Out-of-Order Execution: Scoreboard

Computer Architecture

Wei Wang

Text Book Chapters

- “Computer Architecture” Hennessy and Patterson, Chapter 3.4 and 3.5 about “Dynamic Scheduling”.

Road Map

Out-of-Order Execution and Dynamic Scheduling

The Scoreboard Scheduling Algorithm

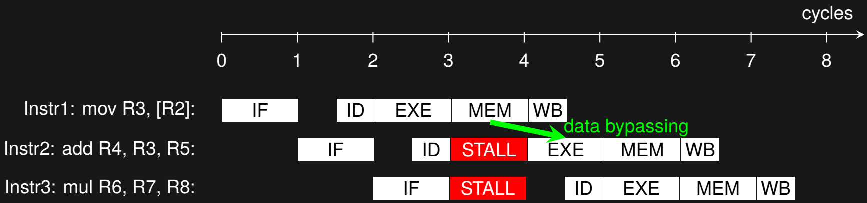
An Example of Scoreboard Scheduling Algorithm

Scoreboarding Limitations

Acknowledgment

Data Hazards That Still Need Stalls

- Recall that for load-use hazards, data forwarding cannot completely remove stalls.
  - The problem is more serious on modern processors, as the memory is much slower than the processors.



cycles

0 1 2 3 4 5 6 7 8

Instr1: mov R3, [R2]: IF ID EXE MEM WB

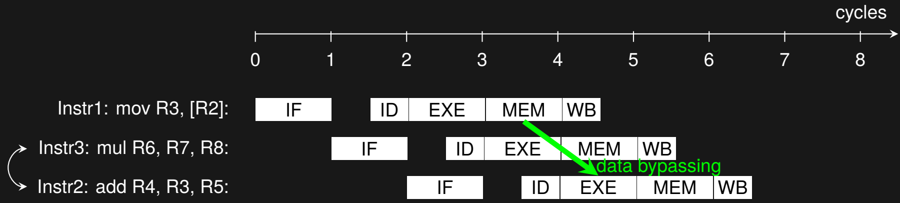
Instr2: add R4, R3, R5: IF ID STALL EXE MEM WB

Instr3: mul R6, R7, R8: IF STALL ID EXE MEM WB

data bypassing

Reordering Instructions to Remove Stalls

- Some stalls can be resolved by reordering the executions of instructions.
  - For example for the previous example, we can switch instruction 2 and instruction 3 to remove the stalls.
    - Note that data bypassing is still required to completely remove the stalls due to the data hazards.



cycles

0 1 2 3 4 5 6 7 8

Instr1: mov R3, [R2]: IF ID EXE MEM WB

Instr3: mul R6, R7, R8: IF ID EXE MEM WB

Instr2: add R4, R3, R5: IF ID EXE MEM WB

data bypassing

Instruction Reordering and Dependencies

- The intuition of the benefit of instructions reordering:
  - Reorder the instructions to keep the processor busy (so fewer stalls).
- What instructions can be reordered:
  - Instructions that do not have dependencies among them.
  - In the previous example, instruction 2 and instruction 3 do not have dependency.
  - Dependencies prevents reordering: RAW, WAR, and WAW.

Instruction-reordering with Compiler and Static Scheduling

- Theoretically, instruction reordering can be done with compilers.
  - Such a reordering is called Static (instruction) Scheduling.
- However, it is very difficult for the compilers to find the optimal order of instructions.
  - Most memory accesses are (register-) indirect. Therefore, compilers do not know the memory addresses at the compilation time, and thus, it is impossible to for the compilers to know the dependencies at compilation time.
    - Many research efforts spent on finding the dependencies with compilers, but eventually failed.
  - Optimal orders are hardware dependent. Consequently, the optimality of statically ordered code is not portable.

Out-of-Order Execution and Dynamic Scheduling

- Instruction reordering can also be done in the processors.
  - Memory addresses are known at run-time.
  - Optimal orders can be determined for each CPU.
- Executing the instructions in an order different than the order specified by the compiler is called Out-of-Order (OoO) Execution.
- Dynamic Scheduling or Dynamic Execution are better terms (than OoO), as they describe the underlying implementation idea of OoO CPUs: the instructions are scheduled to execute dynamically instead of using a fixed “static” order.

OoO Completion

- Besides execution, instructions also go through completion stages (MEM-write and/or WB).
  - Instructions can be Out-of-Order(-ly) completed.
    - OoO Completion makes it difficult to do precise exception and can break the memory/data consistency.
    - Before early 70s, OoO CPUs just ignored these problems...
    - Most modern processors are in-order completion. That is, registers and memory are written following the instruction order specified by the compiler.

