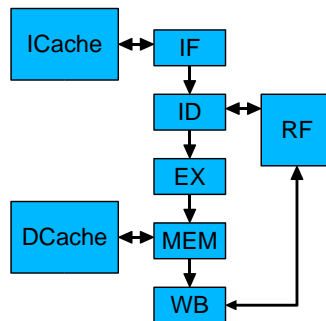


# Lec-05: Speculative Execution and Register Renaming

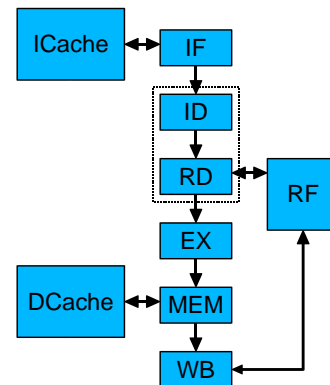
Innovations in Processor Design  
ECE-720-topic4  
2008t1 (Winter)

Mark Aagaard

## Simple Pipelines

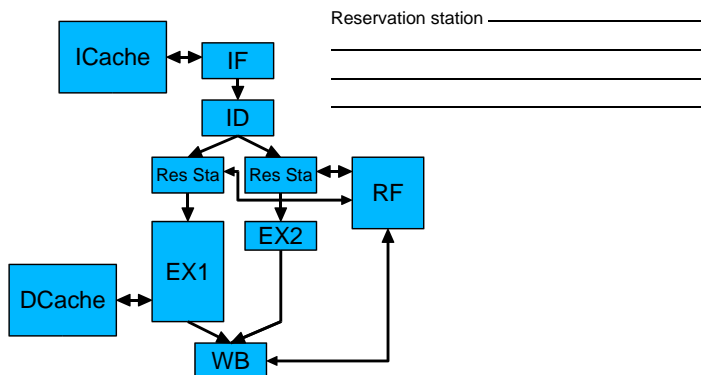


Simple 5-stage pipeline



Split ID into ID + RD

## Multi-Cycle, Parallel Execution



Reservation station \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

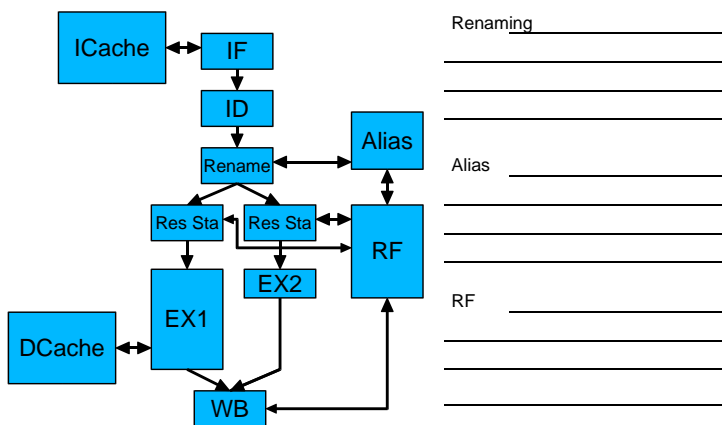
Q: Evaluate design from perspective of performance

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

## Data-Hazard Correctness

- Two factors for data-hazard correctness
  - Producer-consumer: all producer-consumer dependencies in the ISA (assembly code) must be obeyed in the processor
    - RAW: consumer must read after producer writes
    - WAW: earlier producers must write before current producer
    - WAR: later producers must write after consumer reads
  - Snapshot / exception / flushing correctness: if processor flushes (completes) all in-flight instructions, the processor must be in a coherent state with respect to the ISA.

## Register Renaming



Renaming \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Alias \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

RF \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

## Register Renaming Hardware

- Alias table: maps names of architected registers to names of physical registers
- Register file: physical registers
- Reservation station: instructions sit in reservation stations waiting for operands to become valid.

