## Modern Processor Design (III): Can't Hardly Wait

Hung-Wei Tseng

## Recap: But A is faster!

```
 d. /* one line statement using bit-wise operators */ (most efficient)
 a^=b^=a^=b;
```

The order of evaluation is from right to left. This is same as in approach (c) but the three statements are compounded into one statement.

```
void regswap(int* a, int* b) {
   int temp = *a;
   *a = *b;
   *b = temp;
}
```

```
\mathbf{m}
```

```
void xorswap(int* a, int* b) {
    *a ^= *b = *a = *b;
}
```

## **Outline**

- Data hazards
- Hardware optimizations for data hazards

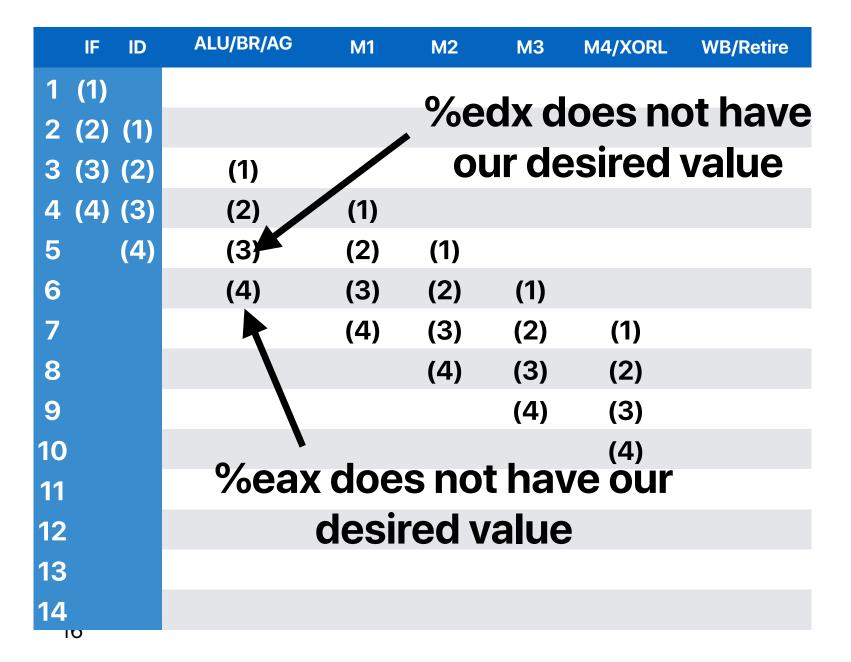
## Data hazards

#### **Data hazards**

- An instruction currently in the pipeline cannot receive the "logically" correct value for execution
- Data dependencies
  - The output of an instruction is the input of a later instruction
  - May sometimes result in data hazard if the later instruction that consumes the result is still in the pipeline

### **Data hazards**

```
① movl (%rdi), %eax
② movl (%rsi), %edx
③ movl %edx, (%rdi)
④ movl %eax, (%rsi)
```



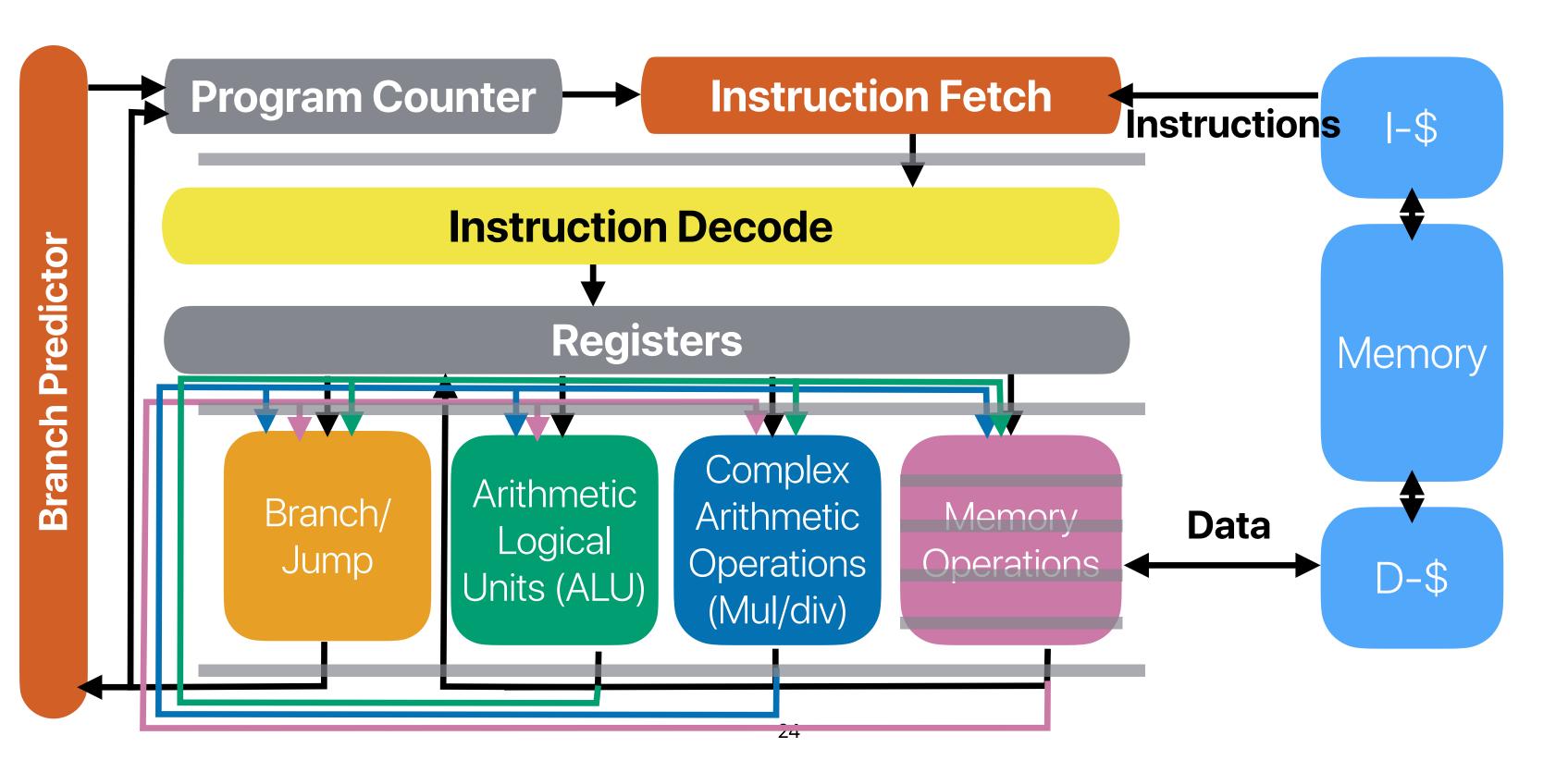
## Solution 1: Let's try "stall" again

 Whenever the input is not ready when the consumer is decoding, just stall — the consumer stays at ID.

## Solution 2: Data forwarding

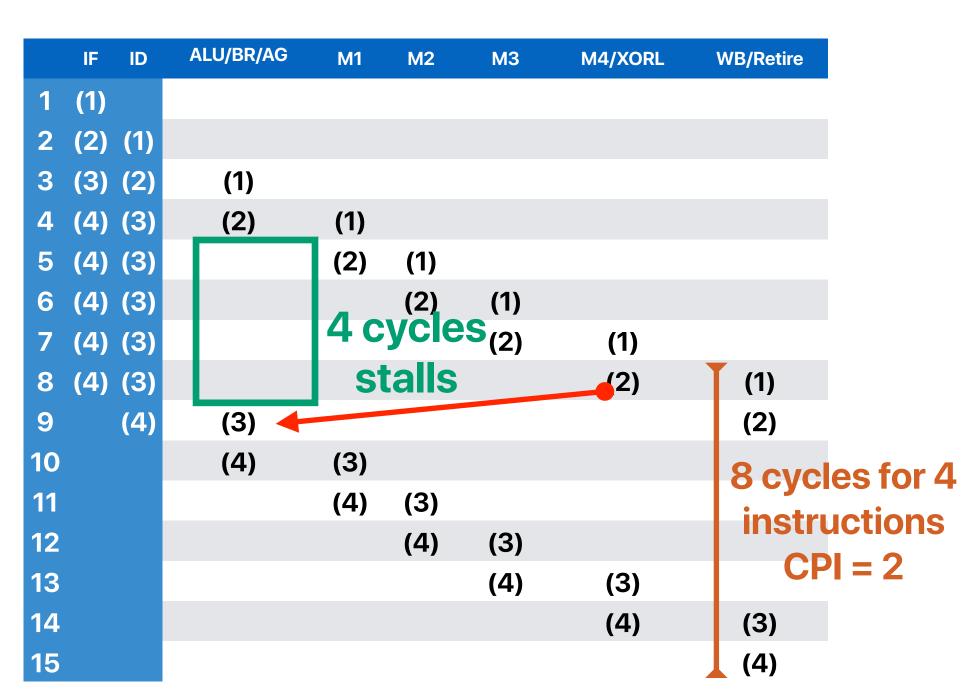
 Add logics/wires to forward the desired values to the demanding instructions

## Data "forwarding"



## The effect of data forwarding

```
① movl (%rdi), %eax
② movl (%rsi), %edx
③ movl %edx, (%rdi)
④ movl %eax, (%rsi)
```



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

			•					
	IF	ID	ALU/BR/AG	M1	M2	МЗ	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)				
11	<b>(7)</b>	(6)	(5)	(4)	(3)			
12	(8)	<b>(7)</b>	(6)	) •	(4)	(3)		
13	(9)	(8)	(7)			(4)	(3)	
14	(10)	(9)	(8)				(4)	(3)
15		(10)	(9)	(8)				(4)
16		(10)		(9)	(8)			(5)
17	(11)		11	cycles	s for	7 (8)		(6)
18		(10)				(9)	(8)	(7)
19		(10)		struct	lions		(9)	(8)
20	(12)		(10)	CDI = 1	1.57			(9)
21		(12)	(11)	(10)				
22	(14)	(13)	(12)	(11)	(10)			
23		(14)	(13)	(12)	(11)	(10)		
24			<b>(14)</b>	(13)	(12)	(11)	(10)	
32								

```
for(i = 0; i < count; i++) {

Compiler optimization

IF ID ALU/BR/
                                                                                                  M3 M4/XORL WB/Retire
                                                                            ID ALU/BR/AG M1
                                                                                              M2
                                                                        (1)
        int64_t temp = a[i];
                                                                                                  9 cycles for 7
                                                                        (2)
                                                                            (1)
        a[i] = b[i];
                                                                        (3)
                                                                                   (1)
                                                                                                   instructions
                                                                        (4)
                                                                                  (2)
                                                                                         (1)
                                                                            (3)
        b[i] = temp;
                                                                                                    CPI = 1.29
                                                                        (5)
                                                                                  (3)
                                                                                         (2)
                                                                                              (1)
                                                                        (5)
                                                                                              (2)
                                                                                                  (1)
                                                                        (5)
                                                                                                  (2)
                                                                                                        (1)
                                                                                                                (1)
                                                                        (6)
                                                                                                        (2)
                                                                                  (4)
                                                                                                                (2)
                                                                        (7)
                                                                                  (5)
                                                                                         (4)
                                  .L9:
                                                                        (8)
                                                                                                                (3)
                                                                                  (6)
                                                                                         (5)
                                                                                              (4)
          (%rdi,%rax), %rsi
                                               (%rcx,%rax), %r8
movq
                                      movq
                                                                        (9)
                                                                                                  (4)
                                                                                   (7)
                                                                                              (5)
                                               (%rdi,%rax), %rsi
          (%rcx,%rax), %r8
movq
                                      mova
                                                                     12 (10)
                                                                                  (8)
                                                                                                  (5)
                                                                                                        (4)
                                               $8, %rax
          %r8, (%rdi,%rax)
                                      addq
                                                                     13 (11)
                                                                                  (9)
movq
                                                                           (10)
                                                                                         (8)
                                                                                                        (5)
                                                                                                                (4)
                                               %r8, -8(%rdi,%rax)
                                                                    14 (11)
                                                                                                                (5)
          %rsi, (%rcx,%rax)
                                      mova
                                                                           (10)
                                                                                  (10)
                                                                                         (9)
                                                                                              (8)
movq
                                               %rsi, -8(%rcx,%rax 15 (11)
                                      mova
                                                                                                                (6)
                                                                                              (9)
                                                                                                  (8)
                                               %r9, %rax
                                                                                                  (9)
                                                                                                                (7)
                                                                     16 (11)
                                      cmpq
                                                                           (10)
                                                                                                        (8)
          %r9, %rax
cmpq
                                               .L9
                                                                     17 (12)
                                      jne
                                                                                                                (8)
                                                                                                        (9)
          .L9
jne
                                               (%rcx,%rax), %r8
                                                                     18 (12)
                                                                                                                (9)
                                  8
                                      mova
          (%rdi,%rax), %rsi
movq
                                               (%rdi,%rax), %rsi
                                                                    19 (13) (12)
                                                                                                                (10)
                                  9
                                                                                  (11)
                                      mova
          (%rcx,%rax), %r8
mova
                                               $8, %rax
                                                                    20 (14) (13)
                                                                                         (11)
                                      addq
                                                                                  (12)
          %r8, (%rdi,%rax)
movq
                                  (11)
                                               %r8, -8(%rdi,%rax)
                                                                                         (12)
                                                                                              (11)
                                                                                  (13)
                                                                           (14)
                                      movq
          %rsi, (%rcx,%rax)
movq
                                  12
                                               %rsi, -8(%rcx,%rax 22
                                                                                  (14)
                                                                                              (12) (11)
                                      movq
addq
          $8, %rax
                                                                                                  (12)
                                                                                                        (11)
                                                                     23
                                               %r9, %rax
                                      cmpq
                                                                    24
                                                                                                                (11)
          %r9, %rax
                                                                                                        (12)
cmpq
                                               .L9
                                      ine
                                                                    25
                                                                                                                (12)
          .L9
jne
                                                          38
                                                                    26
                                                                                                                (13)
```

```
for(i = 0; i < count; i++) {</pre>
                                                                                     M2
                                                                                            M3
                                                                                                 M4/XORL WB/Retire
                                                                                 M1
       int64_t temp = a[i];
                                                                   (1)
       a[i] = b[i];
                                                                          (1)
       b[i]
                = temp;
                                                                          (2)
                                                                                (1)
                        Compiler can only do this when it's 100% for sure count is
                                   always an even number!—Ioop unrolling
                                                                                                   (1)
                                                                                                          (1)
                                                                          (4)
                                                             8 (6) (5)
                             .L9:
         (%rcx,%rax), %r8
movq
                                                                          (5)
                                                                                (4)
                                                                                                         (2)
                                          (%rcx,%rax), %r8
                                 mova
         (%rdi,%rax), %rsi
mova
                                                                                                         (3)
                                          (%rdi,%rax), %rsi 10 (8)
                                                                                     (4)
                                                                          (6)
                                                                                (5)
                                 mova
addq
        $8, %rax
                                                                                (6)
                                                                                     (5)7 cycles for 7
                                                                          (7)
                                                             11 (9)
                                          $8, %rax
                                 addq
        %r8, -8(%rdi,%rax)
movq
                                         %r8, -8(%rdi,%rax)12(10)(9)
                                                                          (8)
                                                                                (7)
        %rsi, -8(%rcx,%rax)<sub>⑤</sub>
                                 mova
movq
                                                                          (9)
                                         %rsi, -8(%rcx,%rax13 (11) (10)
                                                                                                          (4)
                                 movq
        %r9, %rax
cmpq
                                                             14 (12) (11)
                                                                         (10)
                                          (%rcx,%rax), %r8
                             6
                                                                                                          (5)
                                 mova
jne
         .L9
                                          (%rdi,%rax), %rsi 15 (13) (12)
                                                                                (10)
                                                                                                   (7)
                                                                         (11)
                                                                                                          (6)
                             7
                                 mova
         (%rcx,%rax), %r
movq
                                          %r9, %rax
                                                             16 (14) (13)
                                                                         (12)
                                                                                (11)
                                                                                     (10)
                                                                                                          (7)
                                 cmpq
         (%rdi,%rax), %rsi
movq
                                          .L9
                                 jne
                                                                         (13)
                                                                                (12)
                                                                                     (11)
                                                                                           (10)
                                                                                                          (8)
                                                                   (14)
        $8, %rax
addq
                                          $8, %rax
                                 addq
                                                                                                          (9)
                                                                         (14)
                                                                                (13)
                                                                                     (12)
                                                                                           (11)
                                                                                                  (10)
        %r8, -8(%rdi,%rax)
movq
                                         %r8, -8(%rdi,%rax)<sub>19</sub>
                                 mova
                                                                                (14)
                                                                                     (13)
                                                                                           (12)
                                                                                                  (11)
                                                                                                         (10)
        %rsi, -8(%rcx,%rax
movq
                                         %rsi, -8(%rcx,%rax20
                                 movq
                                                                                                         (11)
                                                                                     (14)
                                                                                           (13)
                                                                                                  (12)
        %r9, %rax
cmpq
                                         %19, %1ax
                                 CIIIPY
                                                             21
                                                                                                  (13)
                                                                                                         (12)
                                                                                           (14)
         .L9
jne
                                 jne
                                          .L9
                                                             22
                                                                                                         (13)
                                                                                                  (14)
```

