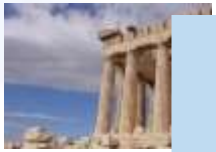
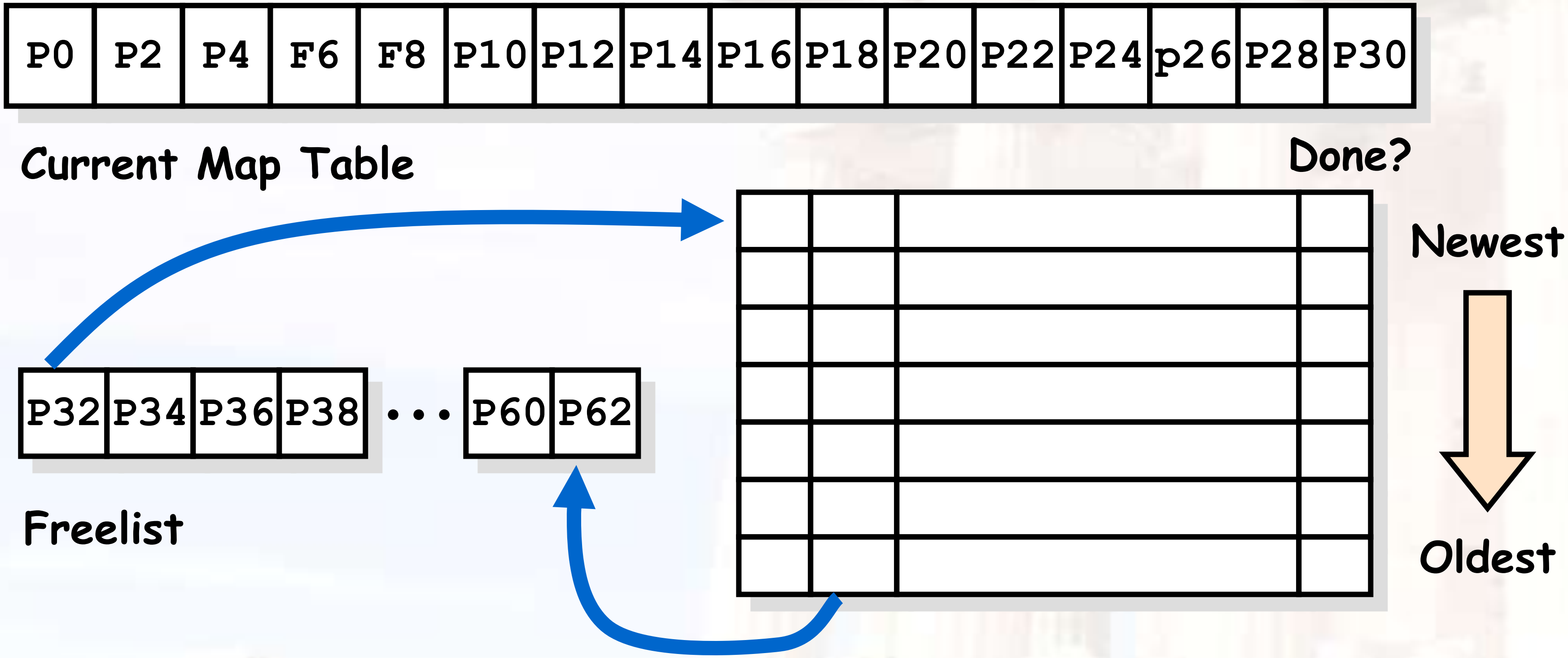


Explicit register renaming: (R1000 Style)



P0	P2	P4	F6	F8	P10	P12	P14	P16	P18	P20	P22	P24	p26	P28	P30

Current Map Table

Done?