Alias phys	
R1	
R2	
R3	

RF mode		val
P1		
P2		
P3		
P4		

## Register Renaming: A Trivial Example

$\alpha$  R1 = #15  
 $\beta$  R2 = R1 + #2

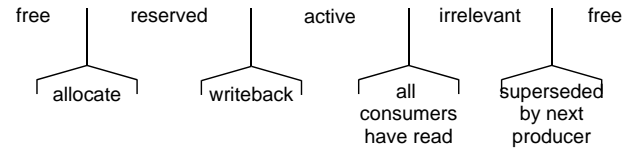
Alias phys		RF mode val	
R1		P1	
R2		P2	
R3		P3	
		P4	

ID										
Ren										
RS1										
RS2										
EX1										
EX2										
WB										
R1										
R2										
P1 (mode,val)										
P2 (mode, val)										
P3 (mode, val)										
P4 (mode, val)										

7

## Modes for Physical Registers

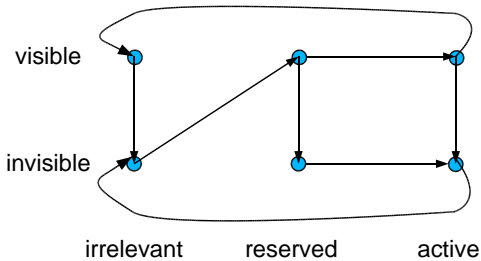
- Free:** physical register is unused; it does not represent any architectural register
- Reserved:** physical register is in use; it represents an architectural register, but the instruction that will write to the register is still in flight.
- Active:** physical register is in use and contains valid data
- Irrelevant:** the data in the physical register is no longer needed, because all consuming instructions have read the data.



8

## Register Renaming

- Data dependencies restrict parallelism, reduce performance
- RAW is only **true** data dependency
- Register renaming gets rid of WAR and WAW data dependencies, which exposes more instruction-level parallelism



9

## Rename Algorithm (1 of 2)

- Rename source index  
 $\text{psrc\_idx} = \text{Alias}[\text{src\_idx}]$
- Set pdst of previous producer to be invisible  
 $\text{RF}[\text{Alias}[\text{dst\_idx}]].\text{visible} = \text{False}$
- Pick physical register for destination  
 $\text{pdst\_idx} = \dots$
- Wait until new pdst is invisible and irrelevant  
 wait until  $(\text{RF}[\text{pdst\_idx}].\text{mode} = \text{irrelevant})$   
 and  $\text{RF}[\text{pdst\_idx}].\text{visible} = \text{False}$
- Set pdst to be visible and reserved  
 $\text{RF}[\text{pdst\_idx}].\text{visible} = \text{True}$   
 $\text{RF}[\text{pdst\_idx}].\text{mode} = \text{reserved}$

10

## Rename Algorithm (2 of 2)

- Update alias table  
 $\text{Alias}[\text{dst\_idx}] = \text{pdst\_idx}$
- Read source values  
 wait until  $\text{RF}[\text{psrc\_idx}].\text{mode} = \text{active}$   
 $\text{src\_val} = \text{RF}[\text{psrc\_idx}].\text{value}$   
 if *last-consumer-to-read*  
 then  $\text{RF}[\text{psrc\_idx}].\text{mode} = \text{irrelevant}$
- Compute result  
 $\text{dst\_val} = \text{func}(\text{src\_val})$
- Write result  
 $\text{RF}[\text{pdst\_idx}].\text{value} = \text{dst\_val}$   
 $\text{RF}[\text{pdst\_idx}].\text{mode} = \text{active}$

11

## Register Renaming: A Simple Example

$\alpha$  R1 = sqrt #16 \_\_\_\_\_  
 $\beta$  R2 = R1 + #2 \_\_\_\_\_  
 $\gamma$  R1 = #95 \_\_\_\_\_  
 $\delta$  R3 = R1 -- #3 \_\_\_\_\_