Hardware Components Required for OoO Execution

- A hardware unit to detect instruction dependencies.
  - Control Unit or the decoder is already doing this for data forwarding.
- A hardware unit to hold un-executed and/or un-issued instructions.
  - The Scoreboard in Scoreborad Algorithm.
  - The Reservation Station in Tomasulo's Algorithm.
- For in-order completion, a hardware unit to buffer results before writing them to the registers or memory.
  - The Reorder Buffer in Tomasulo's Algorithm (can also be used with Scoreboard).

Road Map

Out-of-Order Execution and Dynamic Scheduling

The Scoreboard Scheduling Algorithm

An Example of Scoreboard Scheduling Algorithm

Scoreboarding Limitations

Acknowledgment

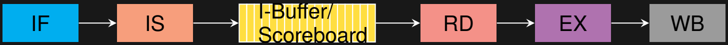
History of Scoreboarding

- Originally proposed in CDC6600 (by Seymour Cray, 1964).
  - The first machine that used OoO execution.
  - The algorithm we learn in this course is based on CDC6600.
- Probably not widely used after CDC6600, since Tomasulo’s Algorithm was invented 3 years later.
- But the Tomasulo’s Algorithm and modern OoO implementations were not fundamentally different than the Scoreboard Algorithm.

The Scoreboard Pipeline

- For Scoreboarding, to support OoO execution, we will split the ID stage into two new stages:
  - Issue (IS) stage: where an instruction is decoded and issued to occupy a function unit for execution.
  - Read Operands (RD) stage: where an instruction reads its source operands, either from registers or from the outputs of other function units (like data forwarding).
- After issue, an instruction's state is stored in the scoreboard or instruction buffer.
- No MEM stage. Memory (load/store) units are simply treated similarly as Arithmetic units. So EX stage also does memory accesses.

New Pipeline for Scoreboarding:



Scoreboarding

- Centralized Scheme.
  - Scoreboard controls the registers and functions units.
  - No data bypassing (can be added).
  - Need to be careful with WAR/WAW hazards.

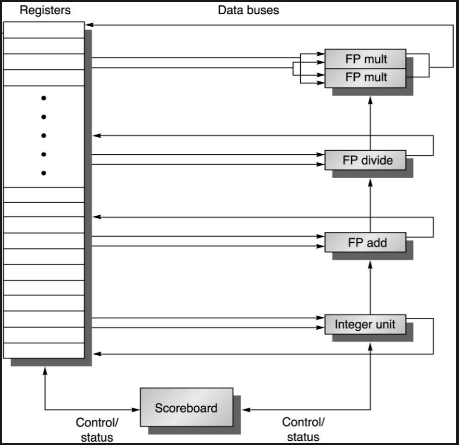


Figure: Hardware Structure for Scoreboarding.

Scoreboarding Stages – Issue

- IF stage: Fetch instruction, same as before.
- Issue stage: Issue the instruction
  - If the functional unit (FU) is free and no other active instructions has same destination register (no WAW), then issue the instruction.
  - Do not issue until FU is free (no structural hazards)
  - Instruction can be issued, but stalled and waiting in the Scoreboard
  - Size of scoreboard is also a structure hazard
    - Fetch and issue may be stalled if scoreboard is filled
  - Note: Issue is in-order: if an instruction cannot be issued, all following instructions cannot be issued.

Scoreboarding Stages – Read Operands

- Read Operands (check data hazards)
  - Check scoreboard for whether source operands are available
  - Available only if
    - Source operand is not produced by previously issued active instructions.
    - No currently active FU is going to write it.
  - Dynamically avoids RAW hazards
    - Instructions may finish RD stage out-of-order, which dynamically avoids RAW hazards.

Scoreboarding Stages – Execution, Memory Access and Write Back

- Execution and Memory Access
  - Execute, read/write memory and update scoreboard status.
- Write Back
  - Scoreboard checks for WAR hazards, and stalls the instruction from writing back (completing) if WAR hazard exists.
  - Unlike the simple pipeline, stalls can now occur at both the beginning and the end of the pipeline.
    - Recall that in the simple pipeline, stalls only occurs before the execution stage.

Scoreboarding and Data Hazards

- A summery of how scoreboarding algorithm avoids data hazards:
  - WAW: IS stage checks WAW hazards. If detected, do not issue instruction
  - RAW: RD stages checks RAW hazards. Instruction is stalled until both source operands are ready, dynamically avoids RAW hazards.
  - WAR: WB stage checks WAR hazards. If detected, stalls WB until WAR hazard is resolved.

The Scoreboard

- The Scoreboard has mainly three components:
  - Instruction status bits:
    - Indicate which of the four stages instruction is in.
  - Functional Unit status bits:
    - Busy bit: FU is occupied by an active instruction or not;
    - Operation (OP) field: Operation to perform in the unit (e.g. MUL, DIV or MOD);
    - F_i field: destination register ID;
    - F_j and F_k fields: source register IDs.
    - Q_j and Q_k fields: FUs that produces source registers F_j and F_k
    - R_j and R_k bits: indicting whether F_j and F_k are ready but not yet read
  - Register results status:
    - Which FU will write each register.

