Read): a later instruction (5) overwrites the source of an earlier one (2)**
- C. (4) and (5) **WAR (Write After Read): a later instruction (5) overwrites the source of an earlier one (2)**
- D. (3) and (8) **WAW (Write After Write): a later instruction (8) overwrites the output of an earlier one (4)**
- E. (6) and (12) **WAR (Write After Read): a later instruction (12) overwrites the source of an earlier one (6)**

# False dependencies

- We are still limited by **false dependencies**
- They are not “true” dependencies because they don’t have an arrow in data dependency graph
  - WAR (Write After Read): a later instruction overwrites the source of an earlier one
  - WAW (Write After Write): a later instruction overwrites the output of an earlier one

①	movq	(%rdi,%rax), %rsi
②	movq	(%rcx,%rax), %r8
③	movq	%r8, (%rdi,%rax)
④	movq	%rsi, (%rcx,%rax)
⑤	addq	\$8, %rax
⑥	cmpq	%r9, %rax
⑦	jne	.L9
⑧	movq	(%rdi,%rax), %rsi
⑨	movq	(%rcx,%rax), %r8
⑩	movq	%r8, (%rdi,%rax)
⑪	movq	%rsi, (%rcx,%rax)
⑫	addq	\$8, %rax
⑬	cmpq	%r9, %rax
⑭	jne	.L9



# Takeaways: data hazards

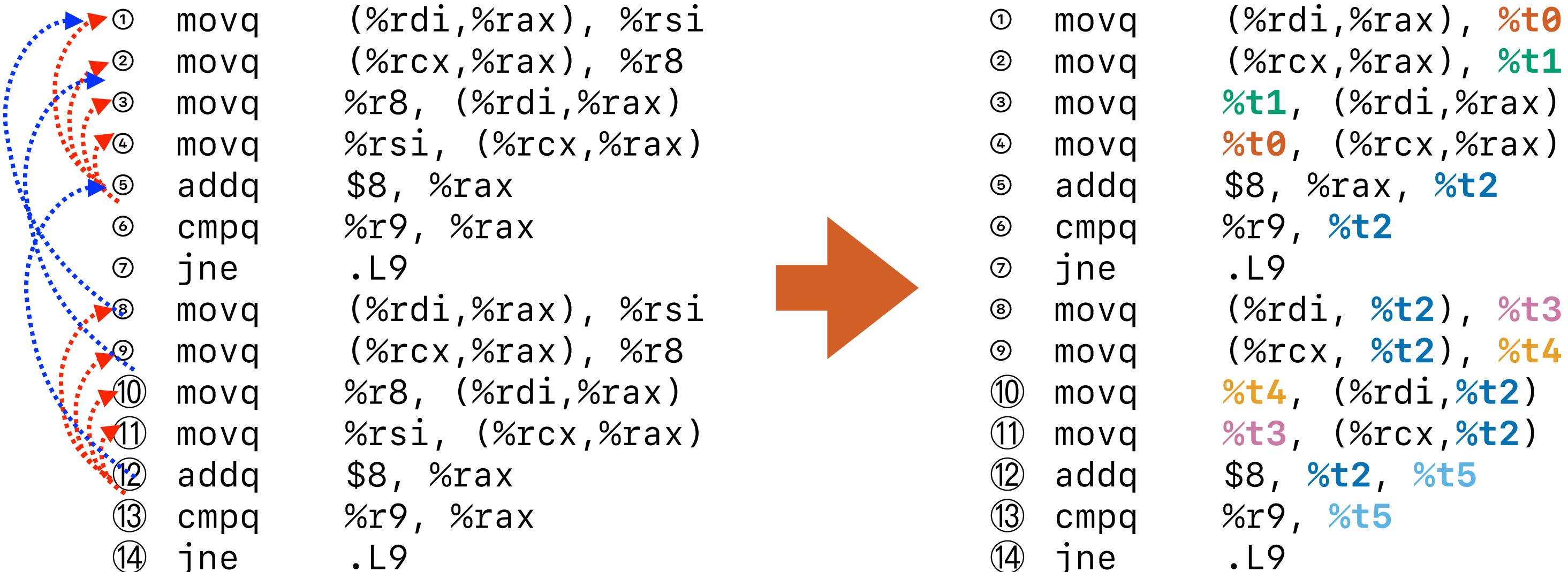
- More data dependencies, more likelihood of data hazards
- Stalls and data forwarding can both address data hazards to generate correct code execution results — but not very efficient
- Compiler optimizations can help, but to a limited extent
- False dependencies limits the freedom of out-of-order execution

# False dependencies

- We are still limited by **false dependencies**
- They are not “true” dependencies because they don’t have an arrow in data dependency graph
  - WAR (Write After Read): a later instruction overwrites the source of an earlier one
  - WAW (Write After Write): a later instruction overwrites the output of an earlier one



# What if we can use more registers...



The diagram illustrates the transformation of assembly code from 14 instructions to 14 instructions, with a large orange arrow pointing from left to right. On the left, the original assembly code is shown with various dependencies indicated by red arrows and circled numbers. On the right, the transformed assembly code is shown with new temporary registers (`%t0` through `%t5`) used to eliminate these dependencies.

①	movq	(%rdi,%rax), %rsi	①	movq	(%rdi,%rax), <b>%t0</b>
②	movq	(%rcx,%rax), %r8	②	movq	(%rcx,%rax), <b>%t1</b>
③	movq	%r8, (%rdi,%rax)	③	movq	<b>%t1</b> , (%rdi,%rax)
④	movq	%rsi, (%rcx,%rax)	④	movq	<b>%t0</b> , (%rcx,%rax)
⑤	addq	\$8, %rax	⑤	addq	\$8, %rax, <b>%t2</b>
⑥	cmpq	%r9, %rax	⑥	cmpq	%r9, <b>%t2</b>
⑦	jne	.L9	⑦	jne	.L9
⑧	movq	(%rdi,%rax), %rsi	⑧	movq	(%rdi, <b>%t2</b> ), <b>%t3</b>
⑨	movq	(%rcx,%rax), %r8	⑨	movq	(%rcx, <b>%t2</b> ), <b>%t4</b>
⑩	movq	%r8, (%rdi,%rax)	⑩	movq	<b>%t4</b> , (%rdi, <b>%t2</b> )
⑪	movq	%rsi, (%rcx,%rax)	⑪	movq	<b>%t3</b> , (%rcx, <b>%t2</b> )
⑫	addq	\$8, %rax	⑫	addq	\$8, <b>%t2</b> , <b>%t5</b>
⑬	cmpq	%r9, %rax	⑬	cmpq	%r9, <b>%t5</b>
⑭	jne	.L9	⑭	jne	.L9

All false dependencies are gone!!!

# **The mechanism of OoO: Register renaming + speculative execution**

- K. C. Yeager, "The MIPS R10000 superscalar microprocessor," in IEEE Micro, vol. 16, no. 2, pp. 28-41, April 1996.

# Register renaming + OoO

- Redirecting the output of an instruction instance to a **physical register**
- Redirecting inputs of an instruction instance from **architectural registers** to correct **physical registers**
  - You need a mapping table between architectural and physical registers
  - You may also need reference counters to reclaim physical registers
- OoO: Executing an instruction all operands are ready (the values of depending physical registers are generated)
  - You will need an **issue logic** to **issue** an instruction to the target functional unit

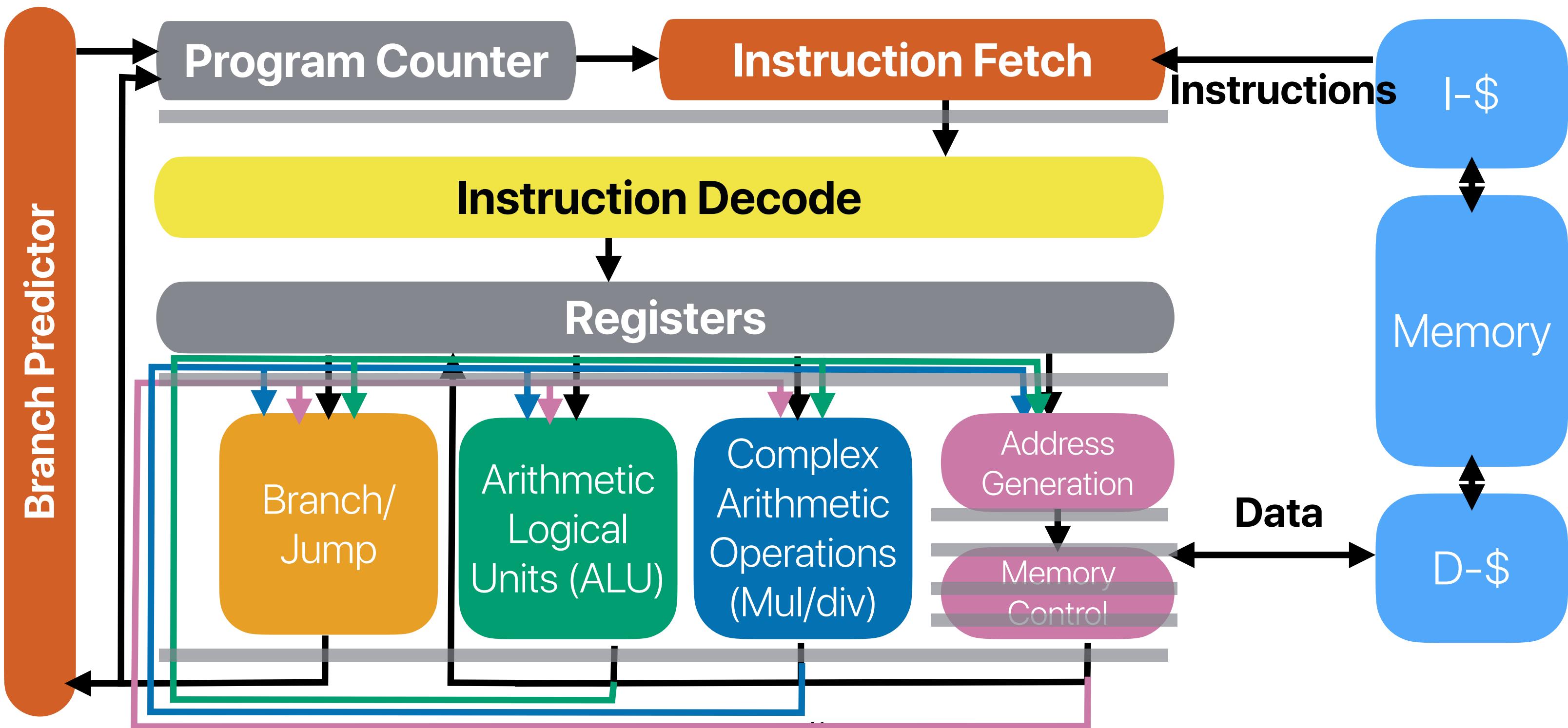
# Can we really execute instructions OoO?

- Exceptions may occur anytime — divided by 0, page fault
  - A later instruction cannot write back its own result otherwise the architectural states won't be correct
  - Instructions after the one causes the exception should not be executed
- Hardware can schedule instruction across branch instructions with the help of branch prediction
  - Fetch instructions according to the branch prediction
  - However, branch predictor can never be perfect

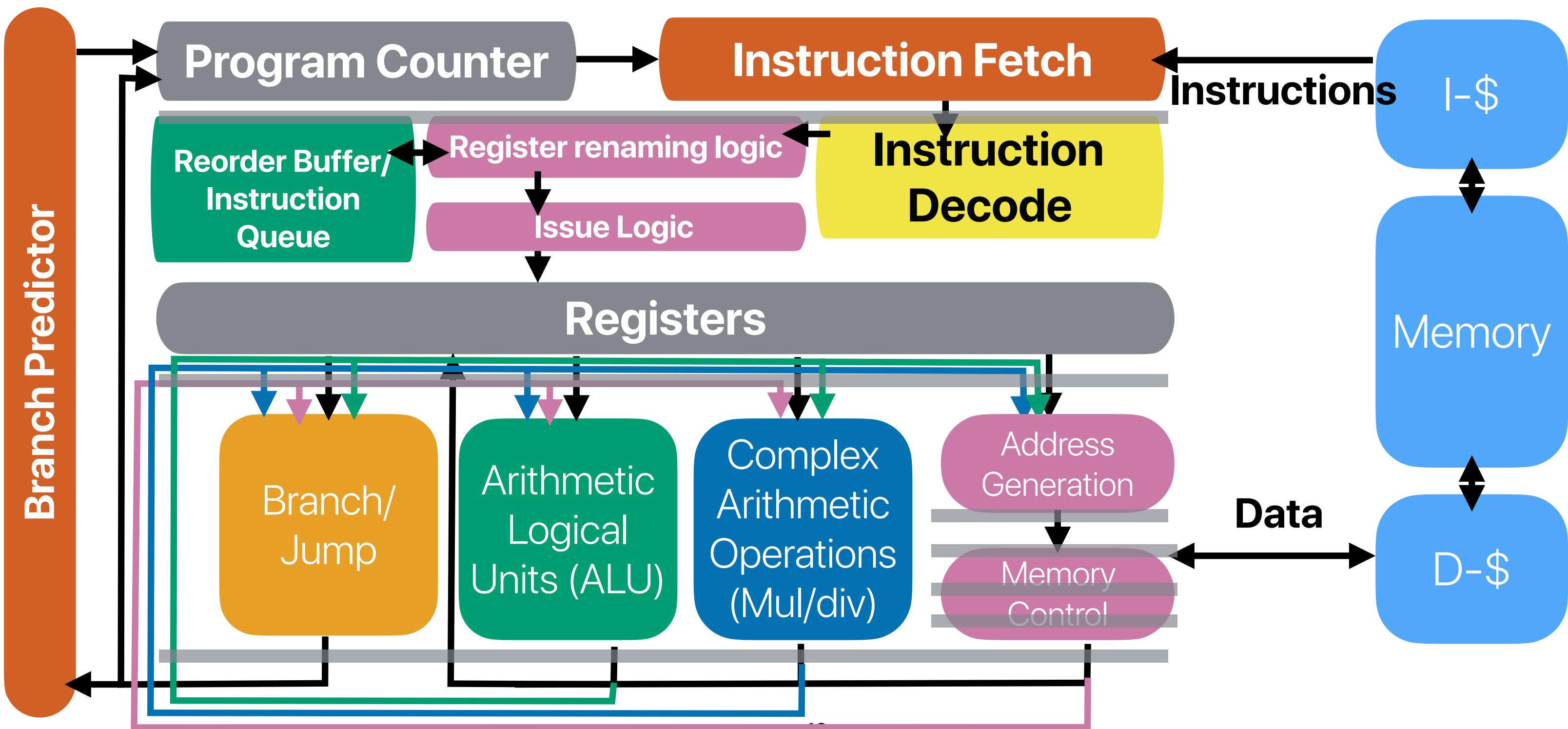
# Speculative Execution

- **Speculative** execution mode: an executing instruction is considered as **speculative** before the processor hasn't determined if the instruction should be executed or not
- Reorder buffer (ROB)
  - The processor allocates an entry for each instruction in a reorder buffer
  - Store results in **reorder buffer and physical registers** when the instruction is still speculative
  - If an earlier instruction failed to commit due to an exception or mis-prediction, the physical registers and all ROB entries after the failed-to-commit instruction are flushed
- Commit/Retire
  - Present the execution result to the running program and in architectural registers when **all prior instructions are non-speculative**
  - Release the ROB entry

# Data “forwarding”



# Register renaming + OoO + RoB



# Register renaming

```
① movq (%rdi,%rax), %rsi
② movq (%rcx,%rax), %r8
③ movq %r8, (%rdi,%rax)
④ movq %rsi, (%rcx,%rax)
⑤ addq $8, %rax
⑥ cmpq %r9, %rax
⑦ jne .L9
⑧ movq (%rdi,%rax), %rsi
⑨ movq (%rcx,%rax), %r8
⑩ movq %r8, (%rdi,%rax)
⑪ movq %rsi, (%rcx,%rax)
⑫ addq $8, %rax
⑬ cmpq %r9, %rax
⑭ jne .L9
```

**Only 1 of them can have a instruction at the same cycle**

Physical Register	
rax	
rcx	
rdi	
rsi	
r8	

	Valid	Value	In use		Valid	Value	In use
P1				P6			
P2				P7			
P3				P8			
P4				P9			
P5				P10			

# Register renaming

Only 1 of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8
- ③ movq %r8, (%rdi,%rax)
- ④ movq %rsi, (%rcx,%rax)
- ⑤ addq \$8, %rax
- ⑥ cmpq %r9, %rax
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi
- ⑨ movq (%rcx,%rax), %r8
- ⑩ movq %r8, (%rdi,%rax)
- ⑪ movq %rsi, (%rcx,%rax)
- ⑫ addq \$8, %rax
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9

	IF	ID	REN	AG	M1	M2	M3	M4	ALU	MUL	BR	ROB
1	(1)											
2	(2)	(1)										
3	(3)	(2)	(1)									
4	(4)	(3)	(2)	(1)								
5												
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												

Physical Register	
rax	
rcx	
rdi	
rsi	P1
r8	

	Valid	Value	In use		Valid	Value	In use
P1	0	1		P6			
P2				P7			
P3				P8			
P4				P9			
P5				P10			

# Register renaming

Only 1 of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8 (P1), (%rdi,%rax)
- ④ movq %rsi, (%rcx,%rax)
- ⑤ addq \$8, %rax
- ⑥ cmpq %r9, %rax
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi
- ⑨ movq (%rcx,%rax), %r8
- ⑩ movq %r8, (%rdi,%rax)
- ⑪ movq %rsi, (%rcx,%rax)
- ⑫ addq \$8, %rax
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9

	IF	ID	REN	AG	M1	M2	M3	M4	ALU	MUL	BR	ROB
1	(1)											
2	(2)	(1)										
3	(3)	(2)	(1)									
4	(4)	(3)	(2)		(1)							
5	(5)	(4)	(3)		(2)	(1)						
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												

Physical Register	
rax	
rcx	
rdi	
rsi	P1
r8	P2

	Valid	Value	In use		Valid	Value	In use
P1	0	1		P6			
P2	0	1		P7			
P3				P8			
P4				P9			
P5				P10			

# Register renaming

Only 1 of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8 (P1), (%rdi,%rax)
- ④ movq %rsi(P2), (%rcx,%rax)
- ⑤ addq \$8, %rax
- ⑥ cmpq %r9, %rax
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi
- ⑨ movq (%rcx,%rax), %r8
- ⑩ movq %r8, (%rdi,%rax)
- ⑪ movq %rsi, (%rcx,%rax)
- ⑫ addq \$8, %rax
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9

	IF	ID	REN	AG	M1	M2	M3	M4	ALU	MUL	BR	ROB
1	(1)											
2	(2)	(1)										
3	(3)	(2)	(1)									
4	(4)	(3)	(2)		(1)							
5	(5)	(4)	(3)		(2)	(1)						
6	(6)	(5)	(3)(4)		(2)	(1)						
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												