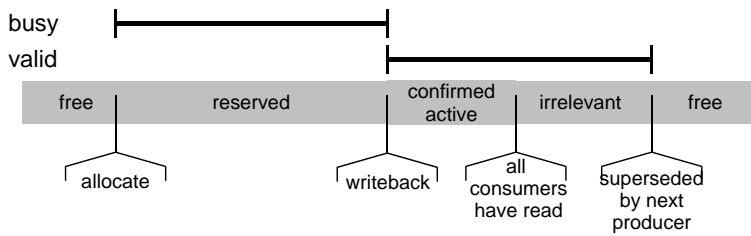
Alias phys		RF mode val	
R1		P1	
R2		P2	
R3		P3	
		P4	

ID	$\alpha$											
Ren												
RS1												
RS2												
EX1												
EX2												
WB												
R1												
R2												
P1 (mode,val)												
P2 (mode, val)												
P3 (mode, val)												
P4 (mode, val)												
		1	2	3	4	5	6	7	8	9	10	12

12

## Modes as Busy and Valid

- busy : the producing instruction has not yet written result
- valid : value in register is available for consumers
- value : data value



13

## Renaming and Hazards

- **Q:** How does register renaming handle each type of data hazard?

- 
- 
- 

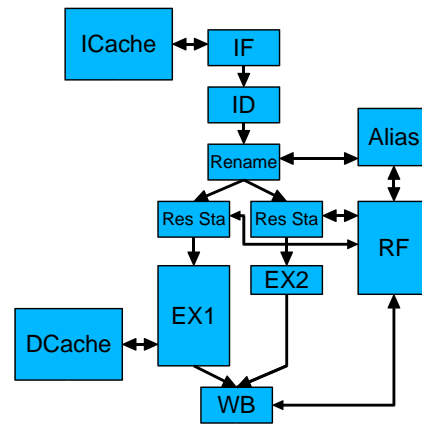
14

## Implementations of Reg Rename

- All forms of register renaming have some version of Alias and RF
- CDC 6600 scoreboard:
  - Alias is identify function
  - Affect:
  -
- IBM 360/91 Tomasulo's algorithm
  - Reservation stations hold instruction until writeback
  - Register file includes "real" RF and reservation stations
  - Affect:
- Modern register renaming
  - Similar to Alias + RF
  - Control hazards are a big complication ...
  - ... wait for the reorder buffer and dual-RAT

busy ← →

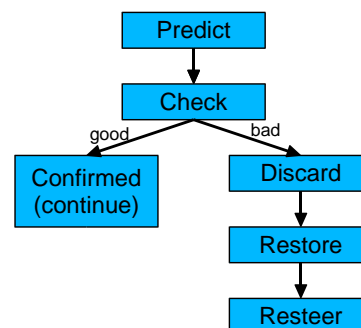
15



16

## Speculative Execution and Out-of-Order Execution

### Speculative Execution



**Predict:** a value without computing it.

**Check:**  
Do the computation whose result was predicted.  
Compare actual and predicted values.

**Confirmed:** do nothing, just continue.

**Discard:** discard or kill instructions that are in the shadow of instruction with mispredicted value.

**Restore:** replace mispredicted value with correct value.

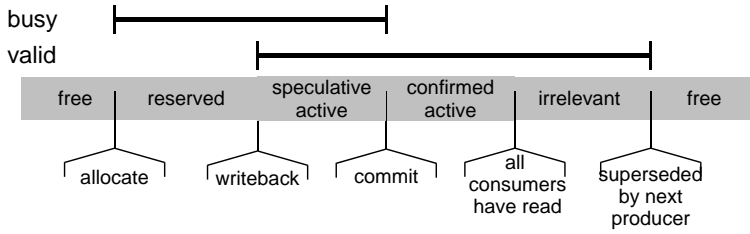
**Resteer:** begin fetching new instructions based on corrected value.

17

18

## Modes as Busy and Valid (w/ Ctrl Haz)

- busy : the producing instruction has not yet committed
- commit: sometimes called "retire"; speculative execution for the instruction and predecessors has been resolved



19

## Speculative and O-o-O Execution

$\alpha$	R1 = sqrt #16
$\beta$	BNZ R1 $\lambda$
$\gamma_1$	R2 = #10
$\gamma_2$	R2 = R2 + #1

O-o-O execution similar to CDC 6600 scoreboard.

