HY425 Lecture 07: Precise Interrupts, Implementing Speculation

Dimitrios S. Nikolopoulos

University of Crete and FORTH-ICS

October 31, 2011

Superscalar execution

Characteristics

- Multiple instruction issue
- Dynamic (scoreboard, Tomasulo) or static (compiler) instruction scheduling
- Logic issues independent instructions, modulo constraints
  - Instructions dependent on pending instructions not issued
  - Instructions not issued due to hazards
  - Other constraints (e.g. on types of instructions issued per cycle)
- Number of instructions issued per cycle variable
- CPI, IPC, limited by dependencies, hazards

Speculation basics

- Multiple-issue processors need more independent instructions
- Wide-issue processors may need to predict one or more branches per cycle
  - Hard to maintain high clock rate
- Speculation attempts to overcome problem
  - Execute program as if all branches were predicted correctly
  - Dynamic scheduling waits for branches instead
  - Need mechanism to undo instructions from incorrect path
- Three elements in speculation
  - Dynamic branch prediction
  - Ability to undo speculative instructions from wrong path
  - Dynamic scheduling

Maintaining precise exceptions

Reorder buffer (ROB)

- Instructions generating exceptions do not commit until they reach the head of the ROB
- In scoreboard, Tomasulo, instructions commit out of order:
  - Out-of-order instructions modify state (register memory)
  - Exceptions from in-order instructions are imprecise
- Recoverable memory exceptions
  - Program must resume execution
    - Example: page faults
  - No speculative state should exist when program resumes

Precise exceptions with speculation

Example

- Assume all instructions in the loop have issued twice
- Assume all instructions have completed execution
- LD, MULD of first iteration have committed

```
Loop:  LD   F0, 0(R1)
       MULD F4, F0, F2
       SD   F4, 0(R1)
       SUBI R1, R1, 8
       BNE  R1, R2, Loop
```

Precise exceptions with speculation (cont.)

ROB snapshot after two iterations

Reorder buffer

Entry	Busy	Instruction	State	Destination	Value
1	no	LD F0, 0(R1)	Commit	F0	M[0+R[R1]]
2	no	MULD F4, F0, F2	Commit	F4	#1 $\times$ R[F2]
3	yes	SD F4, 0(R1)	Write result	0 + R[R1]	#2
4	yes	SUBI R1, R1, 8	Write result	R1	R[R1] - 8
5	yes	BNE R1, R2, Loop	Write result		
6	yes	LD F0, 0(R1)	Write result	F0	M[#4]
7	yes	MULD F4, F0, F2	Write result	F4	#6 $\times$ R[F2]
8	yes	SD F4, 0(R1)	Write result	0 + #4	#7
9	yes	SUBI R1, R1, 8	Write result	R1	#4 - 8
10	yes	BNE R1, R2, Loop	Write result		

Precise exceptions with speculation (cont.)

Assume first BNE not taken

Reorder buffer

Entry	Busy	Instruction	State	Destination	Value
1	no	LD F0, 0(R1)	Commit	F0	M[0+R[R1]]
2	no	MULD F4, F0, F2	Commit	F4	#1 $\times$ R[F2]
3	yes	SD F4, 0(R1)	Commit	0 + R[R1]	#2
4	yes	SUBI R1, R1, 8	Commit	R1	R[R1] - 8
5	yes	BNE R1, R2, Loop	Write result		
6					
7					
8					
9					
10					

- ROB after BNE cleared
- Fetch instructions from fall-through path

Precise exceptions with speculation (cont.)

Using the ROB for exceptions

- Exception recorded in ROB
- Exception not thrown until instruction commits
- If exception happens in speculative instruction and processor mis-speculates:
  - Squash this and other speculative instructions
  - Optimization:
    - Handle exceptions as soon as they arise and earlier branches are resolved

Precise exceptions with speculation (cont.)