Newest

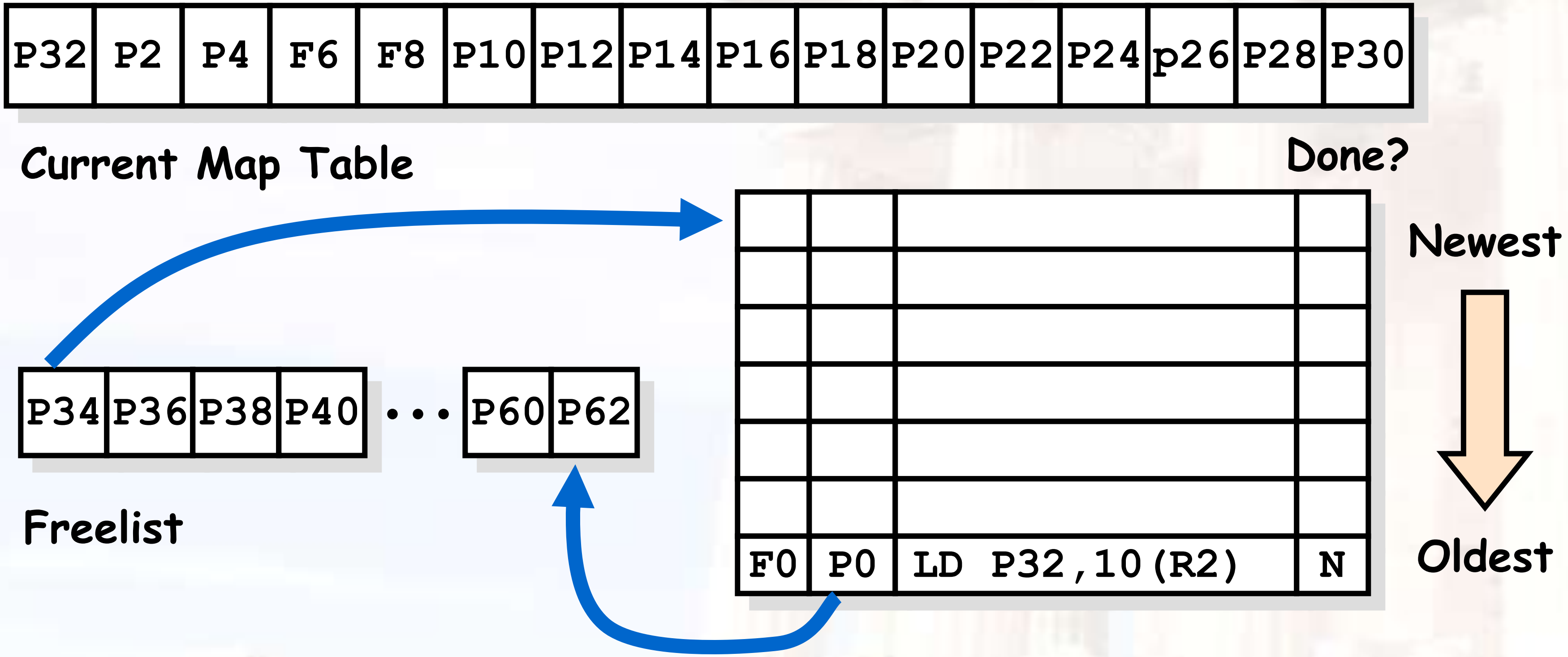
Oldest

P32 P34 P36 P38 ••• P60 P62

Freelist

- ❑Physical register file larger than ISA register file
- ❑On issue, each instruction that modifies a register is allocated new physical register from freelist

Explicit register renaming:
(R1000 Style)



P32	P2	P4	F6	F8	P10	P12	P14	P16	P18	P20	P22	P24	p26	P28	P30

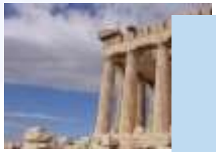
Current Map Table

F0	P0	LD P32,10(R2)	N

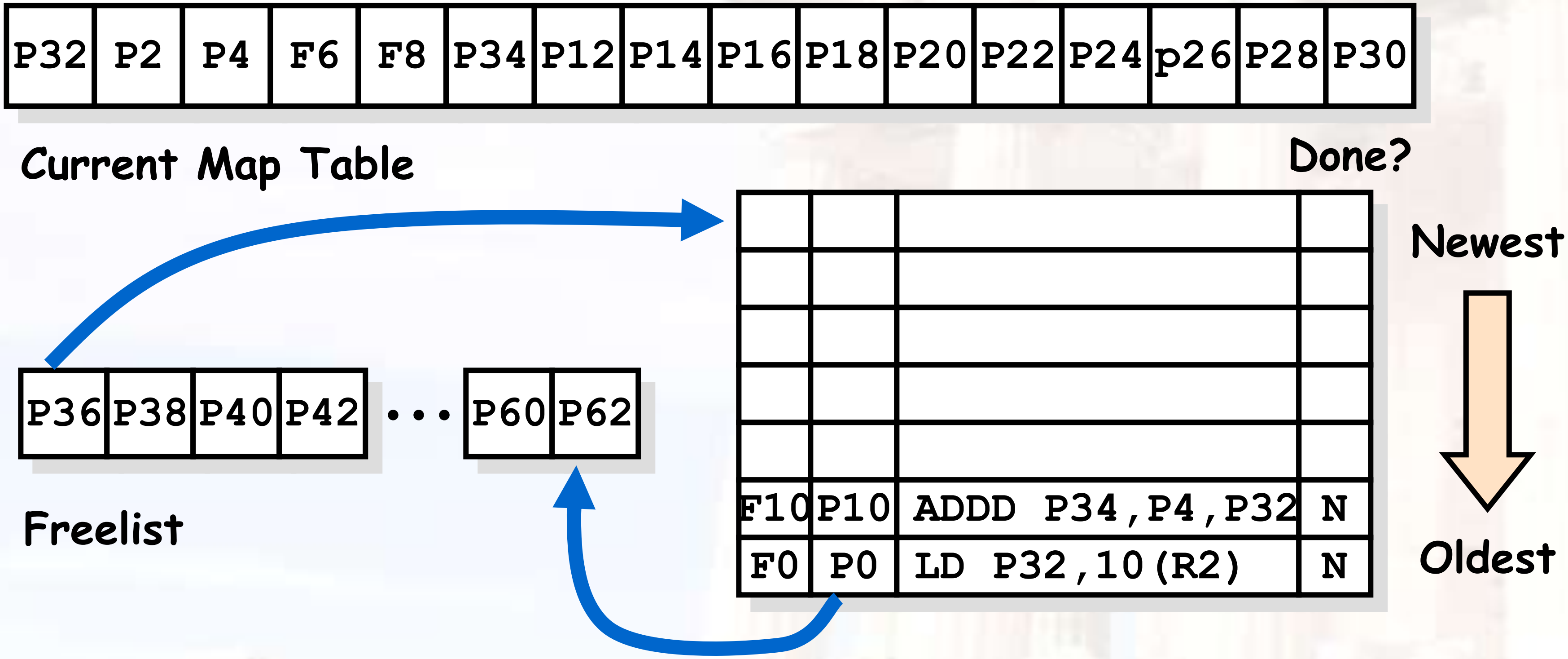
P34	P36	P38	P40	• • •	P60	P62

Freelist

- Note that physical register P0 is "dead" (or not "live") past the point of this load.
  - **When we go to commit the load, we free up**



Explicit register renaming: (R1000 Style)