Physical Register	
rax	
rcx	
rdi	
rsi	P1
r8	P2

	Valid	Value	In use		Valid	Value	In use
P1	0	1		P6			
P2	0	1		P7			
P3				P8			
P4				P9			
P5				P10			

# Register renaming

Only 1 of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8 (P1), (%rdi,%rax)
- ④ movq %rsi(P2), (%rcx,%rax)
- ⑤ addq \$8, %rax
- ⑥ cmpq %r9, %rax
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi
- ⑨ movq (%rcx,%rax), %r8
- ⑩ movq %r8, (%rdi,%rax)
- ⑪ movq %rsi, (%rcx,%rax)
- ⑫ addq \$8, %rax
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9

	IF	ID	REN	AG	M1	M2	M3	M4	ALU	MUL	BR	ROB
1	(1)											
2	(2)	(1)										
3	(3)	(2)	(1)									
4	(4)	(3)	(2)		(1)							
5	(5)	(4)	(3)		(2)	(1)						
6	(6)	(5)	(3)(4)		(2)	(1)						
7	(7)	(6)	(3)(4)(5)		(2)	(1)						
8												
9												
10												
11												
12												
13												
14												
15												
16												

Physical Register	
rax	
rcx	
rdi	
rsi	P1
r8	P2

	Valid	Value	In use		Valid	Value	In use
P1	0	1		P6			
P2	0	1		P7			
P3				P8			
P4				P9			
P5				P10			

# Register renaming

Only 1 of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8 (P1), (%rdi,%rax)
- ④ movq %rsi(P2), (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi
- ⑨ movq (%rcx,%rax), %r8
- ⑩ movq %r8, (%rdi,%rax)
- ⑪ movq %rsi, (%rcx,%rax)
- ⑫ addq \$8, %rax
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9

	IF	ID	REN	AG	M1	M2	M3	M4	ALU	MUL	BR	ROB
	1	(1)										
1	2	(2)	(1)									
2	3	(3)	(2)	(1)								
3	4	(4)	(3)	(2)	(1)							
4	5	(5)	(4)	(3)	(2)	(1)						
5	6	(6)	(5)	(3)(4)	(2)	(1)						
6	7	(7)	(6)	(3)(4)(5)	(2)	(1)						
7	8											
8	9											
9	10											
10	11											
11	12											
12	13											
13	14											
14	15											
15	16											
16												

Physical Register	
rax	P3
rcx	
rdi	
rsi	P1
r8	P2

	Valid	Value	In use		Valid	Value	In use
P1	0		1	P6			
P2	0		1	P7			
P3	0		1	P8			
P4				P9			
P5				P10			

# Register renaming

Only 1 of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8 (P1), (%rdi,%rax)
- ④ movq %rsi(P2), (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax (P3)
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi
- ⑨ movq (%rcx,%rax), %r8
- ⑩ movq %r8, (%rdi,%rax)
- ⑪ movq %rsi, (%rcx,%rax)
- ⑫ addq \$8, %rax
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9

	IF	ID	REN	AG	M1	M2	M3	M4	ALU	MUL	BR	ROB
	1 (1)											
	2 (2)	(1)										
	3 (3)	(2)	(1)									
	4 (4)	(3)	(2)	(1)								
	5 (5)	(4)	(3)	(2) (1)								
	6 (6)	(5)	(3)(4)	(2) (1)								
	7 (7)	(6)	(3)(4)(5)	(2) (1)								
	8 (8)	(7)	(3)(4)(6)	(2) (1)								
	9											
	10											
	11											
	12											
	13											
	14											
	15											
	16											

Physical Register	
rax	P3
rcx	
rdi	
rsi	P1
r8	P2

	Valid	Value	In use		Valid	Value	In use
P1	0		1		P6		
P2	0		1		P7		
P3	0		1		P8		
P4					P9		
P5					P10		

Instruction (5) is running ahead of (3)

# Register renaming

Only 1 of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8 (P1), (%rdi,%rax)
- ④ movq %rsi(P2), (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax (P3)
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi
- ⑨ movq (%rcx,%rax), %r8
- ⑩ movq %r8, (%rdi,%rax)
- ⑪ movq %rsi, (%rcx,%rax)
- ⑫ addq \$8, %rax
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9

	IF	ID	REN	AG	M1	M2	M3	M4	ALU	MUL	BR	ROB
	1	(1)		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	
1	1	(1)		(1)								
2	2	(2)	(1)		(2)							
3	3	(3)	(2)			(1)						
4	4	(4)	(3)				(2)					
5	5	(5)	(4)					(3)				
6	6	(6)	(5)						(1)			
7	7	(7)	(6)							(2)		
8	8	(8)	(7)								(1)	
9	9	(9)	(8)									(1)(5)
10	10											
11	11											
12	12											
13	13											
14	14											
15	15											
16	16											

Physical Register	
rax	P3
rcx	
rdi	
rsi	P1
r8	P2

	Valid	Value	In use		Valid	Value	In use
P1	1		1		P6		
P2	0		1		P7		
P3	1		1		P8		
P4					P9		
P5					P10		

Instruction (4) is running ahead of (3)      Instruction (5) is running ahead of (3)

# Register renaming

Only 1 of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8 (P1), (%rdi,%rax)
- ④ movq %rsi(P2), (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax (P3)
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi
- ⑨ movq (%rcx,%rax), %r8
- ⑩ movq %r8, (%rdi,%rax)
- ⑪ movq %rsi, (%rcx,%rax)
- ⑫ addq \$8, %rax
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9

	IF	ID	REN	AG	M1	M2	M3	M4	ALU	MUL	BR	ROB
	1	(1)										
1	2	(2)	(1)									
2	3	(3)	(2)	(1)								
3	4	(4)	(3)	(2)								
4	5	(5)	(4)	(3)								
5	6	(6)	(5)	(3)(4)								
6	7	(7)	(6)	(3)(4)(5)								
7	8	(8)	(7)	(3)(4)(6)								
8	9	(9)	(8)	(3)(6)(7)								
9	10											
10	11											
11	12											
12	13											
13	14											
14	15											
15	16											
16												

Physical Register	
rax	P3
rcx	
rdi	
rsi	P1
r8	P2

	Valid	Value	In use		Valid	Value	In use
P1	1		1		P6		
P2	0		1		P7		
P3	1		1		P8		
P4					P9		
P5					P10		

# Register renaming

Only 1 of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8 (P1), (%rdi,%rax)
- ④ movq %rsi(P2), (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax (P3)
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi → P4
- ⑨ movq (%rcx,%rax), %r8
- ⑩ movq %r8, (%rdi,%rax)
- ⑪ movq %rsi, (%rcx,%rax)
- ⑫ addq \$8, %rax
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9

	IF	ID	REN	AG	M1	M2	M3	M4	ALU	MUL	BR	ROB
	1	(1)										
1	2	(2)	(1)									
2	3	(3)	(2)	(1)								
3	4	(4)	(3)	(2)								
4	5	(5)	(4)	(3)								
5	6	(6)	(5)	(3)(4)								
6	7	(7)	(6)	(3)(4)(5)								
7	8	(8)	(7)	(3)(4)(6)								
8	9	(9)	(8)	(3)(6)(7)								
9	10	(10)	(9)	(6)(7)(8)								
10	11											
11	12											
12	13											
13	14											
14	15											
15	16											
16												

Physical Register	
rax	P3
rcx	
rdi	
rsi	P4
r8	P2

	Valid	Value	In use		Valid	Value	In use
P1	1		1	P6			
P2	1		1	P7			
P3	1		1	P8			
P4	0		1	P9			
P5				P10			

# Register renaming

Only 1 of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8 (P1), (%rdi,%rax)
- ④ movq %rsi(P2), (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax (P3)
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi → P4
- ⑨ movq (%rcx,%rax), %r8 → P5
- ⑩ movq %r8, (%rdi,%rax)
- ⑪ movq %rsi, (%rcx,%rax)
- ⑫ addq \$8, %rax
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9

	IF	ID	REN	AG	M1	M2	M3	M4	ALU	MUL	BR	ROB
	1	(1)										
1	2	(2)	(1)									
2	3	(3)	(2)									
3	4	(4)	(3)						(1)			
4	5	(5)	(4)						(2)	(1)		
5	6	(6)	(5)						(2)	(1)		
6	7	(7)	(6)						(2)	(1)		
7	8	(8)	(7)						(2)	(1)		
8	9	(9)	(8)						(2)	(1)		
9	10	(10)	(9)						(4)			(1)(5)
10	11	(11)	(10)						(3)	(4)		(2)(5)
11	12								(3)	(4)		
12	13											
13	14											
14	15											
15	16											
16												

Physical Register	
rax	P3
rcx	
rdi	
rsi	P4
r8	P5

	Valid	Value	In use		Valid	Value	In use
P1	1		1		P6		
P2	1		1		P7		
P3	1		1		P8		
P4	0		1		P9		
P5	0		1		P10		

# Register renaming

Only 1 of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8 (P1), (%rdi,%rax)
- ④ movq %rsi(P2), (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax (P3)
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi → P4
- ⑨ movq (%rcx,%rax), %r8 → P5
- ⑩ movq %r8 (P4), (%rdi,%rax)
- ⑪ movq %rsi, (%rcx,%rax)
- ⑫ addq \$8, %rax
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9

	IF	ID	REN	AG	M1	M2	M3	M4	ALU	MUL	BR	ROB
	1	(1)										
1	2	(2)	(1)									
2	3	(3)	(2)	(1)								
3	4	(4)	(3)	(2)	(1)							
4	5	(5)	(4)	(3)	(2)	(1)						
5	6	(6)	(5)	(3)(4)	(2)	(1)						
6	7	(7)	(6)	(3)(4)(5)	(2)	(1)						
7	8	(8)	(7)	(3)(4)(6)	(2)	(1)	(5)					
8	9	(9)	(8)	(3)(6)(7)	(4)	(2)						(1)(5)
9	10	(10)	(9)	(6)(7)(8)	(3)	(4)						(2)(5)
10	11	(11)	(10)	(7)(8)(9)	(3)	(4)	(6)					
11	12	(12)	(11)	(8)(9)(10)	(3)	(4)					(7)	(5)(6)
12	13											
13	14											
14	15											
15	16											

Physical Register	
rax	P3
rcx	
rdi	
rsi	P4
r8	P5

	Valid	Value	In use		Valid	Value	In use
P1	1		1	P6			
P2	1		1	P7			
P3	1		1	P8			
P4	0		1	P9			
P5	0		1	P10			

# Register renaming

Only 1 of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8 (P1), (%rdi,%rax)
- ④ movq %rsi(P2), (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax (P3)
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi → P4
- ⑨ movq (%rcx,%rax), %r8 → P5
- ⑩ movq %r8 (P4), (%rdi,%rax)
- ⑪ movq %rsi(P11), (%rcx,%rax)
- ⑫ addq \$8, %rax
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9

	IF	ID	REN	AG	M1	M2	M3	M4	ALU	MUL	BR	ROB
1	(1)											
2	(2)	(1)										
3	(3)	(2)	(1)									
4	(4)	(3)	(2)	(1)								
5	(5)	(4)	(3)	(2)	(1)							
6	(6)	(5)	(3)(4)	(2)	(1)							
7	(7)	(6)	(3)(4)(5)	(2)	(1)							
8	(8)	(7)	(3)(4)(6)	(2)	(1)	(5)						
9	(9)	(8)	(3)(6)(7)	(4)								(1)(5)
10	(10)	(9)	(6)(7)(8)	(3)	(4)							(2)(5)
11	(11)	(10)	(7)(8)(9)	(3)	(4)	(6)						
12	(12)	(11)	(8)(9)(10)	(3)	(4)						(7)	(5)(6)
13	(13)	(12)	(9)(10)(11)	(8)	(3)	(4)						(5)(6)(7)
14												
15												
16												

Physical Register	
rax	P3
rcx	
rdi	
rsi	P4
r8	P5

	Valid	Value	In use		Valid	Value	In use
P1	1		1	P6			
P2	1		1	P7			
P3	1		1	P8			
P4	0		1	P9			
P5	0		1	P10			

# Register renaming

Only 1 of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8 (P1), (%rdi,%rax)
- ④ movq %rsi(P2), (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax (P3)
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi → P4
- ⑨ movq (%rcx,%rax), %r8 → P5
- ⑩ movq %r8 (P4), (%rdi,%rax)
- ⑪ movq %rsi(P11), (%rcx,%rax)
- ⑫ addq \$8, %rax → P6
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9

	IF	ID	REN	AG	M1	M2	M3	M4	ALU	MUL	BR	ROB
1	(1)											
2	(2)	(1)										
3	(3)	(2)	(1)									
4	(4)	(3)	(2)	(1)								
5	(5)	(4)	(3)	(2)	(1)							
6	(6)	(5)	(3)(4)	(2)	(1)							
7	(7)	(6)	(3)(4)(5)	(2)	(1)							
8	(8)	(7)	(3)(4)(6)	(2)	(1)	(5)						
9	(9)	(8)	(3)(6)(7)	(4)								(1)(5)
10	(10)	(9)	(6)(7)(8)	(3)	(4)							(2)(5)
11	(11)	(10)	(7)(8)(9)	(3)	(4)	(6)						
12	(12)	(11)	(8)(9)(10)	(3)	(4)						(7)	(5)(6)
13	(13)	(12)	(9)(10)(11)	(8)	(3)	(4)						(5)(6)(7)
14	(14)	(13)	(10)(11)(12)	(9)	(8)	(3)						(4)(5)(6)(7)
15												
16												

Physical Register	
rax	P6
rcx	
rdi	
rsi	P4
r8	P5

	Valid	Value	In use		Valid	Value	In use
P1	1		1	P6	0		1
P2	1		1	P7			
P3	1		1	P8			
P4	0		1	P9			
P5	0		1	P10			

# Register renaming

Only 1 of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8 (P1), (%rdi,%rax)
- ④ movq %rsi(P2), (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax (P3)
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi → P4
- ⑨ movq (%rcx,%rax), %r8 → P5
- ⑩ movq %r8 (P4), (%rdi,%rax)
- ⑪ movq %rsi(P11), (%rcx,%rax)
- ⑫ addq \$8, %rax → P6
- ⑬ cmpq %r9, %rax (P6)
- ⑭ jne .L9

	IF	ID	REN	AG	M1	M2	M3	M4	ALU	MUL	BR	ROB
1	(1)											
2	(2)	(1)										
3	(3)	(2)	(1)									
4	(4)	(3)	(2)	(1)								
5	(5)	(4)	(3)	(2)	(1)							
6	(6)	(5)	(3)(4)	(2)	(1)							
7	(7)	(6)	(3)(4)(5)	(2)	(1)							
8	(8)	(7)	(3)(4)(6)	(2)	(1)	(5)						
9	(9)	(8)	(3)(6)(7)	(4)		(2)						(1)(5)
10	(10)	(9)	(6)(7)(8)	(3)	(4)							(2)(5)
11	(11)	(10)	(7)(8)(9)	(3)	(4)	(6)						
12	(12)	(11)	(8)(9)(10)	(3)	(4)		(7)					(5)(6)
13	(13)	(12)	(9)(10)(11)	(8)	(3)	(4)						(5)(6)(7)
14	(14)	(13)	(10)(11)(12)	(9)	(8)	(3)						(4)(5)(6)(7)
15	(15)	(14)	(10)(11)(13)	(9)	(8)		(12)					(3)(4)(5)(6)(7)
16												

Physical Register	
rax	P6
rcx	
rdi	
rsi	P4
r8	P5

	Valid	Value	In use		Valid	Value	In use
P1	1		1	P6	0		1
P2	1		1	P7			
P3	1		1	P8			
P4	0		1	P9			
P5	0		1	P10			

# Register renaming

Only 1 of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8 (P1), (%rdi,%rax)
- ④ movq %rsi(P2), (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax (P3)
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi → P4
- ⑨ movq (%rcx,%rax), %r8 → P5
- ⑩ movq %r8 (P4), (%rdi,%rax)
- ⑪ movq %rsi(P11), (%rcx,%rax)
- ⑫ addq \$8, %rax → P6
- ⑬ cmpq %r9, %rax (P6)
- ⑭ jne .L9

	IF	ID	REN	AG	M1	M2	M3	M4	ALU	MUL	BR	ROB
1	(1)											
2	(2)	(1)										
3	(3)	(2)	(1)									
4	(4)	(3)	(2)	(1)								
5	(5)	(4)	(3)	(2)	(1)							
6	(6)	(5)	(3)(4)	(2)	(1)							
7	(7)	(6)	(3)(4)(5)	(2)	(1)							
8	(8)	(7)	(3)(4)(6)	(2)	(1)	(5)						
9	(9)	(8)	(3)(6)(7)	(4)		(2)						(1)(5)
10	(10)	(9)	(6)(7)(8)	(3)	(4)							(2)(5)
11	(11)	(10)	(7)(8)(9)	(3)	(4)	(6)						
12	(12)	(11)	(8)(9)(10)	(3)	(4)						(7)	(5)(6)
13	(13)	(12)	(9)(10)(11)	(8)	(3)	(4)						(5)(6)(7)
14	(14)	(13)	(10)(11)(12)	(9)	(8)	(3)						(4)(5)(6)(7)
15	(15)	(14)	(10)(11)(13)	(9)	(8)		(12)					(3)(4)(5)(6)(7)
16	(16)	(15)	(10)(11)(14)	(9)	(8)		(13)					(12)

Physical Register	
rax	P6
rcx	
rdi	
rsi	P4
r8	P5

	Valid	Value	In use		Valid	Value	In use
P1	1		1	P6	1		1
P2	1		1	P7			
P3	1		1	P8			
P4	0		1	P9			
P5	0		1	P10			

# Register renaming

Only 1 of them can have

**Only 1 of them can have a instruction at the same cycle**

- ```
① movq (%rdi,%rax), %rsi → P1
② movq (%rcx,%rax), %r8 → P2
③ movq %r8 (P1), (%rdi,%rax)
④ movq %rsi(P2), (%rcx,%rax)
⑤ addq $8, %rax → P3
⑥ cmpq %r9, %rax (P3)
⑦ jne .L9
⑧ movq (%rdi,%rax), %rsi → P4
⑨ movq (%rcx,%rax), %r8 → P5
⑩ movq %r8 (P4), (%rdi,%rax)
⑪ movq %rsi(P11), (%rcx,%rax)
⑫ addq $8, %rax → P6
⑬ cmpq %r9, %rax (P6)
⑭ jne .L9
⑮ movq (%rdi,%rax), %rsi
⑯ movq (%rcx,%rax), %r8
⑰ movq %r8, (%rdi,%rax)
⑱ movq %rsi, (%rcx,%rax)
⑲ addq $8, %rax
⑳ cmpq %r9, %rax
㉑ jne .L9
```