Road Map

Out-of-Order Execution and Dynamic Scheduling

The Scoreboard Scheduling Algorithm

An Example of Scoreboard Scheduling Algorithm

Scoreboarding Limitations

Acknowledgment

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11				
LD	R2	45+	R13				
MUL	R0	R2	R4				
SUB	R8	R6	R2				
DIV	R10	R0	R6				
ADD	R6	R8	R2				

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer	No								
	Multi1	No								
	Multi2	No								
	Add	No								
	Divide	No								

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU												

Clock: cycle

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1			
LD	R2	45+	R13				
MUL	R0	R2	R4				
SUB	R8	R6	R2				
DIV	R10	R0	R6				
ADD	R6	R8	R2				

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer	Yes	Load	R6		R11				Yes
	Multi1	No								
	Multi2	No								
	Add	No								
	Divide	No								

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU							Integer					

Clock: cycle 1

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1			
LD	R2	45+	R13				
MUL	R0	R2	R4				
SUB	R8	R6	R2				
DIV	R10	R0	R6				
ADD	R6	R8	R2				

Issue the first instruction at cycle 1.

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer	Yes	Load	R6		R11				Yes
	Multi1	No								
	Multi2	No								
	Add	No								
	Divide	No								

Issue instruction 1; occupy the Integer unit with the first instruction.

Note that R_k should be "Yes", indicating source operand F_k is read for read.

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	… … …
FU							Integer					

Register R6 will be filled with the result from the Integer unit.

Clock: cycle 1

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2		
LD	R2	45+	R13				
MUL	R0	R2	R4				
SUB	R8	R6	R2				
DIV	R10	R0	R6				
ADD	R6	R8	R2				

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer	Yes	Load	R6		R11				Yes
	Multi1	No								
	Multi2	No								
	Add	No								
	Divide	No								

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU							Integer					

Clock: cycle 2

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2		
LD	R2	45+	R13				
MUL	R0	R2	R4				

Instruction 1 reads registers at cycle 2.

Instruction 2 should not be issued, as it also uses the Integer Function Unit, which is in-use by instruction 1. That is, there is a structural hazard preventing instruction 2 from being issued.

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer	Yes	Load	R6		R11				Yes
	Multi1	No								
	Multi2	No								
	Add	No								
	Divide	No								

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU							Integer					

Clock: cycle 2

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	
LD	R2	45+	R13				
MUL	R0	R2	R4				
SUB	R8	R6	R2				
DIV	R10	R0	R6				
ADD	R6	R8	R2				

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer	Yes	Load	R6		R11				No
	Multi1	No								
	Multi2	No								
	Add	No								
	Divide	No								

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU							Integer					

Clock: cycle 3

A Example of Scoreboard Algorithm

Instruction Unit Status:

Op	Dest i	Src1 j	Src2 k	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	
LD	R2	45+	R13				
MUL	R0	R2	R4				
SUB	R8	R6	R2				

Instruction 1 executes at cycle 3.

If instruction 2 cannot be issued, none of the following instructions can be issued, as Scoreboard is in-order issue.

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer	Yes	Load	R6		R11				No
	Multi1	No								
	Multi2	No								
	Add	No								
	Divide	No								

Note that R_k changed to "No", indicating source operand is consumed.

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	… … …
FU							Integer					

Clock: cycle 3

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13				
MUL	R0	R2	R4				
SUB	R8	R6	R2				
DIV	R10	R0	R6				
ADD	R6	R8	R2				

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer	No								
	Multi1	No								
	Multi2	No								
	Add	No								
	Divide	No								

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU							Integer					

Clock: cycle 4

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13				
MUL	R0	R2	R4				
SUB	R8	R6	R2				
DIV	R10	R0	R6				
ADD	R6	R8	R2				

Instruction 1 writes back to register R6 at cycle 4.

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer	No								
	Multi1	No								
	Multi2									
	Add									
	Divide	No								

Instruction 1 releases the Integer Unit after execution.

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU							Integer					

Clock: cycle 4

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5			
MUL	R0	R2	R4				
SUB	R8	R6	R2				
DIV	R10	R0	R6				
ADD	R6	R8	R2				

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer	Yes	Load	R2		R13				Yes
	Multi1	No								
	Multi2	No								
	Add	No								
	Divide	No								

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	… … …
FU			Integer									

Clock: cycle 5

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest i	Src1 j	Src2 k	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5			
MUL	R0	R2	R4				
SUB	R8	R6	R2				
DIV	R10	R0	R6				
ADD	R6	R8	R2				

Issue instruction 2 at cycle 5.