Using the ROB for exceptions

- Load and stores handled also in ROB
- Register/memory not updated until load/store reaches head of ROB
- WAW, WAR hazards in memory removed due to in-order commit
- RAW hazards handled by:
  - Preventing load from executing if prior store in ROB has same effective address
  - Preserve program order for computation of effective address of load with respect to earlier stores
- Forwarding possible from stores to later loads

Multiple issue with speculation

Mechanisms

- Similar to Tomasulo with multiple issue
- Must support multiple commits in one cycle

Example

```
Loop:  LD       R2,0(R1)
       ADDI     R2,R2,1
       SD       R2,0(R1)
       ADDI     R1,R1,4
       BNE      R2,R3,Loop
```

Multiple issue without speculation

Assume separate LD-ST and INT units

Iteration number	Instruction	Issues at clock cycle number	Executes at clock cycle number	Memory access at clock cycle number	Write CDB at clock cycle number	Comment
1	LD R2,0(R1)	1	2	3	4	First issue
1	ADDI R2,R2,1	1	5		6	Wait for LD
1	SD R2,0(R1)	2	3	7		Wait for ADDI
1	ADDI R1,R1,4	2	3		4	Execute directly
1	BNE R2,R3,Loop	3	7		Wait for ADDI	
2	LD R2,0(R1)	4	8	9	10	Wait for BNE
2	ADDI R2,R2,1	4	11		12	Wait for LD
2	SD R2,0(R1)	5	9	13		Wait for ADDI
2	ADDI R1,R1,4	5	8		9	Execute directly
2	BNE R2,R3,Loop	6	13		Wait for ADDI	
3	LD R2,0(R1)	7	14	15	16	Wait for BNE
3	ADDI R2,R2,1	7	17		18	Wait for LD
3	SD R2,0(R1)	8	15	19		Wait for ADDI
3	ADDI R1,R1,4	8	14		15	Execute directly
3	BNE R2,R3,Loop	9	19		Wait for ADDI	

Multiple issue with speculation

Assume separate LD-ST and INT units

Iteration number	Instruction	Issues at at clock number	Executes at clock number	Read access at clock number	Writes CDB at clock number	Commits at clock number	Comment
1	LD R2,0(R1)	1	2	3	4	5	First issue
1	ADDI R2,R2,1	1	5		6	7	Wait for LD
1	SD R2,0(R1)	2	3			7	Wait for ADDI
1	ADDI R1,R1,4	2	3		4	8	Commit in order
1	BNE R2,R3,Loop	3	7			8	Wait for ADDI
2	LD R2,0(R1)	4	8		9	10	No execute delay
2	ADDI R2,R2,1	4	5	6	7	9	No Wait for LD
2	SD R2,0(R1)	5	6			10	Wait for ADDI
2	ADDI R1,R1,4	5	6		7	11	Commit in order
2	BNE R2,R3,Loop	6	10			11	Wait for ADDI
3	LD R2,0(R1)	7	8	9	10	12	Earliest possible
3	ADDI R2,R2,1	7	11		12	13	Wait for LD
3	SD R2,0(R1)	8	9			13	Wait for ADDI
3	ADDI R1,R1,4	8	9		10	14	Executes earlier
3	BNE R2,R3,Loop	9	13			14	Wait for ADDI

Renaming registers

Alternative renaming mechanisms

- Instruction target renaming done through reservation stations in Tomasulo
- Alternative implementations
  - Use ROB entries
  - Use a larger set of physical registers
    - Separate architecturally visible registers from physical registers
    - Architecture registers are registers visible to programmers
    - Physical registers $>$ Architecture registers (e.g. $2\times$)
    - Physical registers used for renaming, if available

Renaming registers (cont.)

Implementation

- Register renaming done through renaming map
  - Architecture register $\rightarrow$ physical register
  - Status of physical registers
    - Free: Register has not been assigned to instruction
    - Invalid: Register has been assigned to instruction for renaming, but value has not been computed yet
    - Valid: Register has been assigned to instruction for renaming, and value has been computed

Scoreboard with register renaming

Instruction status:

Instruction		j	k	Issue	Read Oper	Exec Comp	Write Result
LD	F6	34+	R2				
LD	F2	45+	R3				
MULTD	F0	F2	F4				
SUBD	F8	F6	F2				
DIVD	F10	F0	F6				
ADDD	F6	F8	F2				

Functional unit status:

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU Qj	FU Qk	Fj? Rj	Fk? Rk
	Integer1	No								
	Integer2	No								
	Mult1	No								
	Add1	No								
	Divide1	No								

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
	Rename	P0	P2	P4	P6	P8	P10	P12	...	P30
	Free list	P32	P34	P36	P38	P40	P42	P44	...	P62
	Old list									

Scoreboard with register renaming – Cycle 1

- P32 allocated from free list
- P6 (old physical register) moved to old list

Instruction status:

Instruction		j	k	Issue	Read Oper	Exec Comp	Write Result
LD	F6	34+	R2	1			
LD	F2	45+	R3				
MULTD	F0	F2	F4				
SUBD	F8	F6	F2				
DIVD	F10	F0	F6				
ADDD	F6	F8	F2				

Functional unit status:

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU Qj	FU Qk	Fj? Rj	Fk? Rk
	Integer1	Yes	Load	P32		R2				Yes
	Integer2	No								
	Mult1	No								
	Add1	No								
	Divide1	No								

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
1	Rename	P0	P2	P4	P32	P8	P10	P12	…	P30
	Free list	P34	P36	P38	P40	P42	P44	….	P60	
	Old list	P6								

Scoreboard with register renaming – Cycle 2

- P34 allocated from free list
- P2 (old physical register) moved to old list

Instruction status:

Instruction		j	k	Issue	Read Oper	Exec Comp	Write Result
LD	F6	34+	R2	1	2		
LD	F2	45+	R3	2			
MULTD	F0	F2	F4				
SUBD	F8	F6	F2				
DIVD	F10	F0	F6				
ADDD	F6	F8	F2				

Functional unit status:

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU Qj	FU Qk	Fj? Rj	Fk? Rk
	Integer1	Yes	Load	P32		R2				Yes
	Integer2	Yes	Load	P34		R3				Yes
	Mult1	No								
	Add1	No								
	Divide1	No								

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
2	Rename	P0	P34	P4	P32	P8	P10	P12	...	P30
	Free list	P36	P38	P40	P42	P44	...	P60		
	Old list	P6	P2							

Scoreboard with register renaming – Cycle 3

- P36 allocated from free list
- P0 (old physical register) moved to old list

Instruction status:

Instruction		j	k	Issue	Read Oper	Exec Comp	Write Result
LD	F6	34+	R2	1	2	3	
LD	F2	45+	R3	2	3		
MULTD	F0	F2	F4	3			
SUBD	F8	F6	F2				
DIVD	F10	F0	F6				
ADDD	F6	F8	F2				

Functional unit status:

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU Qj	FU Qk	Fj? Rj	Fk? Rk
	Integer1	Yes	LD	P32		R2				Yes
	Integer2	Yes	LD	P34		R3				Yes
	Mult1	Yes	MULTD	P36	P34	P4	Integer2		No	Yes
	Add1	No								
	Divide1	No								

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
3	Rename	P36	P34	P4	P32	P8	P10	P12	…	P30
	Free list	P38	P40	P42	P44	…	P60			
	Old list	P6	P2	P0						

Scoreboard with register renaming – Cycle 4

- P38 allocated from free list.
- P6 (old destination of LD) recycled since first LD commits
- P8 (old physical register of SUBD) moved to old list

Instruction status:

Instruction		j	k	Issue	Read Oper	Exec Comp	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	2	3	4	
MULTD	F0	F2	F4	3			
SUBD	F8	F6	F2	4			
DIVD	F10	F0	F6				
ADDD	F6	F8	F2				

Functional unit status:

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU Qj	FU Qk	Fj? Rj	Fk? Rk
	Integer1	No								
	Integer2	Yes	LD	P34		R3				Yes
	Mult1	Yes	MULTD	P36	P34	P4	Integer2		No	Yes
	Add1	Yes	SUBD	P38	P32	P34		Integer2	Yes	No
	Divide1	No								

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
4	Rename	P36	P34	P4	P32	P38	P10	P12	...	P30
	Free list	P40	P42	P44	...	P60	P6			
	Old list	P2	P0	P8						

Scoreboard with register renaming – Cycle 5

- P40 allocated from free list
- P2 (old physical register of second LD) recycled
- P10 (old physical register of SUBD) moved to old list