# Register renaming

Only 1 of them can have

**Only 1 of them can have a instruction at the same cycle**

- ```
① movq (%rdi,%rax), %rsi → P1
② movq (%rcx,%rax), %r8 → P2
③ movq %r8 (P1), (%rdi,%rax)
④ movq %rsi(P2), (%rcx,%rax)
⑤ addq $8, %rax → P3
⑥ cmpq %r9, %rax (P3)
⑦ jne .L9
⑧ movq (%rdi,%rax), %rsi → P4
⑨ movq (%rcx,%rax), %r8 → P5
⑩ movq %r8 (P4), (%rdi,%rax)
⑪ movq %rsi(P11), (%rcx,%rax)
⑫ addq $8, %rax → P6
⑬ cmpq %r9, %rax (P6)
⑭ jne .L9
⑮ movq (%rdi,%rax), %rsi
⑯ movq (%rcx,%rax), %r8
⑰ movq %r8, (%rdi,%rax)
⑱ movq %rsi, (%rcx,%rax)
⑲ addq $8, %rax
⑳ cmpq %r9, %rax
㉑ jne .L9
```

# Register renaming

Only 1 of them can have

**Only 1 of them can have a instruction at the same cycle**

- ```
① movq (%rdi,%rax), %rsi → P1
② movq (%rcx,%rax), %r8 → P2
③ movq %r8 (P1), (%rdi,%rax)
④ movq %rsi(P2), (%rcx,%rax)
⑤ addq $8, %rax → P3
⑥ cmpq %r9, %rax (P3)
⑦ jne .L9
⑧ movq (%rdi,%rax), %rsi → P4
⑨ movq (%rcx,%rax), %r8 → P5
⑩ movq %r8 (P4), (%rdi,%rax)
⑪ movq %rsi(P11), (%rcx,%rax)
⑫ addq $8, %rax → P6
⑬ cmpq %r9, %rax (P6)
⑭ jne .L9
⑮ movq (%rdi,%rax), %rsi
⑯ movq (%rcx,%rax), %r8
⑰ movq %r8, (%rdi,%rax)
⑱ movq %rsi, (%rcx,%rax)
⑲ addq $8, %rax
⑳ cmpq %r9, %rax
㉑ jne .L9
```

# Register renaming

Only 1 of them can have

**Only 1 of them can have a instruction at the same cycle**

- ```
① movq (%rdi,%rax), %rsi → P1
② movq (%rcx,%rax), %r8 → P2
③ movq %r8 (P1), (%rdi,%rax)
④ movq %rsi(P2), (%rcx,%rax)
⑤ addq $8, %rax → P3
⑥ cmpq %r9, %rax (P3)
⑦ jne .L9
⑧ movq (%rdi,%rax), %rsi → P4
⑨ movq (%rcx,%rax), %r8 → P5
⑩ movq %r8 (P4), (%rdi,%rax)
⑪ movq %rsi(P11), (%rcx,%rax)
⑫ addq $8, %rax → P6
⑬ cmpq %r9, %rax (P6)
⑭ jne .L9
⑮ movq (%rdi,%rax), %rsi
⑯ movq (%rcx,%rax), %r8
⑰ movq %r8, (%rdi,%rax)
⑱ movq %rsi, (%rcx,%rax)
⑲ addq $8, %rax
⑳ cmpq %r9, %rax
㉑ jne .L9
```

# Register renaming

Only 1 of them can have

**Only 1 of them can have a instruction at the same cycle**

- ```
① movq (%rdi,%rax), %rsi → P1
② movq (%rcx,%rax), %r8 → P2
③ movq %r8 (P1), (%rdi,%rax)
④ movq %rsi(P2), (%rcx,%rax)
⑤ addq $8, %rax → P3
⑥ cmpq %r9, %rax (P3)
⑦ jne .L9
⑧ movq (%rdi,%rax), %rsi → P4
⑨ movq (%rcx,%rax), %r8 → P5
⑩ movq %r8 (P4), (%rdi,%rax)
⑪ movq %rsi(P11), (%rcx,%rax)
⑫ addq $8, %rax → P6
⑬ cmpq %r9, %rax (P6)
⑭ jne .L9
⑮ movq (%rdi,%rax), %rsi
⑯ movq (%rcx,%rax), %r8
⑰ movq %r8, (%rdi,%rax)
⑱ movq %rsi, (%rcx,%rax)
⑲ addq $8, %rax
⑳ cmpq %r9, %rax
㉑ jne .L9
```

# Register renaming

Only 1 of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8 (P1), (%rdi,%rax)
- ④ movq %rsi(P2), (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax (P3)
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi → P4
- ⑨ movq (%rcx,%rax), %r8 → P5
- ⑩ movq %r8 (P4), (%rdi,%rax)
- ⑪ movq %rsi(P11), (%rcx,%rax)
- ⑫ addq \$8, %rax → P6
- ⑬ cmpq %r9, %rax (P6)
- ⑭ jne .L9
- ⑮ movq (%rdi,%rax), %rsi
- ⑯ movq (%rcx,%rax), %r8
- ⑰ movq %r8, (%rdi,%rax)
- ⑱ movq %rsi, (%rcx,%rax)
- ⑲ addq \$8, %rax
- ⑳ cmpq %r9, %rax
- ㉑ jne .L9

|    | IF   | ID           | REN              | AG   | M1   | M2   | M3   | M4   | ALU | MUL | BR  | ROB                  |
|----|------|--------------|------------------|------|------|------|------|------|-----|-----|-----|----------------------|
| 1  | (1)  |              |                  |      |      |      |      |      |     |     |     |                      |
| 2  | (2)  | (1)          |                  |      |      |      |      |      |     |     |     |                      |
| 3  | (3)  | (2)          | (1)              |      |      |      |      |      |     |     |     |                      |
| 4  | (4)  | (3)          | (2)              | (1)  |      |      |      |      |     |     |     |                      |
| 5  | (5)  | (4)          | (3)              | (2)  | (1)  |      |      |      |     |     |     |                      |
| 6  | (6)  | (5)          | (3)(4)           | (2)  | (1)  |      |      |      |     |     |     |                      |
| 7  | (7)  | (6)          | (3)(4)(5)        | (2)  | (1)  |      |      |      |     |     |     |                      |
| 8  | (8)  | (7)          | (3)(4)(6)        | (2)  | (1)  | (5)  |      |      |     |     |     |                      |
| 9  | (9)  | (8)          | (3)(6)(7)        | (4)  |      |      |      |      |     |     |     | (1)(5)               |
| 10 | (10) | (9)          | (6)(7)(8)        | (3)  | (4)  |      |      |      |     |     |     | (2)(5)               |
| 11 | (11) | (10)         | (7)(8)(9)        | (3)  | (4)  | (6)  |      |      |     |     |     |                      |
| 12 | (12) | (11)         | (8)(9)(10)       | (3)  | (4)  |      |      |      |     |     | (7) | (5)(6)               |
| 13 | (13) | (12)         | (9)(10)(11)      | (8)  | (3)  | (4)  |      |      |     |     |     | (5)(6)(7)            |
| 14 | (14) | (13)         | (10)(11)(12)     | (9)  | (8)  | (3)  |      |      |     |     |     | (4)(5)(6)(7)         |
| 15 | (15) | (14)         | (10)(11)(13)     | (9)  | (8)  |      | (12) |      |     |     |     | (3)(4)(5)(6)(7)      |
| 16 | (16) | (15)         | (10)(11)(14)     | (9)  | (8)  |      | (13) |      |     |     |     | (12)                 |
| 17 | (17) | (16)         | (10)(11)(15)     | (9)  | (8)  |      |      |      |     |     |     | (12)(13)             |
| 18 | (18) | (17)         | (10)(15)(16)     | (11) |      | (9)  | (8)  |      |     |     |     | (14) (8)(12)(13)(14) |
| 19 | (19) | (18)         | (15)(16)(17)     | (10) | (11) |      |      |      |     |     |     | (9)(12)(13)(14)      |
| 20 | (20) | (19)         | (16)(17)(18)     | (15) | (10) | (11) |      |      |     |     |     | (12)(13)(14)         |
| 21 | (21) | (20)         | (17)(18)(19)     | (16) | (15) | (10) | (11) |      |     |     |     | (12)(13)(14)         |
| 22 | (21) | (17)(18)(20) | (16)(15)(10)(11) | (16) | (15) | (10) | (11) | (19) |     |     |     | (12)(13)(14)         |

**Only 1 of them can have a instruction at the same cycle**

**Registration**  **IF ID REN AG M1 M2 M3 M4**

**Only 1 of them can have a instruction at the same cycle**

# Registration

Only 1 of them can have a instruction at the same cycle

# Register renaming

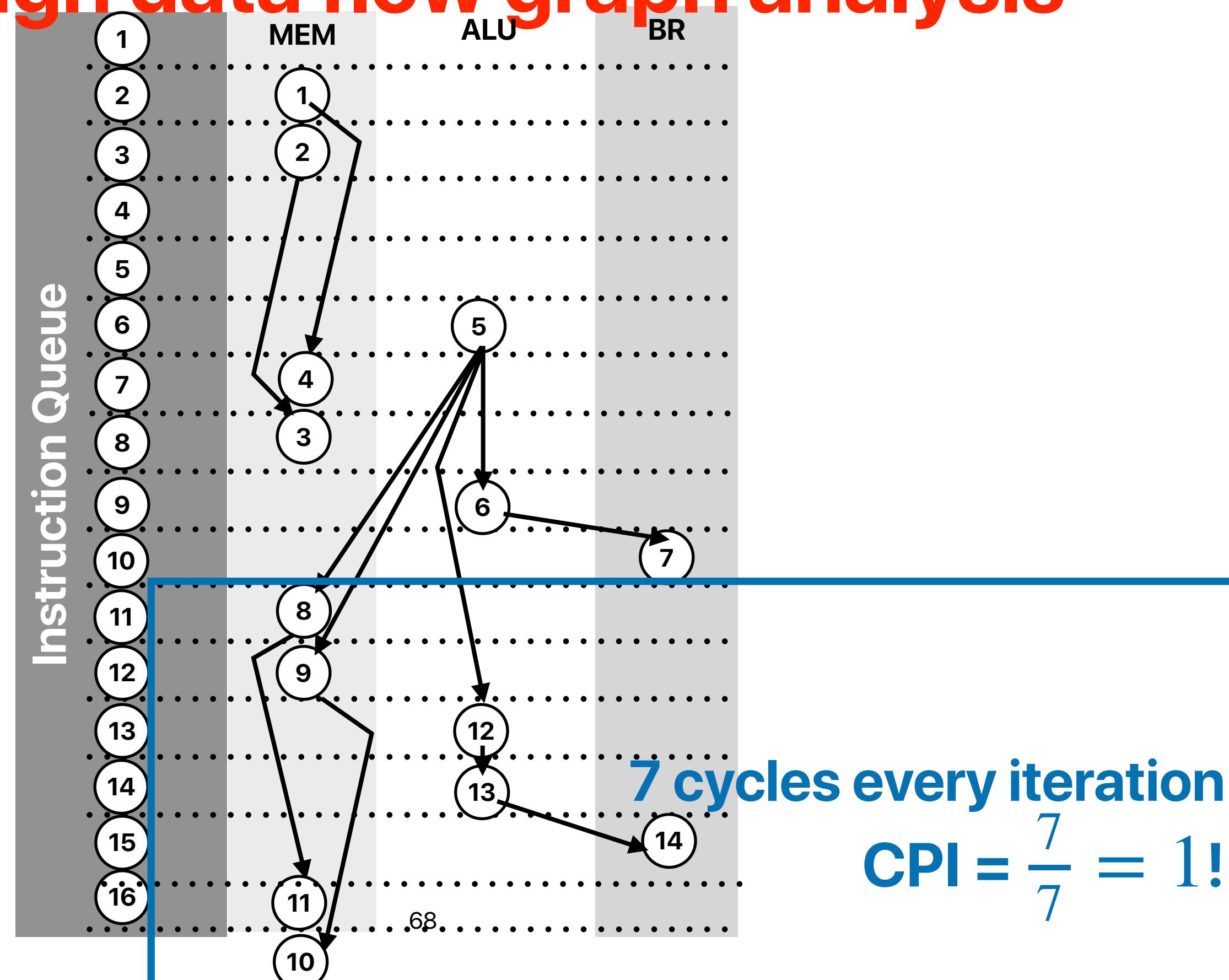
- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8 (P1), (%rdi,%rax)
- ④ movq %rsi(P2), (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax (P3)
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi → P4
- ⑨ movq (%rcx,%rax), %r8 → P5
- ⑩ movq %r8 (P4), (%rdi,%rax)
- ⑪ movq %rsi(P11), (%rcx,%rax)
- ⑫ addq \$8, %rax → P6
- ⑬ cmpq %r9, %rax (P6)
- ⑭ jne .L9
- ⑮ movq (%rdi,%rax), %rsi
- ⑯ movq (%rcx,%rax), %r8
- ⑰ movq %r8, (%rdi,%rax)
- ⑱ movq %rsi, (%rcx,%rax)
- ⑲ addq \$8, %rax
- ⑳ cmpq %r9, %rax
- ㉑ jne .L9

|    | IF   | ID   | REN          | AG   | M1   | M2   | M3   | M4   | ALU | MUL | BR | ROB                      |
|----|------|------|--------------|------|------|------|------|------|-----|-----|----|--------------------------|
| 1  | (1)  |      |              |      |      |      |      |      |     |     |    |                          |
| 2  | (2)  | (1)  |              |      |      |      |      |      |     |     |    |                          |
| 3  | (3)  | (2)  | (1)          |      |      |      |      |      |     |     |    |                          |
| 4  | (4)  | (3)  | (2)          | (1)  |      |      |      |      |     |     |    |                          |
| 5  | (5)  | (4)  | (3)          | (2)  | (1)  |      |      |      |     |     |    |                          |
| 6  | (6)  | (5)  | (3)(4)       | (2)  | (1)  |      |      |      |     |     |    |                          |
| 7  | (7)  | (6)  | (3)(4)(5)    | (2)  | (1)  |      |      |      |     |     |    |                          |
| 8  | (8)  | (7)  | (3)(4)(6)    | (2)  | (1)  | (5)  |      |      |     |     |    |                          |
| 9  | (9)  | (8)  | (3)(6)(7)    | (4)  | (2)  |      |      |      |     |     |    | (1)(5)                   |
| 10 | (10) | (9)  | (6)(7)(8)    | (3)  | (4)  |      |      |      |     |     |    | (2)(5)                   |
| 11 | (11) | (10) | (7)(8)(9)    | (3)  | (4)  | (6)  |      |      |     |     |    |                          |
| 12 | (12) | (11) | (8)(9)(10)   | (3)  | (4)  |      | (7)  |      |     |     |    | (5)(6)                   |
| 13 | (13) | (12) | (9)(10)(11)  | (8)  | (3)  | (4)  |      |      |     |     |    | (5)(6)(7)                |
| 14 | (14) | (13) | (10)(11)(12) | (9)  | (8)  | (3)  |      |      |     |     |    | (4)(5)(6)(7)             |
| 15 | (15) | (14) | (10)(11)(13) | (9)  | (8)  |      | (12) |      |     |     |    | (3)(4)(5)(6)(7)          |
| 16 | (16) | (15) | (10)(11)(14) | (9)  | (8)  | (13) |      |      |     |     |    | (12)                     |
| 17 | (17) | (16) | (10)(11)(15) |      | (9)  | (8)  |      | (14) |     |     |    | (12)(13)                 |
| 18 | (18) | (17) | (10)(15)(16) | (11) |      | (9)  |      |      |     |     |    | (8)(12)(13)(14)          |
| 19 | (19) | (18) | (15)(16)(17) | (10) | (11) |      |      |      |     |     |    | (9)(12)(13)(14)          |
| 20 | (20) | (19) | (16)(17)(18) | (15) | (10) | (11) |      |      |     |     |    | (12)(13)(14)             |
| 21 | (21) | (20) | (17)(18)(19) | (16) | (15) | (11) |      |      |     |     |    | (12)(13)(14)             |
| 22 |      | (21) | (17)(18)(20) | (16) | (15) | (10) | (11) | (19) |     |     |    | (12)(13)(14)             |
| 23 |      |      | (17)(20)(21) | (18) | (16) | (15) | (10) |      |     |     |    | (11)(12)(13)(14)(19)     |
| 24 |      |      | (20)(21)     | (17) | (18) | (16) | (15) |      |     |     |    | (10)(11)(12)(13)(14)(19) |
| 25 |      |      | (21)         | (17) | (18) | (16) | (20) |      |     |     |    | (15)(19)                 |

7 cycles for 7 instructions  
CPI = 1

# Through data flow graph analysis

```
① movq (%rdi,%rax), %rsi
② movq (%rcx,%rax), %r8
③ movq %r8, (%rdi,%rax)
④ movq %rsi, (%rcx,%rax)
⑤ addq $8, %rax
⑥ cmpq %r9, %rax
⑦ jne .L9
⑧ movq (%rdi,%rax), %rsi
⑨ movq (%rcx,%rax), %r8
⑩ movq %r8, (%rdi,%rax)
⑪ movq %rsi, (%rcx,%rax)
⑫ addq $8, %rax
⑬ cmpq %r9, %rax
⑭ jne .L9
⑮ movq (%rdi,%rax), %rsi
⑯ movq (%rcx,%rax), %r8
⑰ movq %r8, (%rdi,%rax)
⑱ movq %rsi, (%rcx,%rax)
⑲ addq $8, %rax
⑳ cmpq %r9, %rax
㉑ jne .L9
```