P32	P2	P4	F6	F8	P34	P12	P14	P16	P18	P20	P22	P24	p26	P28	P30

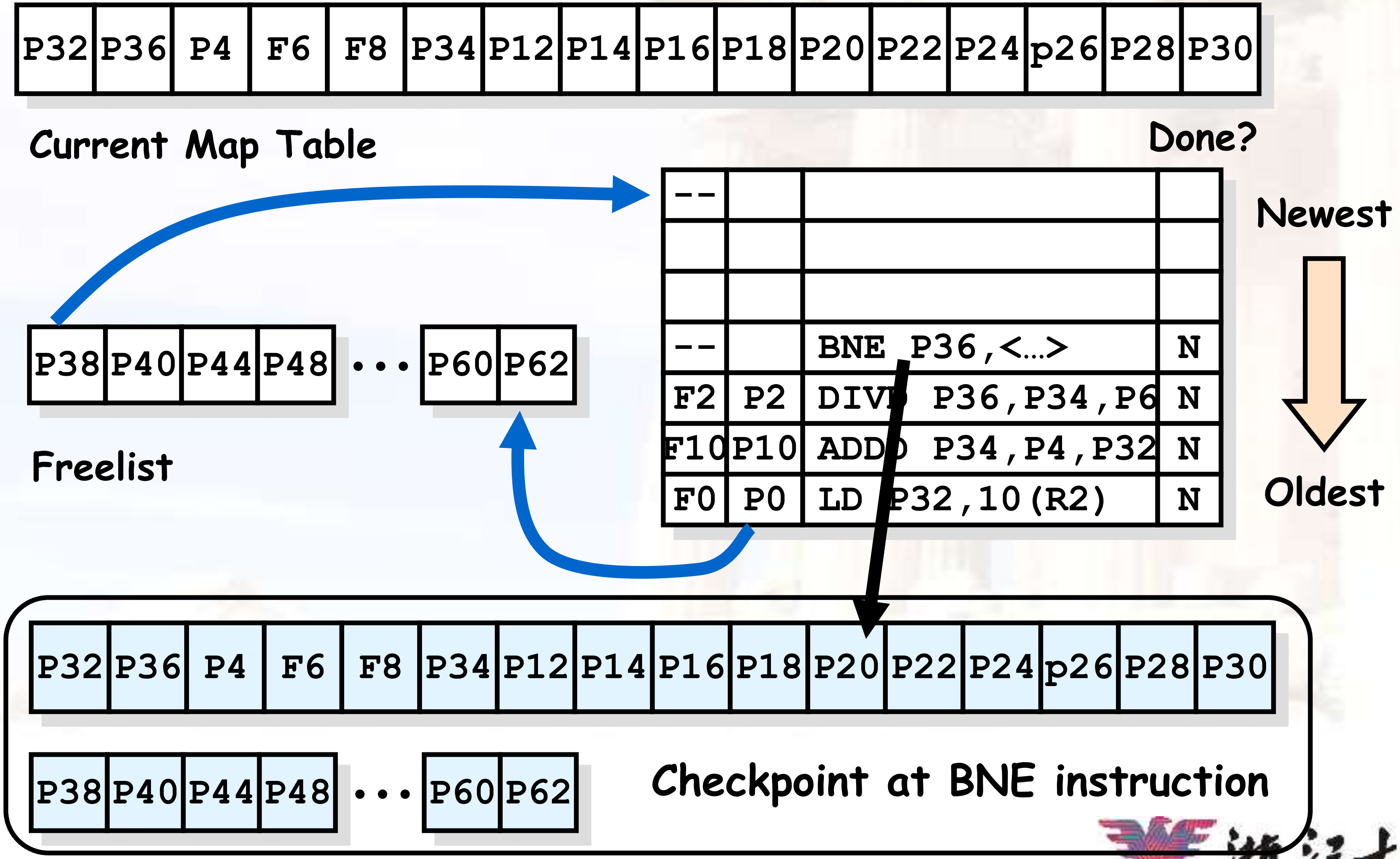
Current Map Table

P36	P38	P40	P42	•••	P60	P62

Freelist

			Done?
F10	P10	ADDD P34,P4,P32	N
F0	P0	LD P32,10(R2)	N

Explicit register renaming: (R1000 Style)