Instruction status:

Instruction		j	k	Issue	Read Oper	Exec Comp	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	2	3	4	5
MULTD	F0	F2	F4	3			
SUBD	F8	F6	F2	4			
DIVD	F10	F0	F6	5			
ADDD	F6	F8	F2				

Functional unit status:

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU Qj	FU Qk	Fj? Rj	Fk? Rk
	Integer1	No								
	Integer2	No								
	Mult1	Yes	MULTD	P36	P34	P4			Yes	Yes
	Add1	Yes	SUBD	P38	P32	P34			Yes	Yes
	Divide1	Yes	DIVD	P40	P36	P32	Mult1		No	Yes

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
5	Rename	P36	P34	P4	P32	P38	P40	P12	...	P30
	Free list	P42	P44	...	P60	P6	P2			
	Old list	P0	P8	P10						

Scoreboard with register renaming – Cycle 6

Instruction status:

Instruction		j	k	Issue	Read Oper	Exec Comp	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	2	3	4	5
MULTD	F0	F2	F4	3	6		
SUBD	F8	F6	F2	4	6		
DIVD	F10	F0	F6	5			
ADDD	F6	F8	F2				

Functional unit status:

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU Qj	FU Qk	Fj? Rj	Fk? Rk
	Integer1	No								
	Integer2	No								
10	Mult1	Yes	MULTD	P36	P34	P4			Yes	Yes
2	Add1	Yes	SUBD	P38	P32	P34			Yes	Yes
	Divide1	Yes	DIVD	P40	P36	P32	Mult1		No	Yes

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
6	Rename	P36	P34	P4	P32	P38	P40	P12	…	P30
	Free list	P42	P44	…	P60	P6	P2			
	Old list	P0	P8	P10						

Scoreboard with register renaming – Cycle 7

Instruction status:

Instruction		j	k	Issue	Read Oper	Exec Comp	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	2	3	4	5
MULTD	F0	F2	F4	3	6		
SUBD	F8	F6	F2	4	6		
DIVD	F10	F0	F6	5			
ADDD	F6	F8	F2				

Functional unit status:

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU Qj	FU Qk	Fj? Rj	Fk? Rk
	Integer1	No								
	Integer2	No								
9	Mult1	Yes	MULTD	P36	P34	P4			Yes	Yes
1	Add1	Yes	SUBD	P38	P32	P34			Yes	Yes
	Divide1	Yes	DIVD	P40	P36	P32	Mult1		No	Yes

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
7	Rename	P36	P34	P4	P32	P38	P40	P12	...	P30
	Free list	P42	P44	...	P60	P6	P2			
	Old list	P0	P8	P10						

Scoreboard with register renaming – Cycle 8

Instruction status:

Instruction		j	k	Issue	Read Oper	Exec Comp	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	2	3	4	5
MULTD	F0	F2	F4	3	6		
SUBD	F8	F6	F2	4	6	8	
DIVD	F10	F0	F6	5			
ADDD	F6	F8	F2				

Functional unit status:

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU Qj	FU Qk	Fj? Rj	Fk? Rk
	Integer1	No								
	Integer2	No								
8	Mult1	Yes	MULTD	P36	P34	P4			Yes	Yes
0	Add1	Yes	SUBD	P38	P32	P34			Yes	Yes
	Divide1	Yes	DIVD	P40	P36	P32	Mult1		No	Yes

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
8	Rename	P36	P34	P4	P32	P38	P40	P12	...	P30
	Free list	P42	P44	...	P60	P6	P2			
	Old list	P0	P8	P10						

Scoreboard with register renaming – Cycle 9

Instruction status:

Instruction		j	k	Issue	Read Oper	Exec Comp	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	2	3	4	5
MULTD	F0	F2	F4	3	6		
SUBD	F8	F6	F2	4	6	8	9
DIVD	F10	F0	F6	5			
ADDD	F6	F8	F2				

Functional unit status:

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU Qj	FU Qk	Fj? Rj	Fk? Rk
	Integer1	No								
	Integer2	No								
7	Mult1	Yes	MULTD	P36	P34	P4			Yes	Yes
	Add1	No								
	Divide1	Yes	DIVD	P40	P36	P32	Mult1		No	Yes