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer	Yes	Load	R2		R13				Yes
	Multi1	No								
	Multi2	No								
	Add	No								
	Divide	No								

Issue instruction 2; occupy the Integer unit.

Note that R_k should be "Yes", indicating source operand F_k is read for read.

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU			Integer									

Register R2 will be filled with the result from the Integer unit.

Clock: cycle 5

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6		
MUL	R0	R2	R4	6			
SUB	R8	R6	R2				
DIV	R10	R0	R6				
ADD	R6	R8	R2				

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer	Yes	Load	R2		R13				Yes
	Multi1	Yes	Multi	R0	R2	R4	Integer		No	Yes
	Multi2	No								
	Add	No								
	Divide	No								

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	… … …
FU	Multi1		Integer									

Clock: cycle 6

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest i	Src1 j	Src2 k	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6		
MUL	R0	R2	R4	6			
SUB	R8	R6	R2				
DIV	R10	R0	R6				
ADD	R6	R8	R2				

At cycle 6, Instruction 2 reads registers, and Instruction 3 is issued.

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer	Yes	Load	R2		R13				Yes
	Multi1	Yes	Multi	R0	R2	R4	Integer		No	Yes
	Multi2	No								
	Add	No								
	Divide	No								

Issue instruction 3; occupy the Mutli1 unit. Note that, instruction 3 uses register R2, which is the output of Instruction 2 and the Integer Unit. Q_j is “Integer”, indicating Src1 is will be produced by the Integer unit. R_j is “No”, indicating Src1 (register R2) is not ready for read.

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU	Multi1		Integer									

Clock: cycle 6

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	
MUL	R0	R2	R4	6			
SUB	R8	R6	R2	7			
DIV	R10	R0	R6				
ADD	R6	R8	R2				

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer	Yes	Load	R2		R13				No
	Multi1	Yes	Multi	R0	R2	R4	Integer		No	Yes
	Multi2	No								
	Add	Yes	SUB	R8	R6	R2		Integer	Yes	No
	Divide	No								

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU	Multi1		Integer						Add			

Clock: cycle 7

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	
MUL	R0	R2	R4	6			
SUB	R8	R6	R2	7			
DIV	R10	R0	R6				
ADD	R6	R8	R2				

Cycle 7: Instruction 2 executes; Instruction 4 is issued; Instruction 3 stalled due to missing Src1 (register R2).

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer	Yes	Load	R2		R13				No
	Multi1	Yes	Multi	R0	R2	R4	Integer		No	Yes
	Multi2	No								
	Add	Yes	SUB	R8	R6	R2		Integer	Yes	No
	Divide	No								

Issue instruction 4; occupy the Add unit. Note that, instruction 4 uses register R2, which is the output of Instruction 2 and the Integer Unit. Q_k is “Integer”, indicating Src2 is will be produced by the Integer unit. R_k is “No”, indicating Src2 (register R2) is not ready for read.

Register Result Sta

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU	Multi1		Integer						Add			

Clock: cycle 7

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6			
SUB	R8	R6	R2	7			
DIV	R10	R0	R6	8			
ADD	R6	R8	R2				

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
	Multi1	Yes	Multi	R0	R2	R4			Yes	Yes
	Multi2	No								
	Add	Yes	SUB	R8	R6	R2			Yes	Yes
	Divide	Yes	Div	R10	R0	R6	Multi1		No	Yes

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU	Multi1		Integer						Add		Divide	

Clock: cycle 8

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6			
SUB	R8	R6	R2	7			
DIV	R10	R0	R6	8			
ADD	R6	R8	R2				

Cycle 8: Instruction 2 finishes execution, writes back to register; Instruction 5 is issued; Instruction 3 and 4 stalled due to missing source operands (register R2).

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
	Multi1	Yes	Multi	R0	R2	R4			Yes	Yes
	Multi2	No								
	Add	Yes	SUB	R8	R6	R2			Yes	Yes
	Divide	Yes	Div	R10	R0	R6	Multi1		No	Yes

Note that Multi1's R_j and Add's R_k switched to "Yes", indicating source operands are ready.

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7				...
FU	Multi1		Integer						Add		Divide	

Clock: cycle 8

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9		
SUB	R8	R6	R2	7	9		
DIV	R10	R0	R6	8			
ADD	R6	R8	R2				

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
	Multi1	Yes	Multi	R0	R2	R4			Yes	Yes
	Multi2	No								
	Add	Yes	SUB	R8	R6	R2			Yes	Yes
	Divide	Yes	Div	R10	R0	R6	Multi1		No	Yes

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU	Multi1								Add		Divide	

Clock: cycle 9

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9		
SUB	R8	R6	R2	7	9		
DIV	R10	R0	R6	8			
ADD	R6	R8	R2				

Cycle 9: Instruction 3 and 4 read operands; Instruction 5 stalled due to missing Src2 (register R0); Instruction 6 cannot be issued due to structural hazard at the Add unit.