39

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 $(i = 0; i < count; i++)$ {													
Tor	<del>-</del>		<del>-</del>			IF	ID	ALU/BR/AG	M1	M2	МЗ	M4/XORL	WB/Retire	
	<pre>int64_t_temp</pre>	) =	= a[i]	•		1 (1)								
	a[i] = Pro			or c	an n	KO		ct v	vh	at				
			<b>G33</b>	UI C	an p	3 (3)	(2,		VII	at				
	b[i] = temp;					4 (4)	(3)	(2)	(1)					
}	b[i] = temp; Shot		d ha	ann	en al	no	4	nro		the	2			
,				app.			<b>(-)</b>			(2)	(1)	442		
•						(5)	(4)	/ /-			(2)	(1)	(4)	
movq	(%rcx,%rax), %r8				/nan			4,11	(4)			(2)	(1)	
movq	(%rdi,%rax), %rsi	1						4.00		(4)			(2) (3)	
addq	\$8, %rax	2	movq		rax), %rsi	10 (8)	(2)	(6) (7)	(5) (6)	(4)		o for		
movq	%r8, <b>-8</b> (%rdi,%rax)	(3)	addq	\$8, %rax				(7) (8)	(6) (7)			es for 7		
movq	%rsi, <b>-8</b> (%rcx,%rax	)	movq		(%rdi,%rax <mark>B(%rcx,%ra</mark>			(9)	(/)	$\frac{(3)}{(7)}$ ir	<b>istru</b>	ctions	(4)	
cmpq	%r9, %rax	6	movq		rax), %r8			(10)		( / )		<b>=</b> (5)	(5)	
jne	.L9	7	movq		rax), %ro rax), %rsi			(11)	(10)		GPI	<b>(7)</b>	(6)	
movq	(%rcx,%rax), %r	8	cmpq	%r9, %r	•	16 (14)		(12)	(11)	(10)		(//	(7)	
movq	(%rdi,%rax), %rsi	9	jne	.L9	<i>-</i> , , ,	17	(14)	(13)	(12)	(11)	(10)	·	(8)	
addq	\$8, %rax	10	addq	\$8, %rax	X	18	( ,	(14)	(13)	(12)	(11)	(10)	(9)	
movq	%r8, -8(%rdi,%rax)		movq	•	(%rdi,%rax	19		<b>(</b> )	(14)	(13)	(12)	(11)	(10)	
movq	%rsi, -8(%rcx,%rax	12	movq	%rsi, -	8(%rcx,%ra	20				(14)	(13)	(12)	(11)	
cmpq jne	%r9, %rax .L9	10	Сшрч	%19, %10	ах	21				- 7	(14)	(13)	(12)	
יונ	• <b>L</b> /	14)	jne	.L9		22					• •	(14)	(13)	
					42	23							(14)	

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

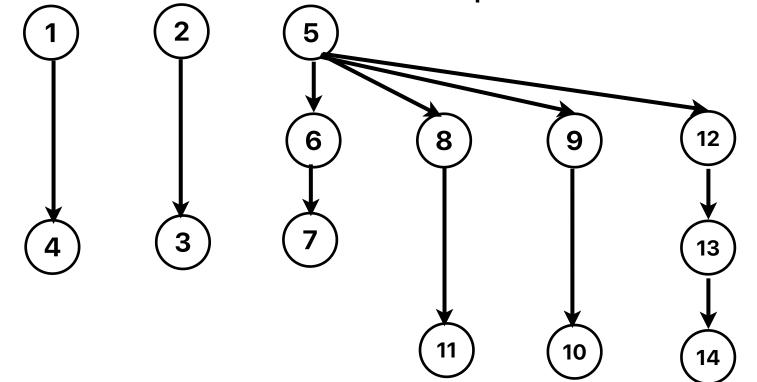
## What do you need to execution an instruction?

- 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.

```
(%rdi,%rax), %rsi
  movq
          (%rcx,%rax), %r8
  movq
          %r8, (%rdi,%rax)
  movq
          %rsi, (%rcx,%rax)
  movq
  addq
          $8, %rax
          %r9, %rax
  cmpq
  jne
           .L9
          (%rdi,%rax), %rsi
  movq
          (%rcx,%rax), %r8
  movq
          %r8, (%rdi,%rax)
10 movq
① movq
          %rsi, (%rcx,%rax)
          $8, %rax
12 addq
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

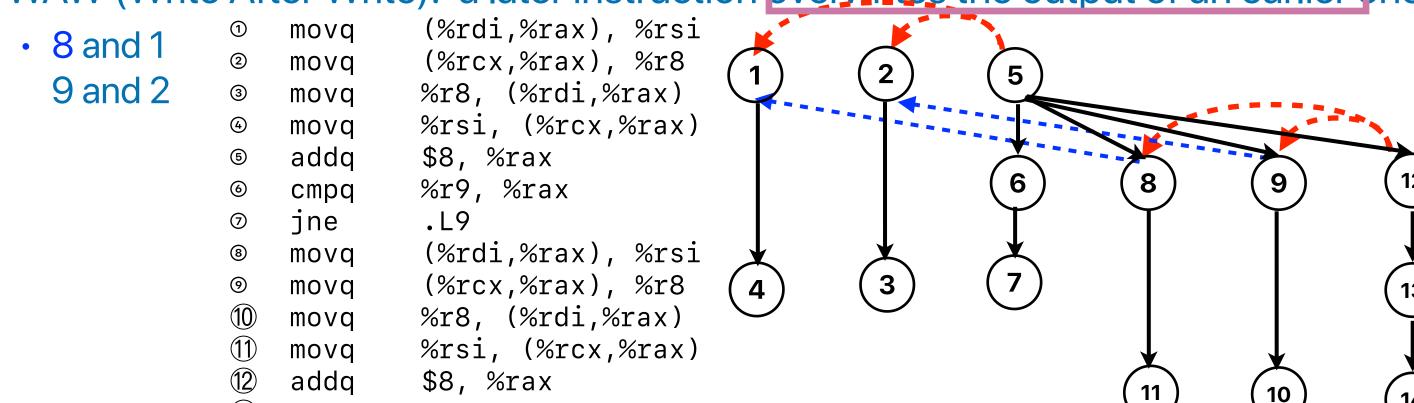
## False dependencies

- 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
    - 5 and 1, 5 and 2, 12 and 8, 12 and 9

cmpq

ine

WAW (Write After Write): a later instruction overwrites the output of an earlier one



51

%r9, %rax

.L9

## False dependencies

We are still limited by false dependencies

.L9

- They are not "true" dependencies because they don't have an arrow in data dependency graph
  - WAR (Write After Read): a later instruction powerwrites the source of an earlier one
    - 5 and 1, 5 and 2, 12 and 8, 12 and 9

ine

 WAW (Write After Write): a later instruction overwrites the output of an earlier one (%rdi,%rax), %rsi movq • 8 and 1 (%rcx,%rax), %r8 mova cmpq jne movq 7 3 (%rcx,%rax), %r8 movq 13 %r8, (%rdi,%rax) movq %rsi, (%rcx,%rax) movq \$8, %rax addq 11 %r9, %rax cmpq 52

## 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

# 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

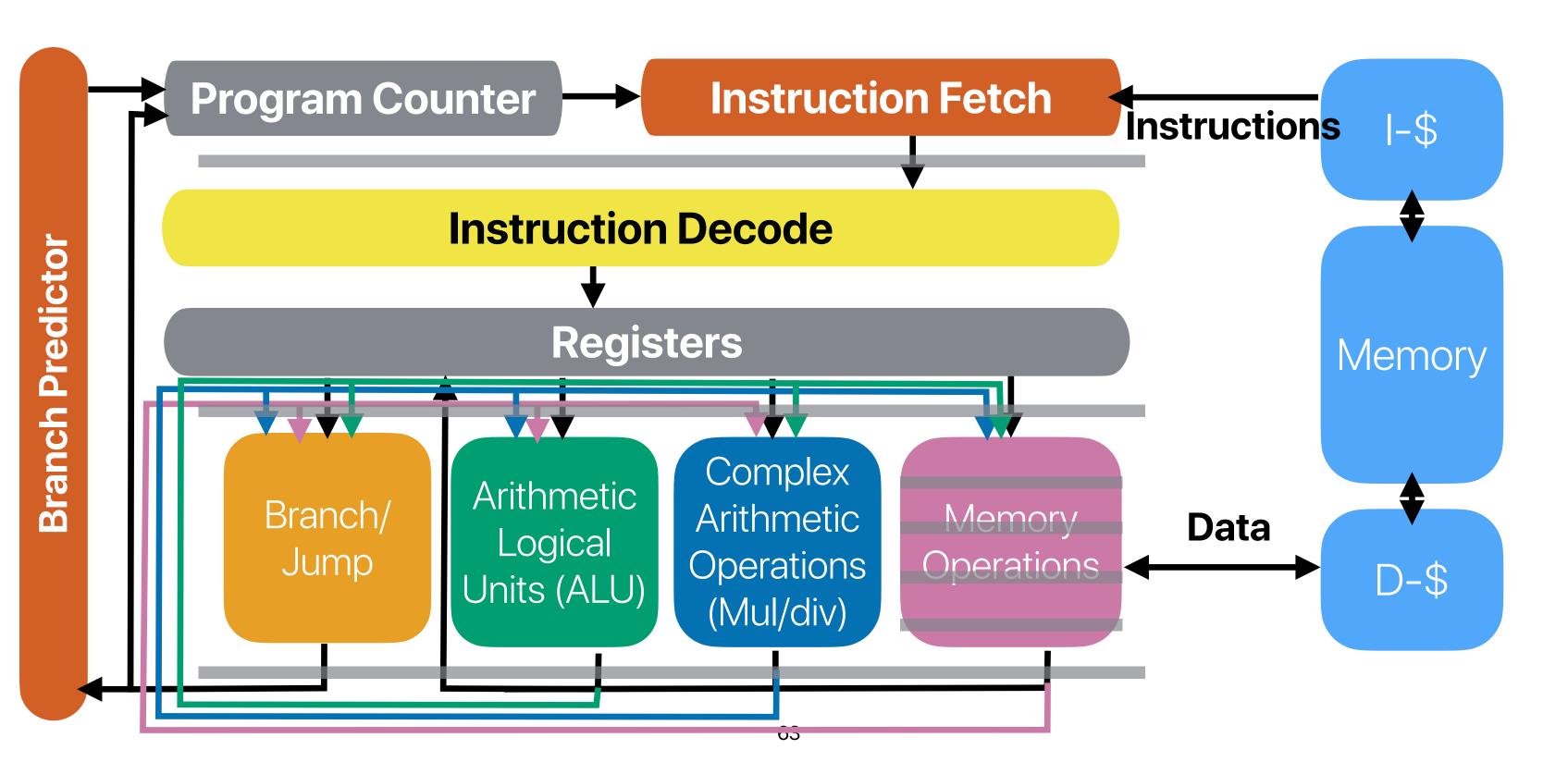
## Can we really execute instructions OoO?

- 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

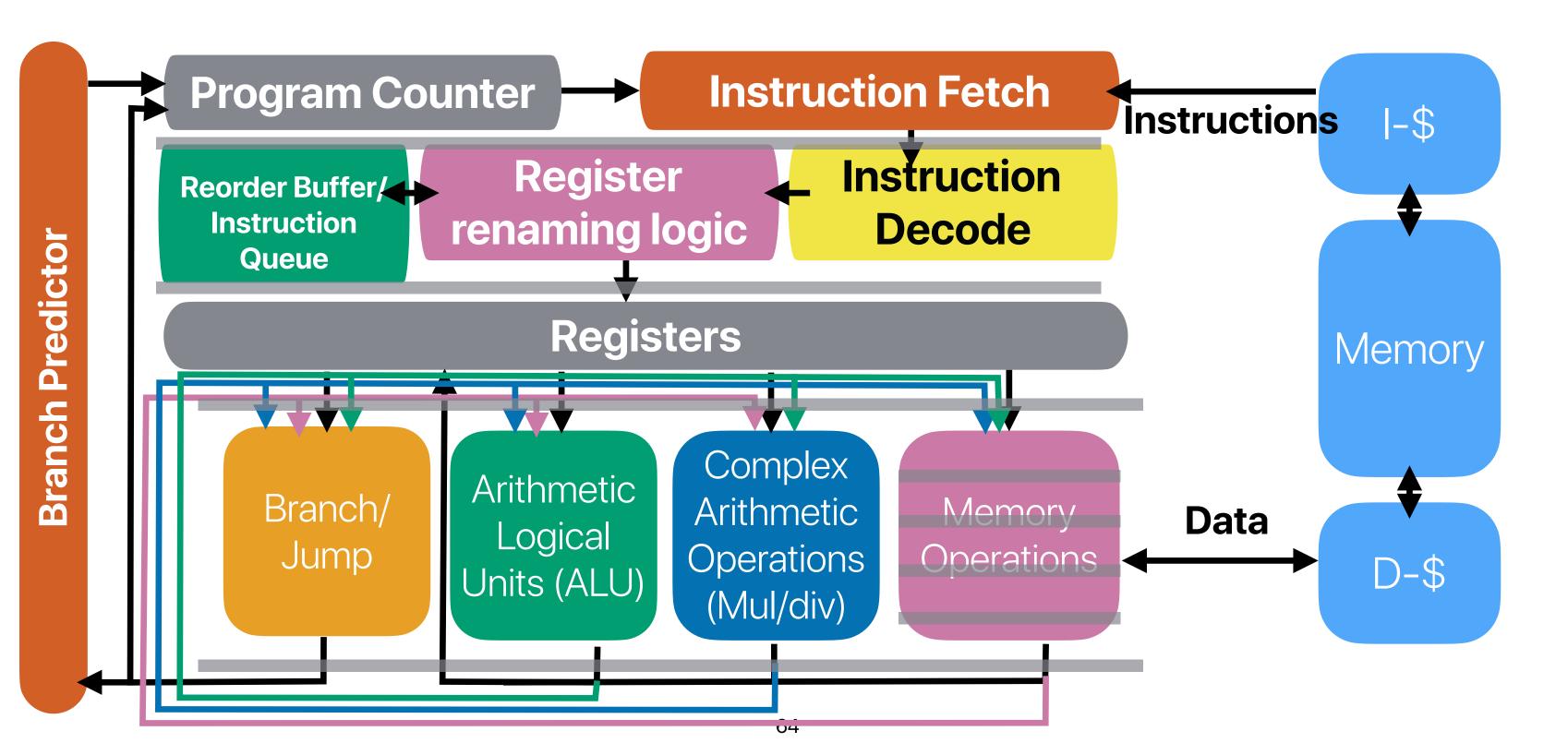
## **Speculative Execution**

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

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

jne .L9

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

	Physical Register
rax	
rcx	
rdi	
rsi	
r8	

	Valid	Value	In use		Valid	Value	In use
P1				P6			
P2				P7			
Р3				P8			
P4				P9			
P5				P10			

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

```
    movq (%rdi,%rax), %rsi → P1
    movq (%rcx,%rax), %r8
    movq %r8, (%rdi,%rax)
    movq %rsi, (%rcx,%rax)
    addq $8, %rax
    cmpq %r9, %rax
```

- <sup>⊙</sup> jne .L9
- ® movq (%rdi,%rax), %rsi
- 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	М3	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	)											
1												
12	2											
13												
14												
1	5											
10	3											

	Physical Register
rax	
rcx	
rdi	
rsi	P1
r8	

	Valid	Value	In use		Valid	Value	In use
P1	0		1	P6			
P2				P7			
Р3				P8			
P4				P9			
P5				P10			

Register renaming

```
  movq (%rdi,%rax), %rsi → P1
```

- ② movq (%rcx,%rax), %r8 → P2
- movq %r8, (%rdi,%rax)
- movq %rsi, (%rcx,%rax)
- ⑤ addq \$8, %rax
- © cmpq %r9, %rax
- <sup>⊙</sup> jne .L9
- ® movq (%rdi,%rax), %rsi
- 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 th	em can have a	instruction	on at th	ne same cy	/cle
					•

	IF.	ID	DEN		<b>N</b> 44	140	MO	<b>N</b> 4 4	ALVI			DOD
	<u>IF</u>	ID	REN	AG	IVI	MZ	M3	M4	ALU	MUL	BR	ROB
1	(1)											
2	(2)	(1)										
3	(3)	(2)	(1)									
4	(4)	(3)	(2)	(1)								
5	(5)		(3)	(2)	(1)							
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												

	Physical Register
rax	
rcx	
rdi	
rsi	P1
r8	<b>P2</b>

	Valid	Value	In use		Valid	Value	In use
P1	0		1	P6			
P2	0		1	P7			
Р3				P8			
P4				P9			
P5				P10			

Register renaming