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
9	Rename	P36	P34	P4	P32	P38	P40	P12	...	P30
	Free list	P42	P44	...	P60	P6	P2			
	Old list	P0	P8	P10						

Scoreboard with register renaming – Cycle 10

- P42 allocated from free list
- P32 pushed to old list, still in use by DIVD

Instruction status:

Instruction		j	k	Issue	Read Oper	Exec Comp	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	2	3	4	5
MULTD	F0	F2	F4	3	6		
SUBD	F8	F6	F2	4	6	8	9
DIVD	F10	F0	F6	5			
ADDD	F6	F8	F2	10			

Functional unit status:

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU Qj	FU Qk	Fj? Rj	Fk? Rk
	Integer1	No								
	Integer2	No								
6	Mult1	Yes	MULTD	P36	P34	P4			Yes	Yes
	Add1	Yes	ADDD	P42	P38	P34			Yes	Yes
	Divide1	Yes	DIVD	P40	P36	P32	Mult1		No	Yes

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
10	Rename	P36	P34	P4	P42	P38	P40	P12	…	P30
	Free list	P44	…	P60	P6	P2				
	Old list	P0	P8	P10	P32					

Scoreboard with register renaming – Cycle 11

Instruction status:

Instruction		j	k	Issue	Read Oper	Exec Comp	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	2	3	4	5
MULTD	F0	F2	F4	3	6		
SUBD	F8	F6	F2	4	6	8	9
DIVD	F10	F0	F6	5			
ADDD	F6	F8	F2	10	11		

Functional unit status:

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU Qj	FU Qk	Fj? Rj	Fk? Rk
	Integer1	No								
	Integer2	No								
5	Mult1	Yes	MULTD	P36	P34	P4			Yes	Yes
2	Add1	Yes	ADDD	P42	P38	P34			Yes	Yes
	Divide1	Yes	DIVD	P40	P36	P32	Mult1		No	Yes

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
11	Rename	P36	P34	P4	P42	P38	P40	P12	…	P30
	Free list	P44	…	P60	P6	P2				
	Old list	P0	P8	P10	P32					

Scoreboard with register renaming – Cycle 12

Instruction status:

Instruction		j	k	Issue	Read Oper	Exec Comp	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	2	3	4	5
MULTD	F0	F2	F4	3	6		
SUBD	F8	F6	F2	4	6	8	9
DIVD	F10	F0	F6	5			
ADDD	F6	F8	F2	10	11		

Functional unit status:

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU Qj	FU Qk	Fj? Rj	Fk? Rk
	Integer1	No								
	Integer2	No								
4	Mult1	Yes	MULTD	P36	P34	P4			Yes	Yes
1	Add1	Yes	ADDD	P42	P38	P34			Yes	Yes
	Divide1	Yes	DIVD	P40	P36	P32	Mult1		No	Yes

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
12	Rename	P36	P34	P4	P42	P38	P40	P12	...	P30
	Free list	P44	...	P60	P6	P2				
	Old list	P0	P8	P10	P32					

Scoreboard with register renaming – Cycle 13

Instruction status:

Instruction		j	k	Issue	Read Oper	Exec Comp	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	2	3	4	5
MULTD	F0	F2	F4	3	6		
SUBD	F8	F6	F2	4	6	8	9
DIVD	F10	F0	F6	5			
ADDD	F6	F8	F2	10	11	13	

Functional unit status:

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU Qj	FU Qk	Fj? Rj	Fk? Rk
	Integer1	No								
	Integer2	No								
3	Mult1	Yes	MULTD	P36	P34	P4			Yes	Yes
0	Add1	Yes	ADDD	P42	P38	P34			Yes	Yes
	Divide1	Yes	DIVD	P40	P36	P32	Mult1		No	Yes

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
13	Rename	P36	P34	P4	P42	P38	P40	P12	…	P30
	Free list	P44	…	P60	P6	P2				
	Old list	P0	P8	P10	P32					

Scoreboard with register renaming – Cycle 14

Instruction status:

Instruction		j	k	Issue	Read Oper	Exec Comp	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	2	3	4	5
MULTD	F0	F2	F4	3	6		
SUBD	F8	F6	F2	4	6	8	9
DIVD	F10	F0	F6	5			
ADDD	F6	F8	F2	10	11	13	14

Functional unit status:

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU Qj	FU Qk	Fj? Rj	Fk? Rk
	Integer1	No								
	Integer2	No								
2	Mult1	Yes	MULTD	P36	P34	P4			Yes	Yes
	Add1	No								
	Divide1	Yes	DIVD	P40	P36	P32	Mult1		No	Yes

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
14	Rename	P36	P34	P4	P42	P38	P40	P12	…	P30
	Free list	P44	…	P60	P6	P2				
	Old list	P0	P8	P10	P32					

Scoreboard with register renaming – Cycle 15

Instruction status:

Instruction		j	k	Issue	Read Oper	Exec Comp	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	2	3	4	5
MULTD	F0	F2	F4	3	6		
SUBD	F8	F6	F2	4	6	8	9
DIVD	F10	F0	F6	5			
ADDD	F6	F8	F2	10	11	13	14

Functional unit status:

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU Qj	FU Qk	Fj? Rj	Fk? Rk
	Integer1	No								
	Integer2	No								
1	Mult1	Yes	MULTD	P36	P34	P4			Yes	Yes
	Add1	No								
	Divide1	Yes	DIVD	P40	P36	P32	Mult1		No	Yes

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
15	Rename	P36	P34	P4	P42	P38	P40	P12	…	P30
	Free list	P44	…	P60	P6	P2				
	Old list	P0	P8	P10	P32					

Scoreboard with register renaming – Cycle 16

Instruction status:

Instruction		j	k	Issue	Read Oper	Exec Comp	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	2	3	4	5
MULTD	F0	F2	F4	3	6	16	
SUBD	F8	F6	F2	4	6	8	9
DIVD	F10	F0	F6	5			
ADDD	F6	F8	F2	10	11	13	14

Functional unit status:

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU Qj	FU Qk	Fj? Rj	Fk? Rk
	Integer1	No								
	Integer2	No								
0	Mult1	Yes	MULTD	P36	P34	P4			Yes	Yes
	Add1	No								
	Divide1	Yes	DIVD	P40	P36	P32	Mult1		No	Yes

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
16	Rename	P36	P34	P4	P42	P38	P40	P12	…	P30
	Free list	P44	…	P60	P6	P2				
	Old list	P0	P8	P10	P32					

Scoreboard with register renaming – Cycle 17

Instruction status:

Instruction		j	k	Issue	Read Oper	Exec Comp	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	2	3	4	5
MULTD	F0	F2	F4	3	6	16	17
SUBD	F8	F6	F2	4	6	8	9
DIVD	F10	F0	F6	5			
ADDD	F6	F8	F2	10	11	13	14

Functional unit status:

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU Qj	FU Qk	Fj? Rj	Fk? Rk
	Integer1	No								
	Integer2	No								
	Mult1	No								
	Add1	No								
	Divide1	Yes	DIVD	P40	P36	P32	Mult1		No	Yes

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
17	Rename	P36	P34	P4	P42	P38	P40	P12	…	P30
	Free list	P44	…	P60	P6	P2	P0			
	Old list	P8	P10	P10						

Scoreboard with register renaming – Cycle 18

- MULTD commits, recycle old register P0

Instruction status:

Instruction		j	k	Issue	Read Oper	Exec Comp	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	2	3	4	5
MULTD	F0	F2	F4	3	6	16	17
SUBD	F8	F6	F2	4	6	8	9
DIVD	F10	F0	F6	5	18		
ADDD	F6	F8	F2	10	11	13	14

Functional unit status:

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU Qj	FU Qk	Fj? Rj	Fk? Rk
	Integer1	No								
	Integer2	No								
	Mult1	No								
	Add1	No								
40	Divide1	Yes	DIVD	P40	P36	P32	Mult1		No	Yes

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
18	Rename	P36	P34	P4	P42	P38	P40	P12	...	P30
	Free list	P44	...	P60	P6	P2	P0			
	Old list	P8	P10	P32						

