Complex Pipelining

Yuri Baida
yuri.baida@gmail.com
yuriy.v.baida@intel.com

February 26, 2011

MDSP Project | Intel Lab
Moscow Institute of Physics and Technology

Agenda

- Review
  - Ideal pipeline
  - Dependencies
  - Hazards, interlocks & stalls
  - Limitations of simple pipeline
- Complex pipelining
  - Out-of-order execution
  - Scoreboard
  - Tomasulo Algorithm
  - Register renaming

Review

Pipelining Idealism

- Uniform suboperations
  - The operation to be pipelined can be evenly partitioned into uniformlatency suboperations
- Repetition of identical operations
  - The same operations are to be performed repeatedly on a large number of different inputs
- Repetition of independent operations
  - All the repetitions of the same operation are mutually independent
  - Good example: automobile assembly line

Hazards, Interlocks & Stalls

- Pipeline hazards
  - Potential violations of program dependences
  - Must ensure program dependences are not violated
- Hazard resolution
  - Static Method: performed at compiled time in software
  - Dynamic Method: performed at run time using hardware
    - Stall
    - Flush
    - Forward
- Pipeline interlock
  - Hardware mechanisms for dynamic hazard resolution
  - Must detect and enforce dependences at run time

Dependencies & Pipeline Hazards

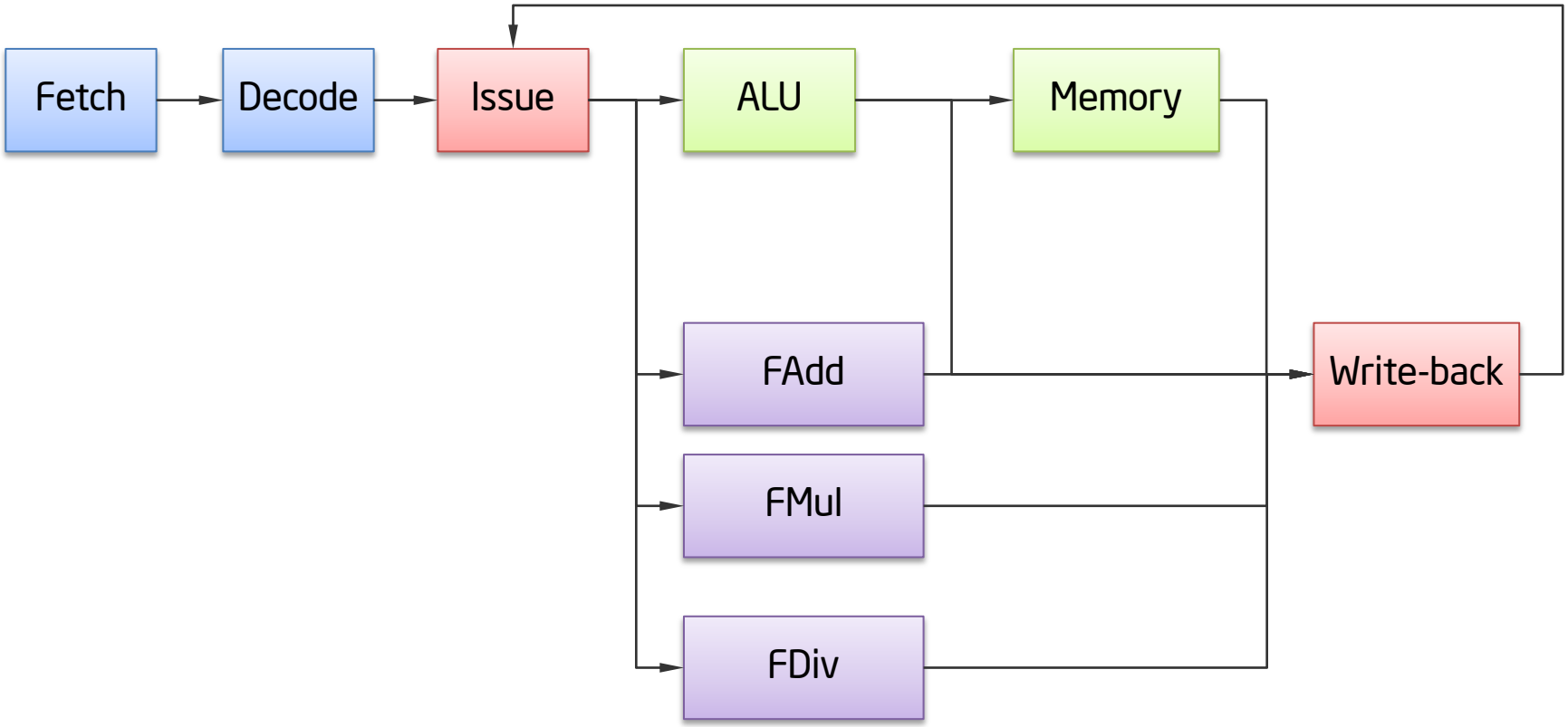
- Data dependence (register or memory)
  - True dependence (RAW)
    - Instruction must wait for all required input operands
  - Anti-dependence (WAR)
    - Later write must not clobber a still-pending earlier read
  - Output dependence (WAW)
    - Earlier write must not clobber an already-finished later write
- Control dependence
  - A “data dependency” on the instruction pointer
  - Conditional branches cause uncertainty to instruction sequencing
- Resource conflicts
  - Two instructions need the same device