```
  movq (%rdi,%rax), %rsi → P1
```

- ② movq (%rcx,%rax), %r8 → P2
- movq %r8, (%rdi,%rax)
- movq %rsi, (%rcx,%rax)
- ⑤ addq \$8, %rax
- © cmpq %r9, %rax
- <sup>⊙</sup> jne .L9
- ® movq (%rdi,%rax), %rsi
- 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 th	iem can ha <del>ve</del> a	instruction	on at th	ne same	cycle

	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)	<b>(2)</b>	(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	<b>P2</b>

	Valid	Value	In use		Valid	Value	In use
P1	0		1	P6			
P2	0		1	P7			
Р3				P8			
P4				P9			
P5				P10			

Register renaming

```
① movq (%rdi,%rax), %rsi → P1
```

- @ movq (%rcx,%rax), %r8  $\rightarrow$  P2
- movq %r8, (%rdi,%rax)
- movq %rsi, (%rcx,%rax)
- ⑤ addq \$8, %rax
- © cmpq %r9, %rax
- <sup>⊙</sup> jne .L9
- ® movq (%rdi,%rax), %rsi
- 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 th	iem can have a	instructio	n at the	same cycl	e

				$\Delta$								
	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	<b>(7)</b>	(6)	(3)(4)(5)			(2)	(1)					
8												
9												
10												
11												
12												
13												
14												
15												
16												

	Physical Register
rax	
rcx	
rdi	
rsi	P1
r8	<b>P2</b>

	Valid	Value	In use		Valid	Value	In use
P1	0		1	P6			
P2	0		1	P7			
Р3				P8			
P4				P9			
P5				P10			

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

```
    movq (%rdi,%rax), %rsi → P1
    movq (%rcx,%rax), %r8 → P2
    movq %r8, (%rdi,%rax)
    movq %rsi, (%rcx,%rax)
    addq $8, %rax → P3
    cmpq %r9, %rax
    jne .L9
    movq (%rdi,%rax), %rsi
```

- 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	<b>(7)</b>	(6)	(3)(4)(5)			(2)	(1)					
8												
9												
10												
11												
12												
13												
14												
15												
16												

	Physical Register
rax	Р3
rcx	
rdi	
rsi	P1
r8	<b>P2</b>

	Valid	Value	In use		Valid	Value	In use
P1	0		1	P6			
P2	0		1	P7			
Р3	0		1	P8			
P4				P9			
P5				P10			

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

```
    movq (%rdi,%rax), %rsi → P1
    movq (%rcx,%rax), %r8 → P2
    movq %r8, (%rdi,%rax)
    movq %rsi, (%rcx,%rax)
    addq $8, %rax → P3
    cmpq %r9, %rax
    jne .L9
    movq (%rdi,%rax), %rsi
```

- movq (%rcx,%rax), %r8
   movq (%rcx,%rax), %r8
- 10 movq %r8, (%rdi,%rax)
- ① movq %rsi, (%rcx,%rax)
- 12 addq \$8, %rax
- 13 cmpq %r9, %rax
- 14 jne .L9

										7		
	IF	ID	REN	AG	M1	M2	<b>M3</b>	M4	ALU	MUL	BR	ROB
1	(1)											
2	(2)	(1)						Ir	netri	ictio	า (5) is	9
3	(3)	(2)	(1)									
4	(4)	(3)	(2)	(1)				run	ning	, ahe	ad of	(3)
5	(5)	(4)	(3)	(2)	(1)							
6	(6)	(5)	(3)(4)		(2)	(1)						
7	<b>(7)</b>	(6)	(3)(4)(5)			(2)	(1)					
8	(8)	<b>(7)</b>	(3)(4)(6)				(2)	(1)	(5)			
9												
10												
11												
12												
13												
14												
15												
16												

Physical Register							
rax	Р3						
rcx							
rdi							
rsi	P1						
r8	P2						

	Valid	Value	In use		Valid	Value	In use
P1	0		1	P6			
P2	0		1	P7			
Р3	0		1	P8			
P4				P9			
P5				P10			

```
    movq (%rdi,%rax), %rsi → P1
    movq (%rcx,%rax), %r8 → P2
    movq %r8, (%rdi,%rax)
    movq %rsi, (%rcx,%rax)
    addq $8, %rax → P3
    cmpq %r9, %rax
    jne .L9
    movq (%rdi,%rax), %rsi
```

10 movq %r8, (%rdi,%rax)

movq (%rcx,%rax), %r8

- ① movq %rsi, (%rcx,%rax)
- 12 addq \$8, %rax
- ① cmpq %r9, %rax
- 14 jne .L9

	IF	ID	REN	AG M1					MUL	BR	ROB
1	(1)			nstru	<b>icti</b>	on	(4	) is			
2	(2)	(1)		ning	ak	100		f (2	\ Ins	struc <sup>*</sup>	tion (5) is
3	(3)	(2)	(1)				iu c	<i>7</i> 1 (3			
4	(4)	(3)	(2)	(1)					runr	ing a	head of (3
5	(5)	(4)	(3)	(2) (1)							
6	(6)	(5)	(3)(4)	(2)	(1)						
7	<b>(7)</b>	(6)	(3)(4)(5)		(2)	(1)					
8	(8)	<b>(7)</b>	(3)(4)(6)			(2)	(1)	(5)			
9	(9)	(8)	(3)(6)(7)	(4)			(2)				(1)(5)
10											
11											
12											
13											
14											
15											
16											

	Physical Register
rax	Р3
rcx	
rdi	
rsi	P1
r8	P2

	Valid	Value	In use		Valid	Value	In use
P1	1		1	P6			
P2	0		1	P7			
Р3	1		1	P8			
P4				P9			
P5				P10			

```
    movq (%rdi,%rax), %rsi → P1
    movq (%rcx,%rax), %r8 → P2
    movq %r8, (%rdi,%rax)
    movq %rsi, (%rcx,%rax)
    addq $8, %rax → P3
    cmpq %r9, %rax
    jne .L9
    movq (%rdi,%rax), %rsi
```

- 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	М3	M4	ALU	MUL	BR	ROB
1	(1)											
2	(2)	(1)										
3	(3)	(2)	(1)					R	atire	/Con	nmit (	<b>(1)</b>
4	(4)	(3)	(2)	(1)						, 0011	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	. • /
5	(5)	(4)	(3)	(2)	(1)							
6	(6)	(5)	(3)(4)		(2)	(1)						
7	<b>(7)</b>	(6)	(3)(4)(5)			(2)	(1)					
8	(8)	<b>(7)</b>	(3)(4)(6)				(2)	(1)	(5)			
9	(9)	(8)	(3)(6)(7)	(4)				(2)				<del>(1)(</del> 5)
10												
11												
12												
13												
14												
15												
16												

	Physical Register
rax	Р3
rcx	
rdi	
rsi	P1
r8	<b>P2</b>

	Valid	Value	In use		Valid	Value	In use
P1	1		1	P6			
P2	0		1	P7			
Р3	1		1	P8			
P4				P9			
P5				P10			

```
movq (%rdi,%rax), %rsi → P1
movq (%rcx,%rax), %r8 → P2
movq %r8, (%rdi,%rax)
movq %rsi, (%rcx,%rax)
addq $8, %rax → P3
cmpq %r9, %rax
```

- ® movq (%rdi,%rax), %rsi → P4
- movq (%rcx,%rax), %r8
- 10 movq %r8, (%rdi,%rax)
- ① movq %rsi, (%rcx,%rax)
- 12 addq \$8, %rax

jne .L9

- 13 cmpq %r9, %rax
- 14 jne .L9

	IF	ID	REN	AG	M1	M2	М3	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)	<b>(7)</b>	(3)(4)(6)				(2)	(1)	(5)			
9	(9)	(8)	(3)(6)(7)	(4)				(2)				<del>(1)(</del> 5)
10	(10)	(9)	(6)(7)(8)	(3)	(4)							(2)(5)
11												
12												
13												
14												
15												

	Physical Register
rax	Р3
rcx	
rdi	
rsi	P4
r8	<b>P2</b>

16

	Valid	Value	In use		Valid	Value	In use
P1	1		1	P6			
P2	1		1	P7			
Р3	1		1	P8			
P4	0		1	P9			
P5				P10			

(7)(8)(9)

```
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
```

- ® movq (%rdi,%rax), %rsi → P4

  ® movq (%rcx,%rax), %r8 → P5
- 10 movq %r8, (%rdi,%rax)
- 10 movq %rsi, (%rcx,%rax)
- 12 addq \$8, %rax
- 13 cmpq %r9, %rax
- 14 jne .L9

	IF	ID	REN	AG	M1	M2	<b>M3</b>	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	<b>(7)</b>	(6)	(3)(4)(5)			(2)	(1)					
8	(8)	(7)	(3)(4)(6)				(2)	(1)	(5)			
9	(9)	(8)	(3)(6)(7)	(4)				(2)				<del>(1)(</del> 5)
10	(10)	(9)	(6)(7)(8)	(3)	(4)							<del>(2)(</del> 5)

(6)

(3) (4)

	Physical Register
rax	P3
rcx	
rdi	
rsi	P4
r8	P5

11 (11) (10)

12

13

14

15

16

	Valid	Value	In use		Valid	Value	In use
P1	1		1	P6			
P2	1		1	P7			
Р3	1		1	P8			
P4	0		1	P9			
P5	0		1	P10			

```
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
```

- ® movq (%rdi,%rax), %rsi → P4

  ® movq (%rcx,%rax), %r8 → P5
- movq %r8, (%rdi,%rax)
- ① movq %rsi, (%rcx,%rax)
- 12 addq \$8, %rax
- 13 cmpq %r9, %rax
- 14 jne .L9

						•							
		IF	ID	REN	AG	M1	M2	М3	M4	ALU	MUL	BR	ROB
•		(1)											
2	2 (	<b>(2)</b>	(1)										
3	3 (	(3)	(2)	(1)									
4	1 (	4)	(3)	(2)	(1)								
Ę	5 (	(5)	(4)	(3)	(2)	(1)							
6	6 (	<b>(6)</b>	(5)	(3)(4)		(2)	(1)						
	7 (	<b>(7)</b>	(6)	(3)(4)(5)			(2)	(1)					
8	3 (	8)	<b>(7)</b>	(3)(4)(6)				(2)	(1)	(5)			
Ş	9 (	9)	(8)	(3)(6)(7)	(4)				(2)				<del>(1)(</del> 5)
1	0 (	10)	(9)	(6)(7)(8)	(3)	(4)							<del>-(2)(</del> 5)
1	1 (	11)	(10)	(7)(8)(9)		(3)	<b>(4)</b>			(6)			
1	2 (	12)	(11)	(8)(9)(10)			(3)	(4)				(7)	(5)(6)
1	3												
1	4												
1	5												
1	6												

	Physical Register
rax	Р3
rcx	
rdi	
rsi	P4
r8	<b>P5</b>

	Valid	Value	In use		Valid	Value	In use
P1	1		1	P6			
P2	1		1	P7			
Р3	1		1	P8			
P4	0		1	P9			
P5	0		1	P10			

```
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
```

- ® movq (%rdi,%rax), %rsi → P4

  ® 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	М3	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	<b>(7)</b>	(6)	(3)(4)(5)			(2)	(1)					
8	(8)	<b>(7)</b>	(3)(4)(6)				(2)	(1)	(5)			
9	(9)	(8)	(3)(6)(7)	(4)				(2)				<del>(1)(</del> 5)
10	(10)	(9)	(6)(7)(8)	(3)	<b>(4)</b>							<del>(2)(</del> 5)
11	(11)	(10)	(7)(8)(9)		(3)	(4)			(6)			
12	(12)	(11)	(8)(9)(10)			(3)	<b>(4)</b>				(7)	(5)(6)
13	(13)	(12)	(9)(10)(11)	(8)			(3)	(4)				(5)(6)(7)
14												
15												

Physical Register
P3
P4
<b>P5</b>

16

	Valid	Value	In use		Valid	Value	In use
P1	1		1	P6			
P2	1		1	P7			
Р3	1		1	P8			
P4	0		1	P9			
P5	0		1	P10			

```
movq (%rdi,%rax), %rsi → P1
movq (%rcx,%rax), %r8 \rightarrow P2
movq %r8, (%rdi,%rax)
movq %rsi, (%rcx,%rax)
addq $8, %rax
cmpq %r9, %rax
jne .L9
movq (%rdi,%rax), %rsi → P4
movq (%rcx,%rax), %r8 \rightarrow P5
movq %r8, (%rdi,%rax)
movq %rsi, (%rcx,%rax)
addq $8, %rax
                        → P6
cmpq %r9, %rax
```

jne .L9

	IF	ID	REN	AG	M1	M2	М3	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	<b>(7)</b>	(6)	(3)(4)(5)			(2)	(1)					
8	(8)	<b>(7)</b>	(3)(4)(6)				(2)	(1)	(5)			
9	(9)	(8)	(3)(6)(7)	(4)				(2)				<del>(1)(</del> 5)
10	(10)	(9)	(6)(7)(8)	(3)	(4)							<del>-(2)(</del> 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
P6
P4
P5

	Valid	Value	In use		Valid	Value	In use
P1	1		1	P6	0		1
P2	1		1	P7			
Р3	1		1	P8			
P4	0		1	P9			
P5	0		1	P10			

```
movq (%rdi,%rax), %rsi → P1
movq (%rcx,%rax), %r8 \rightarrow P2
movq %r8, (%rdi,%rax)
movq %rsi, (%rcx,%rax)
addq $8, %rax
cmpq %r9, %rax
jne .L9
movq (%rdi,%rax), %rsi → P4
movq (%rcx,%rax), %r8 \rightarrow P5
movq %r8, (%rdi,%rax)
movq %rsi, (%rcx,%rax)
addq $8, %rax
                        → P6
cmpq %r9, %rax
```

jne .L9

	IF	ID	REN	AG	M1	M2	М3	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	<b>(7)</b>	(6)	(3)(4)(5)			(2)	(1)					
8	(8)	<b>(7)</b>	(3)(4)(6)				(2)	(1)	(5)			
9	(9)	(8)	(3)(6)(7)	(4)				(2)				<del>(1)(</del> 5)
10	(10)	(9)	(6)(7)(8)	(3)	(4)							<del>-(2)(</del> 5)
11	(11)	(10)	(7)(8)(9)		(3)	<b>(4)</b>			(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
P6
P4
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			