Scoreboard with register renaming – Cycle 19

- SUBD commits, recycle old register P8

Instruction status:

Instruction		j	k	Issue	Read Oper	Exec Comp	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	2	3	4	5
MULTD	F0	F2	F4	3	6	16	17
SUBD	F8	F6	F2	4	6	8	9
DIVD	F10	F0	F6	5	18		
ADDD	F6	F8	F2	10	11	13	14

Functional unit status:

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU Qj	FU Qk	Fj? Rj	Fk? Rk
	Integer1	No								
	Integer2	No								
	Mult1	No								
	Add1	No								
39	Divide1	Yes	DIVD	P40	P36	P32	Mult1		No	Yes

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
19	Rename	P36	P34	P4	P42	P38	P40	P12	...	P30
	Free list	P44	...	P60	P6	P2	P0	P8		
	Old list	P10	P32							

Branches and renaming

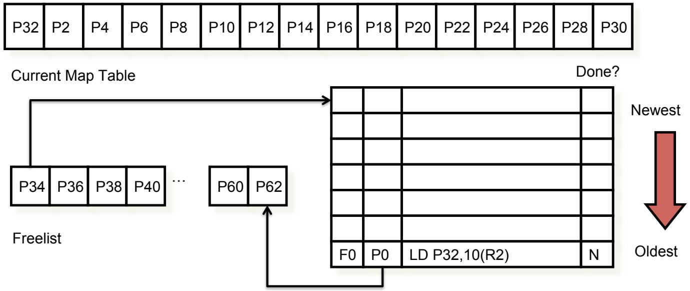
How do we undo speculative commits to registers?

- Commit stage and reorder buffer resolve branches in Tomasulo
  - Instructions commit in-order
  - Reorder buffer holds speculative results
  - Squashing instructions in ROB easy
- If speculative instructions modify physical registers
  - Physical registers hold speculative results
  - How do we undo these modifications?

Register renaming with branches

MIPS R10000 solution

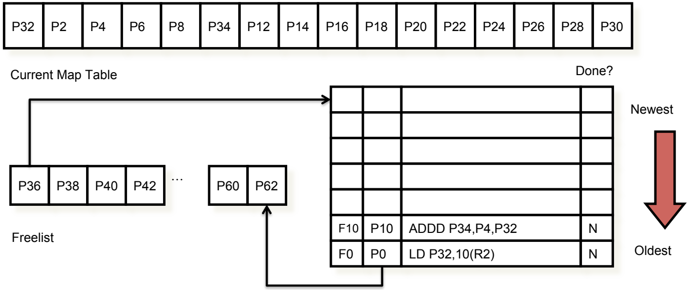
- Rename map stores instruction status
- Reorder buffer maintains instruction order



Register renaming with branches

MIPS R10000 solution

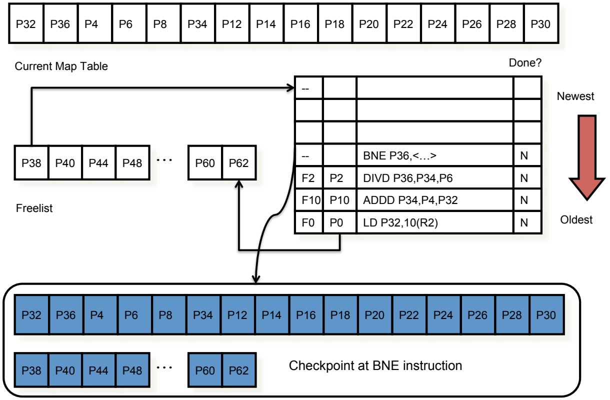
- Reorder buffer maintains old register ID



Register renaming with branches

MIPS R10000 solution

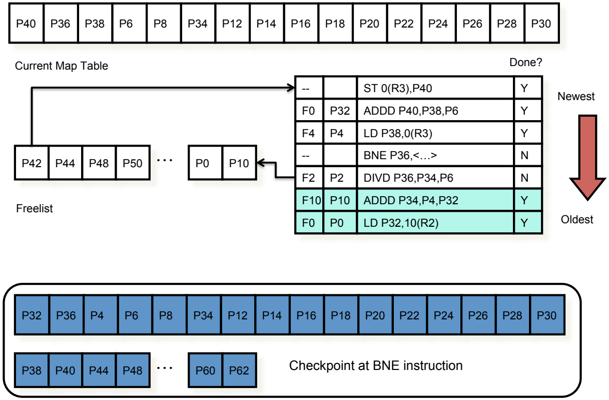
- Map table and freelist checkpointed at branch