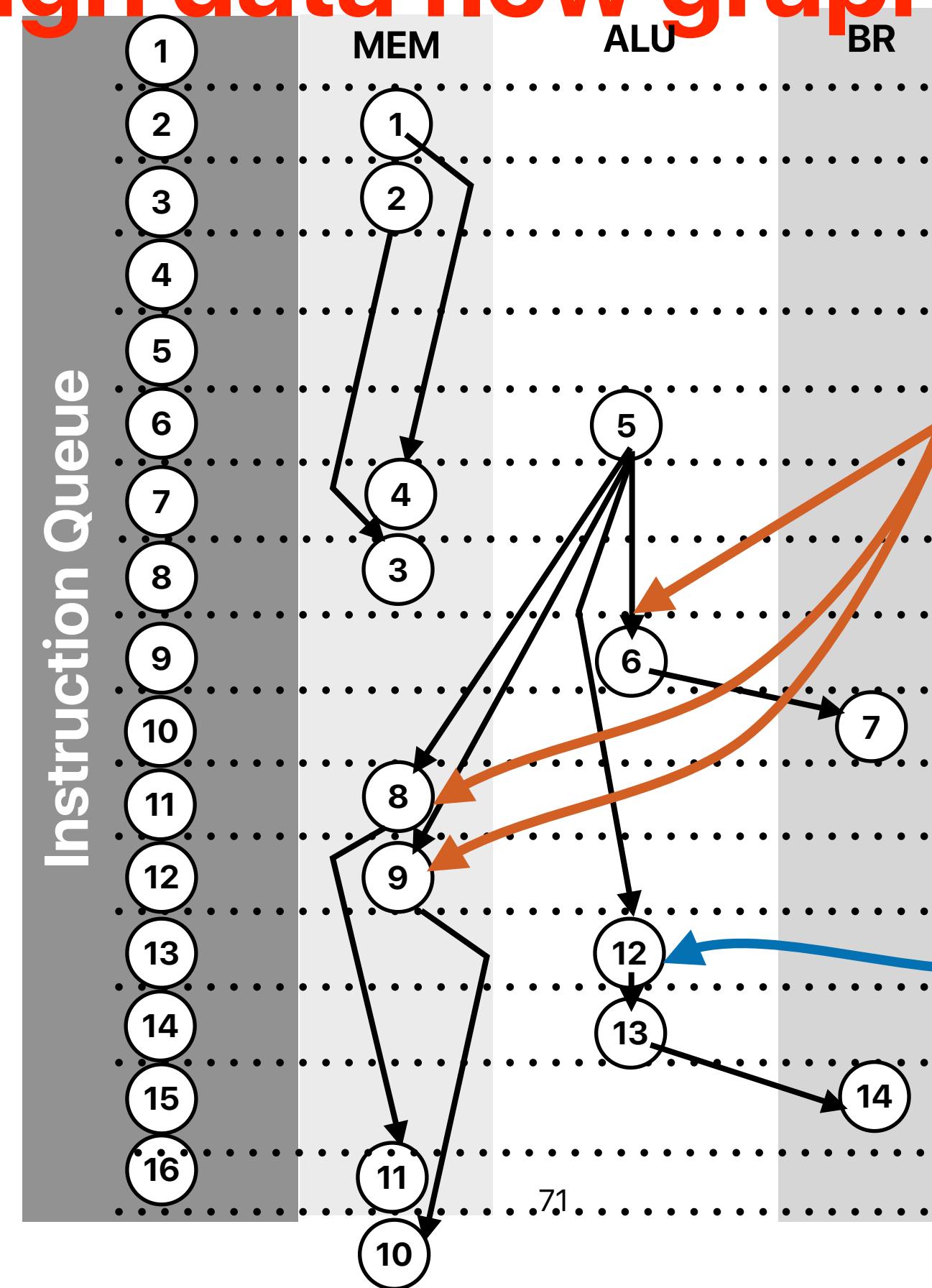
# Takeaways: data hazards

- More data dependencies, more likelihood of data hazards
- Stalls and data forwarding can both address data hazards to generate correct code execution results — but not very efficient
- Compiler optimizations can help, but to a limited extent
- False dependencies limits the freedom of out-of-order execution
- Register renaming + Speculative execution enables more efficient execution by dynamically scheduling instructions whenever their data dependencies are resolved

**If  $CPI == 1$  the limitation?**

# Through data flow graph analysis

```
① movq (%rdi,%rax), %rsi
② movq (%rcx,%rax), %r8
③ movq %r8, (%rdi,%rax)
④ movq %rsi, (%rcx,%rax)
⑤ addq $8, %rax
⑥ cmpq %r9, %rax
⑦ jne .L9
⑧ movq (%rdi,%rax), %rsi
⑨ movq (%rcx,%rax), %r8
⑩ movq %r8, (%rdi,%rax)
⑪ movq %rsi, (%rcx,%rax)
⑫ addq $8, %rax
⑬ cmpq %r9, %rax
⑭ jne .L9
⑮ movq (%rdi,%rax), %rsi
⑯ movq (%rcx,%rax), %r8
⑰ movq %r8, (%rdi,%rax)
⑱ movq %rsi, (%rcx,%rax)
⑲ addq $8, %rax
⑳ cmpq %r9, %rax
㉑ jne .L9
```



We cannot issue them earlier simply because structural hazards!

We could have this executed earlier if it's in the queue earlier

# **Super Scalar**

# Superscalar

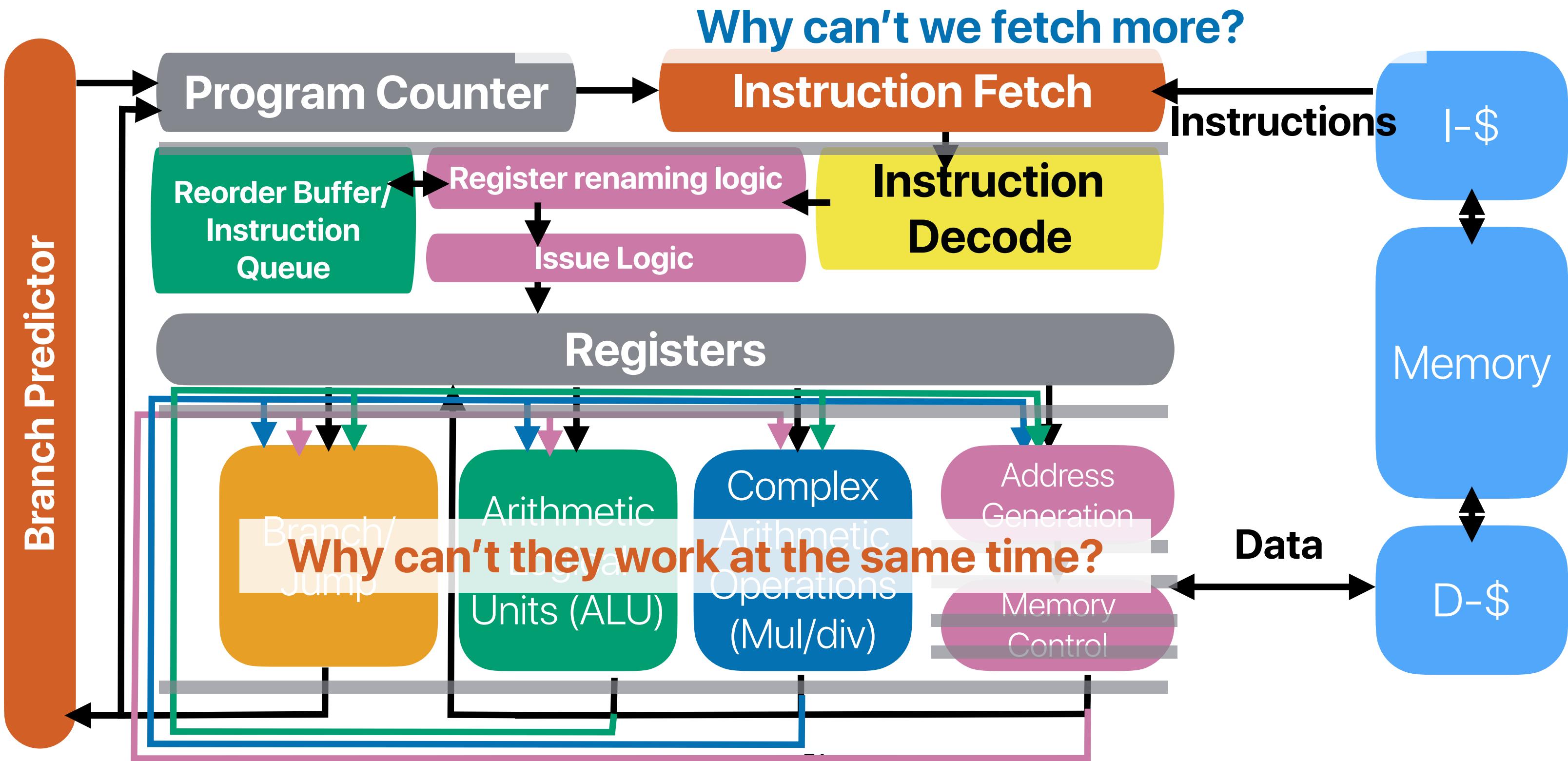
- Since we have many functional units now, we should fetch/decode more instructions each cycle so that we can have more instructions to issue!
- Super-scalar: fetch/decode/issue more than one instruction each cycle
  - **Fetch width:** how many instructions can the processor fetch/decode each cycle
  - **Issue width:** how many instructions can the processor issue each cycle
- The theoretical CPI should now be

1

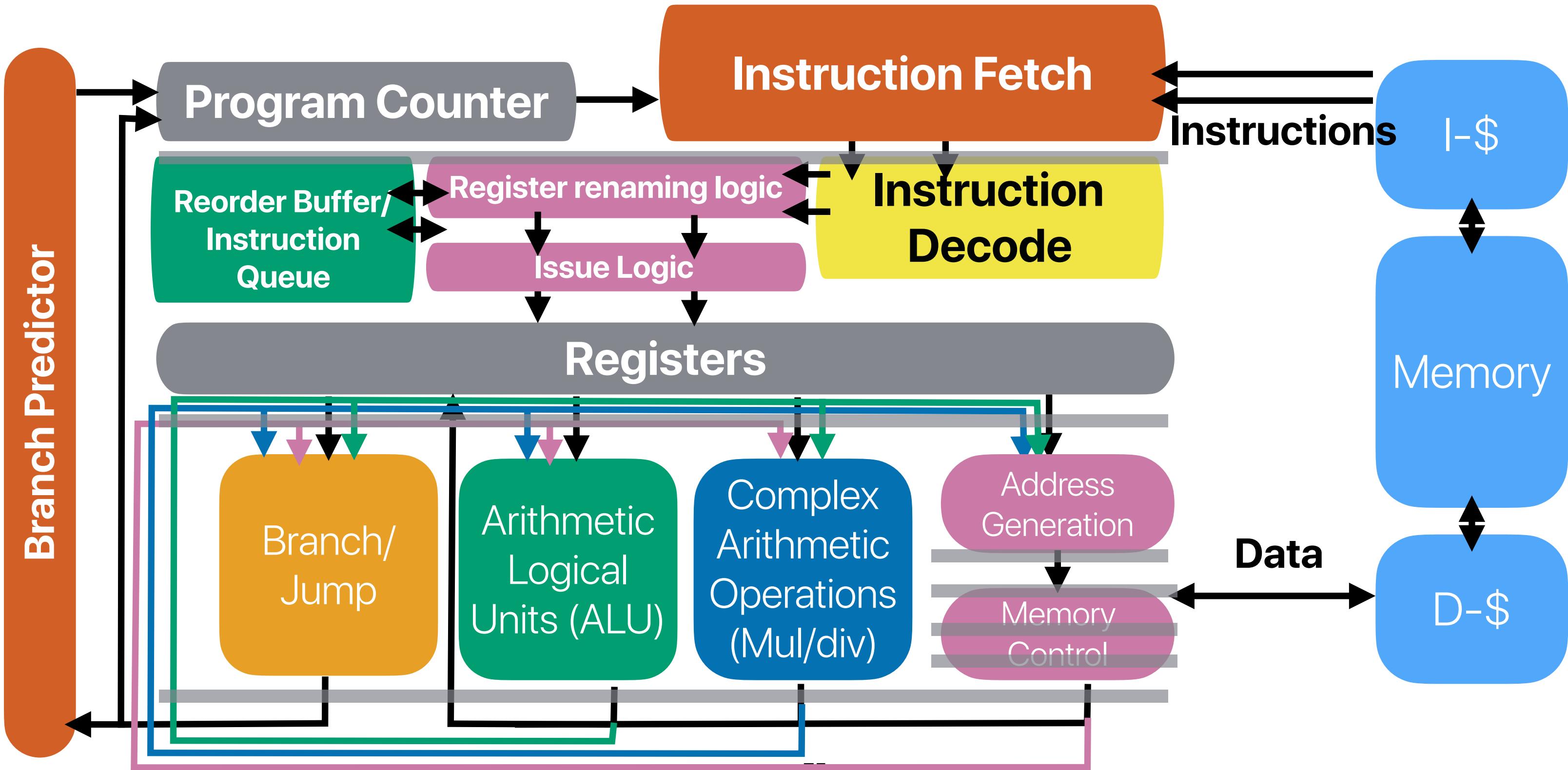
---

*min(issue width, fetch width, decode width)*

# Register renaming + OoO + RoB



# Register renaming + SuperScalar



# 2-issue SS + Register renaming + OoO

2 issue: "2" of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8, (%rdi,%rax)
- ④ movq %rsi, (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi → P4
- ⑨ movq (%rcx,%rax), %r8 → P5
- ⑩ movq %r8, (%rdi,%rax)
- ⑪ movq %rsi, (%rcx,%rax)
- ⑫ addq \$8, %rax → P6
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9
- ⑮ movq (%rdi,%rax), %rsi
- ⑯ movq (%rcx,%rax), %r8
- ⑰ movq %r8, (%rdi,%rax)
- ⑱ movq %rsi, (%rcx,%rax)
- ⑲ addq \$8, %rax
- ⑳ cmpq %r9, %rax
- ㉑ jne .L9

|    | IF     | ID     | REN    | AG | M1 | M2 | M3 | M4 | ALU | MUL | BR | ROB |
|----|--------|--------|--------|----|----|----|----|----|-----|-----|----|-----|
| 1  | (1)(2) |        |        |    |    |    |    |    |     |     |    |     |
| 2  | (3)(4) | (1)(2) |        |    |    |    |    |    |     |     |    |     |
| 3  | (5)(6) | (3)(4) | (1)(2) |    |    |    |    |    |     |     |    |     |
| 4  |        |        |        |    |    |    |    |    |     |     |    |     |
| 5  |        |        |        |    |    |    |    |    |     |     |    |     |
| 6  |        |        |        |    |    |    |    |    |     |     |    |     |
| 7  |        |        |        |    |    |    |    |    |     |     |    |     |
| 8  |        |        |        |    |    |    |    |    |     |     |    |     |
| 9  |        |        |        |    |    |    |    |    |     |     |    |     |
| 10 |        |        |        |    |    |    |    |    |     |     |    |     |
| 11 |        |        |        |    |    |    |    |    |     |     |    |     |
| 12 |        |        |        |    |    |    |    |    |     |     |    |     |
| 13 |        |        |        |    |    |    |    |    |     |     |    |     |
| 14 |        |        |        |    |    |    |    |    |     |     |    |     |
| 15 |        |        |        |    |    |    |    |    |     |     |    |     |
| 16 |        |        |        |    |    |    |    |    |     |     |    |     |
| 17 |        |        |        |    |    |    |    |    |     |     |    |     |
| 18 |        |        |        |    |    |    |    |    |     |     |    |     |
| 19 |        |        |        |    |    |    |    |    |     |     |    |     |
| 20 |        |        |        |    |    |    |    |    |     |     |    |     |
| 21 |        |        |        |    |    |    |    |    |     |     |    |     |

# 2-issue SS + Register renaming + OoO

2 issue: "2" of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8, (%rdi,%rax)
- ④ movq %rsi, (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi → P4
- ⑨ movq (%rcx,%rax), %r8 → P5
- ⑩ movq %r8, (%rdi,%rax)
- ⑪ movq %rsi, (%rcx,%rax)
- ⑫ addq \$8, %rax → P6
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9
- ⑮ movq (%rdi,%rax), %rsi
- ⑯ movq (%rcx,%rax), %r8
- ⑰ movq %r8, (%rdi,%rax)
- ⑱ movq %rsi, (%rcx,%rax)
- ⑲ addq \$8, %rax
- ⑳ cmpq %r9, %rax
- ㉑ jne .L9

|    | IF     | ID     | REN       | AG  | M1 | M2 | M3 | M4 | ALU | MUL | BR | ROB |
|----|--------|--------|-----------|-----|----|----|----|----|-----|-----|----|-----|
| 1  | (1)(2) |        |           |     |    |    |    |    |     |     |    |     |
| 2  | (3)(4) | (1)(2) |           |     |    |    |    |    |     |     |    |     |
| 3  | (5)(6) | (3)(4) | (1)(2)    |     |    |    |    |    |     |     |    |     |
| 4  | (7)(8) | (5)(6) | (2)(3)(4) | (1) |    |    |    |    |     |     |    |     |
| 5  |        |        |           |     |    |    |    |    |     |     |    |     |
| 6  |        |        |           |     |    |    |    |    |     |     |    |     |
| 7  |        |        |           |     |    |    |    |    |     |     |    |     |
| 8  |        |        |           |     |    |    |    |    |     |     |    |     |
| 9  |        |        |           |     |    |    |    |    |     |     |    |     |
| 10 |        |        |           |     |    |    |    |    |     |     |    |     |
| 11 |        |        |           |     |    |    |    |    |     |     |    |     |
| 12 |        |        |           |     |    |    |    |    |     |     |    |     |
| 13 |        |        |           |     |    |    |    |    |     |     |    |     |
| 14 |        |        |           |     |    |    |    |    |     |     |    |     |
| 15 |        |        |           |     |    |    |    |    |     |     |    |     |
| 16 |        |        |           |     |    |    |    |    |     |     |    |     |
| 17 |        |        |           |     |    |    |    |    |     |     |    |     |
| 18 |        |        |           |     |    |    |    |    |     |     |    |     |
| 19 |        |        |           |     |    |    |    |    |     |     |    |     |
| 20 |        |        |           |     |    |    |    |    |     |     |    |     |
| 21 |        |        |           |     |    |    |    |    |     |     |    |     |

# 2-issue SS + Register renaming + OoO

2 issue: "2" of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8, (%rdi,%rax)
- ④ movq %rsi, (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi → P4
- ⑨ movq (%rcx,%rax), %r8 → P5
- ⑩ movq %r8, (%rdi,%rax)
- ⑪ movq %rsi, (%rcx,%rax)
- ⑫ addq \$8, %rax → P6
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9
- ⑮ movq (%rdi,%rax), %rsi
- ⑯ movq (%rcx,%rax), %r8
- ⑰ movq %r8, (%rdi,%rax)
- ⑱ movq %rsi, (%rcx,%rax)
- ⑲ addq \$8, %rax
- ⑳ cmpq %r9, %rax
- ㉑ jne .L9

|    | IF      | ID     | REN          | AG      | M1 | M2 | M3 | M4 | ALU | MUL | BR | ROB |
|----|---------|--------|--------------|---------|----|----|----|----|-----|-----|----|-----|
| 1  | (1)(2)  |        |              |         |    |    |    |    |     |     |    |     |
| 2  | (3)(4)  | (1)(2) |              |         |    |    |    |    |     |     |    |     |
| 3  | (5)(6)  | (3)(4) | (1)(2)       |         |    |    |    |    |     |     |    |     |
| 4  | (7)(8)  | (5)(6) | (2)(3)(4)    | (1)     |    |    |    |    |     |     |    |     |
| 5  | (9)(10) | (7)(8) | (3)(4)(5)(6) | (2) (1) |    |    |    |    |     |     |    |     |
| 6  |         |        |              |         |    |    |    |    |     |     |    |     |
| 7  |         |        |              |         |    |    |    |    |     |     |    |     |
| 8  |         |        |              |         |    |    |    |    |     |     |    |     |
| 9  |         |        |              |         |    |    |    |    |     |     |    |     |
| 10 |         |        |              |         |    |    |    |    |     |     |    |     |
| 11 |         |        |              |         |    |    |    |    |     |     |    |     |
| 12 |         |        |              |         |    |    |    |    |     |     |    |     |
| 13 |         |        |              |         |    |    |    |    |     |     |    |     |
| 14 |         |        |              |         |    |    |    |    |     |     |    |     |
| 15 |         |        |              |         |    |    |    |    |     |     |    |     |
| 16 |         |        |              |         |    |    |    |    |     |     |    |     |
| 17 |         |        |              |         |    |    |    |    |     |     |    |     |
| 18 |         |        |              |         |    |    |    |    |     |     |    |     |
| 19 |         |        |              |         |    |    |    |    |     |     |    |     |
| 20 |         |        |              |         |    |    |    |    |     |     |    |     |
| 21 |         |        |              |         |    |    |    |    |     |     |    |     |