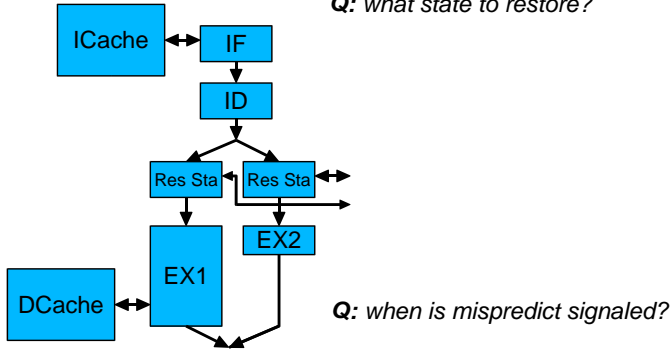
Alias[ x ] = x.

PC	$\alpha$	$\beta$	$\gamma_1$	$\gamma_2$	$\gamma_2$	$\gamma_3$	$\gamma_4$	$\gamma_4$	$\gamma_4$	$\lambda$						
ID		$\alpha$	$\beta$	$\gamma_1$	$\gamma_2$	$\gamma_2$	$\gamma_3$	$\gamma_4$	$\gamma_4$	$\gamma_4$	$\gamma_4$					
RS1			$\alpha$	$\beta$	$\beta$	$\beta$	$\beta$	$\beta$	$\beta$	$\beta$	$\beta$					
RS2					$\gamma_1$	$\gamma_2$	$\gamma_2$	$\gamma_3$	$\gamma_4$	$\gamma_4$	$\gamma_4$					
EX1				$\alpha$	$\alpha$	$\alpha$	$\alpha$	$\alpha$	$\alpha$	$\alpha$	$\beta$					
EX2							$\gamma_1$	$\gamma_2$	$\gamma_3$	$\gamma_4$	$\gamma_4$					
WB								$\gamma_1$	$\gamma_2$	$\gamma_3$	$\gamma_4$					
R1				VR, --								VA, 4				
R2									VA, 10	VA, 11	VA, 12					
		1	2	3	4	5	6	7	8	9	10					

20

## Restoring State After Misprediction

Q: what state to restore?



21

## Speculative and O-o-O Execution

$\alpha$	R1 = sqrt #16
$\beta$	BNZ R1 $\lambda$
$\gamma_1$	R2 = #10
$\gamma_2$	BNZ R2 $\tau$

O-o-O execution similar to CDC 6600 scoreboard.

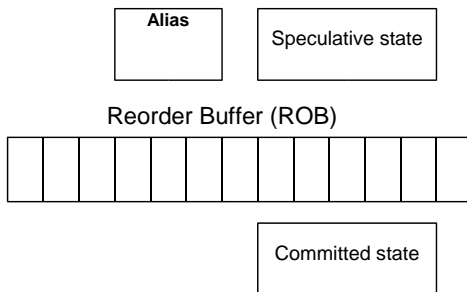
Alias[ x ] = x.

PC	$\alpha$	$\beta$	$\gamma_1$	$\gamma_2$	$\gamma_2$	$\gamma_3$	$\gamma_4$	$\gamma_4$	$\gamma_4$	$\lambda$						
ID		$\alpha$	$\beta$	$\gamma_1$	$\gamma_2$	$\gamma_2$	$\gamma_3$	$\gamma_4$	$\gamma_4$	$\gamma_4$	$\gamma_4$					
RS1			$\alpha$	$\beta$	$\beta$	$\beta$	$\beta$	$\beta$	$\beta$	$\beta$	$\beta$					
RS2					$\gamma_1$	$\gamma_2$	$\gamma_2$	$\gamma_3$	$\gamma_4$	$\gamma_4$	$\gamma_4$					
EX1				$\alpha$	$\alpha$	$\alpha$	$\alpha$	$\alpha$	$\alpha$	$\alpha$	$\beta$					
EX2							$\gamma_1$	$\gamma_2$	$\gamma_3$	$\gamma_4$	$\gamma_4$					
WB								$\gamma_1$	$\gamma_2$	$\gamma_3$	$\gamma_4$					
R1				VR, --								VA, 4				
R2									VA, 10	VA, 11	VA, 12					
		1	2	3	4	5	6	7	8	9	10					