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
	Multi1	Yes	Multi	R0	R2	R4			Yes	Yes
	Multi2	No								
	Add	Yes	SUB	R8	R6	R2			Yes	Yes
	Divide	Yes	Div	R10	R0	R6	Multi1		No	Yes

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU	Multi1								Add		Divide	

Clock: cycle 9

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	
SUB	R8	R6	R2	7	9	10	
DIV	R10	R0	R6	8			
ADD	R6	R8	R2				

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
9	Multi1	Yes	Multi	R0	R2	R4			No	No
	Multi2	No								
1	Add	Yes	SUB	R8	R6	R2			No	No
	Divide	Yes	Div	R10	R0	R6	Multi1		No	Yes

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU	Multi1								Add		Divide	

Clock: cycle 10

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	
SUB	R8	R6	R2	7	9	10	
DIV	R10	R0	R6	8			
ADD	R6	R8	R2				

Cycle 9: Instruction 3 and 4 executes; Instruction 5 stalled due to missing Src2 (register R0); Instruction 6 cannot be issued due to structural hazard at the Add unit.

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
9	Multi1	Yes	Multi	R0	R2	R4			No	No
	Multi2	No								
1	Add	Yes	SUB	R8	R6	R2			No	No
	Divide	Yes	Div	R10	R0	R6	Multi1		No	Yes

Multi1 unit takes 10 cycles to execute (9 cycles remaining), and Add unit takes 2 cycles to execute (1 cycles remaining).

Regist

		5	R6	R7	R8	R9	R10	... ...
FU	Multi1				Add		Divide	

Clock: cycle 10

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	
SUB	R8	R6	R2	7	9	10	
DIV	R10	R0	R6	8			
ADD	R6	R8	R2				

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
8	Multi1	Yes	Multi	R0	R2	R4			No	No
	Multi2	No								
0	Add	Yes	SUB	R8	R6	R2			No	No
	Divide	Yes	Div	R10	R0	R6	Multi1		No	Yes

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU	Multi1								Add		Divide	

Clock: cycle 11

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	
SUB	R8	R6	R2	7	9	10	
DIV	R10	R0	R6	8			
ADD	R6	R8	R2				

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
8	Multi1	Yes	Multi	R0	R2	R4			No	No
	Multi2	No								
0	Add	Yes	SUB	R8	R6	R2			No	No
	Divide	Yes	Div	R10	R0	R6	Multi1		No	Yes

Cycle 11: Both Multi1 and Add units are still executing (remaining cycles reduced by 1).

Reg

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	… … …
FU	Multi1								Add		Divide	

Clock: cycle 11

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	
SUB	R8	R6	R2	7	9	10	12
DIV	R10	R0	R6	8			
ADD	R6	R8	R2				

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
7	Multi1	Yes	Multi	R0	R2	R4			No	No
	Multi2	No								
	Add	No								
	Divide	Yes	Div	R10	R0	R6	Multi1		No	Yes

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU	Multi1								Add		Divide	

Clock: cycle 12

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	
SUB	R8	R6	R2	7	9	10	12
DIV	R10	R0	R6	8			
ADD	R6	R8	R2				

Cycle 12: Instruction 3 remains executing; Instruction 4 writes back; Instruction 5 still stalled.

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
7	Multi1	Yes	Multi	R0	R2	R4			No	No
	Multi2	No								
	Add	No								
	Divide	Yes	Div	R10	R0	R6	Multi1		No	Yes

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU	Multi1								Add		Divide	

Clock: cycle 12

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	
SUB	R8	R6	R2	7	9	10	12
DIV	R10	R0	R6	8			
ADD	R6	R8	R2	13			

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
6	Multi1	Yes	Multi	R0	R2	R4			No	No
	Multi2	No								
	Add	Yes	Add	R6	R8	R2			Yes	Yes
	Divide	Yes	Div	R10	R0	R6	Multi1		No	Yes

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU	Multi1						Add				Divide	

Clock: cycle 13

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	
SUB	R8	R6	R2	7	9	10	12
DIV	R10	R0	R6	8			
ADD	R6	R8	R2	13			

Cycle 13: Instruction 3 remains executing; Instruction 6 issued; Instruction 5 still stalled.