```
movq (%rdi,%rax), %rsi → P1
movq (%rcx,%rax), %r8 \rightarrow P2
movq %r8, (%rdi,%rax)
movq %rsi, (%rcx,%rax)
addq $8, %rax
cmpq %r9, %rax
jne .L9
movq (%rdi,%rax), %rsi → P4
movq (%rcx,%rax), %r8 \rightarrow P5
movq %r8, (%rdi,%rax)
movq %rsi, (%rcx,%rax)
addq $8, %rax
                        → P6
cmpq %r9, %rax
jne .L9
```

	IF	ID	REN	AG	M1	M2	М3	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	<b>(7)</b>	(6)	(3)(4)(5)			(2)	(1)					
8	(8)	<b>(7)</b>	(3)(4)(6)				(2)	(1)	(5)			
9	(9)	(8)	(3)(6)(7)	(4)				(2)				<del>(1)(</del> 5)
10	(10)	(9)	(6)(7)(8)	(3)	(4)							<del>-(2)(</del> 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			
Р3	1		1	P8			
P4	0		1	P9			
P5	0		1	P10			

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

			Only 1 of	tne	m c	an	nav	ea	Instru	iction a	t the	same cycle
	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	<b>(7)</b>	(6)	(3)(4)(5)			(2)	(1)					
8	(8)	<b>(7)</b>	(3)(4)(6)				(2)	(1)	(5)			
9	(9)	(8)	(3)(6)(7)	(4)				(2)				<del>-(1)(</del> 5)
10	(10)	(9)	(6)(7)(8)	(3)	(4)							<del>(2)(</del> 5)
11	(11)	(10)	(7)(8)(9)		(3)	(4)			(6)			
12	(12)	(11)	(8)(9)(10)			(3)	<b>(4)</b>				(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)			(2)(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												
19												
20												
21												
22												
00				A .								

		•	
1	movq	(%rdi,%rax), %rsi →	<b>P1</b>
2	movq	(%rcx,%rax), %r8 →	<b>P2</b>
3	movq	%r8, (%rdi,%rax)	
4	movq	%rsi, (%rcx,%rax)	
<b>5</b>	addq	\$8, %rax →	<b>P3</b>
6	cmpq	%r9, %rax	
7	jne	.L9	
8	movq	(%rdi,%rax), %rsi→	<b>P4</b>
9	movq	(%rcx,%rax), %r8 →	<b>P5</b>
10	movq	%r8, (%rdi,%rax)	
1	movq	%rsi, (%rcx,%rax)	
12	addq	\$8, %rax →	<b>P6</b>
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	
<b>21</b>	jne	.L9	

			Only 1 of	tne	m c	an	nav	ea	Instru	iction a	t the	same cycle
	IF	ID	REN	AG	M1	M2	М3	M4	ALU	MUL	ЬR	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	<b>(7)</b>	(6)	(3)(4)(5)			(2)	(1)					
8	(8)	<b>(7)</b>	(3)(4)(6)				(2)	(1)	(5)			
9	(9)	(8)	(3)(6)(7)	(4)				(2)				<del>-(1)(</del> 5)
10	(10)	(9)	(6)(7)(8)	(3)	(4)							<del>(2)(</del> 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)			<del>(3)(4)(5)(6)(7)</del>
16	(16)	(15)	(10) (11) (14)			(9)	(8)		(13)			(12)
<b>17</b>	(17)	(16)	(10) (11) (15)				(9)	(8)			(14)	(12)(13)
18	(18)	(17)	(10) (15) (16)	(11)				(9)				(8)(12)(13)(14)
19												
20												
21												
22												
22												

1	movq	(%rdi,%rax), %rsi →	<b>P1</b>
2	movq	(%rcx,%rax), %r8 →	<b>P2</b>
3	movq	%r8, (%rdi,%rax)	
4	movq	%rsi, (%rcx,%rax)	
5	addq	\$8, %rax →	<b>P3</b>
6	cmpq	%r9, %rax	
7	jne	.L9	
8	movq	(%rdi,%rax), %rsi →	<b>P4</b>
9	movq	(%rcx,%rax), %r8 →	<b>P5</b>
10	movq	%r8, (%rdi,%rax)	
11	movq	%rsi, (%rcx,%rax)	
12	addq	\$8, %rax →	<b>P6</b>
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	
<b>(21)</b>	jne	.L9	

			Only 1 of	tne	m c	an I	nav	ea	Instru	iction a	ttne	same cycle
	IF	ID	REN	AG	M1	M2	М3	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	<b>(7)</b>	(6)	(3)(4)(5)			(2)	(1)					
8	(8)	<b>(7)</b>	(3)(4)(6)				(2)	(1)	(5)			
9	(9)	(8)	(3)(6)(7)	(4)				(2)				<del>-(1)(</del> 5)
10	(10)	(9)	(6)(7)(8)	(3)	(4)							<del>-(2)</del> (5)
11	(11)	(10)	(7)(8)(9)		(3)	(4)			(6)			
12	(12)	(11)	(8)(9)(10)			(3)	<b>(4)</b>				(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)			(2)(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)				<del>(8)</del> (12)(13)(14)
19	(19)	(18)	(15)(16)(17)	(10)	(11)							(9)(12)(13)(14)
20												
21												
22												
22												

		•	
1	movq	(%rdi,%rax), %rsi →	<b>P1</b>
2	movq	(%rcx,%rax), %r8 →	<b>P2</b>
3	movq	%r8, (%rdi,%rax)	
4	movq	%rsi, (%rcx,%rax)	
5	addq	\$8, %rax →	<b>P3</b>
6	cmpq	%r9, %rax	
7	jne	.L9	
8	movq	(%rdi,%rax), %rsi →	<b>P4</b>
9	movq	(%rcx,%rax), %r8 →	<b>P5</b>
10	movq	%r8, (%rdi,%rax)	
1	movq	%rsi, (%rcx,%rax)	
12	addq	\$8, %rax →	<b>P6</b>
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	
<b>21</b> )	jne	.L9	

			Only 1 of	the	m c	an I	hav	ea	instru	uction a	t the	same cycle
	IF	ID	REN	AG	M1	M2	М3	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	<b>(7)</b>	(6)	(3)(4)(5)			(2)	(1)					
8	(8)	<b>(7)</b>	(3)(4)(6)				(2)	(1)	(5)			
9	(9)	(8)	(3)(6)(7)	(4)				(2)				<del>-(1)(</del> 5)
10	(10)	(9)	(6)(7)(8)	(3)	<b>(4)</b>							<del>(2)(</del> 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)			(2)(4)(5)(6)(7)
16	(16)	(15)	(10)(11)(14)			(9)	(8)		(13)			(12)
<b>17</b>	(17)	(16)	(10) (11) (15)				(9)	(8)			(14)	(12)(13)
18	(18)	(17)	(10) (15) (16)	(11)				(9)				<del>(8)(</del> 12)(13)(14)
19	(19)	(18)	(15)(16)(17)	(10)	(11)							<del>(9)(</del> 12)(13)(14)
20	(20)	(19)	(16) (17) (18)	(15)	(10)	(11)						(12)(13)(14)
21												
22												

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

			Only 1 of	the	m c	an	hav	ea	instru	uction a	t the	same cycle
	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	<b>(7)</b>	(6)	(3)(4)(5)			(2)	(1)					
8	(8)	<b>(7)</b>	(3)(4)(6)				(2)	(1)	(5)			
9	(9)	(8)	(3)(6)(7)	(4)				(2)				<del>-(1)(</del> 5)
10	(10)	(9)	(6)(7)(8)	(3)	(4)							<del>-(2)(</del> 5)
11	(11)	(10)	(7)(8)(9)		(3)	(4)			(6)			
12	(12)	(11)	(8)(9)(10)			(3)	(4)				<b>(7)</b>	(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)			(2)(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)				<del>(8)(</del> 12)(13)(14)
19	(19)	(18)	(15) (16) (17)	(10)	(11)							<del>(9)(</del> 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												

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 →	<b>P3</b>
6	cmpq	%r9, %rax	
7	jne	.L9	
8	movq	(%rdi,%rax), %rsi →	<b>P4</b>
9	movq	(%rcx,%rax), %r8 →	P5
10	movq	%r8, (%rdi,%rax)	
1	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	
<b>21</b>	jne	.L9	

			Only 1 of	the	m c	an	hav	ea	instr	uction at	the	same cycle
	IF	ID	REN	AG	M1	M2	М3	M4	ALU	MUL	ЬR	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	<b>(7)</b>	(6)	(3)(4)(5)			(2)	(1)					
8	(8)	<b>(7)</b>	(3)(4)(6)				(2)	(1)	(5)			
9	(9)	(8)	(3)(6)(7)	(4)				(2)				<del>-(1)(</del> 5)
10	(10)	(9)	(6)(7)(8)	(3)	(4)							<del>-(2)(</del> 5)
11	(11)	(10)	(7)(8)(9)		(3)	(4)			(6)			
12	(12)	(11)	(8)(9)(10)			(3)	(4)				<b>(7)</b>	(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)			(2)(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)				<del>(8)(</del> 12)(13)(14)
19	(19)	(18)	(15) (16) (17)	(10)	(11)							<del>(9)</del> (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)	(19)			(12)(13)(14)
00												

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

			•						liav	<del>С</del> а					
			Ke		F ID	REN	AG	M1	M2	М3	M4	ALU	MUL	ЬR	ROB
1	mova	(%rdi,%rax), %rsi		1 (	1)										
2		(%rcx,%rax), %r8			2) (1)										
3	•	%r8, (%rdi,%rax)	<i>,</i> , ,	3 (			(4)								
	-	· · · · · · · · · · · · · · · · · · ·		4 (			(1)	141							
4	•	%rsi, (%rcx,%rax)	<b>N</b> DO	5 (			(2)	(1)							
5	addq	\$8, %rax	→ P3	6 (				(2)	(1)	(4)					
6	cmpq	%r9, %rax			7) (6				(2)		(4)	<b>(C</b> )			
7	jne	.L9			8) (7 <sub>)</sub>		(4)			(2)		(5)			(4) (5)
8	mova	(%rdi,%rax), %rsi	→ P4	•	9) (8		(4)	(4)			(2)				<del>(1)(</del> 5)
9	-	(%rcx,%rax), %r8			(0) (9 (1) (10		(3)	(4)				<b>(6)</b>			<del>-(2)(</del> 5)
_	•	·	<i>,</i> L2					(3)	(4)	(4)		(6)		(7)	(E)(G)
10	•	%r8, (%rdi,%rax)		_	2) (11  2) (12		(0)		(3)	(4)	(4)			(7)	(5)(6) (5)(6)(7)