22

## O-o-O Speculative Execution

- $\alpha$ : R3 = #19
- $\beta$ : R1 = R3 div #3
- $\gamma$ : BNZ R1  $\lambda$
- $\delta$ : R2 = R3 -- #1
- $\epsilon$ : R2 = R2 -- #2
- $\zeta$ : R2 = R2 -- #3
- $\eta$ : R2 = R2 -- #4



- GOAL (with predict not-taken): allow  $\delta$ ,  $\epsilon$ ,  $\zeta$ ,  $\eta$ , etc to execute before knowing whether  $\gamma$  is taken.

23

## Rollback (Undo)

R1=10, R2=20, R3=30

- $\delta$ : R2 = 29
- $\epsilon$ : R2 = 27
- $\zeta$ : R2 = 24
- $\eta$ : R2 = 20

R1																
R2																
R3																

Reorder Buffer (ROB)

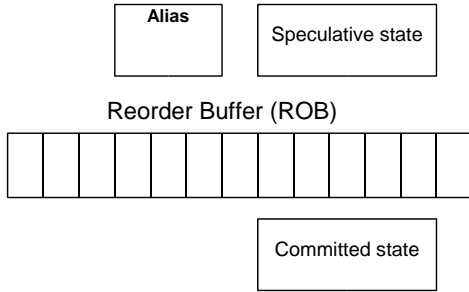
dst_idx																
prev_val																

R1	
R2	
R3	

24

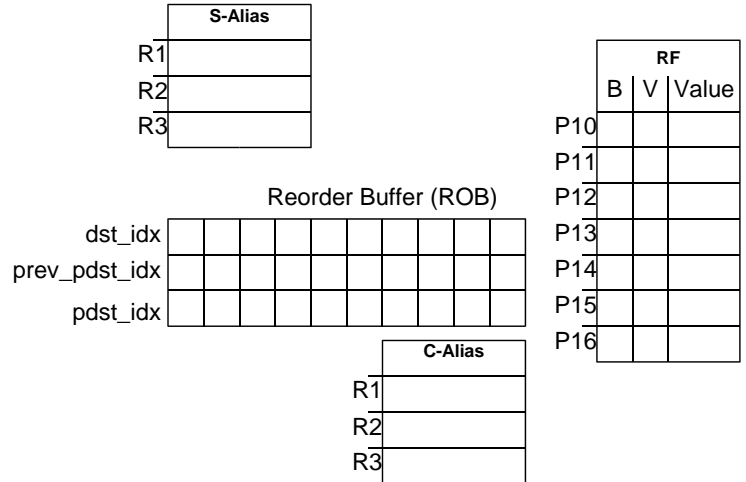
## When to Signal Mispredict?

$\alpha$ : R3 = #19  
 $\beta$ : R1 = R3 div #3  
 $\gamma$ : BNZ R1  $\lambda$   
 $\delta$ : R2 = R3 -- #1  
 $\epsilon$ : R2 = R2 -- #2  
 $\zeta$ : R2 = R2 -- #3  
 $\eta$ : R2 = R2 -- #4



25

## Dual Alias Tables



26

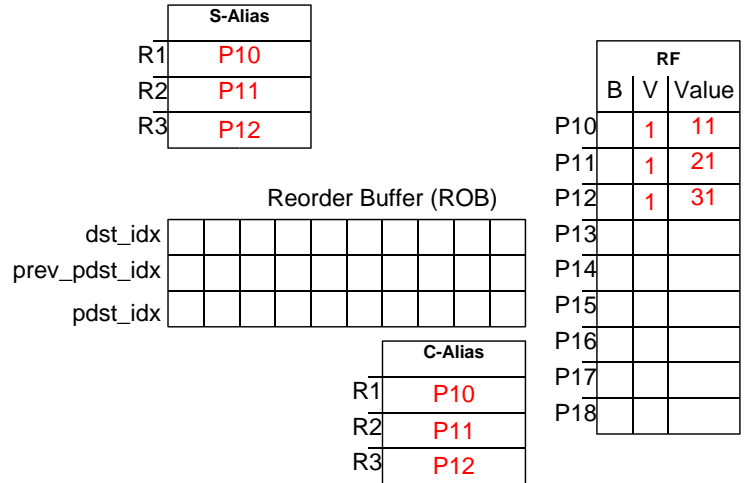
## Dual Alias Tables (1 of 5)