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
6	Multi1	Yes	Multi	R0	R2	R4			No	No
	Multi2	No								
	Add	Yes	Add	R6	R8	R2			Yes	Yes
	Divide	Yes	Div	R10	R0	R6	Multi1		No	Yes

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU	Multi1						Add				Divide	

Clock: cycle 13

A Example of Scoreboard Algorithm

Instruction Unit Status:

	Instructions			Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	
SUB	R8	R6	R2	7	9	10	12
DIV	R10	R0	R6	8			
ADD	R6	R8	R2	13	14		

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
5	Multi1	Yes	Multi	R0	R2	R4			No	No
	Multi2	No								
	Add	Yes	Add	R6	R8	R2			Yes	Yes
	Divide	Yes	Div	R10	R0	R6	Multi1		No	Yes

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU	Multi1						Add				Divide	

Clock: cycle 14

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	
SUB	R8	R6	R2	7	9	10	12
DIV	R10	R0	R6	8			
ADD	R6	R8	R2	13	14		

Cycle 14: Instruction 3 remains executing; Instruction 6 read operands; Instruction 5 still stalled.

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
5	Multi1	Yes	Multi	R0	R2	R4			No	No
	Multi2	No								
	Add	Yes	Add	R6	R8	R2			Yes	Yes
	Divide	Yes	Div	R10	R0	R6	Multi1		No	Yes

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU	Multi1						Add				Divide	

Clock: cycle 14

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	
SUB	R8	R6	R2	7	9	10	12
DIV	R10	R0	R6	8			
ADD	R6	R8	R2	13	14	15	

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
4	Multi1	Yes	Multi	R0	R2	R4			No	No
	Multi2	No								
1	Add	Yes	Add	R6	R8	R2			No	No
	Divide	Yes	Div	R10	R0	R6	Multi1		No	Yes

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU	Multi1						Add				Divide	

Clock: cycle 15

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	
SUB	R8	R6	R2	7	9	10	12
DIV	R10	R0	R6	8			
ADD	R6	R8	R2	13	14	15	

Cycle 15: Instruction 3 remains executing; Instruction 6 starts executing; Instruction 5 still stalled.

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
4	Multi1	Yes	Multi	R0	R2	R4			No	No
	Multi2	No								
1	Add	Yes	Add	R6	R8	R2			No	No
	Divide	Yes	Div	R10	R0	R6	Multi1		No	Yes

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU	Multi1						Add				Divide	

Clock: cycle 15

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	
SUB	R8	R6	R2	7	9	10	12
DIV	R10	R0	R6	8			
ADD	R6	R8	R2	13	14	15	

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
3	Multi1	Yes	Multi	R0	R2	R4			No	No
	Multi2	No								
0	Add	Yes	Add	R6	R8	R2			No	No
	Divide	Yes	Div	R10	R0	R6	Multi1		No	Yes

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU	Multi1						Add				Divide	

Clock: cycle 16

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	
SUB	R8	R6	R2	7	9	10	12
DIV	R10	R0	R6	8			
ADD	R6	R8	R2	13	14	15	

Cycle 16: Instruction 3 remains executing; Instruction 6 remains executing; Instruction 5 still stalled.

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
3	Multi1	Yes	Multi	R0	R2	R4			No	No
	Multi2	No								
0	Add	Yes	Add	R6	R8	R2			No	No
	Divide	Yes	Div	R10	R0	R6	Multi1		No	Yes

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU	Multi1						Add				Divide	

Clock: cycle 16

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	
SUB	R8	R6	R2	7	9	10	12
DIV	R10	R0	R6	8			
ADD	R6	R8	R2	13	14	15	

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
2	Multi1	Yes	Multi	R0	R2	R4			No	No
	Multi2	No								
	Add	Yes	Add	R6	R8	R2			No	No
	Divide	Yes	Div	R10	R0	R6	Multi1		No	Yes

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU	Multi1						Add				Divide	

Clock: cycle 17

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	
SUB	R8	R6	R2	7	9	10	12
DIV	R10	R0	R6	8			
ADD	R6	R8	R2	13	14	15	

Cycle 17: Instruction 3 remains executing; Instruction 6 stalled due to WAR hazard; Instruction 5 still stalled.

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
2	Multi1	Yes	Multi	R0	R2	R4			No	No
	Multi2	No								
	Add	Yes	Add	R6	R8	R2			No	No
	Divide	Yes	Div	R10	R0	R6	Multi1		No	Yes

Instruction 6 cannot write back to R6, because R6 is used by Instruction 5 as a source operand. That is, there is an WAR hazard. In the Function Unit Status table, this hazard is identified by the Divide Unit's F_k field being R6 and R_K field being "Yes".