P32	P36	P4	F6	F8	P34	P12	P14	P16	P18	P20	P22	P24	p26	P28	P30

Current Map Table

			Done?
--			
--		BNE P36,<…>	N
F2	P2	DIVD P36,P34,P6	N
F10	P10	ADDD P34,P4,P32	N
F0	P0	LD P32,10(R2)	N

P38	P40	P44	P48	•••	P60	P62

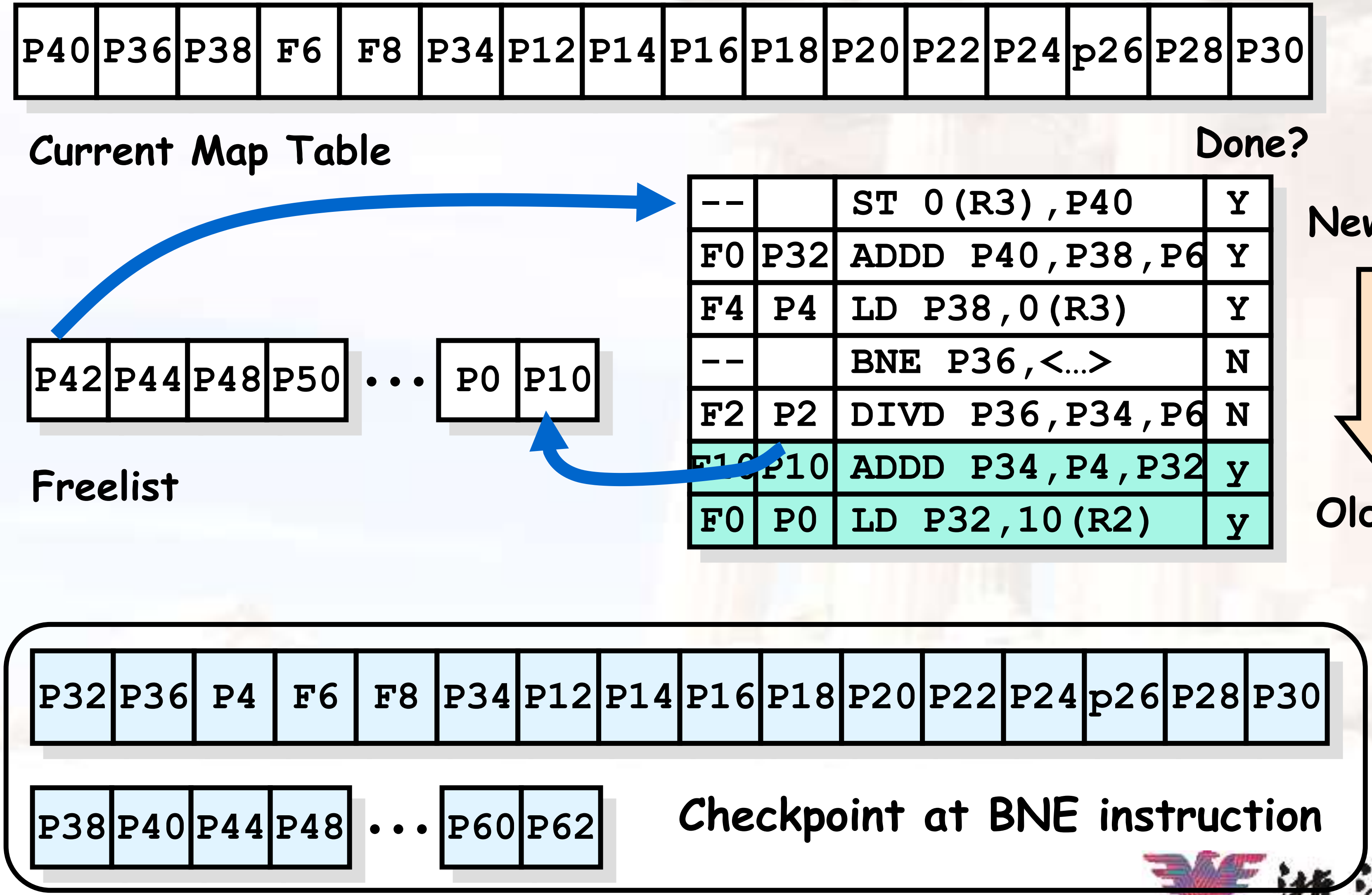
Freelist

P32	P36	P4	F6	F8	P34	P12	P14	P16	P18	P20	P22	P24	p26	P28	P30

P38	P40	P44	P48	•••	P60	P62

Checkpoint at BNE instruction

Explicit register renaming:
(R1000 Style)



P40	P36	P38	F6	F8	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

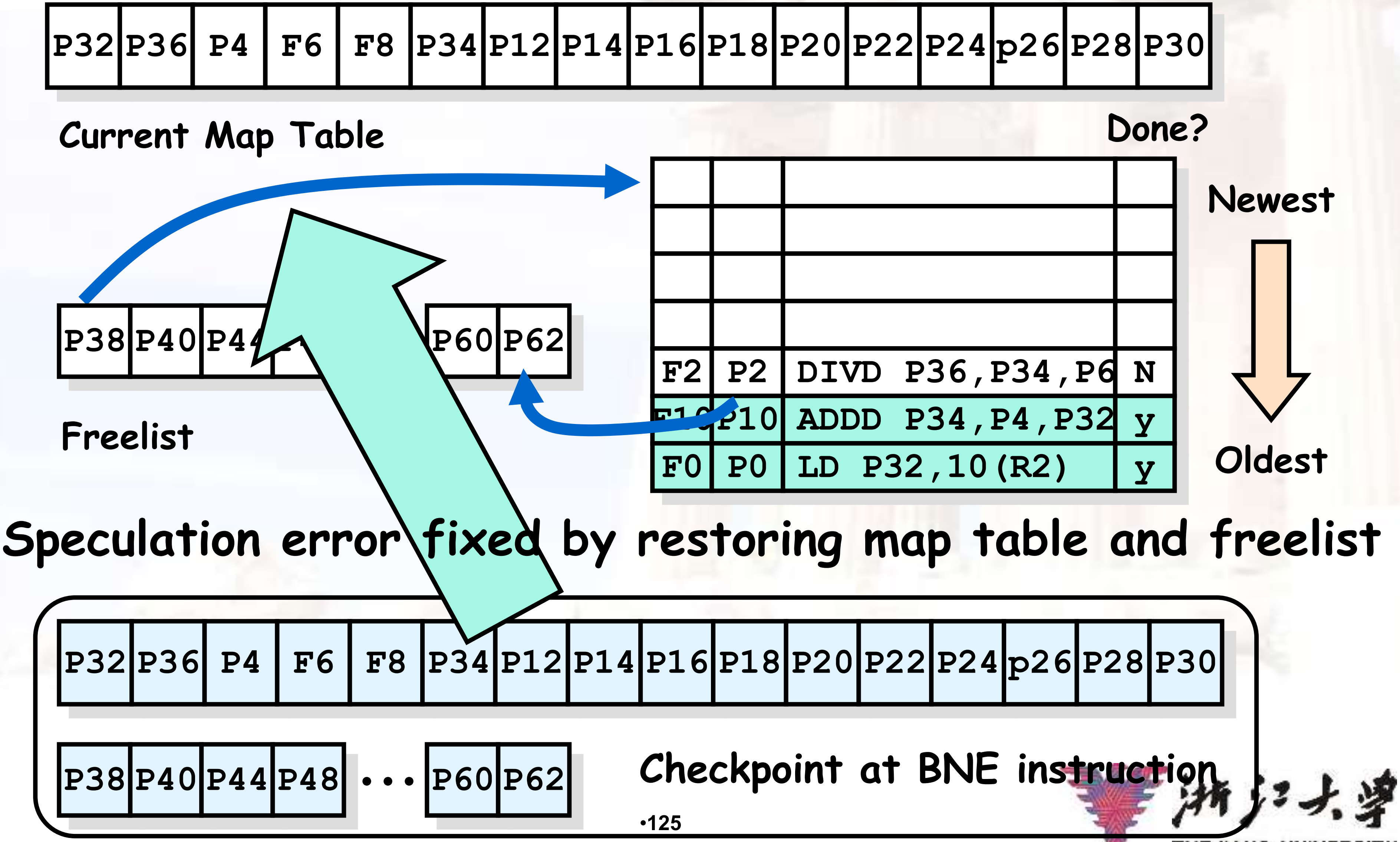
Freelist

P32	P36	P4	F6	F8	P34	P12	P14	P16	P18	P20	P22	P24	p26	P28	P30

P38	P40	P44	P48	•••	P60	P62

Checkpoint at BNE instruction

Explicit register renaming: (R1000 Style)