Initial values for RF R1=11, R2=21, R3=31  
 Initial values for Alias R1→P10, R2→P11, R3→P12

Program  $\alpha$ : R3 = #30 30  
 $\beta$ : R1 = R3 div #3  
 $\gamma$ : BNZ R1  $\lambda$   
 $\delta$ : R2 = R3 -- #1 29  
 $\epsilon$ : R2 = R2 -- #2 27  
 $\zeta$ : R2 = R2 -- #3 24  
 $\eta$ : R2 = R2 -- #4 20

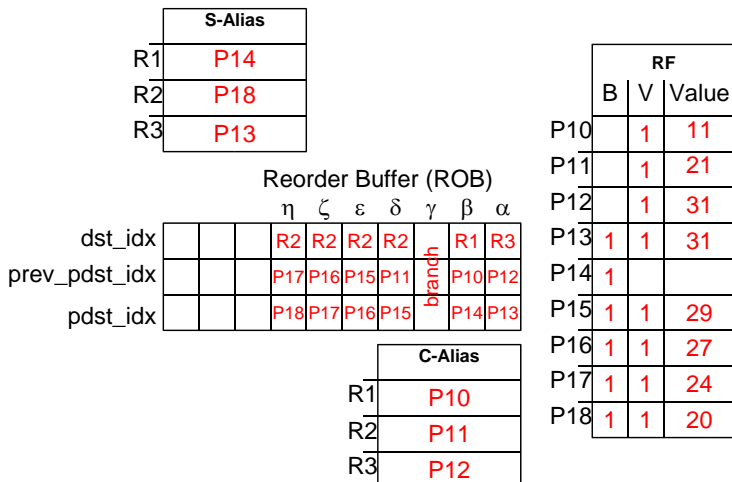
27

## Dual Alias Tables: Initial State



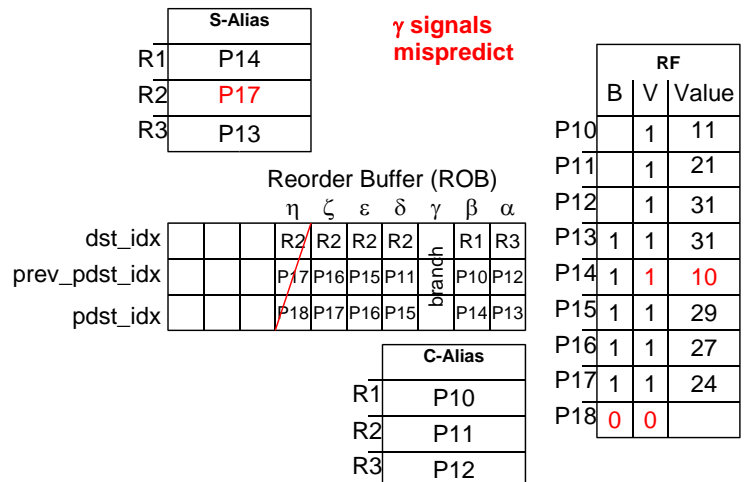
28

## Dual Alias Tables: Execute $\alpha$ -- $\eta$



29

## Dual Alias Tables: Undo $\eta$



30

## Dual Alias Tables: Undo ζ—δ

S-Alias	
R1	P14
R2	P11
R3	P13

RF		
B	V	Value
P10	1	11
P11	1	21
P12	1	31
P13	1	31
P14	1	
P15	0	0
P16	0	0
P17	0	0
P18	0	0