Register Result Status:

	R0	R1	R2	...	...
FU	Multi1				

Clock: cycle 17

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	
SUB	R8	R6	R2	7	9	10	12
DIV	R10	R0	R6	8			
ADD	R6	R8	R2	13	14	15	

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
1	Multi1	Yes	Multi	R0	R2	R4			No	No
	Multi2	No								
	Add	Yes	Add	R6	R8	R2			No	No
	Divide	Yes	Div	R10	R0	R6	Multi1		No	Yes

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	… … …
FU	Multi1						Add				Divide	

Clock: cycle 18

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	
SUB	R8	R6	R2	7	9	10	12
DIV	R10	R0	R6	8			
ADD	R6	R8	R2	13	14	15	

Cycle 18-19: Instruction 3 remains executing; Instruction 5 and 6 still stalled.

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
1	Multi1	Yes	Multi	R0	R2	R4			No	No
	Multi2	No								
	Add	Yes	Add	R6	R8	R2			No	No
	Divide	Yes	Div	R10	R0	R6	Multi1		No	Yes

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU	Multi1						Add				Divide	

Clock: cycle 18

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	
SUB	R8	R6	R2	7	9	10	12
DIV	R10	R0	R6	8			
ADD	R6	R8	R2	13	14	15	

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
0	Multi1	Yes	Multi	R0	R2	R4			No	No
	Multi2	No								
	Add	Yes	Add	R6	R8	R2			No	No
	Divide	Yes	Div	R10	R0	R6	Multi1		No	Yes

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU	Multi1						Add				Divide	

Clock: cycle 19

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	
SUB	R8	R6	R2	7	9	10	12
DIV	R10	R0	R6	8			
ADD	R6	R8	R2	13	14	15	

Cycle 18-19: Instruction 3 remains executing; Instruction 5 and 6 still stalled.

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
0	Multi1	Yes	Multi	R0	R2	R4			No	No
	Multi2	No								
	Add	Yes	Add	R6	R8	R2			No	No
	Divide	Yes	Div	R10	R0	R6	Multi1		No	Yes

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU	Multi1						Add				Divide	

Clock: cycle 19

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	20
SUB	R8	R6	R2	7	9	10	12
DIV	R10	R0	R6	8			
ADD	R6	R8	R2	13	14	15	

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
	Multi1	No								
	Multi2	No								
	Add	Yes	Add	R6	R8	R2			No	No
	Divide	Yes	Div	R10	R0	R6			Yes	Yes

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU	Multi1						Add				Divide	

Clock: cycle 20

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	20
SUB	R8	R6	R2	7	9	10	12
DIV	R10	R0	R6	8			
ADD	R6	R8	R2	13	14	15	

Cycle 20: Instruction 3 writes back to R0; Instruction 5 and 6 still stalled.

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
	Multi1	No								
	Multi2	No								
	Add	Yes	Add	R6	R8	R2			No	No
	Divide	Yes	Div	R10	R0	R6			Yes	Yes

After Instruction 3 writes back to R0, Instruction 5's (and Divide Unit's) source operands are both ready.

Register Result Status:

	R0	R1	R2	R3	R4	R5	... ...
FU	Multi1						

Clock: cycle 20

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	20
SUB	R8	R6	R2	7	9	10	12
DIV	R10	R0	R6	8	21		
ADD	R6	R8	R2	13	14	15	

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
	Multi1	No								
	Multi2	No								
	Add	Yes	Add	R6	R8	R2			No	No
	Divide	Yes	Div	R10	R0	R6			Yes	Yes

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU							Add				Divide	

Clock: cycle 21

A Example of Scoreboard Algorithm

Instruction Unit Status:

	Instructions			Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	20
SUB	R8	R6	R2	7	9	10	12
DIV	R10	R0	R6	8	21		
ADD	R6	R8	R2	13	14	15	

Cycle 21: Instruction 5 reads operands; Instruction 6 still stalled.

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
	Multi1	No								
	Multi2	No								
	Add	Yes	Add	R6	R8	R2			No	No
	Divide	Yes	Div	R10	R0	R6			Yes	Yes

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU							Add				Divide	

Clock: cycle 21

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	20
SUB	R8	R6	R2	7	9	10	12
DIV	R10	R0	R6	8	21	22	
ADD	R6	R8	R2	13	14	15	22

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
	Multi1	No								
	Multi2	No								
	Add	No								
39	Divide	Yes	Div	R10	R0	R6			No	No

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU							Add				Divide	

Clock: cycle 22

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	20
SUB	R8	R6	R2	7	9	10	12
DIV	R10	R0	R6	8	21	22	
ADD	R6	R8	R2	13	14	15	22

Cycle 22: Instruction 5 starts executing; Instruction 6 writes back to R6 (WAR hazard is gone).