# 2-issue SS + Register renaming + OoO

2 issue: "2" of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8, (%rdi,%rax)
- ④ movq %rsi, (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi → P4
- ⑨ movq (%rcx,%rax), %r8 → P5
- ⑩ movq %r8, (%rdi,%rax)
- ⑪ movq %rsi, (%rcx,%rax)
- ⑫ addq \$8, %rax → P6
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9
- ⑮ movq (%rdi,%rax), %rsi
- ⑯ movq (%rcx,%rax), %r8
- ⑰ movq %r8, (%rdi,%rax)
- ⑱ movq %rsi, (%rcx,%rax)
- ⑲ addq \$8, %rax
- ⑳ cmpq %r9, %rax
- ㉑ jne .L9

|    | IF       | ID      | REN             | AG | M1  | M2  | M3 | M4 | ALU | MUL | BR | ROB |
|----|----------|---------|-----------------|----|-----|-----|----|----|-----|-----|----|-----|
| 1  | (1)(2)   |         |                 |    |     |     |    |    |     |     |    |     |
| 2  | (3)(4)   | (1)(2)  |                 |    |     |     |    |    |     |     |    |     |
| 3  | (5)(6)   | (3)(4)  | (1)(2)          |    |     |     |    |    |     |     |    |     |
| 4  | (7)(8)   | (5)(6)  | (2)(3)(4)       |    | (1) |     |    |    |     |     |    |     |
| 5  | (9)(10)  | (7)(8)  | (3)(4)(5)(6)    |    | (2) | (1) |    |    |     |     |    |     |
| 6  | (11)(12) | (9)(10) | (3)(4)(6)(7)(8) |    | (2) | (1) |    |    |     |     |    |     |
| 7  |          |         |                 |    |     |     |    |    |     |     |    |     |
| 8  |          |         |                 |    |     |     |    |    |     |     |    |     |
| 9  |          |         |                 |    |     |     |    |    |     |     |    |     |
| 10 |          |         |                 |    |     |     |    |    |     |     |    |     |
| 11 |          |         |                 |    |     |     |    |    |     |     |    |     |
| 12 |          |         |                 |    |     |     |    |    |     |     |    |     |
| 13 |          |         |                 |    |     |     |    |    |     |     |    |     |
| 14 |          |         |                 |    |     |     |    |    |     |     |    |     |
| 15 |          |         |                 |    |     |     |    |    |     |     |    |     |
| 16 |          |         |                 |    |     |     |    |    |     |     |    |     |
| 17 |          |         |                 |    |     |     |    |    |     |     |    |     |
| 18 |          |         |                 |    |     |     |    |    |     |     |    |     |
| 19 |          |         |                 |    |     |     |    |    |     |     |    |     |
| 20 |          |         |                 |    |     |     |    |    |     |     |    |     |
| 21 |          |         |                 |    |     |     |    |    |     |     |    |     |

# 2-issue SS + Register renaming + OoO

2 issue: "2" of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8, (%rdi,%rax)
- ④ movq %rsi, (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi → P4
- ⑨ movq (%rcx,%rax), %r8 → P5
- ⑩ movq %r8, (%rdi,%rax)
- ⑪ movq %rsi, (%rcx,%rax)
- ⑫ addq \$8, %rax → P6
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9
- ⑮ movq (%rdi,%rax), %rsi
- ⑯ movq (%rcx,%rax), %r8
- ⑰ movq %r8, (%rdi,%rax)
- ⑱ movq %rsi, (%rcx,%rax)
- ⑲ addq \$8, %rax
- ⑳ cmpq %r9, %rax
- ㉑ jne .L9

|    | IF       | ID       | REN                       | AG | M1  | M2  | M3  | M4 | ALU | MUL | BR | ROB |
|----|----------|----------|---------------------------|----|-----|-----|-----|----|-----|-----|----|-----|
| 1  | (1)(2)   |          |                           |    |     |     |     |    |     |     |    |     |
| 2  | (3)(4)   | (1)(2)   |                           |    |     |     |     |    |     |     |    |     |
| 3  | (5)(6)   | (3)(4)   | (1)(2)                    |    |     |     |     |    |     |     |    |     |
| 4  | (7)(8)   | (5)(6)   | (2)(3)(4)                 |    | (1) |     |     |    |     |     |    |     |
| 5  | (9)(10)  | (7)(8)   | (3)(4)(5)(6)              |    | (2) | (1) |     |    |     |     |    |     |
| 6  | (11)(12) | (9)(10)  | (3)(4)((6)(7)(8))         |    | (2) | (1) |     |    | (5) |     |    |     |
| 7  | (13)(14) | (11)(12) | (3)(4)((6)(7)(9))<br>(10) |    | (8) | (2) | (1) |    |     |     |    | (5) |
| 8  |          |          |                           |    |     |     |     |    |     |     |    |     |
| 9  |          |          |                           |    |     |     |     |    |     |     |    |     |
| 10 |          |          |                           |    |     |     |     |    |     |     |    |     |
| 11 |          |          |                           |    |     |     |     |    |     |     |    |     |
| 12 |          |          |                           |    |     |     |     |    |     |     |    |     |
| 13 |          |          |                           |    |     |     |     |    |     |     |    |     |
| 14 |          |          |                           |    |     |     |     |    |     |     |    |     |
| 15 |          |          |                           |    |     |     |     |    |     |     |    |     |
| 16 |          |          |                           |    |     |     |     |    |     |     |    |     |
| 17 |          |          |                           |    |     |     |     |    |     |     |    |     |
| 18 |          |          |                           |    |     |     |     |    |     |     |    |     |
| 19 |          |          |                           |    |     |     |     |    |     |     |    |     |
| 20 |          |          |                           |    |     |     |     |    |     |     |    |     |
| 21 |          |          |                           |    |     |     |     |    |     |     |    |     |

# 2-issue SS + Register renaming + OoO

2 issue: "2" of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8, (%rdi,%rax)
- ④ movq %rsi, (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi → P4
- ⑨ movq (%rcx,%rax), %r8 → P5
- ⑩ movq %r8, (%rdi,%rax)
- ⑪ movq %rsi, (%rcx,%rax)
- ⑫ addq \$8, %rax → P6
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9
- ⑮ movq (%rdi,%rax), %rsi
- ⑯ movq (%rcx,%rax), %r8
- ⑰ movq %r8, (%rdi,%rax)
- ⑱ movq %rsi, (%rcx,%rax)
- ⑲ addq \$8, %rax
- ⑳ cmpq %r9, %rax
- ㉑ jne .L9

|    | IF       | ID       | REN                   | AG  | M1  | M2  | M3  | M4 | ALU | MUL | BR | ROB    |
|----|----------|----------|-----------------------|-----|-----|-----|-----|----|-----|-----|----|--------|
| 1  | (1)(2)   |          |                       |     |     |     |     |    |     |     |    |        |
| 2  | (3)(4)   | (1)(2)   |                       |     |     |     |     |    |     |     |    |        |
| 3  | (5)(6)   | (3)(4)   | (1)(2)                |     |     |     |     |    |     |     |    |        |
| 4  | (7)(8)   | (5)(6)   | (2)(3)(4)             | (1) |     |     |     |    |     |     |    |        |
| 5  | (9)(10)  | (7)(8)   | (3)(4)(5)(6)          | (2) | (1) |     |     |    |     |     |    |        |
| 6  | (11)(12) | (9)(10)  | (3)(4)((6)(7)(8))     | (2) | (1) |     |     |    | (5) |     |    |        |
| 7  | (13)(14) | (11)(12) | (3)(4)((6)(7)(9)(10)) | (8) | (2) | (1) |     |    | (6) |     |    | (5)    |
| 8  | (15)(16) | (13)(14) | (3)(4)(7)(10)(11)(12) | (9) | (8) | (2) | (1) |    |     | (7) |    | (5)(6) |
| 9  |          |          |                       |     |     |     |     |    |     |     |    |        |
| 10 |          |          |                       |     |     |     |     |    |     |     |    |        |
| 11 |          |          |                       |     |     |     |     |    |     |     |    |        |
| 12 |          |          |                       |     |     |     |     |    |     |     |    |        |
| 13 |          |          |                       |     |     |     |     |    |     |     |    |        |
| 14 |          |          |                       |     |     |     |     |    |     |     |    |        |
| 15 |          |          |                       |     |     |     |     |    |     |     |    |        |
| 16 |          |          |                       |     |     |     |     |    |     |     |    |        |
| 17 |          |          |                       |     |     |     |     |    |     |     |    |        |
| 18 |          |          |                       |     |     |     |     |    |     |     |    |        |
| 19 |          |          |                       |     |     |     |     |    |     |     |    |        |
| 20 |          |          |                       |     |     |     |     |    |     |     |    |        |
| 21 |          |          |                       |     |     |     |     |    |     |     |    |        |

# 2-issue SS + Register renaming + OoO

2 issue: "2" of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8, (%rdi,%rax)
- ④ movq %rsi, (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi → P4
- ⑨ movq (%rcx,%rax), %r8 → P5
- ⑩ movq %r8, (%rdi,%rax)
- ⑪ movq %rsi, (%rcx,%rax)
- ⑫ addq \$8, %rax → P6
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9
- ⑮ movq (%rdi,%rax), %rsi
- ⑯ movq (%rcx,%rax), %r8
- ⑰ movq %r8, (%rdi,%rax)
- ⑱ movq %rsi, (%rcx,%rax)
- ⑲ addq \$8, %rax
- ⑳ cmpq %r9, %rax
- ㉑ jne .L9

|    | IF       | ID       | REN                   | AG  | M1  | M2  | M3  | M4   | ALU | MUL | BR | ROB          |
|----|----------|----------|-----------------------|-----|-----|-----|-----|------|-----|-----|----|--------------|
| 1  | (1)(2)   |          |                       |     |     |     |     |      |     |     |    |              |
| 2  | (3)(4)   | (1)(2)   |                       |     |     |     |     |      |     |     |    |              |
| 3  | (5)(6)   | (3)(4)   | (1)(2)                |     |     |     |     |      |     |     |    |              |
| 4  | (7)(8)   | (5)(6)   | (2)(3)(4)             | (1) |     |     |     |      |     |     |    |              |
| 5  | (9)(10)  | (7)(8)   | (3)(4)(5)(6)          | (2) | (1) |     |     |      |     |     |    |              |
| 6  | (11)(12) | (9)(10)  | (3)(4)((6)(7)(8))     | (2) | (1) |     |     |      | (5) |     |    |              |
| 7  | (13)(14) | (11)(12) | (3)(4)((6)(7)(9)(10)) | (8) | (2) | (1) |     |      | (6) |     |    | (5)          |
| 8  | (15)(16) | (13)(14) | (3)(4)(7)(10)(11)(12) | (9) | (8) | (2) | (1) |      |     | (7) |    | (5)(6)       |
| 9  | (17)(18) | (15)(16) | (3)(10)(11)(13)(14)   | (4) | (9) | (8) | (2) | (12) |     |     |    | (1)(5)(6)(7) |
| 10 |          |          |                       |     |     |     |     |      |     |     |    |              |
| 11 |          |          |                       |     |     |     |     |      |     |     |    |              |
| 12 |          |          |                       |     |     |     |     |      |     |     |    |              |
| 13 |          |          |                       |     |     |     |     |      |     |     |    |              |
| 14 |          |          |                       |     |     |     |     |      |     |     |    |              |
| 15 |          |          |                       |     |     |     |     |      |     |     |    |              |
| 16 |          |          |                       |     |     |     |     |      |     |     |    |              |
| 17 |          |          |                       |     |     |     |     |      |     |     |    |              |
| 18 |          |          |                       |     |     |     |     |      |     |     |    |              |
| 19 |          |          |                       |     |     |     |     |      |     |     |    |              |
| 20 |          |          |                       |     |     |     |     |      |     |     |    |              |
| 21 |          |          |                       |     |     |     |     |      |     |     |    |              |

# 2-issue SS + Register renaming + OoO

2 issue: "2" of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8, (%rdi,%rax)
- ④ movq %rsi, (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi → P4
- ⑨ movq (%rcx,%rax), %r8 → P5
- ⑩ movq %r8, (%rdi,%rax)
- ⑪ movq %rsi, (%rcx,%rax)
- ⑫ addq \$8, %rax → P6
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9
- ⑮ movq (%rdi,%rax), %rsi
- ⑯ movq (%rcx,%rax), %r8
- ⑰ movq %r8, (%rdi,%rax)
- ⑱ movq %rsi, (%rcx,%rax)
- ⑲ addq \$8, %rax
- ⑳ cmpq %r9, %rax
- ㉑ jne .L9

|    | IF       | ID       | REN                   | AG  | M1  | M2  | M3  | M4   | ALU | MUL | BR | ROB              |
|----|----------|----------|-----------------------|-----|-----|-----|-----|------|-----|-----|----|------------------|
| 1  | (1)(2)   |          |                       |     |     |     |     |      |     |     |    |                  |
| 2  | (3)(4)   | (1)(2)   |                       |     |     |     |     |      |     |     |    |                  |
| 3  | (5)(6)   | (3)(4)   | (1)(2)                |     |     |     |     |      |     |     |    |                  |
| 4  | (7)(8)   | (5)(6)   | (2)(3)(4)             | (1) |     |     |     |      |     |     |    |                  |
| 5  | (9)(10)  | (7)(8)   | (3)(4)(5)(6)          | (2) | (1) |     |     |      |     |     |    |                  |
| 6  | (11)(12) | (9)(10)  | (3)(4)((6)(7)(8))     | (2) | (1) |     |     |      | (5) |     |    |                  |
| 7  | (13)(14) | (11)(12) | (3)(4)((6)(7)(9)(10)) | (8) | (2) | (1) |     |      | (6) |     |    | (5)              |
| 8  | (15)(16) | (13)(14) | (3)(4)(7)(10)(11)(12) | (9) | (8) | (2) | (1) |      |     | (7) |    | (5)(6)           |
| 9  | (17)(18) | (15)(16) | (3)(10)(11)(13)(14)   | (4) | (9) | (8) | (2) | (12) |     |     |    | (1)(5)(6)(7)     |
| 10 | (19)(20) | (17)(18) | (10)(11)(14)(15)(16)  | (3) | (4) | (9) | (8) | (13) |     |     |    | (2)(5)(6)(7)(12) |
| 11 |          |          |                       |     |     |     |     |      |     |     |    |                  |
| 12 |          |          |                       |     |     |     |     |      |     |     |    |                  |
| 13 |          |          |                       |     |     |     |     |      |     |     |    |                  |
| 14 |          |          |                       |     |     |     |     |      |     |     |    |                  |
| 15 |          |          |                       |     |     |     |     |      |     |     |    |                  |
| 16 |          |          |                       |     |     |     |     |      |     |     |    |                  |
| 17 |          |          |                       |     |     |     |     |      |     |     |    |                  |
| 18 |          |          |                       |     |     |     |     |      |     |     |    |                  |
| 19 |          |          |                       |     |     |     |     |      |     |     |    |                  |
| 20 |          |          |                       |     |     |     |     |      |     |     |    |                  |
| 21 |          |          |                       |     |     |     |     |      |     |     |    |                  |

# 2-issue SS + Register renaming + OoO

2 issue: "2" of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8, (%rdi,%rax)
- ④ movq %rsi, (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi → P4
- ⑨ movq (%rcx,%rax), %r8 → P5
- ⑩ movq %r8, (%rdi,%rax)
- ⑪ movq %rsi, (%rcx,%rax)
- ⑫ addq \$8, %rax → P6
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9
- ⑮ movq (%rdi,%rax), %rsi
- ⑯ movq (%rcx,%rax), %r8
- ⑰ movq %r8, (%rdi,%rax)
- ⑱ movq %rsi, (%rcx,%rax)
- ⑲ addq \$8, %rax
- ⑳ cmpq %r9, %rax
- ㉑ jne .L9

|    | IF       | ID       | REN                   | AG   | M1  | M2  | M3  | M4   | ALU | MUL  | BR                | ROB              |
|----|----------|----------|-----------------------|------|-----|-----|-----|------|-----|------|-------------------|------------------|
| 1  | (1)(2)   |          |                       |      |     |     |     |      |     |      |                   |                  |
| 2  | (3)(4)   | (1)(2)   |                       |      |     |     |     |      |     |      |                   |                  |
| 3  | (5)(6)   | (3)(4)   | (1)(2)                |      |     |     |     |      |     |      |                   |                  |
| 4  | (7)(8)   | (5)(6)   | (2)(3)(4)             | (1)  |     |     |     |      |     |      |                   |                  |
| 5  | (9)(10)  | (7)(8)   | (3)(4)(5)(6)          | (2)  | (1) |     |     |      |     |      |                   |                  |
| 6  | (11)(12) | (9)(10)  | (3)(4)((6)(7)(8))     | (2)  | (1) |     |     |      | (5) |      |                   |                  |
| 7  | (13)(14) | (11)(12) | (3)(4)((6)(7)(9)(10)) | (8)  | (2) | (1) |     |      | (6) |      |                   | (5)              |
| 8  | (15)(16) | (13)(14) | (3)(4)(7)(10)(11)(12) | (9)  | (8) | (2) | (1) |      |     | (7)  |                   | (5)(6)           |
| 9  | (17)(18) | (15)(16) | (3)(10)(11)(13)(14)   | (4)  | (9) | (8) | (2) | (12) |     |      |                   | (1)(5)(6)(7)     |
| 10 | (19)(20) | (17)(18) | (10)(11)(14)(15)(16)  | (3)  | (4) | (9) | (8) | (13) |     |      |                   | (2)(5)(6)(7)(12) |
| 11 | (21)(22) | (19)(20) | (10)(11)(16)(17)(18)  | (15) | (3) | (4) | (9) | (8)  |     | (14) | (5)(6)(7)(12)(13) |                  |
| 12 |          |          |                       |      |     |     |     |      |     |      |                   |                  |
| 13 |          |          |                       |      |     |     |     |      |     |      |                   |                  |
| 14 |          |          |                       |      |     |     |     |      |     |      |                   |                  |
| 15 |          |          |                       |      |     |     |     |      |     |      |                   |                  |
| 16 |          |          |                       |      |     |     |     |      |     |      |                   |                  |
| 17 |          |          |                       |      |     |     |     |      |     |      |                   |                  |
| 18 |          |          |                       |      |     |     |     |      |     |      |                   |                  |
| 19 |          |          |                       |      |     |     |     |      |     |      |                   |                  |
| 20 |          |          |                       |      |     |     |     |      |     |      |                   |                  |
| 21 |          |          |                       |      |     |     |     |      |     |      |                   |                  |