### Reorder Buffer (ROB)

	η	ζ	ε	δ	γ	β	α
dst_idx		R2	R2	R2	R2	R1	R3
prev_pdst_idx		P17	P16	P15	P11	P10	P12
pdst_idx		P18	P17	P16	P15	P14	P13

C-Alias	
R1	P10
R2	P11
R3	P12

31

## ROB as S-RF (1 of 5)

Also known as: **Rob with renaming and speculative execution.**

Initial values for C-RF R1=11, R2=21, R3=31

Initial values for Alias R1=(Y,--), R2=(Y,--), R3=(Y,--)

Program

α: R3 = #30 30  
←O1

β: R1 = R3 div #3  
←O2 ←O1

γ: BNZ R1 λ  
O3 ←O2

δ: R2 = R3 -- #1 29  
←O4 ←O1

ε: R2 = R2 -- #2 27  
←O5 ←O4

ζ: R2 = R2 -- #3 24  
←O6 ←O5

η: R2 = R2 -- #4 20  
←O7 ←O6

32

## ROB as S-RF

Alias	
cmt	rob_idx
R1	
R2	
R3	

### Reorder Buffer (ROB)

	O11	O10	O9	O8	O7	O6	O5	O4	O3	O2	O1
dst_idx											
prev_commit											
prev_rob_idx											
dst_val											
valid											

C-RF	
R1	
R2	
R3	

33

## ROB as S-RF: Initial State

Alias	
cmt	rob_idx
R1	Y
R2	Y
R3	Y

### Reorder Buffer (ROB)

	O11	O10	O9	O8	O7	O6	O5	O4	O3	O2	O1
dst_idx											
prev_commit											
prev_rob_idx											
dst_val											
valid											

C-RF	
R1	11
R2	21
R3	31

34

## ROB as S-RF: Execute α--η

Alias	
cmt	rob_idx
R1	n O2
R2	n O7
R3	n O1

### Reorder Buffer (ROB)

	η	ζ	ε	δ	γ	β	α
dst_idx		R2	R2	R2	R2	R1	R3
prev_commit		n	n	n	Y	Y	Y
prev_rob_idx		O6	O5	O4	branch		
dst_val		20	24	27	29		30
valid		Y	Y	Y	Y	n	Y

C-RF	
R1	11
R2	21
R3	31

35

## ROB as S-RF: Undo η

Alias	
cmt	rob_idx
R1	n O2
R2	n O6
R3	n O1

γ signals  
mispredict

### Reorder Buffer (ROB)

	η	ζ	ε	δ	γ	β	α
dst_idx		R2	R2	R2	R2	R1	R3
prev_commit		n	n	n	Y	Y	Y
prev_rob_idx		O6	O5	O4	branch		
dst_val		20	24	27	29	10	30
valid		Y	Y	Y	Y	Y	Y

C-RF	
R1	11
R2	21
R3	31

36

## ROB as S-RF: Undo $\zeta$ — $\delta$

		Alias	
		cmt	rob_idx
R1	n		O2
R2	Y		
R3	n		O1

Reorder Buffer (ROB)

		$\eta$	$\zeta$	$\epsilon$	$\delta$	$\gamma$	$\beta$	$\alpha$				
dst_idx			R2	R2	R2	R2		R1 R3				
prev_commit			n	n	n	Y		Y Y				
prev_rob_idx			O6	O5	O4		branch					
dst_val			20	24	27	29		30				
valid			Y	Y	Y	Y		n Y				
		O11	O10	O9	O8	O7	O6	O5	O4	O3	O2	O1

		C-RF
R1		11
R2		21
R3		31

37

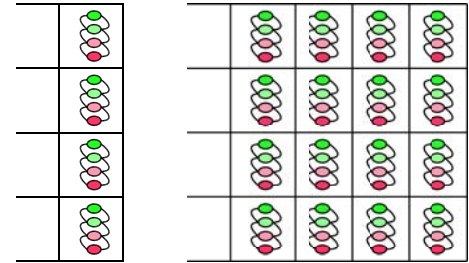
## Branch Prediction

strongly taken

weakly taken

weakly not taken

strongly not taken



38