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
	Multi1	No								
	Multi2	No								
	Add	No								
39	Divide	Yes	Div	R10	R0	R6			No	No

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU							Add				Divide	

Clock: cycle 22

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	20
SUB	R8	R6	R2	7	9	10	12
DIV	R10	R0	R6	8	21	22	
ADD	R6	R8	R2	13	14	15	22

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
	Multi1	No								
	Multi2	No								
	Add	No								
38	Divide	Yes	Div	R10	R0	R6			No	No

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU											Divide	

Clock: cycle 23

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	20
SUB	R8	R6	R2	7	9	10	12
DIV	R10	R0	R6	8	21	22	
ADD	R6	R8	R2	13	14	15	22

Cycle 23: Instruction 5 remains executing.

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
	Multi1	No								
	Multi2	No								
	Add	No								
38	Divide	Yes	Div	R10	R0	R6			No	No

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU											Divide	

Clock: cycle 23

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	20
SUB	R8	R6	R2	7	9	10	12
DIV	R10	R0	R6	8	21	22	
ADD	R6	R8	R2	13	14	15	22

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
	Multi1	No								
	Multi2	No								
	Add	No								
0	Divide	Yes	Div	R10	R0	R6			No	No

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU											Divide	

Clock: cycle 61 (skipped a couple of cycles)

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	20
SUB	R8	R6	R2	7	9	10	12
DIV	R10	R0	R6	8	21	22	
ADD	R6	R8	R2	13	14	15	22

Cycle 61: Instruction 5's last execution cycle.

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
	Multi1	No								
	Multi2	No								
	Add	No								
0	Divide	Yes	Div	R10	R0	R6			No	No

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU											Divide	

Clock: cycle 61 (skipped a couple of cycles)

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	20
SUB	R8	R6	R2	7	9	10	12
DIV	R10	R0	R6	8	21	22	62
ADD	R6	R8	R2	13	14	15	22

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
	Multi1	No								
	Multi2	No								
	Add	No								
	Divide	No								

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU											Divide	

Clock: cycle 62

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	20
SUB	R8	R6	R2	7	9	10	12
DIV	R10	R0	R6	8	21	22	62
ADD	R6	R8	R2	13	14	15	22

Cycle 62: Instruction 5 writes back to R10.

Function Unit Status:

Time	Name	Busy	Op	Dest F_i	Src1 F_j	Src2 F_k	FU1 Q_j	FU2 Q_k	F_j? R_j	F_k? R_k
	Integer									
	Multi1	No								
	Multi2	No								
	Add	No								
	Divide	No								

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU											Divide	

Clock: cycle 62

A Example of Scoreboard Algorithm

Instruction Unit Status:

Instructions				Stages			
Op	Dest *i*	Src1 *j*	Src2 *k*	IS	RD	EX	WB
LD	R6	34+	R11	1	2	3	4
LD	R2	45+	R13	5	6	7	8
MUL	R0	R2	R4	6	9	10	20
SUB	R8	R6	R2	7	9	10	12
DIV	R10	R0	R6	8	21	22	62
ADD	R6	R8	R2	13	14	15	22

Out-of-order completion.

In-order issue.

Out-of-order execution.

Function Unit Status:

Time	Name	Busy	Op	st F_i	Src1 F_j			U2 Q_k	F_j? R_j	F_k? R_k
	Integer									
	Multi1	No								
	Multi2	No								
	Add	No								
	Divide	No								

Register Result Status:

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	... ...
FU												

Clock: cycle 63

Road Map

Out-of-Order Execution and Dynamic Scheduling

The Scoreboard Scheduling Algorithm

An Example of Scoreboard Scheduling Algorithm

Scoreboarding Limitations

Acknowledgment

Scoreboarding is Limited By

- Number and type of functional units.
- Number of instruction buffer entries (scoreboard size)
- Amount of application ILP (RAW hazards)
- Presence of anti-dependencies (WAR) and output dependencies (WAW)
  - In-order issue for WAW/Structural Hazards limits scheduler.
  - WAR stalls are critical for loops (WAR prevents hardware unrolling in Scoreboarding).

The WAR Limitation of Scoreboarding

- WAR stalls are critical for loops (WAR prevents hardware unrolling in Scoreboarding).
  - For example, for the following two iterations of a loop, instruction 5 cannot be finished due to WAR hazard between instructions 3 and instruction 5, preventing the instructions following it from execution,

```
1  mov R3, [R12+R14]
2  mul R3, R3, 11
3  add R5, R3, 10
4  add R14, R14, 4
5  mov R3, [R12+R14]
6  mul R3, R3, 11
7  add R5, R3, 10
8  add R14, R14, 4
```

Road Map

Out-of-Order Execution and Dynamic Scheduling

The Scoreboard Scheduling Algorithm

An Example of Scoreboard Scheduling Algorithm

Scoreboarding Limitations

Acknowledgment

Acknowledgment

This lecture is partially based on the slides from Dr. David Brooks. The Scoreboard example was designed by Dr. Broderson, for CS152 at UCB (Copyright (C) 2001 UCB).