# 2-issue SS + Register renaming + OoO

2 issue: "2" of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8, (%rdi,%rax)
- ④ movq %rsi, (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi → P4
- ⑨ movq (%rcx,%rax), %r8 → P5
- ⑩ movq %r8, (%rdi,%rax)
- ⑪ movq %rsi, (%rcx,%rax)
- ⑫ addq \$8, %rax → P6
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9
- ⑮ movq (%rdi,%rax), %rsi
- ⑯ movq (%rcx,%rax), %r8
- ⑰ movq %r8, (%rdi,%rax)
- ⑱ movq %rsi, (%rcx,%rax)
- ⑲ addq \$8, %rax
- ⑳ cmpq %r9, %rax
- ㉑ jne .L9

|    | IF       | ID       | REN                   | AG   | M1   | M2  | M3  | M4   | ALU | MUL  | BR | ROB                  |
|----|----------|----------|-----------------------|------|------|-----|-----|------|-----|------|----|----------------------|
| 1  | (1)(2)   |          |                       |      |      |     |     |      |     |      |    |                      |
| 2  | (3)(4)   | (1)(2)   |                       |      |      |     |     |      |     |      |    |                      |
| 3  | (5)(6)   | (3)(4)   | (1)(2)                |      |      |     |     |      |     |      |    |                      |
| 4  | (7)(8)   | (5)(6)   | (2)(3)(4)             | (1)  |      |     |     |      |     |      |    |                      |
| 5  | (9)(10)  | (7)(8)   | (3)(4)(5)(6)          | (2)  | (1)  |     |     |      |     |      |    |                      |
| 6  | (11)(12) | (9)(10)  | (3)(4)((6)(7)(8))     | (2)  | (1)  |     |     |      | (5) |      |    |                      |
| 7  | (13)(14) | (11)(12) | (3)(4)((6)(7)(9)(10)) | (8)  | (2)  | (1) |     |      | (6) |      |    | (5)                  |
| 8  | (15)(16) | (13)(14) | (3)(4)(7)(10)(11)(12) | (9)  | (8)  | (2) | (1) |      |     | (7)  |    | (5)(6)               |
| 9  | (17)(18) | (15)(16) | (3)(10)(11)(13)(14)   | (4)  | (9)  | (8) | (2) | (12) |     |      |    | (1)(5)(6)(7)         |
| 10 | (19)(20) | (17)(18) | (10)(11)(14)(15)(16)  | (3)  | (4)  | (9) | (8) | (13) |     |      |    | (2)(5)(6)(7)(12)     |
| 11 | (21)(22) | (19)(20) | (10)(11)(16)(17)(18)  | (15) | (3)  | (4) | (9) | (8)  |     | (14) |    | (5)(6)(7)(12)(13)    |
| 12 |          | (21)(22) | (16)(17)(18)(19)(20)  | (11) | (15) | (3) | (4) | (9)  |     |      |    | (5)(6)(7)(8)(12)(13) |
| 13 |          |          |                       |      |      |     |     |      |     |      |    |                      |
| 14 |          |          |                       |      |      |     |     |      |     |      |    |                      |
| 15 |          |          |                       |      |      |     |     |      |     |      |    |                      |
| 16 |          |          |                       |      |      |     |     |      |     |      |    |                      |
| 17 |          |          |                       |      |      |     |     |      |     |      |    |                      |
| 18 |          |          |                       |      |      |     |     |      |     |      |    |                      |
| 19 |          |          |                       |      |      |     |     |      |     |      |    |                      |
| 20 |          |          |                       |      |      |     |     |      |     |      |    |                      |
| 21 |          |          |                       |      |      |     |     |      |     |      |    |                      |

# 2-issue SS + Register renaming + OoO

2 issue: "2" of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8, (%rdi,%rax)
- ④ movq %rsi, (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi → P4
- ⑨ movq (%rcx,%rax), %r8 → P5
- ⑩ movq %r8, (%rdi,%rax)
- ⑪ movq %rsi, (%rcx,%rax)
- ⑫ addq \$8, %rax → P6
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9
- ⑮ movq (%rdi,%rax), %rsi
- ⑯ movq (%rcx,%rax), %r8
- ⑰ movq %r8, (%rdi,%rax)
- ⑱ movq %rsi, (%rcx,%rax)
- ⑲ addq \$8, %rax
- ⑳ cmpq %r9, %rax
- ㉑ jne .L9

|    | IF       | ID       | REN                      | AG   | M1   | M2   | M3  | M4   | ALU  | MUL  | BR | ROB                         |
|----|----------|----------|--------------------------|------|------|------|-----|------|------|------|----|-----------------------------|
| 1  | (1)(2)   |          |                          |      |      |      |     |      |      |      |    |                             |
| 2  | (3)(4)   | (1)(2)   |                          |      |      |      |     |      |      |      |    |                             |
| 3  | (5)(6)   | (3)(4)   | (1)(2)                   |      |      |      |     |      |      |      |    |                             |
| 4  | (7)(8)   | (5)(6)   | (2)(3)(4)                | (1)  |      |      |     |      |      |      |    |                             |
| 5  | (9)(10)  | (7)(8)   | (3)(4)(5)(6)             | (2)  | (1)  |      |     |      |      |      |    |                             |
| 6  | (11)(12) | (9)(10)  | (3)(4)((6)(7)(8))        | (2)  | (1)  |      |     |      | (5)  |      |    |                             |
| 7  | (13)(14) | (11)(12) | (3)(4)((6)(7)(9)(10))    | (8)  | (2)  | (1)  |     |      | (6)  |      |    | (5)                         |
| 8  | (15)(16) | (13)(14) | (3)(4)(7)(10)(11)(12)    | (9)  | (8)  | (2)  | (1) |      |      | (7)  |    | (5)(6)                      |
| 9  | (17)(18) | (15)(16) | (3)(10)(11)(13)(14)      | (4)  | (9)  | (8)  | (2) | (12) |      |      |    | (1)(5)(6)(7)                |
| 10 | (19)(20) | (17)(18) | (10)(11)(14)(15)(16)     | (3)  | (4)  | (9)  | (8) | (13) |      |      |    | (2)(5)(6)(7)(12)            |
| 11 | (21)(22) | (19)(20) | (10)(11)(16)(17)(18)     | (15) | (3)  | (4)  | (9) | (8)  |      | (14) |    | (5)(6)(7)(12)(13)           |
| 12 |          | (21)(22) | (16)(17)(18)(19)(20)     | (11) | (15) | (3)  | (4) | (9)  |      |      |    | (5)(6)(7)(8)(12)(13)        |
| 13 |          |          | (16)(17)(18)(20)(21)(22) | (10) | (11) | (15) | (3) | (4)  | (19) |      |    | (5)(6)(7)(8)(9)(12)(13)(14) |
| 14 |          |          |                          |      |      |      |     |      |      |      |    |                             |
| 15 |          |          |                          |      |      |      |     |      |      |      |    |                             |
| 16 |          |          |                          |      |      |      |     |      |      |      |    |                             |
| 17 |          |          |                          |      |      |      |     |      |      |      |    |                             |
| 18 |          |          |                          |      |      |      |     |      |      |      |    |                             |
| 19 |          |          |                          |      |      |      |     |      |      |      |    |                             |
| 20 |          |          |                          |      |      |      |     |      |      |      |    |                             |
| 21 |          |          |                          |      |      |      |     |      |      |      |    |                             |

# 2-issue SS + Register renaming + OoO

2 issue: "2" of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8, (%rdi,%rax)
- ④ movq %rsi, (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi → P4
- ⑨ movq (%rcx,%rax), %r8 → P5
- ⑩ movq %r8, (%rdi,%rax)
- ⑪ movq %rsi, (%rcx,%rax)
- ⑫ addq \$8, %rax → P6
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9
- ⑮ movq (%rdi,%rax), %rsi
- ⑯ movq (%rcx,%rax), %r8
- ⑰ movq %r8, (%rdi,%rax)
- ⑱ movq %rsi, (%rcx,%rax)
- ⑲ addq \$8, %rax
- ⑳ cmpq %r9, %rax
- ㉑ jne .L9

|    | IF       | ID       | REN                      | AG   | M1   | M2   | M3   | M4   | ALU  | MUL  | BR | ROB                                |
|----|----------|----------|--------------------------|------|------|------|------|------|------|------|----|------------------------------------|
| 1  | (1)(2)   |          |                          |      |      |      |      |      |      |      |    |                                    |
| 2  | (3)(4)   | (1)(2)   |                          |      |      |      |      |      |      |      |    |                                    |
| 3  | (5)(6)   | (3)(4)   | (1)(2)                   |      |      |      |      |      |      |      |    |                                    |
| 4  | (7)(8)   | (5)(6)   | (2)(3)(4)                | (1)  |      |      |      |      |      |      |    |                                    |
| 5  | (9)(10)  | (7)(8)   | (3)(4)(5)(6)             | (2)  | (1)  |      |      |      |      |      |    |                                    |
| 6  | (11)(12) | (9)(10)  | (3)(4)((6)(7)(8))        | (2)  | (1)  |      |      |      | (5)  |      |    |                                    |
| 7  | (13)(14) | (11)(12) | (3)(4)((6)(7)(9)(10))    | (8)  | (2)  | (1)  |      |      | (6)  |      |    | (5)                                |
| 8  | (15)(16) | (13)(14) | (3)(4)(7)(10)(11)(12)    | (9)  | (8)  | (2)  | (1)  |      |      | (7)  |    | (5)(6)                             |
| 9  | (17)(18) | (15)(16) | (3)(10)(11)(13)(14)      | (4)  | (9)  | (8)  | (2)  | (12) |      |      |    | (1)(5)(6)(7)                       |
| 10 | (19)(20) | (17)(18) | (10)(11)(14)(15)(16)     | (3)  | (4)  | (9)  | (8)  | (13) |      |      |    | (2)(5)(6)(7)(12)                   |
| 11 | (21)(22) | (19)(20) | (10)(11)(16)(17)(18)     | (15) | (3)  | (4)  | (9)  | (8)  |      | (14) |    | (5)(6)(7)(12)(13)                  |
| 12 |          | (21)(22) | (16)(17)(18)(19)(20)     | (11) | (15) | (3)  | (4)  | (9)  |      |      |    | (5)(6)(7)(8)(12)(13)               |
| 13 |          |          | (16)(17)(18)(20)(21)(22) | (10) | (11) | (15) | (3)  | (4)  | (19) |      |    | (5)(6)(7)(8)(9)(12)(13)(14)        |
| 14 |          |          |                          | (16) | (10) | (11) | (15) | (3)  | (20) |      |    | (4)(5)(6)(7)(8)(9)(12)(13)(14)(19) |
| 15 |          |          |                          |      |      |      |      |      |      |      |    |                                    |
| 16 |          |          |                          |      |      |      |      |      |      |      |    |                                    |
| 17 |          |          |                          |      |      |      |      |      |      |      |    |                                    |
| 18 |          |          |                          |      |      |      |      |      |      |      |    |                                    |
| 19 |          |          |                          |      |      |      |      |      |      |      |    |                                    |
| 20 |          |          |                          |      |      |      |      |      |      |      |    |                                    |
| 21 |          |          |                          |      |      |      |      |      |      |      |    |                                    |

# 2-issue SS + Register renaming + OoO

2 issue: "2" of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8, (%rdi,%rax)
- ④ movq %rsi, (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi → P4
- ⑨ movq (%rcx,%rax), %r8 → P5
- ⑩ movq %r8, (%rdi,%rax)
- ⑪ movq %rsi, (%rcx,%rax)
- ⑫ addq \$8, %rax → P6
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9
- ⑮ movq (%rdi,%rax), %rsi
- ⑯ movq (%rcx,%rax), %r8
- ⑰ movq %r8, (%rdi,%rax)
- ⑱ movq %rsi, (%rcx,%rax)
- ⑲ addq \$8, %rax
- ⑳ cmpq %r9, %rax
- ㉑ jne .L9

|    | IF       | ID       | REN                      | AG   | M1   | M2   | M3   | M4   | ALU  | MUL  | BR   | ROB                                       |
|----|----------|----------|--------------------------|------|------|------|------|------|------|------|------|-------------------------------------------|
| 1  | (1)(2)   |          |                          |      |      |      |      |      |      |      |      |                                           |
| 2  | (3)(4)   | (1)(2)   |                          |      |      |      |      |      |      |      |      |                                           |
| 3  | (5)(6)   | (3)(4)   | (1)(2)                   |      |      |      |      |      |      |      |      |                                           |
| 4  | (7)(8)   | (5)(6)   | (2)(3)(4)                | (1)  |      |      |      |      |      |      |      |                                           |
| 5  | (9)(10)  | (7)(8)   | (3)(4)(5)(6)             | (2)  | (1)  |      |      |      |      |      |      |                                           |
| 6  | (11)(12) | (9)(10)  | (3)(4)((6)(7)(8))        | (2)  | (1)  |      |      |      | (5)  |      |      |                                           |
| 7  | (13)(14) | (11)(12) | (3)(4)((6)(7)(9)(10))    | (8)  | (2)  | (1)  |      |      | (6)  |      |      | (5)                                       |
| 8  | (15)(16) | (13)(14) | (3)(4)(7)(10)(11)(12)    | (9)  | (8)  | (2)  | (1)  |      |      | (7)  |      | (5)(6)                                    |
| 9  | (17)(18) | (15)(16) | (3)(10)(11)(13)(14)      | (4)  | (9)  | (8)  | (2)  | (12) |      |      |      | (1)(5)(6)(7)                              |
| 10 | (19)(20) | (17)(18) | (10)(11)(14)(15)(16)     | (3)  | (4)  | (9)  | (8)  | (13) |      |      |      | (2)(5)(6)(7)(12)                          |
| 11 | (21)(22) | (19)(20) | (10)(11)(16)(17)(18)     | (15) | (3)  | (4)  | (9)  | (8)  |      | (14) |      | (5)(6)(7)(12)(13)                         |
| 12 |          | (21)(22) | (16)(17)(18)(19)(20)     | (11) | (15) | (3)  | (4)  | (9)  |      |      |      | (5)(6)(7)(8)(12)(13)                      |
| 13 |          |          | (16)(17)(18)(20)(21)(22) | (10) | (11) | (15) | (3)  | (4)  | (19) |      |      | (5)(6)(7)(8)(9)(12)(13)(14)               |
| 14 |          |          |                          | (16) | (10) | (11) | (15) | (3)  | (20) |      |      | (4)(5)(6)(7)(8)(9)(12)(13)(14)(19)        |
| 15 |          |          |                          |      | (16) | (10) | (11) | (15) |      |      | (21) | (3)(4)(5)(6)(7)(8)(9)(12)(13)(14)(19)(20) |
| 16 |          |          |                          |      |      |      |      |      |      |      |      |                                           |
| 17 |          |          |                          |      |      |      |      |      |      |      |      |                                           |
| 18 |          |          |                          |      |      |      |      |      |      |      |      |                                           |
| 19 |          |          |                          |      |      |      |      |      |      |      |      |                                           |
| 20 |          |          |                          |      |      |      |      |      |      |      |      |                                           |
| 21 |          |          |                          |      |      |      |      |      |      |      |      |                                           |

# 2-issue SS + Register renaming + OoO

2 issue: "2" of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8, (%rdi,%rax)
- ④ movq %rsi, (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi → P4
- ⑨ movq (%rcx,%rax), %r8 → P5
- ⑩ movq %r8, (%rdi,%rax)
- ⑪ movq %rsi, (%rcx,%rax)
- ⑫ addq \$8, %rax → P6
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9
- ⑮ movq (%rdi,%rax), %rsi
- ⑯ movq (%rcx,%rax), %r8
- ⑰ movq %r8, (%rdi,%rax)
- ⑱ movq %rsi, (%rcx,%rax)
- ⑲ addq \$8, %rax
- ⑳ cmpq %r9, %rax
- ㉑ jne .L9

|    | IF       | ID       | REN                      | AG   | M1   | M2   | M3   | M4   | ALU  | MUL  | BR   | ROB                                       |
|----|----------|----------|--------------------------|------|------|------|------|------|------|------|------|-------------------------------------------|
| 1  | (1)(2)   |          |                          |      |      |      |      |      |      |      |      |                                           |
| 2  | (3)(4)   | (1)(2)   |                          |      |      |      |      |      |      |      |      |                                           |
| 3  | (5)(6)   | (3)(4)   | (1)(2)                   |      |      |      |      |      |      |      |      |                                           |
| 4  | (7)(8)   | (5)(6)   | (2)(3)(4)                | (1)  |      |      |      |      |      |      |      |                                           |
| 5  | (9)(10)  | (7)(8)   | (3)(4)(5)(6)             | (2)  | (1)  |      |      |      |      |      |      |                                           |
| 6  | (11)(12) | (9)(10)  | (3)(4)((6)(7)(8))        | (2)  | (1)  |      |      |      | (5)  |      |      |                                           |
| 7  | (13)(14) | (11)(12) | (3)(4)((6)(7)(9)(10))    | (8)  | (2)  | (1)  |      |      | (6)  |      |      | (5)                                       |
| 8  | (15)(16) | (13)(14) | (3)(4)(7)(10)(11)(12)    | (9)  | (8)  | (2)  | (1)  |      |      | (7)  |      | (5)(6)                                    |
| 9  | (17)(18) | (15)(16) | (3)(10)(11)(13)(14)      | (4)  | (9)  | (8)  | (2)  | (12) |      |      |      | (1)(5)(6)(7)                              |
| 10 | (19)(20) | (17)(18) | (10)(11)(14)(15)(16)     | (3)  | (4)  | (9)  | (8)  | (13) |      |      |      | (2)(5)(6)(7)(12)                          |
| 11 | (21)(22) | (19)(20) | (10)(11)(16)(17)(18)     | (15) | (3)  | (4)  | (9)  | (8)  |      | (14) |      | (5)(6)(7)(12)(13)                         |
| 12 |          | (21)(22) | (16)(17)(18)(19)(20)     | (11) | (15) | (3)  | (4)  | (9)  |      |      |      | (5)(6)(7)(8)(12)(13)                      |
| 13 |          |          | (16)(17)(18)(20)(21)(22) | (10) | (11) | (15) | (3)  | (4)  | (19) |      |      | (5)(6)(7)(8)(9)(12)(13)(14)               |
| 14 |          |          |                          | (16) | (10) | (11) | (15) | (3)  | (20) |      |      | (4)(5)(6)(7)(8)(9)(12)(13)(14)(19)        |
| 15 |          |          |                          |      | (16) | (10) | (11) | (15) |      |      | (21) | (3)(4)(5)(6)(7)(8)(9)(12)(13)(14)(19)(20) |
| 16 |          |          |                          |      |      | (17) | (16) | (10) | (11) |      |      | (12)(13)(14)(15)(19)(20)(21)              |
| 17 |          |          |                          |      |      |      |      |      |      |      |      |                                           |
| 18 |          |          |                          |      |      |      |      |      |      |      |      |                                           |
| 19 |          |          |                          |      |      |      |      |      |      |      |      |                                           |
| 20 |          |          |                          |      |      |      |      |      |      |      |      |                                           |
| 21 |          |          |                          |      |      |      |      |      |      |      |      |                                           |