Limitations of Simple Pipelined Processors
(aka Scalar Processors)

- Upper bound on scalar pipeline throughput
  - Limited by IPC = 1
- Inefficiencies of very deep pipelines
  - Clocking overheads
  - Longer hazards and stalls
- Performance lost due to in-order pipeline
  - Unnecessary stalls
- Inefficient unification Into single pipeline
  - Long latency for each instruction
  - Hazards and associated stalls

Complex Pipelining

Complex Pipeline Structure



Complex Pipeline Control Issues

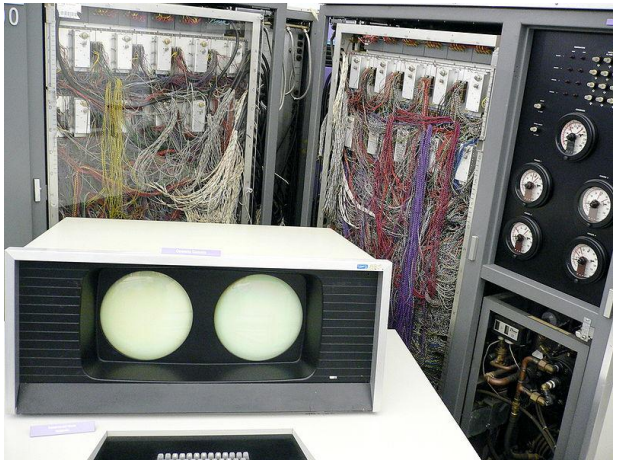
- Structural conflicts
  - At the execution stage
    - Some FPU or memory unit is not pipelined and takes more than one cycle
  - At the write-back stage
    - Variable latencies of different function units
- Can we solve write hazards without equalizing all pipeline depths and without bypassing?

When is it Safe to Issue an Instruction?

- Suppose a data structure keeps track of all the instructions in all the functional units
- The following checks need to be made before the Issue stage can dispatch an instruction
  - Is the required function unit available?
  - Is the input data available? ⇒ RAW?
  - Is it safe to write the destination? ⇒ WAR? WAW?
  - Is there a structural conflict at the Write-back stage?

Scoreboard

CDC 6600 built by Seymour Cray in 1963



Picture: Steve Jurvetson

Picture: Hullie

- Fast pipelined machine
  - Very fast clock, 10 MHz
  - Ten functional units (parallel, unpipelined)
  - 400,000 transistors
  - Took up 70 m^2, weighted 5 tons
  - Consumed 150 kW of power
- Dynamic scheduling of instructions
  - Using a scoreboard
- Fastest machine in world for 5 years
  - Until CDC 7600
  - Over 100 sold ($7-10M each)

Four Stages of Scoreboard Control (1)

- Issue (ID1) — decode instructions, check for structural hazards
  - Instructions issued in program order (for hazard checking)
  - Don't issue if structural hazard
  - Don't issue if instruction is output dependent on any previously issued but uncompleted instruction (no WAW hazards)
- Read operands (ID2) — wait until no data hazards, then read operands
  - All real dependencies (RAW hazards) resolved in this stage, since we wait for instructions to write back data
  - No forwarding of data in this model!

Four Stages of Scoreboard Control (2)

- Execution (EX) — operate on operands
  - The functional unit begins execution upon receiving operands
  - When the result is ready, it notifies the scoreboard that it has completed execution
- Write-back result (WB) — finish execution
  - Stall until no WAR hazards with previous instructions
  - Example:
    - `DIVD F0,  F2, F4`
    - `ADDD F10, F0, F8`
    - `SUBD F8,  F8, F14`
  - CDC 6600 scoreboard would stall SUBD until ADDD reads operands

Three Parts of the Scoreboard

- Instruction status
  - Which of 4 steps the instruction is in
- Functional unit status
  - Indicates the state of the functional unit (FU)
  - 9 fields for each functional unit
    - Busy: Indicates whether the unit is busy or not
    - Op: Operation to perform in the unit (e.g., + or –)
    - D: Destination register number
    - S_1, S_2: Source register numbers
    - FU_1, FU_2: Functional units producing source registers S_1, S_2
    - R_1, R_2: Flags indicating when S_1, S_2 are ready
- Register result status
  - Indicates which functional unit will write each register, if one exists. Blank when no pending instructions will write that register