$\underbrace{11}_{\bigcirc}$	•	%rsi, (%rcx,%rax)			(12) (13)		(8)	(0)		(3)					(5)(6)(7)
(12)	addq	\$8, %rax	→ P6		4) (13  5) (1 <i>4</i>		(9)	(8)			(3)	(12)			(4)(5)(6)(7)
13	cmpq	%r9, %rax			5) (14  6) (15			(9)	(8)	(9)		(12)			(12)
14	jne	.L9			17) (16				(9)	(8) (9)	(8)	(13)		(14)	(12)(13)
15	_	(%rdi,%rax), %rsi			8) (17		(11)			(3)	(9)			(14)	(12)(13) (8)(12)(13)(14)
16	_	(%rcx,%rax), %r8			9) (18			(11)			(5)				<del>(9)(</del> 12)(13)(14)
	•	•			20) (19				(11)						(12)(13)(14)
17	•	%r8, (%rdi,%rax)			21) (20				(10)						(12)(13)(14)
18	movq	%rsi, (%rcx,%rax)		22	(21		(10)				(11)	(19)			(12)(13)(14)
19	addq	\$8, %rax		23		(17)(20)(21)	(18)			(15)		(10)			(11)(12)(13)(14)(19)
20	cmpq	%r9, %rax		24			(10)		()	(:-)	( · • )				,
<b>21</b> )	jne	.L9		25											

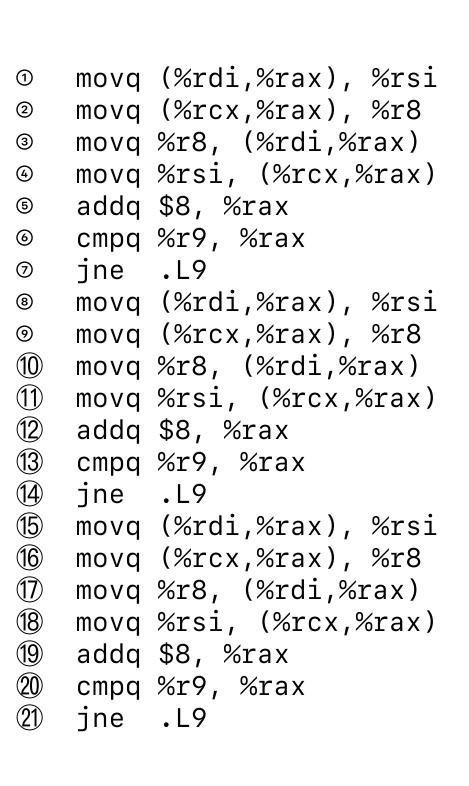
Only 1 of them can have a instruction at the same cycle

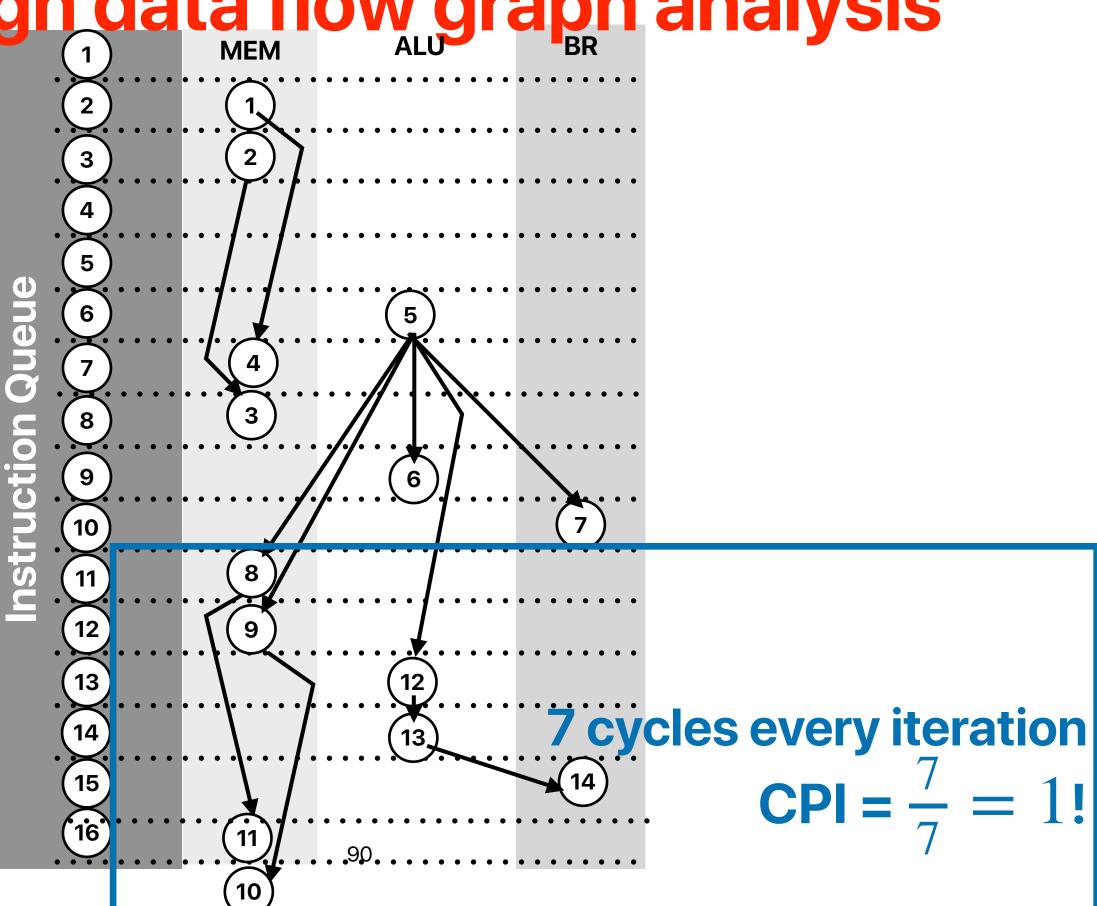
							· · · ·	•		<b>-</b>				
		Ke		F ID	REN	AG	M1	M2	МЗ	M4	ALU	MUL	ЬR	ROB
1	movq (%rdi,%rax), %rsi-	• D1	1 (1											
2	movq (%rcx,%rax), %r8 -		2 (2											
3	movq (%16X,%1dX), %10 movq %r8, (%rdi,%rax)	<i>,</i> , ,	3 (3		(1)	(4)								
	•		4 (4		(2)	(1)	(4)							
4	movq %rsi, (%rcx,%rax)	<b>N</b> D2	5 (5		(3)	(2)	(1)	(4)						
<b>5</b>		<b>→</b> P3	6 (6		(3)(4)		(2)		(1)					
6	cmpq %r9, %rax		7 (7 8 (8		(3)(4)(5)			(2)		(1)	<b>(5)</b>			
7	jne .L9		9 (9		(3)(4)(6) (3)(6)(7)	(4)			(2)	(2)	(5)			<del>-(1)(</del> 5)
8	movq (%rdi,%rax), %rsi	<b>&gt;</b> P4		0) (9)	(6)(7)(8)		(4)			(2)				<del>(1)</del> (5) <del>-(2)(</del> 5)
9	movq (%rcx,%rax), %r8 -	→ P5		1) (10)		(0)		(4)			(6)			(2)(0)
10	movq %r8, (%rdi,%rax)			2) (11)	(8)(9)(10)		(-)		(4)		(0)		(7)	(5)(6)
$\underbrace{11}$	movq %rsi, (%rcx,%rax)			3) (12)		(8)			(3)	(4)			,	(5)(6)(7)
12		> P6		4) (13)			(8)			(3)				(4)(5)(6)(7)
_	• •	7 PO	15 (1	5) (14)	(10)(11)(13)		(9)	(8)			(12)			(2)(4)(5)(6)(7)
13	cmpq %r9, %rax		16 (1	6) (15)	(10)(11)(14)			(9)	(8)		(13)			(12)
_	jne .L9		17 (1	7) (16)	(10)(11)(15)				(9)	(8)			(14)	(12)(13)
15	movq (%rdi,%rax), %rsi		18 (1	8) (17)	(10)(15)(16)	(11)				(9)				<del>(8)</del> (12)(13)(14)
16	movq (%rcx,%rax), %r8		19 (1	9) (18)	(15)(16)(17)	(10)	(11)							<del>(9)(</del> 12)(13)(14)
17)	<pre>movq %r8, (%rdi,%rax)</pre>		20 (2	0) (19)		(15)	(10)	(11)						(12)(13)(14)
18	movq %rsi, (%rcx,%rax)			1) (20)		(16)		(10)						(12)(13)(14)
19	addq \$8, %rax		22	(21)	(17)(18)(20)	4.5.5					(19)			(12)(13)(14)
20	cmpq %r9, %rax		23		(17)(20)(21)	(18)			(15)					(11)(12)(13)(14)(19)
_			24		(20)(21)	(17)	(18)		(16)	(15)				(10)(11)(12)(13)(14)(19)
(21)	ine .L9		25											

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

				•	_	Only 1 of	tne	m c	an i	nav	e a		ion at ti	1e s L	same cycle
			Rec	IF	ID	REN	AG	M1	M2	M3	M4	ALU N	MUL B	R	ROB
1	movq	(%rdi,%rax), %rsi	→ P1	1 (1)											
2		(%rcx,%rax), %r8		2 (2) 3 (3)	(1) (2)	(1)									
3	movq	%r8, (%rdi,%rax)		4 (4)	(3)	(2)	(1)								
4	movq	%rsi, (%rcx,%rax)		5 (5)	(4)	(3)		(1)							
<b>5</b>	addq	\$8, %rax	<b>P3</b>	6 (6)	(5)	(3)(4)		(2)	(1)						
6	cmpq	%r9, %rax		7 (7)	(6)	(3)(4)(5)			(2)						
7	jne	.L9		8 (8)	(7)	(3)(4)(6)				<b>(2)</b>	(1)	(5)			44.45
8		(%rdi,%rax), %rsi	<b>P4</b>	9 (9)	(8)	(3)(6)(7)	(4)	(4)			(2)				<del>-(1)(</del> 5)
9	•	(%rcx,%rax), %r8		10 (10)		(6)(7)(8)	(3)	(4)	(4)			(0)			<del>-(2)(</del> 5)
_	-	-	PJ	11 (11)		(7)(8)(9)		(3)	(4)	(4)		(6)	<i>(</i> -	<b>7</b> \	(E)(G)
10	•	%r8, (%rdi,%rax)		12 (12)		(8)(9)(10)	(0)		(3)		(4)			7)	(5)(6) (5)(6)(7)
$\underbrace{11}_{\widehat{\Omega}}$	•	%rsi, (%rcx,%rax)		13 (13)		(9)(10)(11)	(8)	(0)		(3)					(5)(6)(7)
(12)	addq	\$8, %rax	<b>P6</b>	14 (14)		(10)(11)(12)	(9)	(8)	(0)		(3)	(12)			(4)(5)(6)(7)
13	cmpq	%r9, %rax		15 (15)		(10)(11)(13)		(9)	(8)	(0)		(12)			(0)(-)(0)(0)(/)
14	jne	.L9		16 (16) 17 (17)		(10)(11)(14) (10)(11)(15)			(9)		(8)	(13)	(1	<b>/</b> 1)	(12) (12)(13)
15	_	(%rdi,%rax), %rsi		18 (18)		(10)(11)(13)	(11)				(8) (9)			<b>4)</b>	(12)(13) (12)(13)(14)
_	•	(%rcx,%rax), %r8		19 (19)			(11) (10)				(3)	7 cycle	es tor		<del>(9)(</del> 12)(13)(14)
<b>16</b>	•			20 (20)					(11)			instru	ctions	3	(12)(13)(14)
<u>17</u>	•	%r8, (%rdi,%rax)		21 (21)											(12)(13)(14)
18	movq	%rsi, (%rcx,%rax)		22	(21)	(17)(18)(20)	(10)	(16)	(15)	(10)	(11)	(19) <sup>CP</sup>	I = 1		(12)(13)(14)
19	addq	\$8, %rax		23	(21)		(18)	(10)	(16)	(15)	(10)	(10)			(11)(12)(13)(14)(19)
20	cmpq	%r9, %rax		24		(20)(21)		(18)			(15)				<del>(10)(11)(12)(13)(14)(</del> 19)
<u>21</u>	jne	.L9		25		(21)	()		(18)			(20)			(15)(19)

Through data flow graph analysis



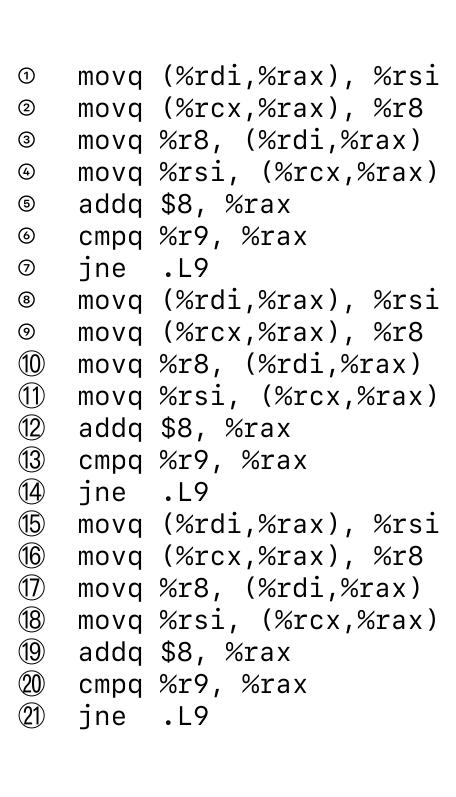


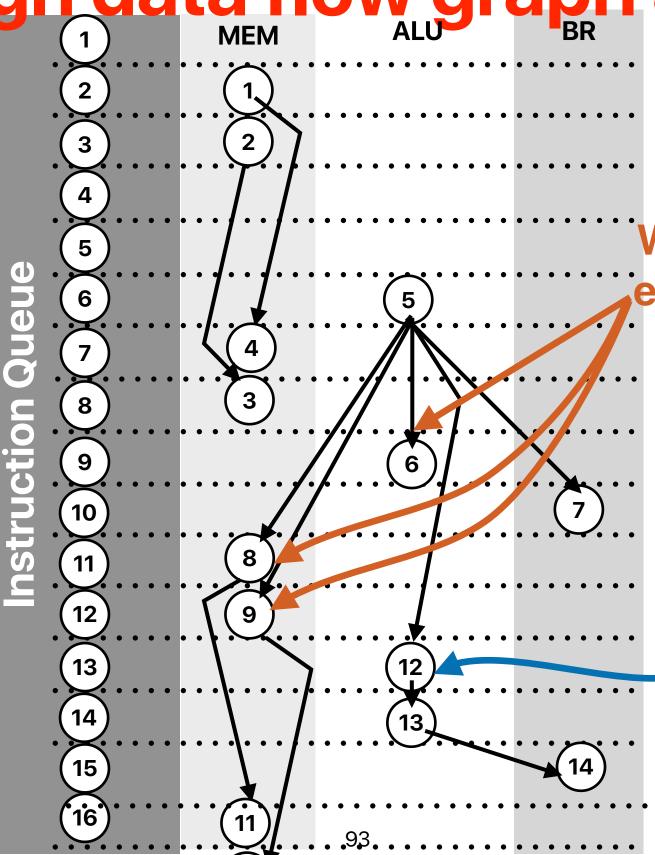
#### Takeaways: data hazards

- 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





10

We cannot issue them earlier simply because structural hazards!

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

# Super Scalar

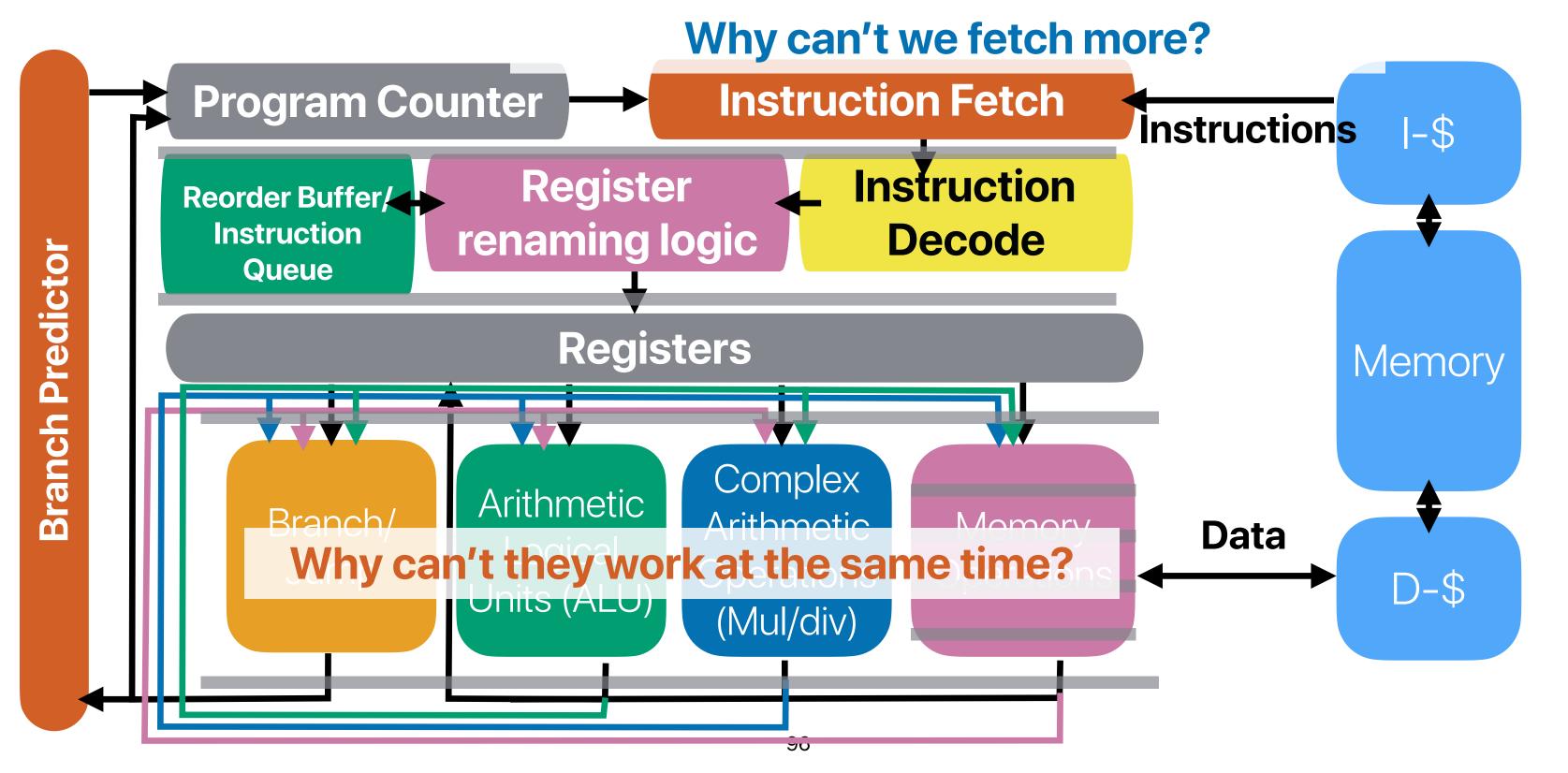
#### Superscalar

- 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

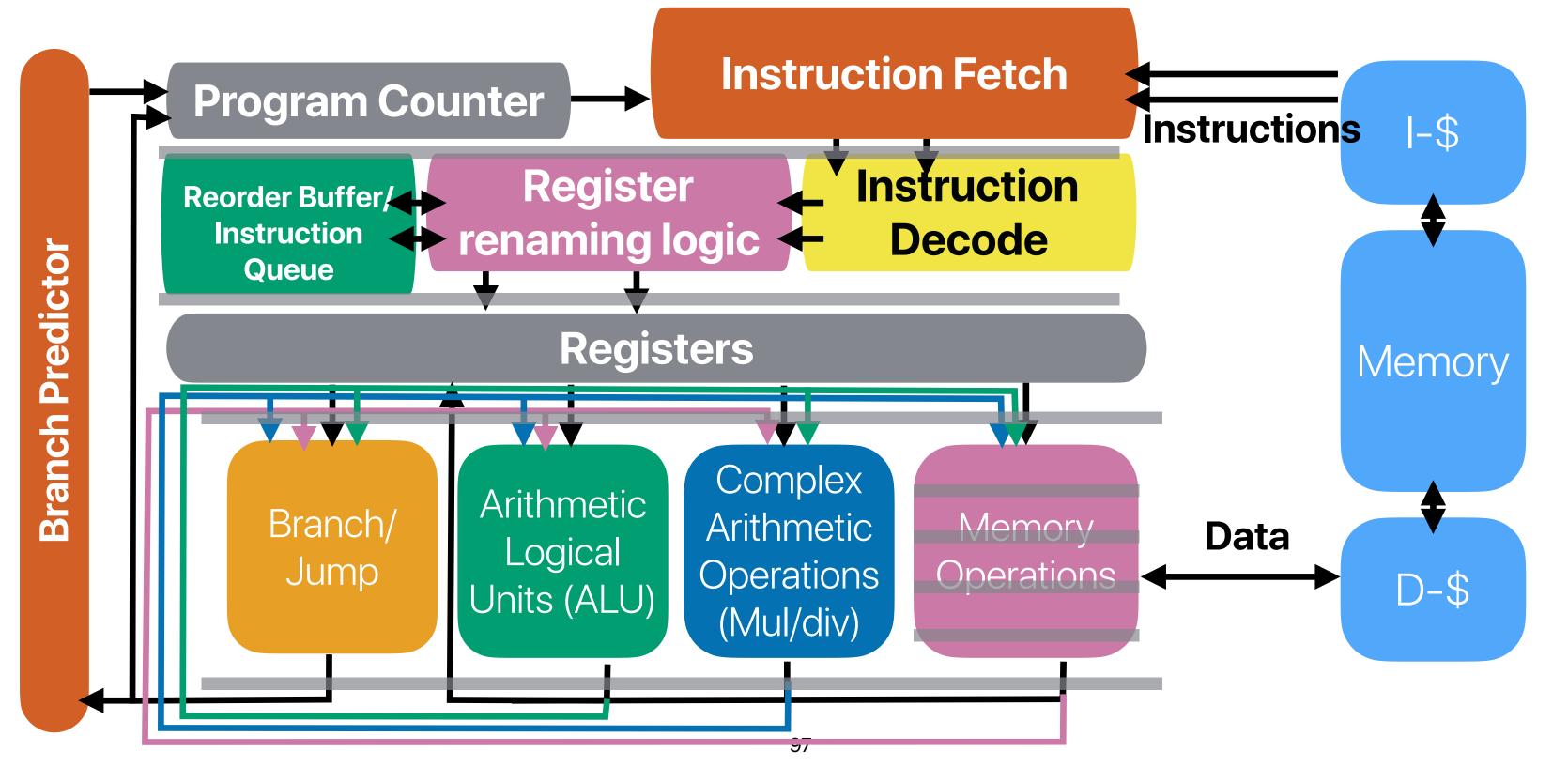
1

min(issue width, fetch width, decode width)

## Register renaming + OoO + RoB



## Register renaming + SuperScalar



1	movq	(%rdi,%rax), %rsi	<ul><li>P1</li></ul>				issue:	4		ınem	Can	ınav	eal	nstruc	tion at	the same cy	cie
2	movq	(%rcx,%rax), %r8	▶ P2		IF	ID	REN	AG	M	1 M2	МЗ	M4	ALU	MUL	BR	ROB	
3	movq	%r8, (%rdi,%rax)		1	(1)(2)												
4	movq	%rsi, (%rcx,%rax)			(3)(4)												
5	addq	\$8, %rax	<b>P3</b>	3	(5)(6)	(3)(4)	(1)(2)										
6	cmpq	%r9, %rax		4													
7	jne			5													
8	movq	(%rdi,%rax), %rsi	<b>P</b> 4	7													
9	movq	(%rcx,%rax), %r8	<b>P5</b>	8													
10	movq	%r8, (%rdi,%rax)		9													
11	movq	%rsi, (%rcx,%rax)		10													
12	addq	\$8, %rax	▶ P6	11													
13	cmpq	%r9, %rax		12													
14	jne	.L9		13													
15	movq	(%rdi,%rax), %rsi		14													
16	movq	(%rcx,%rax), %r8		15													
		%r8, (%rdi,%rax)		16													
18	movq	%rsi, (%rcx,%rax)		17													
19	addq	\$8, %rax		18 19													
20	cmpq	%r9, %rax		20													
(21)	ine	. L9					98										

1	movq	(%rdi,%rax), %rsi	> P1				issue:	4	or then	ı car	ınav	eal	nstruc	tion at	the same cycle
2	movq	(%rcx,%rax), %r8	→ P2		IF	ID	REN	AG	M1 M2	М3	M4	ALU	MUL	BR	ROB
3	movq	%r8, (%rdi,%rax)		1	(1)(2)										
4	movq	%rsi, (%rcx,%rax)			(3)(4)										
<b>⑤</b>	addq	\$8, %rax	→ P3				(1)(2)								
6	cmpq	%r9, %rax		4	(7)(8)	(5)(6)	(2)(3)(4)	(1)							
7	jne	.L9		5											
8	movq	(%rdi,%rax), %rsi	→ P4	6											
9	movq	(%rcx,%rax), %r8	→ P5	2											
		%r8, (%rdi,%rax)		9											
11	movq	%rsi, (%rcx,%rax)		10											
12	addq	\$8, %rax	→ P6	11											
13	cmpq	%r9, %rax		12											
14	jne	.L9		13											
15	movq	(%rdi,%rax), %rsi		14											
16	movq	(%rcx,%rax), %r8		15											
17)	movq	%r8, (%rdi,%rax)		16											
18	movq	%rsi, (%rcx,%rax)		17											
19	addq	\$8, %rax		18 19											
20	cmpq	%r9, %rax		20											
(21)	ine	<b>.</b> L 9					99								

1	movq	(%rdi,%rax), %rsi	→ P1	ı			13346.	1	or ti		Cai	HIGN				tric same cy	CIC
2	movq	(%rcx,%rax), %r8	→ P2		IF	ID	REN	AG	M1	M2	М3	M4	ALU	MUL	BR	ROB	
3	movq	%r8, (%rdi,%rax)		1	(1)(2)												
4	movq	%rsi, (%rcx,%rax)			(3)(4)												
(5)	addq	\$8, %rax	→ P3				(1)(2)										
6	cmpq	%r9, %rax			(7)(8)		(2)(3)(4)	(1)									
7	jne	.L9			(9)(10)	(7)(8)	(3)(4)(5)(6)	(2)	(1)								