# 2-issue SS + Register renaming + OoO

2 issue: "2" of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8, (%rdi,%rax)
- ④ movq %rsi, (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi → P4
- ⑨ movq (%rcx,%rax), %r8 → P5
- ⑩ movq %r8, (%rdi,%rax)
- ⑪ movq %rsi, (%rcx,%rax)
- ⑫ addq \$8, %rax → P6
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9
- ⑮ movq (%rdi,%rax), %rsi
- ⑯ movq (%rcx,%rax), %r8
- ⑰ movq %r8, (%rdi,%rax)
- ⑱ movq %rsi, (%rcx,%rax)
- ⑲ addq \$8, %rax
- ⑳ cmpq %r9, %rax
- ㉑ jne .L9

|    | IF       | ID       | REN                      | AG   | M1   | M2   | M3   | M4   | ALU  | MUL  | BR   | ROB                                       |
|----|----------|----------|--------------------------|------|------|------|------|------|------|------|------|-------------------------------------------|
| 1  | (1)(2)   |          |                          |      |      |      |      |      |      |      |      |                                           |
| 2  | (3)(4)   | (1)(2)   |                          |      |      |      |      |      |      |      |      |                                           |
| 3  | (5)(6)   | (3)(4)   | (1)(2)                   |      |      |      |      |      |      |      |      |                                           |
| 4  | (7)(8)   | (5)(6)   | (2)(3)(4)                | (1)  |      |      |      |      |      |      |      |                                           |
| 5  | (9)(10)  | (7)(8)   | (3)(4)(5)(6)             | (2)  | (1)  |      |      |      |      |      |      |                                           |
| 6  | (11)(12) | (9)(10)  | (3)(4)((6)(7)(8))        | (2)  | (1)  |      |      |      | (5)  |      |      |                                           |
| 7  | (13)(14) | (11)(12) | (3)(4)((6)(7)(9)(10))    | (8)  | (2)  | (1)  |      |      | (6)  |      |      | (5)                                       |
| 8  | (15)(16) | (13)(14) | (3)(4)(7)(10)(11)(12)    | (9)  | (8)  | (2)  | (1)  |      |      | (7)  |      | (5)(6)                                    |
| 9  | (17)(18) | (15)(16) | (3)(10)(11)(13)(14)      | (4)  | (9)  | (8)  | (2)  | (12) |      |      |      | (1)(5)(6)(7)                              |
| 10 | (19)(20) | (17)(18) | (10)(11)(14)(15)(16)     | (3)  | (4)  | (9)  | (8)  | (13) |      |      |      | (2)(5)(6)(7)(12)                          |
| 11 | (21)(22) | (19)(20) | (10)(11)(16)(17)(18)     | (15) | (3)  | (4)  | (9)  | (8)  |      | (14) |      | (5)(6)(7)(12)(13)                         |
| 12 |          | (21)(22) | (16)(17)(18)(19)(20)     | (11) | (15) | (3)  | (4)  | (9)  |      |      |      | (5)(6)(7)(8)(12)(13)                      |
| 13 |          |          | (16)(17)(18)(20)(21)(22) | (10) | (11) | (15) | (3)  | (4)  | (19) |      |      | (5)(6)(7)(8)(9)(12)(13)(14)               |
| 14 |          |          |                          | (16) | (10) | (11) | (15) | (3)  | (20) |      |      | (4)(5)(6)(7)(8)(9)(12)(13)(14)(19)        |
| 15 |          |          |                          |      | (16) | (10) | (11) | (15) |      |      | (21) | (12)(13)(14)(19)(20)                      |
| 16 |          |          |                          |      | (17) | (16) | (10) | (11) |      |      |      | (3)(4)(5)(6)(7)(8)(9)(12)(13)(14)(19)(20) |
| 17 |          |          |                          |      |      | (17) | (16) | (10) |      |      |      | (11)(12)(13)(14)(15)(19)(20)(21)          |
| 18 |          |          |                          |      |      |      | (17) | (16) | (10) |      |      |                                           |
| 19 |          |          |                          |      |      |      |      |      |      |      |      |                                           |
| 20 |          |          |                          |      |      |      |      |      |      |      |      |                                           |
| 21 |          |          |                          |      |      |      |      |      |      |      |      |                                           |

# 2-issue SS + Register renaming + OoO

2 issue: "2" of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8, (%rdi,%rax)
- ④ movq %rsi, (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi → P4
- ⑨ movq (%rcx,%rax), %r8 → P5
- ⑩ movq %r8, (%rdi,%rax)
- ⑪ movq %rsi, (%rcx,%rax)
- ⑫ addq \$8, %rax → P6
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9
- ⑮ movq (%rdi,%rax), %rsi
- ⑯ movq (%rcx,%rax), %r8
- ⑰ movq %r8, (%rdi,%rax)
- ⑱ movq %rsi, (%rcx,%rax)
- ⑲ addq \$8, %rax
- ⑳ cmpq %r9, %rax
- ㉑ jne .L9

|    | IF       | ID       | REN                      | AG   | M1   | M2   | M3   | M4   | ALU  | MUL  | BR   | ROB                                       |
|----|----------|----------|--------------------------|------|------|------|------|------|------|------|------|-------------------------------------------|
| 1  | (1)(2)   |          |                          |      |      |      |      |      |      |      |      |                                           |
| 2  | (3)(4)   | (1)(2)   |                          |      |      |      |      |      |      |      |      |                                           |
| 3  | (5)(6)   | (3)(4)   | (1)(2)                   |      |      |      |      |      |      |      |      |                                           |
| 4  | (7)(8)   | (5)(6)   | (2)(3)(4)                | (1)  |      |      |      |      |      |      |      |                                           |
| 5  | (9)(10)  | (7)(8)   | (3)(4)(5)(6)             | (2)  | (1)  |      |      |      |      |      |      |                                           |
| 6  | (11)(12) | (9)(10)  | (3)(4)((6)(7)(8))        | (2)  | (1)  |      |      |      | (5)  |      |      |                                           |
| 7  | (13)(14) | (11)(12) | (3)(4)((6)(7)(9)(10))    | (8)  | (2)  | (1)  |      |      | (6)  |      |      | (5)                                       |
| 8  | (15)(16) | (13)(14) | (3)(4)(7)(10)(11)(12)    | (9)  | (8)  | (2)  | (1)  |      |      | (7)  |      | (5)(6)                                    |
| 9  | (17)(18) | (15)(16) | (3)(10)(11)(13)(14)      | (4)  | (9)  | (8)  | (2)  | (12) |      |      |      | (1)(5)(6)(7)                              |
| 10 | (19)(20) | (17)(18) | (10)(11)(14)(15)(16)     | (3)  | (4)  | (9)  | (8)  | (13) |      |      |      | (2)(5)(6)(7)(12)                          |
| 11 | (21)(22) | (19)(20) | (10)(11)(16)(17)(18)     | (15) | (3)  | (4)  | (9)  | (8)  |      | (14) |      | (5)(6)(7)(12)(13)                         |
| 12 |          | (21)(22) | (16)(17)(18)(19)(20)     | (11) | (15) | (3)  | (4)  | (9)  |      |      |      | (5)(6)(7)(8)(12)(13)                      |
| 13 |          |          | (16)(17)(18)(20)(21)(22) | (10) | (11) | (15) | (3)  | (4)  | (19) |      |      | (5)(6)(7)(8)(9)(12)(13)(14)               |
| 14 |          |          |                          | (16) | (10) | (11) | (15) | (3)  | (20) |      |      | (4)(5)(6)(7)(8)(9)(12)(13)(14)(19)        |
| 15 |          |          |                          |      | (16) | (10) | (11) | (15) |      |      | (21) | (3)(4)(5)(6)(7)(8)(9)(12)(13)(14)(19)(20) |
| 16 |          |          |                          |      |      | (17) | (16) | (10) | (11) |      |      | (12)(13)(14)(15)(19)(20)(21)              |
| 17 |          |          |                          |      |      |      | (17) | (16) | (10) |      |      | (11)(12)(13)(14)(15)(19)(20)(21)          |
| 18 |          |          |                          |      |      |      |      | (17) | (16) |      |      | (10)(11)(12)(13)(14)(15)(19)(20)(21)      |
| 19 |          |          |                          |      |      |      |      |      | (17) | (16) |      | (20)(21)                                  |
| 20 |          |          |                          |      |      |      |      |      |      |      |      |                                           |
| 21 |          |          |                          |      |      |      |      |      |      |      |      |                                           |

# 2-issue SS + Register renaming + OoO

2 issue: "2" of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8, (%rdi,%rax)
- ④ movq %rsi, (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi → P4
- ⑨ movq (%rcx,%rax), %r8 → P5
- ⑩ movq %r8, (%rdi,%rax)
- ⑪ movq %rsi, (%rcx,%rax)
- ⑫ addq \$8, %rax → P6
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9
- ⑮ movq (%rdi,%rax), %rsi
- ⑯ movq (%rcx,%rax), %r8
- ⑰ movq %r8, (%rdi,%rax)
- ⑱ movq %rsi, (%rcx,%rax)
- ⑲ addq \$8, %rax
- ⑳ cmpq %r9, %rax
- ㉑ jne .L9

|    | IF       | ID       | REN                      | AG   | M1   | M2   | M3   | M4   | ALU  | MUL  | BR   | ROB                                       |
|----|----------|----------|--------------------------|------|------|------|------|------|------|------|------|-------------------------------------------|
| 1  | (1)(2)   |          |                          |      |      |      |      |      |      |      |      |                                           |
| 2  | (3)(4)   | (1)(2)   |                          |      |      |      |      |      |      |      |      |                                           |
| 3  | (5)(6)   | (3)(4)   | (1)(2)                   |      |      |      |      |      |      |      |      |                                           |
| 4  | (7)(8)   | (5)(6)   | (2)(3)(4)                | (1)  |      |      |      |      |      |      |      |                                           |
| 5  | (9)(10)  | (7)(8)   | (3)(4)(5)(6)             | (2)  | (1)  |      |      |      |      |      |      |                                           |
| 6  | (11)(12) | (9)(10)  | (3)(4)((6)(7)(8))        | (2)  | (1)  |      |      |      | (5)  |      |      |                                           |
| 7  | (13)(14) | (11)(12) | (3)(4)((6)(7)(9)(10))    | (8)  | (2)  | (1)  |      |      | (6)  |      |      | (5)                                       |
| 8  | (15)(16) | (13)(14) | (3)(4)(7)(10)(11)(12)    | (9)  | (8)  | (2)  | (1)  |      |      | (7)  |      | (5)(6)                                    |
| 9  | (17)(18) | (15)(16) | (3)(10)(11)(13)(14)      | (4)  | (9)  | (8)  | (2)  | (12) |      |      |      | (1)(5)(6)(7)                              |
| 10 | (19)(20) | (17)(18) | (10)(11)(14)(15)(16)     | (3)  | (4)  | (9)  | (8)  | (13) |      |      |      | (2)(5)(6)(7)(12)                          |
| 11 | (21)(22) | (19)(20) | (10)(11)(16)(17)(18)     | (15) | (3)  | (4)  | (9)  | (8)  |      | (14) |      | (5)(6)(7)(12)(13)                         |
| 12 |          | (21)(22) | (16)(17)(18)(19)(20)     | (11) | (15) | (3)  | (4)  | (9)  |      |      |      | (5)(6)(7)(8)(12)(13)                      |
| 13 |          |          | (16)(17)(18)(20)(21)(22) | (10) | (11) | (15) | (3)  | (4)  | (19) |      |      | (5)(6)(7)(8)(9)(12)(13)(14)               |
| 14 |          |          |                          | (16) | (10) | (11) | (15) | (3)  | (20) |      |      | (4)(5)(6)(7)(8)(9)(12)(13)(14)(19)        |
| 15 |          |          |                          |      | (16) | (10) | (11) | (15) |      |      | (21) | (3)(4)(5)(6)(7)(8)(9)(12)(13)(14)(19)(20) |
| 16 |          |          |                          |      |      | (17) | (16) | (10) | (11) |      |      | (12)(13)(14)(15)(19)(20)(21)              |
| 17 |          |          |                          |      |      |      | (17) | (16) | (10) |      |      | (11)(12)(13)(14)(15)(19)(20)(21)          |
| 18 |          |          |                          |      |      |      |      | (17) | (16) |      |      | (10)(11)(12)(13)(14)(15)(19)(20)(21)      |
| 19 |          |          |                          |      |      |      |      |      | (17) | (16) |      | (16)(19)(20)(21)                          |
| 20 |          |          |                          |      |      |      |      |      |      | (18) | (17) |                                           |
| 21 |          |          |                          |      |      |      |      |      |      |      |      |                                           |

# 2-issue SS + Register renaming + OoO

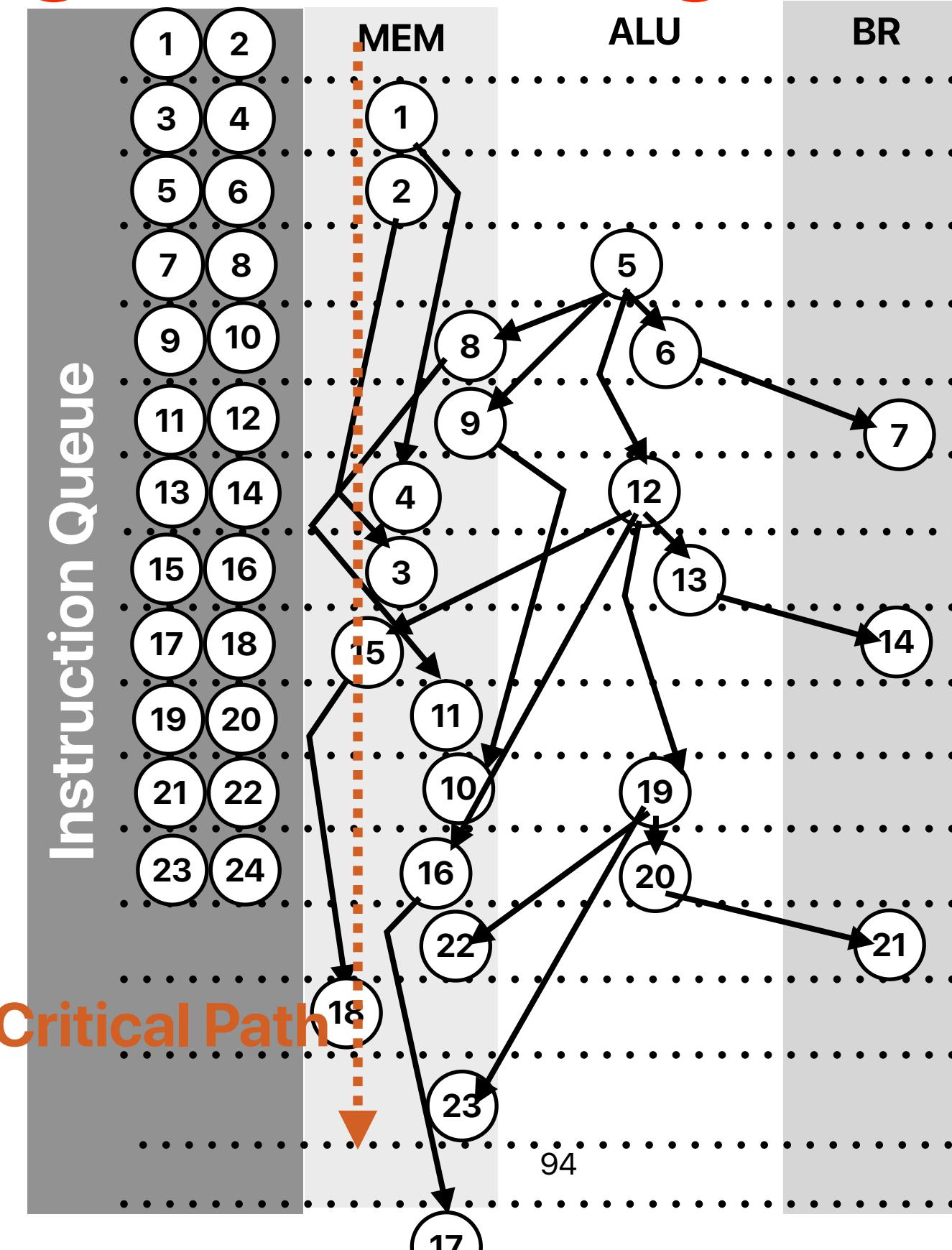
2 issue: "2" of them can have a instruction at the same cycle