P32	P36	P4	F6	F8	P34	P12	P14	P16	P18	P20	P22	P24	p26	P28	P30

Current Map Table

			Done?
F2	P2	DIVD P36,P34,P6	N
F10	P10	ADDD P34,P4,P32	y
F0	P0	LD P32,10(R2)	y

P38	P40	P44			P60	P62

Freelist

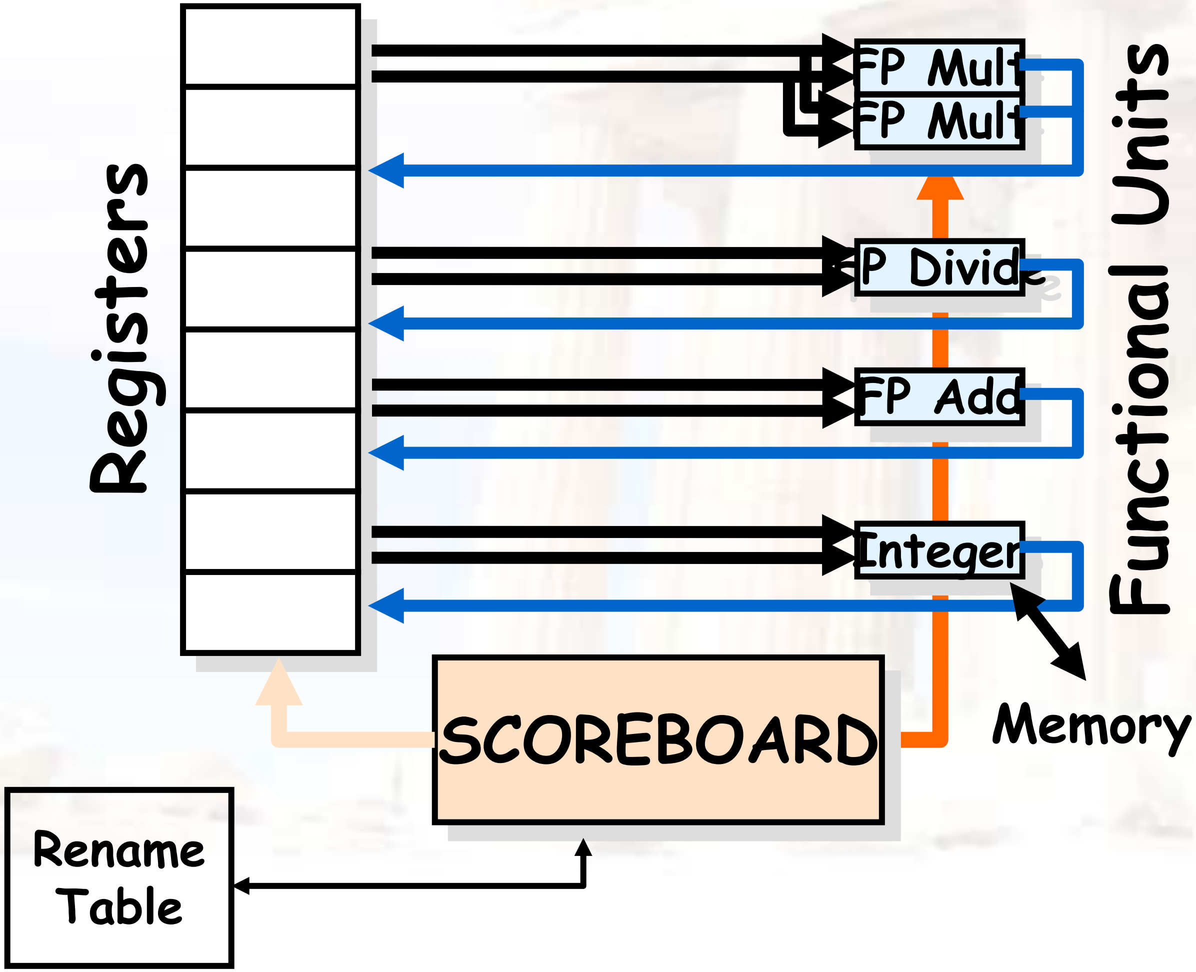
Speculation error fixed by restoring map table and freelist

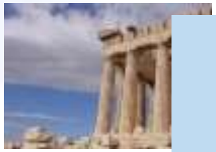
P32	P36	P4	F6	F8	P34	P12	P14	P16	P18	P20	P22	P24	p26	P28	P30

P38	P40	P44	P48	···	P60	P62

Checkpoint at BNE instruction

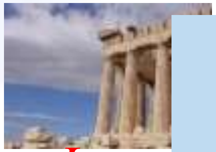
Can we use explicit register renaming with scoreboard?





Four Stages of Scoreboard Control With Explicit Renaming

- ❑Issue—decode instructions & check for structural hazards & allocate new physical register for result
  - ➢ Instructions issued in program order (for hazard checking)
  - ➢ Don't issue if no free physical registers
  - ➢ Don't issue if structural hazard
- ❑Read operands—wait until no hazards, read operands
  - ➢ All real dependencies (RAW hazards) resolved in this stage, since we wait for instructions to write back data.
- ❑Execution—operate on operands
  - ➢ The functional unit begins execution upon receiving operands. When the result is ready, it notifies the scoreboard
- ❑Write result—finish execution
- ❑Note: No checks for WAR or WAW hazards!



Scoreboard With Explicit 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:

				dest	S1	S2	FU	FU	Fj?	Fk?
Time	Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
	Int1	No								
	Int2	No								
	Mult1	No								
	Add	No								
	Divide	No								

Register Rename and Result

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
FU	P0	P2	P4	P6	P8	P10	P12		P30

- **Initialized Rename Table**

- Each instruction allocates free register
- Similar to single-assignment compiler transformation

Instruction status:

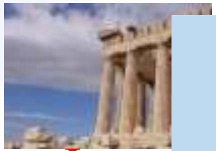
Instruction		j	k	Issue	*Read* *Oper*	Exec Start	*Exec* *Comp*	*Write* *Result*
LD	F6	34+	R2	1				
LD	F2	45+	R3					
MULTD	F0	F2	F4					
SUBD	F8	F6	F2					
DIVD	F10	F0	F6					
ADDD	F6	F8	F2					

Functional unit status:

Time	*Name*	*Busy*	*Op*	*dest* *Fi*	*S1* *Fj*	*S2* *Fk*	*FU* *Qj*	*FU* *Qk*	*Fj?* *Rj*	*Fk?* *Rk*
	Int1	Yes	Load	P32		R2				Yes
	Int2	No								
	Mult1	No								
	Add	No								
	Divide	No								