8	movq	(%rdi,%rax), %rsi	→ P4	5													
9	movq	(%rcx,%rax), %r8	→ P5	2													
		%r8, (%rdi,%rax)		9													
11	movq	%rsi, (%rcx,%rax)		10													
12	addq	\$8, %rax	→ P6	11													
13	cmpq	%r9, %rax		12													
14	jne	.L9		13													
15	movq	(%rdi,%rax), %rsi		14													
16	movq	(%rcx,%rax), %r8		15													
17	movq	%r8, (%rdi,%rax)		16													
18	movq	%rsi, (%rcx,%rax)		17													
19	addq	\$8, %rax		18 19													
		%r9, %rax		20													
<b>(21)</b>	ine	.L9		20			100										

1	movq	(%rdi,%rax), %rsi	→ P1	ı			13346.	4	oi ti		Cai	HIGN			CIOTICAL	tric same cy	CIC
2	movq	(%rcx,%rax), %r8	→ P2		IF	ID	REN	AG	M1	M2	МЗ	M4	ALU	MUL	BR	ROB	
3	movq	%r8, (%rdi,%rax)		1	(1)(2)												
4	movq	%rsi, (%rcx,%rax)			(3)(4)												
<b>⑤</b>	addq	\$8, %rax	→ P3														
6	cmpq	%r9, %rax			(7)(8)			(1)									
7	jne						(3)(4)(5)(6)	(2)		(4)							
8	movq	(%rdi,%rax), %rsi	→ P4	6	(11)(12)	(9)(10)	(3)(4)((6)(7)(8)		(2)	(1)							
9	movq	(%rcx,%rax), %r8	→ P5	9													
		%r8, (%rdi,%rax)		9													
11	movq	%rsi, (%rcx,%rax)		10													
12	addq	\$8, %rax	→ P6	11													
13	cmpq	%r9, %rax		12													
14	jne	.L9		13													
15	movq	(%rdi,%rax), %rsi		14													
16	movq	(%rcx,%rax), %r8		15													
17)	movq	%r8, (%rdi,%rax)		16													
18	movq	%rsi, (%rcx,%rax)		17													
19	addq	\$8, %rax		18 19													
		%r9, %rax		20													
	ine			20			101										

1	movq	(%rdi,%rax), %rsi	<ul><li>P1</li></ul>				. 10040.	7			- Cai	iiiav				ine dame by	
2	movq	(%rcx,%rax), %r8	P2		IF	ID	REN	AG	M1	M2	МЗ	M4	ALU	MUL	BR	ROB	
3	movq	%r8, (%rdi,%rax)		1	(1)(2)												
4	movq	%rsi, (%rcx,%rax)			(3)(4)												
<b>(5)</b>	addq	\$8, %rax	• P3														
6	cmpq	%r9, %rax				(5)(6)		(1)									
	jne					(7)(8)		(2)		141							
8	movq	(%rdi,%rax), %rsi	<b>P4</b>	-			(3)(4)((6)(7)(8)		(2)		(1)		<b>(E)</b>			<b>(5)</b>	
9	movq	(%rcx,%rax), %r8	<b>P5</b>	2	(13)(14)	(11)(12)	(3)(4)((6)(7)(9) (10)	(0)		(2)	(1)		(5)			(5)	
		%r8, (%rdi,%rax)		9													
11	movq	%rsi, (%rcx,%rax)		10													
12	addq	\$8, %rax	<b>P6</b>	11													
13	cmpq	%r9, %rax		12													
14	jne	.L9		13													
15	movq	(%rdi,%rax), %rsi		14													
16	movq	(%rcx,%rax), %r8		15													
17)	movq	%r8, (%rdi,%rax)		16													
18	movq	%rsi, (%rcx,%rax)		17													
19	addq	\$8, %rax		18													
		%r9, %rax		19 20													
	ine			20			102										

1	movq	(%rdi,%rax), %rsi	$\rightarrow$ P	1			. 155ue.	4	oi ti	lem	Cai	IIIav	ear	ristinic	tion at	tile Saiii	e Cycle
2	movq	(%rcx,%rax), %r8	$\rightarrow$ P	2	IF	ID	REN	AG	M1	M2	МЗ	M4	ALU	MUL	BR	ROB	
3	movq	%r8, (%rdi,%rax)		1	(1)(2)												
4	movq	%rsi, (%rcx,%rax)			(3)(4)												
(5)	addq	\$8, %rax	→ P		(5)(6)												
6	cmpq	%r9, %rax			(7)(8)			(1)									
7	jne		_		(9)(10)			(2)		(4)							
8	movq	(%rdi,%rax), %rsi	$\rightarrow$ P	<b>T</b>			(3)(4)((6)(7)(8)		(2)		(1)		<b>(</b> E)			<i>(</i> E)	
9	movq	(%rcx,%rax), %r8	$\rightarrow$ P	5 <b>1</b>			(3)(4)((6)(7)(9) (10) (3)(4)(7)(10)(11) (12)			(2)		(1)	(5) (6)			(5) (5)(6	
		%r8, (%rdi,%rax)		9	(10)(10)	(15)(1.1)	(12)	(3)	(0)		(2)	(1)	(0)			(3)(0	
11	movq	%rsi, (%rcx,%rax)		10													
12	addq	\$8, %rax	$\rightarrow$ P	6 11													
13	cmpq	%r9, %rax		12	2												
14	jne	.L9		13													
15	movq	(%rdi,%rax), %rsi		14	l e												
16	movq	(%rcx,%rax), %r8		15													
<u>17</u> )	movq	%r8, (%rdi,%rax)		16													
18	movq	%rsi, (%rcx,%rax)		17													
19	addq	\$8, %rax		18 19													
20	cmpq	%r9, %rax		20													
<b>21</b> )	jne	.L9					103										

1	movq	(%rdi,%rax), %rsi	→ P1	•			. 10040.	1			Odi					ile saille cycl
2	movq	(%rcx,%rax), %r8	→ P2		IF	ID	REN	AG	M1	M2	МЗ	M4	ALU	MUL	BR	ROB
3	movq	%r8, (%rdi,%rax)		1	(1)(2)											
4	movq	%rsi, (%rcx,%rax)			(3)(4)											
<b>⑤</b>	addq	\$8, %rax	→ P3				(1)(2)									
6	cmpq	%r9, %rax			(7)(8)			(1)								
7	jne	.L9					(3)(4)(5)(6)	(2)								
8	movq	(%rdi,%rax), %rsi	$\rightarrow$ P <sup>2</sup>				(3)(4)((6)(7)(8)		(2)		(4)		<b>(C</b> )			
9	movq	(%rcx,%rax), %r8	→ P5				(3)(4)((6)(7)(9) (10) (3)(4)(7)(10)(11)			(2)		(1)	(5)			(5)
		%r8, (%rdi,%rax)					(3)(4)(7)(10)(11) (12) (3)(10)(11)(12)				(2)		(6)		(7)	(5)(6) (1)(5)(6)
_	-	%rsi, (%rcx,%rax)		10	(17)(10)	(10)(10)	(3)(10)(11)(12) (13)(14)	(4)	(9)	(0)		(2)			(7)	(1)(5)(6)
_	-	•	→ P6													
	-	%r9, %rax		12												
_	jne	-		13												
_		(%rdi,%rax), %rsi		14												
_	-	(%rcx,%rax), %r8		15												
_	<del>-</del>	%r8, (%rdi,%rax)		16												
	<del>-</del>	%rsi, (%rcx,%rax)		17												
		\$8, %rax		18												
		%r9, %rax		19												
	ine			20			104									

1	movq	(%rdi,%rax), %rsi	<b>&gt;</b>	<b>P1</b>			2	' issue: "	4	of th	nem	car	n hav	еап	nstruc	tion at i	the same cycle
2	movq	(%rcx,%rax), %r8	$\rightarrow$	<b>P2</b>		IF	ID	REN	AG	M1	M2	МЗ	M4	ALU	MUL	BR	ROB
3		%r8, (%rdi,%rax)				(1)(2)											
4	movq	%rsi, (%rcx,%rax)			2	(3)(4)	(1)(2)										
5	addq	\$8, %rax	$\rightarrow$	<b>P3</b>	3	(5)(6)	(3)(4)	(1)(2)									
6	<del>-</del>	%r9, %rax				(7)(8)			(1)								
7	jne	•				(9)(10)			(2)								
8	_	(%rdi,%rax), %rsi	$\rightarrow$	<b>P4</b>				(3)(4)((6)(7)(8)			(1)						
9	mova	(%rcx,%rax), %r8	<b>&gt;</b>	P5		(13)(14)		(10)			(2)		(4)	(5)			(5)
10		%r8, (%rdi,%rax)						(3)(4)(7)(10)(11) (12) (3)(10)(11)(12)			(0)	(2)	(1)	(6)		(7)	(5)(6) (1)(5)(6)
_	•	%rsi, (%rcx,%rax)						(3)(10)(11)(12) (13)(14) (10)(11)(13)(14)			(8) (9)	(8)	(2)	(12)		(7)	(1)(5)(6) (2)(5)(6)(7)
	•	\$8, %rax	$\rightarrow$	P6	11	(10)(20)	(17)(10)	(10)(11)(13)(14) (15)(16)	(3)	(4)	(9)	(0)		(12)			(2)(3)(0)(7)
_	•	%r9, %rax			12												
_	jne	-			13												
_	_	(%rdi,%rax), %rsi			14												
_	•	(%rcx,%rax), %r8			15												
_	•	%r8, (%rdi,%rax)			16												
_	•	%rsi, (%rcx,%rax)			17												
	_	\$8, %rax			18												
	-	%r9, %rax			19												
	ine				20			105									

1	movq	(%rdi,%rax), %rsi	→ P1					7								
2	movq	(%rcx,%rax), %r8	→ P2		IF	ID	REN	AG	M1	M2	МЗ	M4	ALU	MUL	BR	ROB
3	movq	%r8, (%rdi,%rax)		1	(1)(2)											
4	movq	%rsi, (%rcx,%rax)			(3)(4)											
(5)	addq	\$8, %rax	→ P3													
6	cmpq	%r9, %rax			(7)(8)			(1)								
7	jne	.L9			(9)(10)			(2)								
8	movq	(%rdi,%rax), %rsi	→ P4				(3)(4)((6)(7)(8)		(2)		(4)		<b>(C</b> )			<b>(E)</b>
9	movq	(%rcx,%rax), %r8	→ P5				(3)(4)((6)(7)(9) (10) (3)(4)(7)(10)(11)			(2)		(1)	(5) (6)			(5) (5)(6)
10		%r8, (%rdi,%rax)					(3)(4)(7)(10)(11) (12) (3)(10)(11)(12)		(8) (9)	(8)	(2)	(1) (2)	(6)		(7)	(5)(6) (1)(5)(6)
$\widetilde{11}$	•	%rsi, (%rcx,%rax)					(3)(10)(11)(12) (13)(14) (10)(11)(13)(14) (15)(16)		(4)		(8)	(2)	(12)		(7)	(2)(5)(6)(7)
	-		→ P6	11	(21)(22)	(19)(20)	(15)(16) (10)(11)(14)(16) (17)(18)	(15)				(8)	(12)			(5)(6)(7)(12)
	•	%r9, %rax		12			(17)(18)	,	(0)	( - /	(0)	(0)	(10)			(0)(0)(1)(1-)
_	jne	-		13												
_	_	(%rdi,%rax), %rsi		14												
	-	(%rcx,%rax), %r8		15												
_	=	%r8, (%rdi,%rax)		16												
	-	%rsi, (%rcx,%rax)		17												
		\$8, %rax		18												
		%r9, %rax		19												
	jne			20			106									

1	movq	(%rdi,%rax), %rsi	$\rightarrow$ P	1				7								
2	movq	(%rcx,%rax), %r8	$\rightarrow$ P	2	IF	ID	REN	AG	M1	M2	МЗ	M4	ALU	MUL	BR	ROB