P32 P36 P4 P6 P8 P34 P12 P14 P16 P18 P20 P22 P24 P26 P28 P30
Current Map Table
Done?
Newest
Oldest
P38 P40 P44 P48 ... P60 P62
Freelist
--
-- BNE P36,<...> N
F2 P2 DIVD P36,P34,P6 N
F10 P10 ADDD P34,P4,P32 N
F0 P0 LD P32,10(R2) N
P32 P36 P4 P6 P8 P34 P12 P14 P16 P18 P20 P22 P24 P26 P28 P30
P38 P40 P44 P48 ... P60 P62
Checkpoint at BNE instruction

Register renaming with branches

MIPS R10000 solution

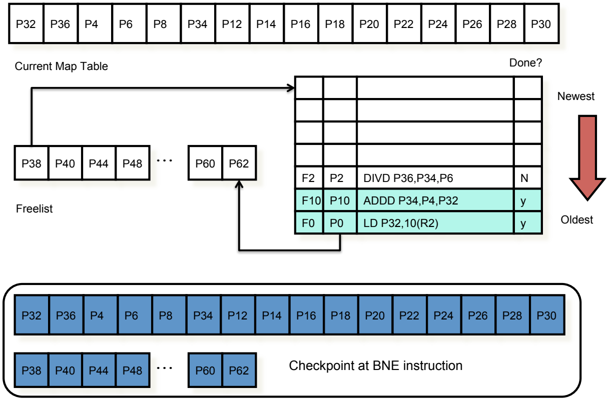
- Old registers recycled when instructions commit


P40 P36 P38 P6 P8 P34 P12 P14 P16 P18 P20 P22 P24 P26 P28 P30
Current Map Table
Done?
-- | | ST 0(R3),P40 | Y
F0 | P32 | ADDD P40,P38,P6 | Y
F4 | P4 | LD P38,0(R3) | Y
-- | | BNE P36,<...> | N
F2 | P2 | DIVD P36,P34,P6 | N
F10 | P10 | ADDD P34,P4,P32 | Y
F0 | P0 | LD P32,10(R2) | Y
Newest
Oldest
P42 P44 P48 P50 ··· P0 P10
Freelist
P32 P36 P4 P6 P8 P34 P12 P14 P16 P18 P20 P22 P24 P26 P28 P30
P38 P40 P44 P48 ··· P60 P62
Checkpoint at BNE instruction

Register renaming with branches

MIPS R10000 solution

- Checkpoint restored when misprediction is detected


P32 P36 P4 P6 P8 P34 P12 P14 P16 P18 P20 P22 P24 P26 P28 P30
Current Map Table
Done?
Newest
Oldest
P38 P40 P44 P48 ··· P60 P62
Freelist
F2 P2 DIVD P36,P34,P6 N
F10 P10 ADDD P34,P4,P32 y
F0 P0 LD P32,10(R2) y
P32 P36 P4 P6 P8 P34 P12 P14 P16 P18 P20 P22 P24 P26 P28 P30
P38 P40 P44 P48 ··· P60 P62
Checkpoint at BNE instruction

Concept of in-order instruction commit

Fixing speculation errors

- Out-of-order execution and speculation boost available ILP
- In-order instruction commit helps:
  - Resolve WAW, WAR hazards
  - Resolve memory hazards
  - Speculate past branches
  - Preserve precise exceptions
- Various implementations (ROB, register map, reservation stations)
- Used in most modern high-end processors

Implications of out-of-order execution

Complexity

- Expensive hardware
  - Large storage structures, associative searches
  - Complex issue logic
  - Complex branch prediction and speculation hardware
  - Hard to handle expensive exceptions
- Diminishing performance returns
  - Limits of ILP topic of next lecture
  - Finding more parallelism while reducing complexity topic of following lectures