Register Rename and Result

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
1	*FU*	P0	P2	P4	P32	P8	P10	P12		P30



Renamed Scoreboard 2

Instruction status:

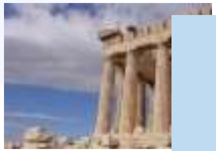
Instruction		*j*	*k*	*Issue*	*Read Oper*	Exec start	*Exec Comp*	*Write Result*
LD	F6	34+	R2	1	2			
LD	F2	45+	R3	2				
MULTD	F0	F2	F4					
SUBD	F8	F6	F2					
DIVD	F10	F0	F6					
ADDD	F6	F8	F2					

Functional unit status:

Time	*Name*	*Busy*	*Op*	*dest Fi*	*S1 Fj*	*S2 Fk*	*FU Qj*	*FU Qk*	*Fj? Rj*	*Fk? Rk*
	Int1	Yes	Load	P32		R2				No
	Int2	Yes	Load	P34		R3				Yes
	Mult1	No								
	Add	No								
	Divide	No								

Register Rename and Result

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
2	*FU*	P0	P34	P4	P32	P8	P10	P12		P30



Renamed Scoreboard 3

Instruction status:

Instruction		*j*	*k*	*Issue*	*Read Oper*	Exec start	*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*
	Int1	Yes	Load	P32		R2				No
	Int2	Yes	Load	P34		R3				No
	Mult1	Yes	Multd	P36	P34	P4	Int2		No	Yes
	Add	No								
	Divide	No								

Register Rename and Result

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
3	*FU*	P36	P34	P4	P32	P8	P10	P12		P30

• Next step Int1 will write result, where need the value?

Instruction status:

Instruction		*j*	*k*	*Issue*	*Read* *Oper*	Exec start	*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:

				dest	*S1*	*S2*	*FU*	*FU*	*Fj?*	*Fk?*
Time	*Name*	*Busy*	*Op*	*Fi*	*Fj*	*Fk*	*Qj*	*Qk*	*Rj*	*Rk*
	Int1	Yes	Load	P32		R2				No
	Int2	Yes	Load	P34		R3				No
	Mult1	Yes	Multd	P36	P34	P4	Int2		No	Yes
	Add	Yes	Sub	P38	P32	P34	Int1	Int2	No	No
	Divide	No								

Register Rename and Result

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
4	*FU*	P36	P34	P4	P32	P38	P10	P12		P30

• Next step Int2 will write result, where need the value?

Instruction status:

Instruction		j	k	*Issue*	*Read Oper*	Exec start	*Exec Comp*	*Write Result*
LD	F6	34+	R2	1	2	3	4	5
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:

				dest	*S1*	*S2*	*FU*	*FU*	*Fj?*	*Fk?*
Time	*Name*	*Busy*	*Op*	*Fi*	*Fj*	*Fk*	*Qj*	*Qk*	*Rj*	*Rk*
	Int1	No								
	Int2	Yes	Load	P34		R3				No
	Mult1	Yes	Multd	P36	P34	P4	Int2		No	Yes
	Add	Yes	Sub	P38	P32v	P34		Int2	Yes	No
	Divide	Yes	Divd	P40	P36	P32v	Mult1		No	Yes

Register Rename and Result

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
5	*FU*	P36	P34	P4	P32	P38	P40	P12		P30

- Why ADDD not issue ? Structure hazard ! Adder is occupied by with SUBD.

Instruction status:

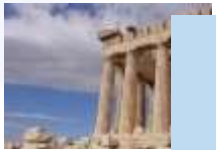
Instruction		*j*	*k*	*Issue*	*Read Oper*	Exec start	*Exec Comp*	*Write Result*
LD	F6	34+	R2	1	2	3	4	5
LD	F2	45+	R3	2	3	4	5	6
MULTD	F0	F2	F4	3				
SUBD	F8	F6	F2	4				
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2					

Functional unit status:

Time	*Name*	*Busy*	*Op*	*dest Fi*	*S1 Fj*	*S2 Fk*	*FU Qj*	*FU Qk*	*Fj? Rj*	*Fk? Rk*
	Int1	No								
	Int2	No								
	Mult1	Yes	Multd	P36	P34v	P4			Yes	Yes
	Add	Yes	Sub	P38	P32v	P34v			Yes	Yes
	Divide	Yes	Divd	P40	P36	P32v	Mult1		No	Yes

Register Rename and Result

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
6	*FU*	P36	P34	P4	P32	P38	P40	P12		P30



Renamed Scoreboard 7

Instruction status:

Instruction		*j*	*k*	*Issue*	*Read Oper*	Exec start	*Exec Comp*	*Write Result*
LD	F6	34+	R2	1	2	3	4	5
LD	F2	45+	R3	2	3	4	5	6
MULTD	F0	F2	F4	3	7			
SUBD	F8	F6	F2	4	7			
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2					

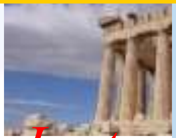
Functional unit status:

Time	*Name*	*Busy*	*Op*	*dest Fi*	*S1 Fj*	*S2 Fk*	*FU Qj*	*FU Qk*	*Fj? Rj*	*Fk? Rk*
	Int1	No								
	Int2	No								
	Mult1	Yes	Multd	P36	P34v	P4			No	No
	Add	Yes	Sub	P38	P32v	P34v			No	No
	Divide	Yes	Divd	P40	P36	P32	Mult1		No	Yes

Register Rename and Result

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
7	*FU*	P36	P34	P4	P32	P38	P40	P12		P30

Next step Adder will complete execution



Renamed Scoreboard 8

Instruction status:

Instruction		*j*	*k*	*Issue*	*Read* *Oper*	Exec start	*Exec* *Comp*	*Write* *Result*
LD	F6	34+	R2	1	2	3	4	5
LD	F2	45+	R3	2	3	4	5	6
MULTD	F0	F2	F4	3	7	8		
SUBD	F8	F6	F2	4	7	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*
	Int1	No								
	Int2	No								
9	Mult1	Yes	Multd	P36	P34v	P4			No	No
1	Add	Yes	Sub	P38	P32v	P34v			No	No
	Divide	Yes	Divd	P40	P36	P32v	Mult1		No	Yes

Register Rename and Result

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
8	*FU*	P36	P34	P4	P32	P38	P40	P12		P30

Renamed Scoreboard 9

- Next step Adder will write result, where need the value?