3	movq	%r8, (%rdi,%rax)		1	(1)(2)											
4	movq	%rsi, (%rcx,%rax)	_	2	2 (3)(4)	(1)(2)										
<b>5</b>	addq	\$8, %rax	→ P		(5)(6)											
6	cmpq	%r9, %rax			(7)(8)			(1)								
7	jne	.L9			(9)(10)			(2)								
8	movq	(%rdi,%rax), %rsi	$\rightarrow$ P	_			(3)(4)((6)(7)(8)		(2)		(4)		<b>(C</b> )			<b>(E)</b>
9	movq	(%rcx,%rax), %r8	$\rightarrow$ P	5 <b>E</b>			(3)(4)((6)(7)(9) (10) (3)(4)(7)(10)(11)			(2)		(1)	(5) (6)			(5) (5)(6)
_		%r8, (%rdi,%rax)					(3)(4)(7)(10)(11) (12) (3)(10)(11)(12) (13)(14)		(8) (9)	(8)	(2)	(2)	(6)		(7)	(5)(6) (1)(5)(6)
_	•	%rsi, (%rcx,%rax)					(13)(14) (10)(11)(13)(14) (15)(16)		(4)		(8)	(2)	(12)		(7)	(2)(5)(6)(7)
_	•	· · · · · · · · · · · · · · · · · · ·	$\rightarrow$ P				(15)(16) (10)(11)(14)(16) (17)(18)				(9)	(8)	(12)			(5)(6)(7)(12)
	•	%r9, %rax			2		(17)(18) (16)(17)(18)(19) (20						(10)		(14)	(5)(6)(7)(8)(12)(13)
_	jne	•		1	3		(20									
_	_	(%rdi,%rax), %rsi		1	4											
_	-	(%rcx,%rax), %r8		1	5											
	_	%r8, (%rdi,%rax)		1	6											
_	-	%rsi, (%rcx,%rax)		1												
		\$8, %rax		1												
	=	%r9, %rax		1												
<u>21</u> )				2	0		107									

1	movq	(%rdi,%rax), %rsi	$\rightarrow$	<b>P1</b>			_	. 155UC.	4	Ji ti	IEIII	Cai	IIIav		istique	tion at	the same cycli
2	movq	(%rcx,%rax), %r8	$\rightarrow$	<b>P2</b>		IF	ID	REN	AG	M1	M2	М3	M4	ALU	MUL	BR	ROB
3	movq	%r8, (%rdi,%rax)			1	(1)(2)											
4	movq	%rsi, (%rcx,%rax)				(3)(4)											
<b>⑤</b>	addq	\$8, %rax	$\rightarrow$	<b>P3</b>	3	(5)(6)	(3)(4)	(1)(2)									
6	cmpq	%r9, %rax			4	(7)(8)		(2)(3)(4)	(1)								
7	jne	.L9				(9)(10)		(3)(4)(5)(6)	(2)								
8	movq	(%rdi,%rax), %rsi	$\rightarrow$	<b>P4</b>				(3)(4)((6)(7)(8)		(2)		(4)		<b>(</b> 5)			
9	movq	(%rcx,%rax), %r8	<b>&gt;</b>	<b>P5</b>				(3)(4)((6)(7)(9) (10) (3)(4)(7)(10)(11)			(2)		(1)	(5)			(5)
10		%r8, (%rdi,%rax)				(17)(18)		(3)(4)(7)(10)(11) (12) (3)(10)(11)(12)		(8)	(9)	(2)	(1)	(6)		(7)	(5)(6) (1)(5)(6)
$\widetilde{11}$	•	%rsi, (%rcx,%rax)						(3)(10)(11)(12) (13)(14) (10)(11)(13)(14) (15)(16)		(9) (4)	(9)	(8)	(2)	(12)		(7)	(1)(5)(6) (2)(5)(6)(7)
_	•	•	$\rightarrow$	P6	11	(21)(22)		(15)(16) (10)(11)(14)(16) (17)(18)			(4)		(8)	(12)			(5)(6)(7)(12)
13	•	%r9, %rax			12			(17) (18) (16) (17) (18) (19) (20			(3)			(10)		(14)	(5)(6)(7)(8)(12)(13)
14	jne	•			13			(20 (16)(17)(18)(20) (21)(22)					(4)	(19)		( /	(5)(6)(7)(8)(9)(12)(13)(14)
15		(%rdi,%rax), %rsi			14			(21)(22)									
_	•	(%rcx,%rax), %r8			15												
_	=	%r8, (%rdi,%rax)			16												
		%rsi, (%rcx,%rax)			17												
		\$8, %rax			18												
		%r9, %rax			19												
<u>20</u>	jne				20			108									

1	movq	(%rdi,%rax), %rsi	$\rightarrow$	<b>P1</b>			_	. 133UC.	4	or ti	ICIII	Cari	IIIav		istruc	tion a	r the same cycle
2	movq	(%rcx,%rax), %r8	$\rightarrow$	<b>P2</b>		IF	ID	REN	AG	M1	M2	М3	M4	ALU	MUL	BR	ROB
3	movq	%r8, (%rdi,%rax)			1	(1)(2)											
4	movq	%rsi, (%rcx,%rax)				(3)(4)											
<b>5</b>	addq	\$8, %rax	$\rightarrow$	<b>P3</b>	3	(5)(6)	(3)(4)	(1)(2)									
6	cmpq	%r9, %rax				(7)(8)		(2)(3)(4)	(1)								
7	jne	.L9				(9)(10)		(3)(4)(5)(6)	(2)								
8	movq	(%rdi,%rax), %rsi	$\rightarrow$	<b>P4</b>				(3)(4)((6)(7)(8)		(2)		(4)		<b>(</b> E)			<b>(5)</b>
9	movq	(%rcx,%rax), %r8	$\rightarrow$	<b>P5</b>				(3)(4)((6)(7)(9) (10) (3)(4)(7)(10)(11)			(2)		(1)	(5)			(5)
10		%r8, (%rdi,%rax)				(17)(18)		(3)(4)(7)(10)(11) (12) (3)(10)(11)(12)		(8) (9)	(8)	(2)	(1) (2)	(6)		(7)	(5)(6) (1)(5)(6)
$\widetilde{11}$	•	%rsi, (%rcx,%rax)				(19)(20)		(13)(14)			(9)	(8)	(2)	(12)		(7)	(2)(5)(6)(7)
	•	\$8, %rax	$\rightarrow$	<b>P6</b>	11	(21)(22)		(15)(16) (10)(11)(14)(16) (17)(18)			(4)		(8)	(13)			(5)(6)(7)(12)
_	•	%r9, %rax			12						(3)		(9)	(10)		(14)	(5)(6)(7)(8)(12)(13)
		.L9			13			(16)(17)(18)(20) (21)(22)			(15)		(4)	(19)			(5)(6)(7)(8)(9)(12)(13)(14)
<u>(15)</u>		(%rdi,%rax), %rsi			14			(=:)(==)				(15)	(3)	(20)			(4)(5)(6)(7)(8)(9)(12)(13)(14) (19)
_	•	(%rcx,%rax), %r8			15												
	-	%r8, (%rdi,%rax)			16												
	<del>-</del>	%rsi, (%rcx,%rax)			17												
	•	\$8, %rax			18												
		%r9, %rax			19												
<u>21</u> )	_	.L9			20			109									

1	movq	(%rdi,%rax), %rsi	$\rightarrow$ P	1			. 10040.	7			oai	····				
2	movq	(%rcx,%rax), %r8	$\rightarrow$ P	2	IF	ID	REN	AG	M1	M2	МЗ	M4	ALU	MUL	BR	ROB
3	movq	%r8, (%rdi,%rax)		1	(1)(2)											
4	movq	%rsi, (%rcx,%rax)	_	_	(3)(4)	(1)(2)										
<b>⑤</b>	addq	\$8, %rax	$\rightarrow$ P	3	(5)(6)	(3)(4)	(1)(2)									
6	cmpq	%r9, %rax		4		(5)(6)	(2)(3)(4)	(1)								
7	jne	.L9		5			(3)(4)(5)(6)	(2)		(4)						
8	movq	(%rdi,%rax), %rsi	$\rightarrow$ P	_	(11)(12)		(3)(4)((6)(7)(8)		(2)	(1)	(4)		<b>(C</b> )			<b>(5)</b>
9	movq	(%rcx,%rax), %r8	$\rightarrow$ P	5 6	(13)(14)	(11)(12)	(3)(4)((6)(7)(9) (10) (3)(4)(7)(10)(11)	(0)	(0)	(2)		(1)	(5) (6)			(5) (5)(6)
10		%r8, (%rdi,%rax)		9			(3)(4)(7)(10)(11) (12) (3)(10)(11)(12) (13)(14)		(8) (9)	(8)	(2)	(1) (2)	(6)		(7)	(5)(6) (1)(5)(6)
(11)	movq	%rsi, (%rcx,%rax)		10		(17)(18)	(13)(14) (10)(11)(13)(14) (15)(16)			(9)	(8)	(2)	(12)		(7)	(2)(5)(6)(7)
12	-	•	$\rightarrow$ P			(19)(20)	(15)(16) (10)(11)(14)(16) (17)(18)			(4)		(8)	(13)			(5)(6)(7)(12)
13	•	%r9, %rax		12		(21)(22)	(17)(18) (16)(17)(18)(19) (20			(3)			()		(14)	(5)(6)(7)(8)(12)(13)
14	jne	.L9		13	3		(16)(17)(18)(20) (21)(22)			(15)		(4)	(19)			(5)(6)(7)(8)(9)(12)(13)(14)
15		(%rdi,%rax), %rsi		14			(=:/(==/		(10)	(11)	(15)	(3)	(20)			(4)(5)(6)(7)(8)(9)(12)(13)(14) (19)
_	•	(%rcx,%rax), %r8		15	5				(16)	(10)	(11)	(15)			(21)	(3)(4)(5)(6)(7)(8)(9)(12)(13) (14)(19)(20)
	•	%r8, (%rdi,%rax)		16	5											
_		%rsi, (%rcx,%rax)		17	,											
		\$8, %rax		18												
		%r9, %rax		19												
_	jne			20			110									

1	movq	(%rdi,%rax), %rsi	$\rightarrow$ P	1		_	. 133uc.	4	or ti	ICIII	Cai	Hav		istrac	CIOILS	r tille Saille Cycle
2	movq	(%rcx,%rax), %r8	$\rightarrow$ P	2	IF	ID	REN	AG	M1	M2	МЗ	M4	ALU	MUL	BR	ROB
3	movq	%r8, (%rdi,%rax)		1	(1)(2)											
4	movq	%rsi, (%rcx,%rax)	_		(3)(4)	(1)(2)										
<b>5</b>	addq	\$8, %rax	→ P	3 3	(5)(6)	(3)(4)	(1)(2)									
6	cmpq	%r9, %rax		4	(7)(8)	(5)(6)	(2)(3)(4)	(1)								
7	jne	.L9		5	(9)(10)		(3)(4)(5)(6)	(2)		7.43						
8	movq	(%rdi,%rax), %rsi	$\rightarrow$ P				(3)(4)((6)(7)(8)		(2)		(4)		<b>(</b> E)			<b>(E)</b>
9	movq	(%rcx,%rax), %r8	$\rightarrow$ P		(13)(14)		(3)(4)((6)(7)(9) (10) (3)(4)(7)(10)(11)		(0)	(2)		(1)	(5) (6)			(5) (5)(6)
10		%r8, (%rdi,%rax)		C	(17)(18)		(12)	(0)	(8) (9)	(8)	(2)	(1) (2)	(6)		(7)	(5)(6) (1)(5)(6)
11	movq	%rsi, (%rcx,%rax)		10	(19)(20)					(9)	(8)	(2)	(12)		(7)	(2)(5)(6)(7)
12	addq	\$8, %rax	$\rightarrow$ P	_		(19)(20)	(15)(16) (10)(11)(14)(16) (17)(18)			(4)		(8)	(13)			(5)(6)(7)(12)
13	cmpq	%r9, %rax		1:	2	(21)(22)			(15)		(4)	(9)			(14)	(5)(6)(7)(8)(12)(13)
		.L9		13	3		(16)(17)(18)(20) (21)(22)			(15)	(3)	(4)	(19)			(5)(6)(7)(8)(9)(12)(13)(14)
<u>(15)</u>	movq	(%rdi,%rax), %rsi		14	1			(16)	(10)	(11)	(15)	(3)	(20)			(4)(5)(6)(7)(8)(9)(12)(13)(14) (19)
_	movq	(%rcx,%rax), %r8		1!	5				(16)	(10)	(11)	(15)			(21)	(3)(4)(5)(6)(7)(8)(9)(12)(13) (14)(19)(20)
	-	%r8, (%rdi,%rax)		16	5			(17)		(16)	(10)	(11)				(12)(13)(14)(15)(19)(20(21)
	'='	%rsi, (%rcx,%rax)		17												
_	-	\$8, %rax		18												
_	<del>-</del>	%r9, %rax		19												
<u>21</u>		.L9		20			111									

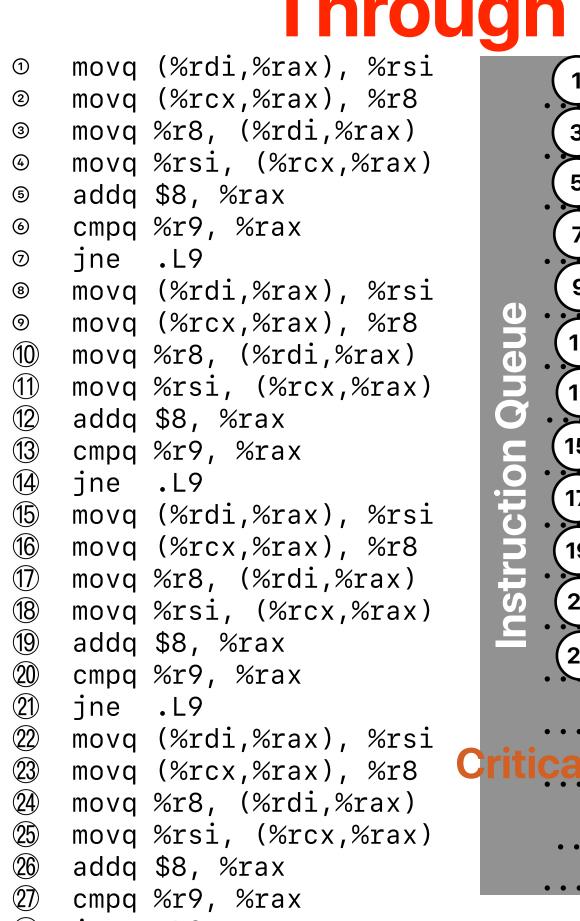