- ① movq (%rdi,%rax), %rsi → P1
- ② movq (%rcx,%rax), %r8 → P2
- ③ movq %r8, (%rdi,%rax)
- ④ movq %rsi, (%rcx,%rax)
- ⑤ addq \$8, %rax → P3
- ⑥ cmpq %r9, %rax
- ⑦ jne .L9
- ⑧ movq (%rdi,%rax), %rsi → P4
- ⑨ movq (%rcx,%rax), %r8 → P5
- ⑩ movq %r8, (%rdi,%rax)
- ⑪ movq %rsi, (%rcx,%rax)
- ⑫ addq \$8, %rax → P6
- ⑬ cmpq %r9, %rax
- ⑭ jne .L9
- ⑮ movq (%rdi,%rax), %rsi
- ⑯ movq (%rcx,%rax), %r8
- ⑰ movq %r8, (%rdi,%rax)
- ⑱ movq %rsi, (%rcx,%rax)
- ⑲ addq \$8, %rax
- ⑳ cmpq %r9, %rax
- ㉑ jne .L9

|    | IF       | ID       | REN                      | AG   | M1   | M2   | M3   | M4   | ALU  | MUL  | BR   | ROB                                       |                                      |
|----|----------|----------|--------------------------|------|------|------|------|------|------|------|------|-------------------------------------------|--------------------------------------|
| 1  | (1)(2)   |          |                          |      |      |      |      |      |      |      |      |                                           |                                      |
| 2  | (3)(4)   | (1)(2)   |                          |      |      |      |      |      |      |      |      |                                           |                                      |
| 3  | (5)(6)   | (3)(4)   | (1)(2)                   |      |      |      |      |      |      |      |      |                                           |                                      |
| 4  | (7)(8)   | (5)(6)   | (2)(3)(4)                | (1)  |      |      |      |      |      |      |      |                                           |                                      |
| 5  | (9)(10)  | (7)(8)   | (3)(4)(5)(6)             | (2)  | (1)  |      |      |      |      |      |      |                                           |                                      |
| 6  | (11)(12) | (9)(10)  | (3)(4)((6)(7)(8))        | (2)  | (1)  |      |      |      | (5)  |      |      |                                           |                                      |
| 7  | (13)(14) | (11)(12) | (3)(4)((6)(7)(9)(10))    | (8)  | (2)  | (1)  |      |      | (6)  |      |      | (5)                                       |                                      |
| 8  | (15)(16) | (13)(14) | (3)(4)(7)(10)(11)(12)    | (9)  | (8)  | (2)  | (1)  |      |      | (7)  |      | (5)(6)                                    |                                      |
| 9  | (17)(18) | (15)(16) | (3)(10)(11)(13)(14)      | (4)  | (9)  | (8)  | (2)  | (12) |      |      |      | (1)(5)(6)(7)                              |                                      |
| 10 | (19)(20) | (17)(18) | (10)(11)(14)(15)(16)     | (3)  | (4)  | (9)  | (8)  | (13) |      |      |      | (2)(5)(6)(7)(12)                          |                                      |
| 11 | (21)(22) | (19)(20) | (10)(11)(16)(17)(18)     | (15) | (3)  | (4)  | (9)  | (8)  |      | (14) |      | (5)(6)(7)(12)(13)                         |                                      |
| 12 |          | (21)(22) | (16)(17)(18)(19)(20)     | (11) | (15) | (3)  | (4)  | (9)  |      |      |      | (5)(6)(7)(8)(12)(13)                      |                                      |
| 13 |          |          | (16)(17)(18)(20)(21)(22) | (10) | (11) | (15) | (3)  | (4)  | (19) |      |      | (5)(6)(7)(8)(9)(12)(13)(14)               |                                      |
| 14 |          |          |                          | (16) | (10) | (11) | (15) | (3)  | (20) |      |      | (4)(5)(6)(7)(8)(9)(12)(13)(14)(19)        |                                      |
| 15 |          |          |                          |      | (16) | (10) | (11) | (15) |      |      | (21) | (3)(4)(5)(6)(7)(8)(9)(12)(13)(14)(19)(20) |                                      |
| 16 |          |          |                          |      |      | (17) | (16) | (10) | (11) |      |      |                                           | (12)(13)(14)(15)(19)(20)(21)         |
| 17 |          |          |                          |      |      |      | (17) | (16) | (10) |      |      |                                           | (11)(12)(13)(14)(15)(19)(20)(21)     |
| 18 |          |          |                          |      |      |      |      | (17) | (16) |      |      |                                           | (10)(11)(12)(13)(14)(15)(19)(20)(21) |
| 19 |          |          |                          |      |      |      |      |      | (17) | (16) |      |                                           |                                      |
| 20 |          |          |                          |      |      |      |      |      |      | (18) | (17) |                                           | (16)(19)(20)(21)                     |
| 21 |          |          |                          |      |      |      |      |      |      |      | (18) | (17)                                      | (19)(20)(21)                         |

# Through data flow graph analysis

```
① movq (%rdi,%rax), %rsi
② movq (%rcx,%rax), %r8
③ movq %r8, (%rdi,%rax)
④ movq %rsi, (%rcx,%rax)
⑤ addq $8, %rax
⑥ cmpq %r9, %rax
⑦ jne .L9
⑧ movq (%rdi,%rax), %rsi
⑨ movq (%rcx,%rax), %r8
⑩ movq %r8, (%rdi,%rax)
⑪ movq %rsi, (%rcx,%rax)
⑫ addq $8, %rax
⑬ cmpq %r9, %rax
⑭ jne .L9
⑮ movq (%rdi,%rax), %rsi
⑯ movq (%rcx,%rax), %r8
⑰ movq %r8, (%rdi,%rax)
⑱ movq %rsi, (%rcx,%rax)
⑲ addq $8, %rax
⑳ cmpq %r9, %rax
㉑ jne .L9
㉒ movq (%rdi,%rax), %rsi
㉓ movq (%rcx,%rax), %r8
㉔ movq %r8, (%rdi,%rax)
㉕ movq %rsi, (%rcx,%rax)
㉖ addq $8, %rax
㉗ cmpq %r9, %rax
㉘
```



12 cycles for every 11 memory instructions

If we have 11 loops, it will have 44 memory instructions, 77 instructions in total and take 48 cycles

CPI:

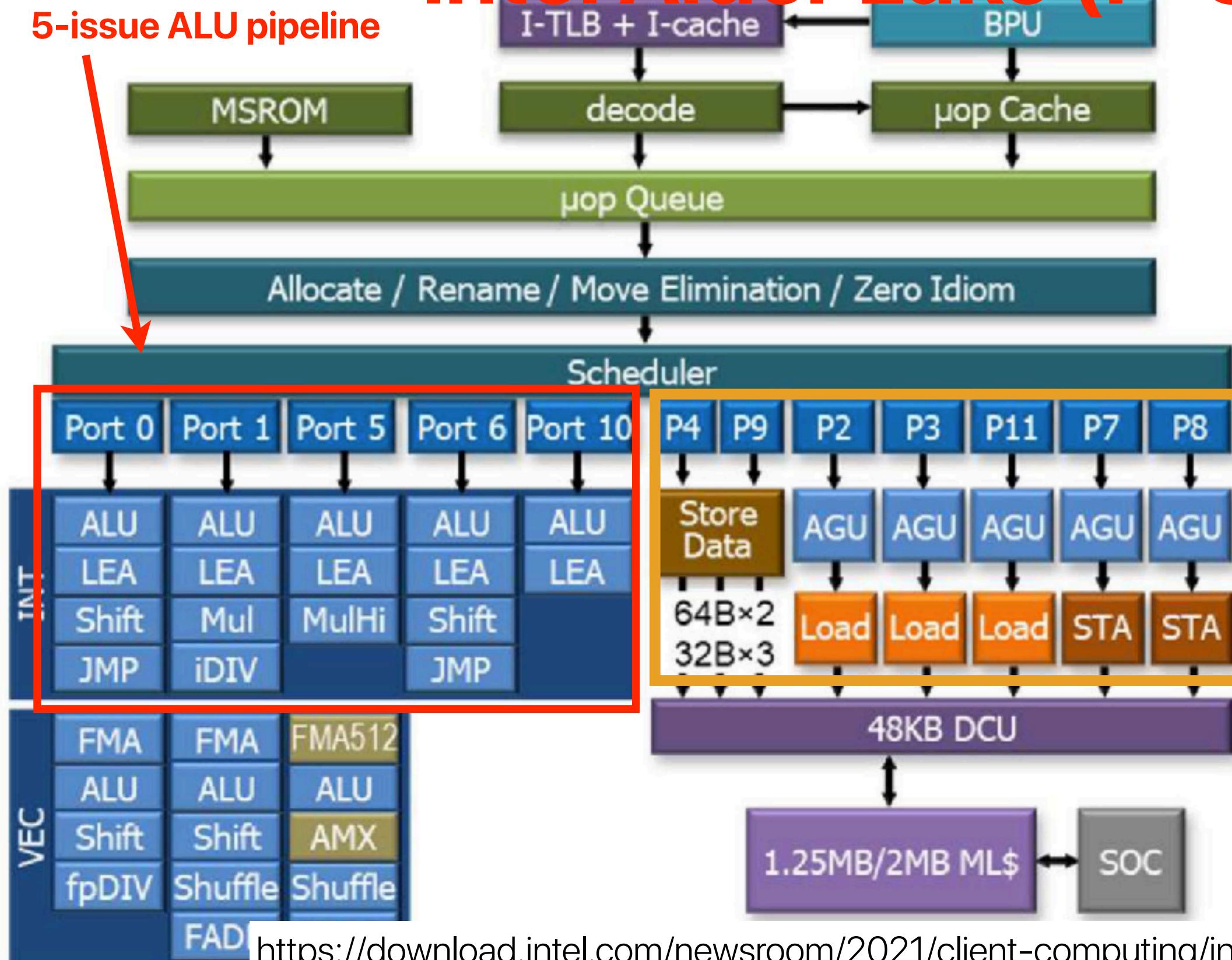
$$\frac{48}{77} = 0.62$$

# Takeaways: data hazards

- More data dependencies, more likelihood of data hazards
- Stalls and data forwarding can both address data hazards to generate correct code execution results — but not very efficient
- Compiler optimizations can help, but to a limited extent
- False dependencies limits the freedom of out-of-order execution
- Register renaming + Speculative execution enables more efficient execution by dynamically scheduling instructions whenever their data dependencies are resolved
- Super scalar further improves the utilization of hardware and throughput

# **The pipelines of Modern Processors**

# Intel Alder Lake (P-Core)



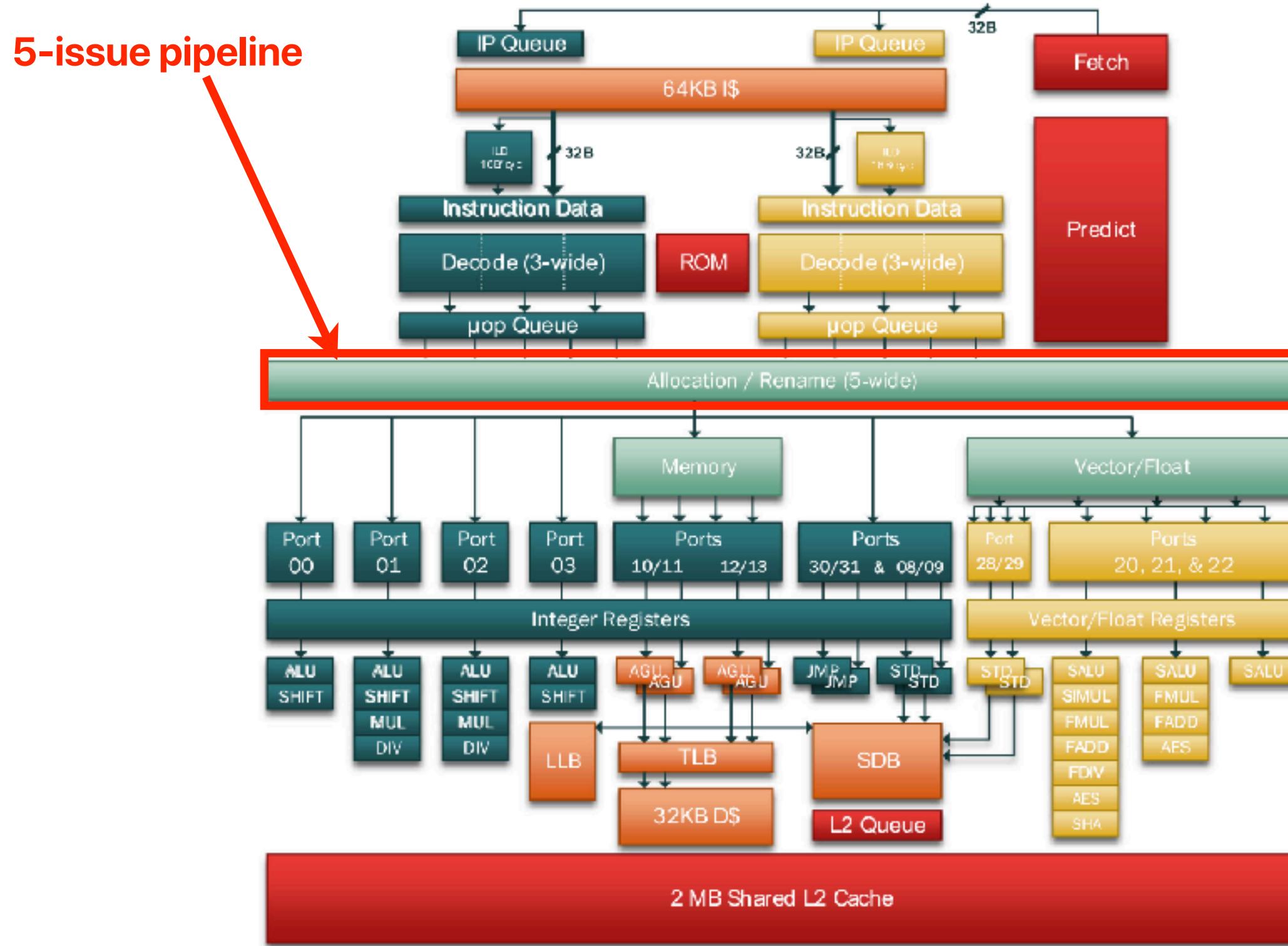
$$MinCPI = \frac{1}{12}$$

$$MinINTInst . CPI = \frac{1}{5}$$

$$MinMEMInst . CPI = \frac{1}{7}$$

$$MinBRInst . CPI = \frac{1}{2}$$

# Intel Alder Lake (E-Core)



# AMD Zen 3 (RyZen 5000 Series)

**3-issue memory pipeline**

**4-issue integer pipeline + 1 additional branch**

$$MinCPI = \frac{1}{8}$$

$$MinINTInst . CPI = \frac{1}{4}$$

$$MinMEMInst . CPI = \frac{1}{3}$$

$$MinBRIInst . CPI = \frac{1}{2}$$

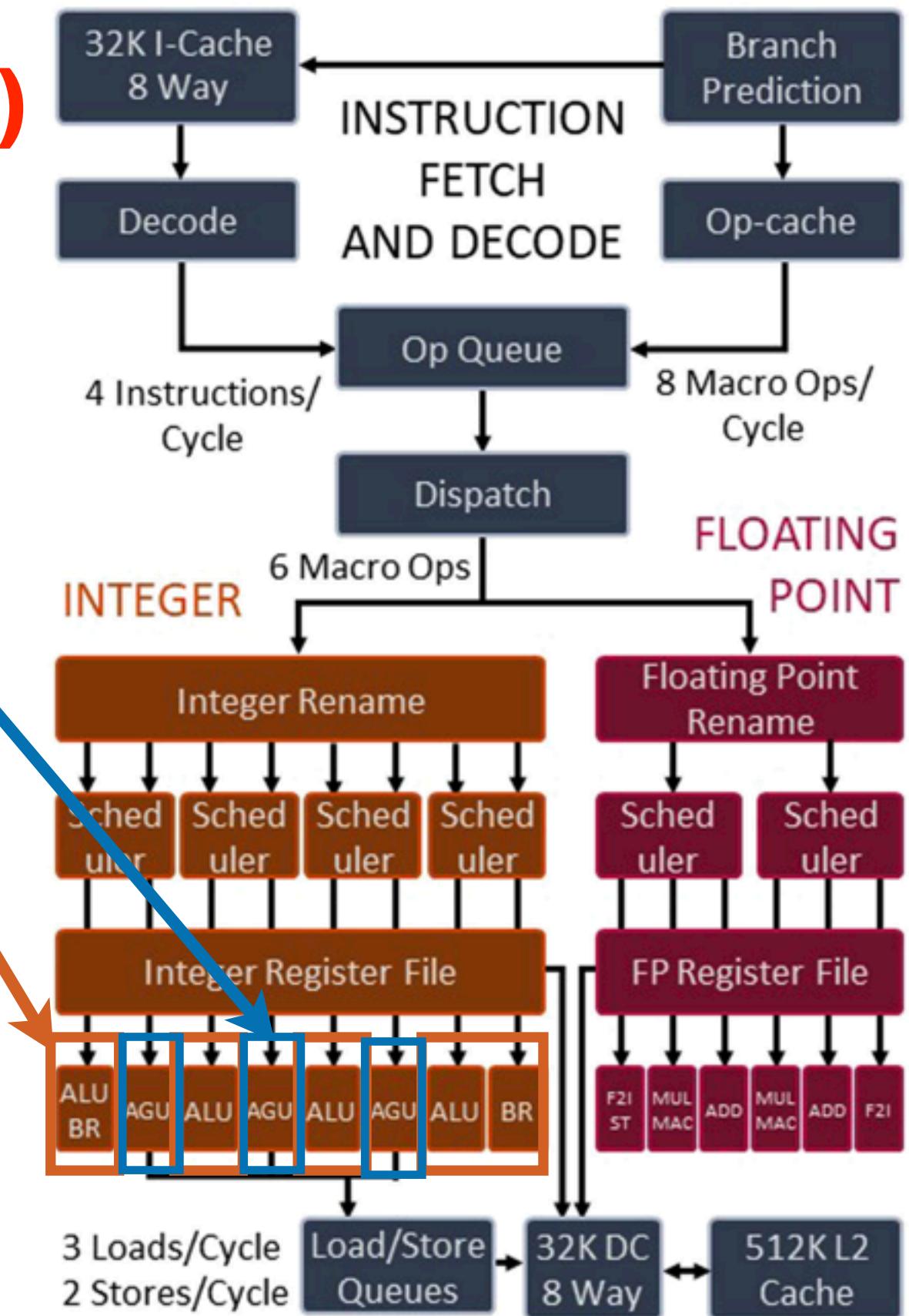


FIGURE 1. "Zen 3" block diagram.

# Summary: Characteristics of modern processor architectures

- Multiple-issue pipelines with multiple functional units available
  - Multiple ALUs
  - Multiple Load/store units
  - Dynamic OoO scheduling to reorder instructions whenever possible
- Cache — very high hit rate **if your code has good locality**
  - Very matured data/instruction prefetcher
- Branch predictors — very high accuracy **if your code is predictable**
  - Perceptron
  - TAGE

# Takeaways: data hazards

- More data dependencies, more likelihood of data hazards
- Stalls and data forwarding can both address data hazards to generate correct code execution results — but not very efficient
- Compiler optimizations can help, but to a limited extent
- False dependencies limits the freedom of out-of-order execution
- Register renaming + Speculative execution enables more efficient execution by dynamically scheduling instructions whenever their data dependencies are resolved
- Super scalar further improves the utilization of hardware and throughput
- Modern processors are all very wide-issue super scalar processors with OoO capabilities

# Announcements

- **Assignment 4** due next **Thursday**
- **Reading Quiz 8** due **next Tuesday** before the lecture
- **Programming assignment 3** released

# Computer Science & Engineering

203

つづく