LD	F2	45+	R3	2	3	4	5	6
MULTD	F0	F2	F4	3	7	8		
SUBD	F8	F6	F2	4	7	8	9	
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2					

Functional unit status:

				dest	*S1*	*S2*	*FU*	*FU*	*Fj?*	*Fk?*
Time	*Name*	*Busy*	*Op*	*Fi*	*Fj*	*Fk*	*Qj*	*Qk*	*Rj*	*Rk*
	Int1	No								
	Int2	No								
8	Mult1	Yes	Multd	P36	P34v	P4			No	No
0	Add	Yes	Sub	P38	P32v	P34v			No	No
	Divide	Yes	Divd	P40	P36	P32v	Mult1		No	Yes

Register Rename and Result

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
9	*FU*	P36	P34	P4	P32	P38	P40	P12		P30

ZHEJIANG UNIVERSITY

Adder is cleared, so ADDD can be issued next cycle.

Instruction status:

Instruction		j	k	Issue	Read Oper		Exec Comp	Write Result
LD	F6	34+	R2	1	2	3	4	5
LD	F2	45+	R3	2	3	4	5	6
MULTD	F0	F2	F4	3	7	8		
SUBD	F8	F6	F2	4	7	8	9	10
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
	Int1	No								
	Int2	No								
7	Mult1	Yes	Multd	P36	P34v	P4			No	No
	Add	No								
	Divide	Yes	Divd	P40	P36	P32v	Mult1		No	Yes

Register Rename and Result

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
10	*FU*	P36	P34	P4	P32	P38	P40	P12		P30

Notice that P32 not listed in Rename Table
– Still live. Must not be reallocated by accident

Instruction status:

Instruction		*j*	*k*	*Issue*	*Read Oper*		*Exec Comp*	*Write Result*
LD	F6	34+	R2	1	2	3	4	5
LD	F2	45+	R3	2	3	4	5	6
MULTD	F0	F2	F4	3	7	8		
SUBD	F8	F6	F2	4	7	8	9	10
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2	11				

WAR dependence

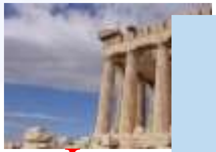
Functional unit status:

Time	*Name*	*Busy*	*Op*	*dest Fi*	*S1 Fj*	*S2 Fk*	*FU Qj*	*FU Qk*	*Fj? Rj*	*Fk? Rk*
	Int1	No								
	Int2	No								
6	Mult1	Yes	Multd	P36	P34	P4			No	No
	Add	Yes	Addd	P42	P38	P34			Yes	Yes
	Divide	Yes	Divd	P40	P36	P32	Mult1		No	Yes

WAR Hazard gone!

Register Rename and Result

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
11	*FU*	P36	P34	P4	P42	P38	P40	P12		P30



Renamed Scoreboard 12

Instruction status:

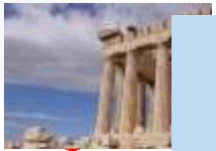
Instruction		*j*	*k*	*Issue*	*Read Oper*	Exec start	*Exec Comp*	*Write Result*
LD	F6	34+	R2	1	2	3	4	5
LD	F2	45+	R3	2	3	4	5	6
MULTD	F0	F2	F4	3	7	8		
SUBD	F8	F6	F2	4	7	8	9	10
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2	11	12			

Functional unit status:

Time	*Name*	*Busy*	*Op*	*dest Fi*	*S1 Fj*	*S2 Fk*	*FU Qj*	*FU Qk*	*Fj? Rj*	*Fk? Rk*
	Int1	No								
	Int2	No								
5	Mult1	Yes	Multd	P36	P34	P4			No	No
2	Add	Yes	Addd	P42	P38	P34			Yes	Yes
	Divide	Yes	Divd	P40	P36	P32	Mult1		No	Yes

Register Rename and Result

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
12	*FU*	P36	P34	P4	P42	P38	P40	P12		P30



Renamed Scoreboard 13

Instruction status:

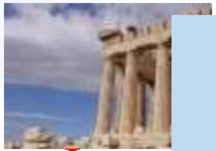
Instruction		j	k	*Issue*	*Read Oper*	Exec start	*Exec Comp*	*Write Result*
LD	F6	34+	R2	1	2	3	4	5
LD	F2	45+	R3	2	3	4	5	6
MULTD	F0	F2	F4	3	7	8		
SUBD	F8	F6	F2	4	7	8	9	10
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2	11	12	13		

Functional unit status:

Time	*Name*	*Busy*	*Op*	*dest Fi*	*S1 Fj*	*S2 Fk*	*FU Qj*	*FU Qk*	*Fj? Rj*	*Fk? Rk*
	Int1	No								
	Int2	No								
4	Mult1	Yes	Multd	P36	P34	P4			No	No
1	Add	Yes	Addd	P42	P38	P34			Yes	Yes
	Divide	Yes	Divd	P40	P36	P32	Mult1		No	Yes

Register Rename and Result

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
13	*FU*	P36	P34	P4	P42	P38	P40	P12		P30



Renamed Scoreboard 14

Instruction status:

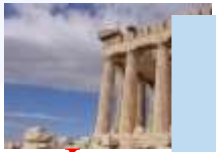
Instruction		*j*	*k*	*Issue*	*Read* *Oper*	Exec start	*Exec* *Comp*	*Write* *Result*
LD	F6	34+	R2	1	2	3	4	5
LD	F2	45+	R3	2	3	4	5	6
MULTD	F0	F2	F4	3	7	8		
SUBD	F8	F6	F2	4	7	8	9	10
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2	11	12	13	14	

Functional unit status:

Time	*Name*	*Busy*	*Op*	*dest* *Fi*	*S1* *Fj*	*S2* *Fk*	*FU* *Qj*	*FU* *Qk*	*Fj?* *Rj*	*Fk?* *Rk*
	Int1	No								
	Int2	No								
3	Mult1	Yes	Multd	P36	P34	P4			No	No
0	Add	Yes	Addd	P42	P38	P34			Yes	Yes
	Divide	Yes	Divd	P40	P36	P32	Mult1		No	Yes

Register Rename and Result

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
14	*FU*	P36	P34	P4	P42	P38	P40	P12		P30



Renamed Scoreboard 15

Instruction status:

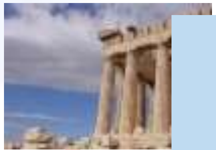
Instruction		*j*	*k*	*Issue*	*Read Oper*	Exec start	*Exec Comp*	*Write Result*
LD	F6	34+	R2	1	2	3	4	5
LD	F2	45+	R3	2	3	4	5	6
MULTD	F0	F2	F4	3	7	8		
SUBD	F8	F6	F2	4	7	8	9	10
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2	11	12	13	14	15

Functional unit status:

Time	*Name*	*Busy*	*Op*	*dest Fi*	*S1 Fj*	*S2 Fk*	*FU Qj*	*FU Qk*	*Fj? Rj*	*Fk? Rk*
	Int1	No								
	Int2	No								
2	Mult1	Yes	Multd	P36	P34	P4			No	No
	Add	No								
	Divide	Yes	Divd	P40	P36	P32	Mult1		No	Yes

Register Rename and Result

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
15	*FU*	P36	P34	P4	P42	P38	P40	P12		P30



Renamed Scoreboard 16

Instruction status:

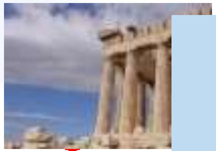
					Read	Exec	*Exec*	*Write*
Instruction		*j*	*k*	*Issue*	*Oper*	start	*Comp*	*Result*
LD	F6	34+	R2	1	2	3	4	5
LD	F2	45+	R3	2	3	4	5	6
MULTD	F0	F2	F4	3	7	8		
SUBD	F8	F6	F2	4	7	8	9	10
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2	11	12	13	14	15

Functional unit status:

				dest	*S1*	*S2*	*FU*	*FU*	*Fj?*	*Fk?*
Time	*Name*	*Busy*	*Op*	*Fi*	*Fj*	*Fk*	*Qj*	*Qk*	*Rj*	*Rk*
	Int1	No								
	Int2	No								
1	Mult1	Yes	Multd	P36	P34	P4			No	No
	Add	No								
	Divide	Yes	Divd	P40	P36	P32	Mult1		No	Yes

Register Rename and Result

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
16	*FU*	P36	P34	P4	P42	P38	P40	P12		P30



Renamed Scoreboard 17

Instruction status:

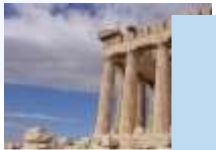
Instruction		j	k	Issue	Read Oper	Exec start	Exec Comp	Write Result
LD	F6	34+	R2	1	2	3	4	5
LD	F2	45+	R3	2	3	4	5	6
MULTD	F0	F2	F4	3	7	8	17	
SUBD	F8	F6	F2	4	7	8	9	10
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2	11	12	13	14	15

Functional unit status:

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU Qj	FU Qk	Fj? Rj	Fk? Rk
	Int1	No								
	Int2	No								
0	Mult1	Yes	Multd	P36	P34	P4			No	No
	Add	No								
	Divide	Yes	Divd	P40	P36	P32	Mult1		No	Yes

Register Rename and Result

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
17	FU	P36	P34	P4	P42	P38	P40	P12		P30



Renamed Scoreboard 18

Instruction status:

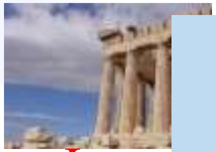
Instruction		*j*	*k*	*Issue*	*Read Oper*	*Exec Start*	*Exec Comp*	*Write Result*
LD	F6	34+	R2	1	2	3	4	5
LD	F2	45+	R3	2	3	4	5	6
MULTD	F0	F2	F4	3	7	8	17	18
SUBD	F8	F6	F2	4	7	8	9	10
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2	11	12	13	14	15

Functional unit status:

Time	*Name*	*Busy*	*Op*	*dest Fi*	*S1 Fj*	*S2 Fk*	*FU Qj*	*FU Qk*	*Fj? Rj*	*Fk? Rk*
	Int1	No								
	Int2	No								
	Mult1	No								
	Add	No								
	Divide	Yes	Divd	P40	P36v	P32	~~Mult1~~		**Yes**	**Yes**

Register Rename and Result

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
18	*FU*	P36	P34	P4	P42	P38	P40	P12		P30



Renamed Scoreboard 19

Instruction status:

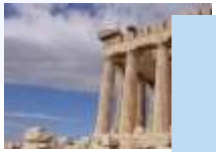
Instruction		*j*	*k*	*Issue*	*Read Oper*	Exec start	*Exec Comp*	*Write Result*
LD	F6	34+	R2	1	2	3	4	5
LD	F2	45+	R3	2	3	4	5	6
MULTD	F0	F2	F4	3	7	8	17	18
SUBD	F8	F6	F2	4	7	8	9	10
DIVD	F10	F0	F6	5	19			
ADDD	F6	F8	F2	11	12	13	14	15

Functional unit status:

Time	*Name*	*Busy*	*Op*	*dest Fi*	*S1 Fj*	*S2 Fk*	*FU Qj*	*FU Qk*	*Fj? Rj*	*Fk? Rk*
	Int1	No								
	Int2	No								
	Mult1	No								
	Add	No								
40	Divide	Yes	Divd	P40	P36	P32	Mult1		**NO**	**NO**

Register Rename and Result

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
19	*FU*	P36	P34	P4	P42	P38	P40	P12		P30



Summary #2

- ❑Explicit Renaming: more physical registers than needed by ISA.
  - ➢Separates *renaming* from *scheduling*
    - Opens up lots of options for resolving RAW hazards
  - ➢Rename table: tracks current association between architectural registers and physical registers
  - ➢Potentially complicated rename table management