1	movq	(%rdi,%rax), %rsi	$\rightarrow$	<b>P1</b>			_	. 133UC.	4	or ti	ICIII	Cai	IIIav			LIOILA	'
2	movq	(%rcx,%rax), %r8	$\rightarrow$	<b>P2</b>		IF	ID	REN	AG	M1	M2	МЗ	M4	ALU	MUL	BR	ROB
3	movq	%r8, (%rdi,%rax)			1	(1)(2)											
4	movq	%rsi, (%rcx,%rax)				(3)(4)											
<b>5</b>	addq	\$8, %rax	$\rightarrow$	<b>P3</b>	3	(5)(6)	(3)(4)	(1)(2)									
6	cmpq	%r9, %rax				(7)(8)		(2)(3)(4)	(1)								
7		.L9				(9)(10)		(3)(4)(5)(6)	(2)		445						
8	movq	(%rdi,%rax), %rsi	$\rightarrow$	<b>P4</b>				(3)(4)((6)(7)(8)			(1)	(4)		<b>(C</b> )			<b>(F)</b>
9	movq	(%rcx,%rax), %r8	$\rightarrow$	P5				(3)(4)((6)(7)(9) (10) (3)(4)(7)(10)(11)			(2)		(1)	(5) (6)			(5) (5)(6)
10		%r8, (%rdi,%rax)				(17)(18)		(3)(4)(7)(10)(11) (12) (3)(10)(11)(12) (13)(14)		(8) (9)	(8)	(2)	(1) (2)	(6)		(7)	(5)(6) (1)(5)(6)
(1)	movq	%rsi, (%rcx,%rax)				(19)(20)		(13)(14) (10)(11)(13)(14) (15)(16)			(9)	(8)	(2)	(12)		(/)	(2)(5)(6)(7)
(12)	addq	\$8, %rax	$\rightarrow$	P6	11	(21)(22)		(15)(16) (10)(11)(14)(16) (17)(18)			(4)		(8)	(13)			(5)(6)(7)(12)
	•	%r9, %rax			12		(21)(22)	(17)(18) (16)(17)(18)(19) (20					(9)	, ,		(14)	(5)(6)(7)(8)(12)(13)
		.L9			13			(16)(17)(18)(20) (21)(22)			(15)		(4)	(19)			(5)(6)(7)(8)(9)(12)(13)(14)
<u>(15)</u>		(%rdi,%rax), %rsi			14					(10)	(11)	(15)	(3)	(20)			(4)(5)(6)(7)(8)(9)(12)(13)(14) (19)
_	•	(%rcx,%rax), %r8			15					(16)	(10)	(11)	(15)			(21)	(12)(13)(14)(19)(20)
_	_	%r8, (%rdi,%rax)			16				(17)		(16)	(10)	(11)				(3)(4)(5)(6)(7)(8)(9)(12)(13) (14)(19)(20)
_	-	%rsi, (%rcx,%rax)			17					(17)		(16)	(10)				(11)(12)(13)(14)(15)(19)(20(21)
_	•	\$8, %rax			18												
20	-	%r9, %rax			19												
<u>21</u>		.L9			20			112									

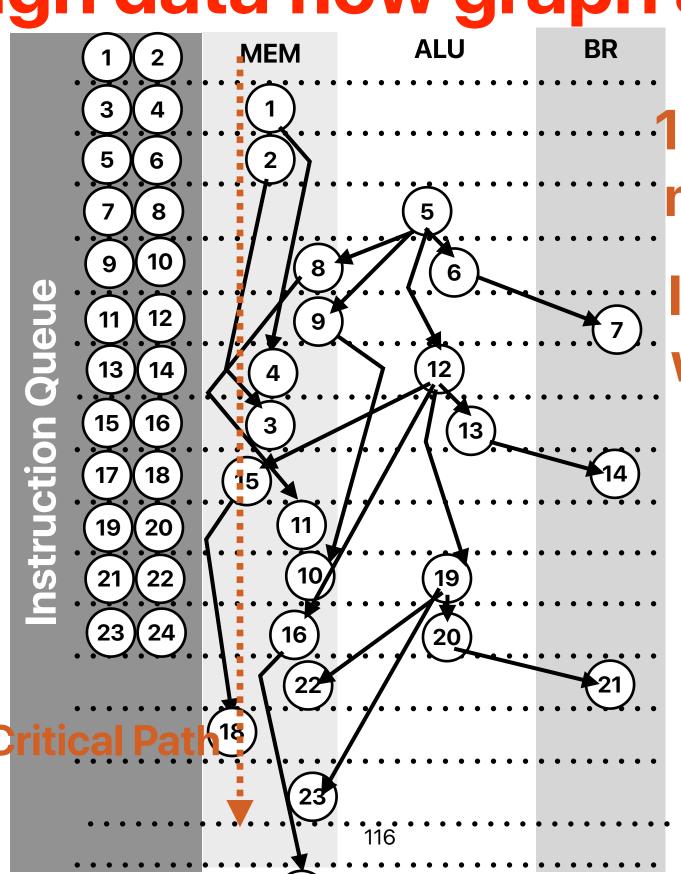
1	movq	(%rdi,%rax), %rsi	$\rightarrow$	<b>P1</b>				. 155ue.	4	oi ti	lem	Gai	IIIav	e a II	IISTILIC	tion a	' The Same Cycle
2	movq	(%rcx,%rax), %r8	$\rightarrow$	<b>P2</b>		IF	ID	REN	AG	M1	M2	МЗ	M4	ALU	MUL	BR	ROB
3	movq	%r8, (%rdi,%rax)			1	(1)(2)											
4	movq	%rsi, (%rcx,%rax)			2	(3)(4)	(1)(2)										
5	addq	\$8, %rax	$\rightarrow$	<b>P3</b>	3	(5)(6)		(1)(2)									
6	cmpq	%r9, %rax			4		(5)(6)	(2)(3)(4)	(1)								
7		.L9				(9)(10)		(3)(4)(5)(6)		(1)	(4)						
8	movq	(%rdi,%rax), %rsi	<b>→</b>	<b>P4</b>	7	(11)(12)		(3)(4)((6)(7)(8)			(1)	(1)		<b>(E)</b>			<b>(E)</b>
9	movq	(%rcx,%rax), %r8	$\rightarrow$	<b>P5</b>		(15)(14)		(3)(4)((6)(7)(9) (10) (3)(4)(7)(10)(11)		(8)	(2)	(1)	(1)	(5) (6)			(5) (5)(6)
10		%r8, (%rdi,%rax)				(17)(18)		(12) (3)(10)(11)(12) (13)(14)	(0)	(9)	(8)	(2)	(2)	(0)		(7)	(1)(5)(6)
11	movq	%rsi, (%rcx,%rax)						(13)(14) (10)(11)(13)(14) (15)(16)			(9)	(8)	(-)	(12)		(2)	(2)(5)(6)(7)
12	addq	\$8, %rax	$\rightarrow$	<b>P6</b>	11	(21)(22)		(15)(16) (10)(11)(14)(16) (17)(18)		(3)		(9)	(8)	(13)			(5)(6)(7)(12)
13	cmpq	%r9, %rax			12			(16)(17)(18)(19) (20			(3)		(9)			(14)	(5)(6)(7)(8)(12)(13)
14	jne	.L9			13			(16)(17)(18)(20) (21)(22)		(11)	(15)	(3)	(4)	(19)			(5)(6)(7)(8)(9)(12)(13)(14)
15)	movq	(%rdi,%rax), %rsi			14				(16)	(10)	(11)	(15)	(3)	(20)			(4)(5)(6)(7)(8)(9)(12)(13)(14) (19)
16	movq	(%rcx,%rax), %r8			15					(16)	(10)	(11)	(15)			(21)	(3)(4)(5)(6)(7)(8)(9)(12)(13) (14)(19)(20)
<u>17</u> )	<u>-</u>	%r8, (%rdi,%rax)			16				(17)			(10)					(12)(13)(14)(15)(19)(20(21)
18	movq	%rsi, (%rcx,%rax)			17					(17)			(10)				(11)(12)(13)(14)(15)(19)(20(21)
19	•	\$8, %rax			18						(17)		(16)				(10)(11)(12)(13)(14)(15)(19) (20)(21)
<u>20</u>	<u>-</u>	%r9, %rax			19												
<u>21</u>	jne	.L9			20			113									

1	movq	(%rdi,%rax), %rsi	→ P:	L		_	. 133uc.	4	oi ti		Cai	IIIAV			CHOILS	' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' '
2	movq	(%rcx,%rax), %r8	→ P2	2	IF	ID	REN	AG	M1	M2	МЗ	M4	ALU	MUL	BR	ROB
3	movq	%r8, (%rdi,%rax)		1	(1)(2)											
4	movq	%rsi, (%rcx,%rax)			(3)(4)											
<b>5</b>	addq	\$8, %rax	→ P:	3	(5)(6)	(3)(4)	(1)(2)									
6	cmpq	%r9, %rax		4	(7)(8)	(5)(6)	(2)(3)(4)	(1)								
7	jne	.L9			(9)(10)		(3)(4)(5)(6)		(1)							
8	movq	(%rdi,%rax), %rsi	$\rightarrow P$	•	(11)(12)					(1)	(4)		<b>(</b> E)			
9	movq	(%rcx,%rax), %r8	→ P!		(13)(14)		(3)(4)((6)(7)(9) (10) (3)(4)(7)(10)(11)				(1)	(1)	(5)			(5)
10		%r8, (%rdi,%rax)			(15)(16)		(12)		(8)		(2)		(6)		(7)	(5)(6) (1)(5)(6)
$\widetilde{11}$	•	%rsi, (%rcx,%rax)			(19)(20)		(13)(14)		(9) (4)	(8) (9)	(8)	(2)	(12)		(7)	(1)(5)(6) (2)(5)(6)(7)
~	•	•	$\rightarrow$ P(	_		(19)(20)				(4)		(8)	(13)			(5)(6)(7)(12)
13	•	%r9, %rax		12		(21)(22)		_		(3)		(9)	(10)		(14)	(5)(6)(7)(8)(12)(13)
14	jne	.L9		13			(20 (16)(17)(18)(20) (21)(22)			(15)		(4)	(19)		,	(5)(6)(7)(8)(9)(12)(13)(14)
15		(%rdi,%rax), %rsi		14			(21)(22)					(3)	(20)			(4)(5)(6)(7)(8)(9)(12)(13)(14) (19)
_	•	(%rcx,%rax), %r8		15					(16)	(10)	(11)	(15)			(21)	(3)(4)(5)(6)(7)(8)(9)(12)(13) (14)(19)(20)
17	-	%r8, (%rdi,%rax)		16				(17)		(16)	(10)	(11)				(12)(13)(14)(15)(19)(20(21)
18	=	%rsi, (%rcx,%rax)		17					(17)		(16)	(10)				(11)(12)(13)(14)(15)(19)(20(21)
19	•	\$8, %rax		18						(17)		(16)				(10)(11)(12)(13)(14)(15)(19) (20)(21)
20	-	%r9, %rax		19				(18)			(17)					(16)(19)(20)(21)
<u>21</u>	jne	.L9		20			114									

1	movq	(%rdi,%rax), %rsi	→ P1				. 155ue.	4	oi u	iem	Cai	IIIav	e a li	ristique	tiona	' '' '' Same Cycle
2	movq	(%rcx,%rax), %r8	→ P2		IF	ID	REN	AG	M1	M2	МЗ	M4	ALU	MUL	BR	ROB
3	movq	%r8, (%rdi,%rax)		1	(1)(2)											
4	movq	%rsi, (%rcx,%rax)			(3)(4)											
5	addq	\$8, %rax	→ P3	3			(1)(2)									
6	cmpq	%r9, %rax		4	(7)(8)		(2)(3)(4)	(1)								
7		.L9			(9)(10)		(3)(4)(5)(6)		(1)	(4)						
8	movq	(%rdi,%rax), %rsi	→ P4	-	(11)(12)					(1)	(1)		<b>(</b> E)			<b>(E)</b>
9	movq	(%rcx,%rax), %r8	→ P5				(3)(4)((6)(7)(9) (10) (3)(4)(7)(10)(11) (12)		(8)	(2)	(1) (2)	(1)	(5) (6)			(5) (5)(6)
10		%r8, (%rdi,%rax)			(17)(18)				(9)	(8)	(2)	(2)	(0)		(7)	<del>(1)</del> (5)(6)
11	movq	%rsi, (%rcx,%rax)			(19)(20)					(9)	(8)	(-/	(12)		(*)	<del>(2)</del> (5)(6)(7)
12	addq	\$8, %rax	→ P6			(19)(20)	(15)(16) (10)(11)(14)(16) (17)(18)		(3)		(9)	(8)	(13)			(5)(6)(7)(12)
13	cmpq	%r9, %rax		12		(21)(22)	(16)(17)(18)(19) (20			(3)		(9)			(14)	(5)(6)(7)(8)(12)(13)
14	jne	.L9		13			(16)(17)(18)(20) (21)(22)		(11)	(15)	(3)	(4)	(19)			(5)(6)(7)(8)(9)(12)(13)(14)
15	movq	(%rdi,%rax), %rsi		14					(10)	(11)	(15)	(3)	(20)			(4)(5)(6)(7)(8)(9)(12)(13)(14) (19)
16	movq	(%rcx,%rax), %r8		15					(16)	(10)	(11)	(15)			(21)	(2)(4)(5)(6)(7)(9)(9)(12)(13) (14)(19)(20)
17	<del>-</del>	%r8, (%rdi,%rax)		16				(17)		(16)	(10)	(11)				(12)(13)(14)(15)(19)(20(21)
18	-	%rsi, (%rcx,%rax)		17					(17)			(10)				(11)(12)(13)(14)(15)(19)(20(21)
19	-	\$8, %rax		18				(40)		(17)		(16)				(10)(11)(12)(13)(14)(15)(19) (20)(21)
<u>20</u>	<del>-</del>	%r9, %rax		19				(18)			(17)	(17)				<del>(16)</del> (19)(20)(21)
<u>21</u>	jne	.L9		20			115		(18)			(17)				(19)(20)(21)

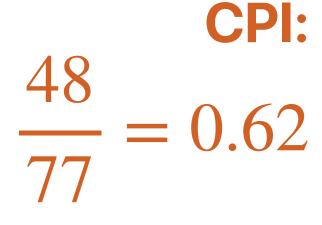
Through data flow graph analysis





12 cycles for every 11memory instructions

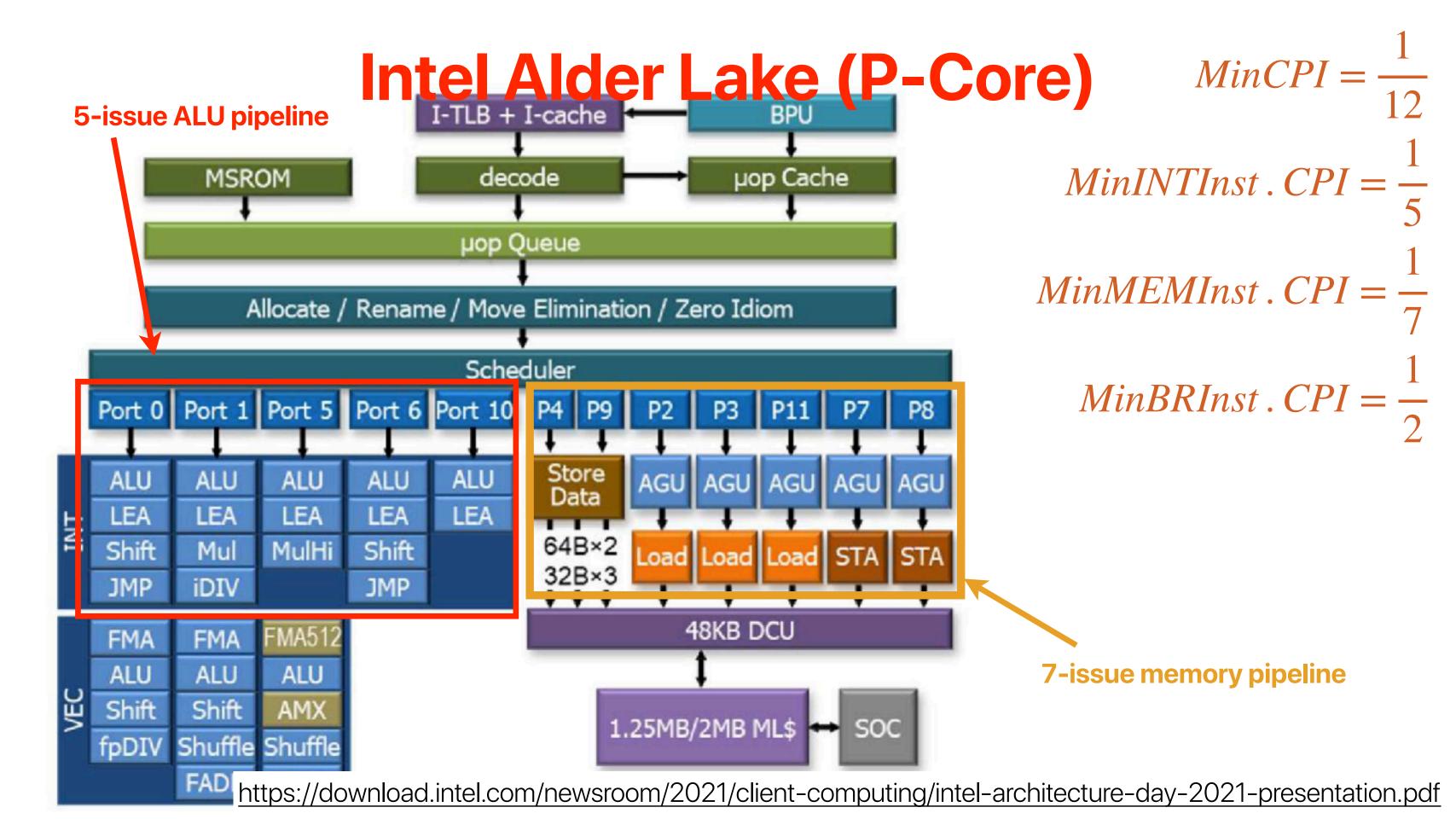
If we have 11 loops, it will have 44 memory instructions, 77 instructions in total and take 48 cycles



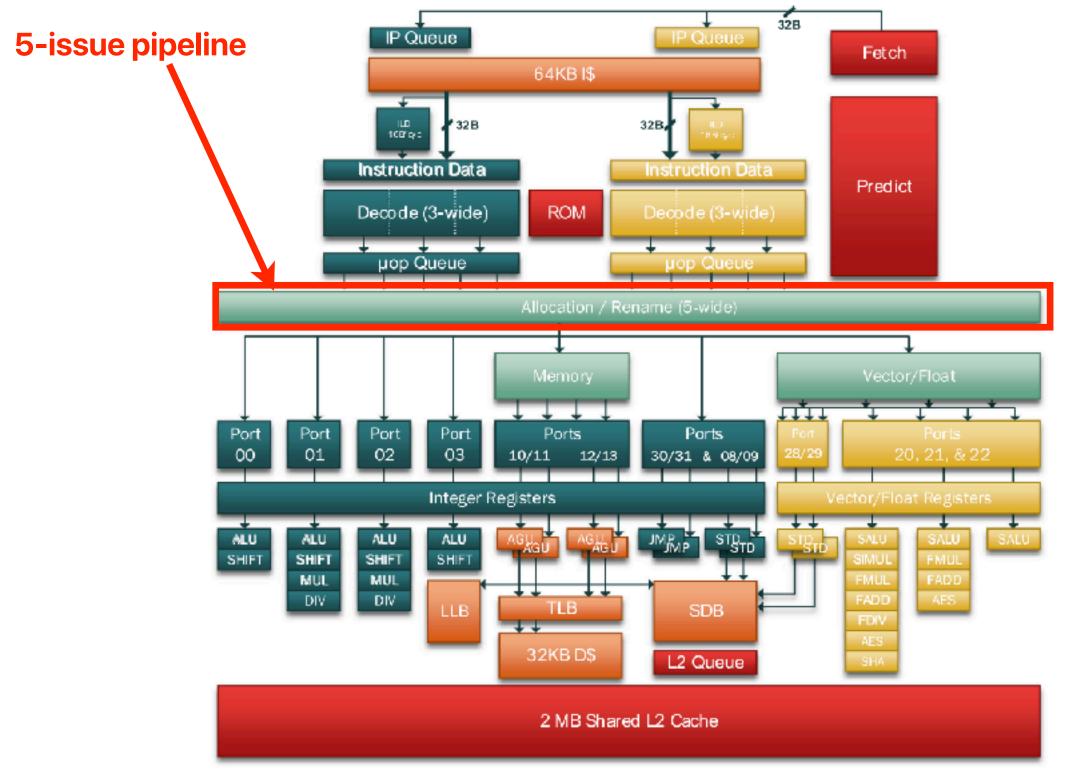
## Takeaways: data hazards

- 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

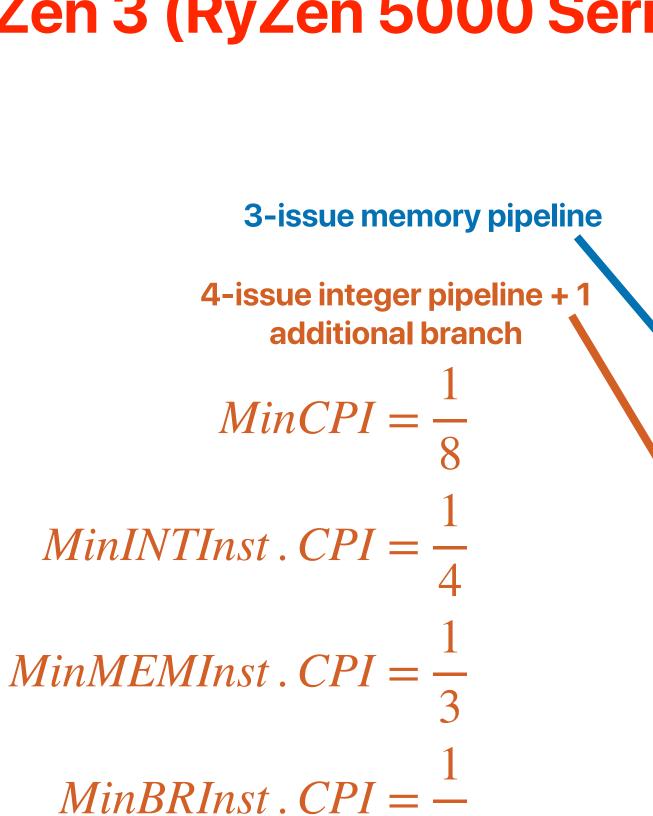
# The pipelines of Modern Processors

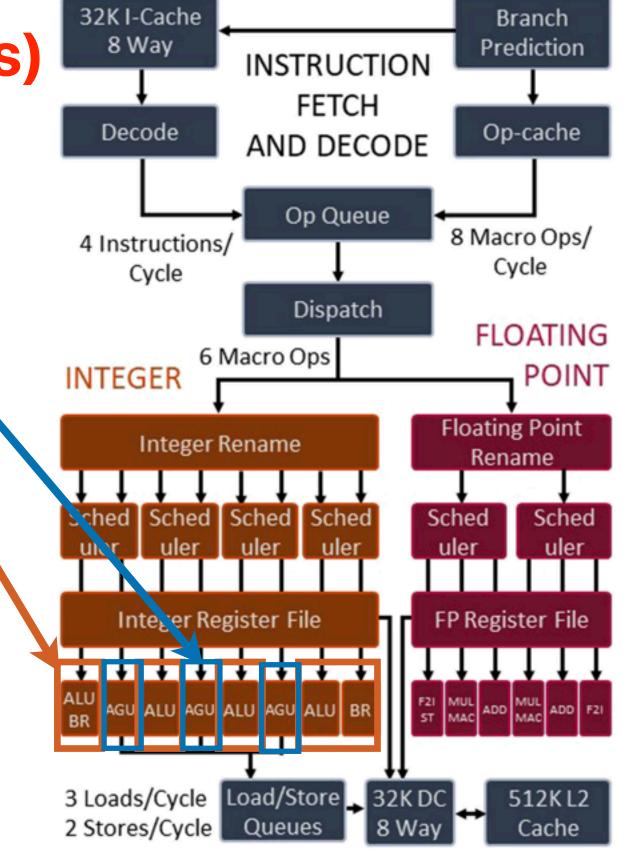


## Intel Alder Lake (E-Core)



### AMD Zen 3 (RyZen 5000 Series)





#### 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