Scoreboard Example

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Read Oper	Exec Comp	Write Result

- Functional unit status

Time	Name	Busy	Op	D	S_1	S_2	FU_1	FU_2	R_1	R_2
	Integer	No								
	Mult1	No								
	Mult2	No								
	Add	No								
	Divide	No								

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
0									

Scoreboard Example: Cycle 1

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Read Oper	Exec Comp	Write Result
1			

- Functional unit status

Time	Name	Busy	Op	D	S_1	S_2	FU_1	FU_2	R_1	R_2
	Integer	Yes	Load	F6		R2				Yes
	Mult1	No								
	Mult2	No								
	Add	No								
	Divide	No								

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
1				Integer					

Scoreboard Example: Cycle 2

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Read Oper	Exec Comp	Write Result
1	2		

- Functional unit status

Time	Name	Busy	Op	D	S_1	S_2	FU_1	FU_2	R_1	R_2
	Integer	Yes	Load	F6		R2				Yes
	Mult1	No								
	Mult2	No								
	Add	No								
	Divide	No								

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
2				Integer					

Scoreboard Example: Cycle 3

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Read Oper	Exec Comp	Write Result
1	2	3	

- Functional unit status

Time	Name	Busy	Op	D	S_1	S_2	FU_1	FU_2	R_1	R_2
	Integer	Yes	Load	F6		R2				No
	Mult1	No								
	Mult2	No								
	Add	No								
	Divide	No								

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
3				Integer					

Scoreboard Example: Cycle 4

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Read Oper	Exec Comp	Write Result
1	2	3	4

- Functional unit status

Time	Name	Busy	Op	D	S_1	S_2	FU_1	FU_2	R_1	R_2
	Integer	No								
	Mult1	No								
	Mult2	No								
	Add	No								
	Divide	No								

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
4				Integer					

Scoreboard Example: Cycle 5

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Read Oper	Exec Comp	Write Result
1	2	3	4
5			

- Functional unit status

Time	Name	Busy	Op	D	S_1	S_2	FU_1	FU_2	R_1	R_2
	Integer	Yes	Load	F2		R3				Yes
	Mult1	No								
	Mult2	No								
	Add	No								
	Divide	No								

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
5		Integer							

Scoreboard Example: Cycle 6

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Read Oper	Exec Comp	Write Result
1	2	3	4
5	6		
6			

- Functional unit status

Time	Name	Busy	Op	D	S_1	S_2	FU_1	FU_2	R_1	R_2
	Integer	Yes	Load	F2		R3				Yes
	Mult1	Yes	Mult	F0	F2	F4	Integer		No	Yes
	Mult2	No								
	Add	No								
	Divide	No								

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
6	Mult1	Integer							

Scoreboard Example: Cycle 7

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Read Oper	Exec Comp	Write Result
1	2	3	4
5	6	7	
6			
7			

- Functional unit status

Time	Name	Busy	Op	D	S_1	S_2	FU_1	FU_2	R_1	R_2
	Integer	Yes	Load	F2		R3				No
	Mult1	Yes	Mult	F0	F2	F4	Integer		No	Yes
	Mult2	No								
	Add	Yes	Sub	F8	F6	F2		Integer	Yes	No
	Divide	No								

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
7	Mult1	Integer			Add				

Scoreboard Example: Cycle 8a

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Read Oper	Exec Comp	Write Result
1	2	3	4
5	6	7	
6			
7			
8			

- Functional unit status

Time	Name	Busy	Op	D	S_1	S_2	FU_1	FU_2	R_1	R_2
	Integer	Yes	Load	F2		R3				No
	Mult1	Yes	Mult	F0	F2	F4	Integer		No	Yes
	Mult2	No								
	Add	Yes	Sub	F8	F6	F2		Integer	Yes	No
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
8	Mult1	Integer			Add	Divide			

Scoreboard Example: Cycle 8b

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Read Oper	Exec Comp	Write Result
1	2	3	4
5	6	7	8
6			
7			
8			

- Functional unit status

Time	Name	Busy	Op	D	S_1	S_2	FU_1	FU_2	R_1	R_2
	Integer	No								
	Mult1	Yes	Mult	F0	F2	F4			Yes	Yes
	Mult2	No								
	Add	Yes	Sub	F8	F6	F2			Yes	Yes
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
8	Mult1				Add	Divide			

Scoreboard Example: Cycle 9

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Read Oper	Exec Comp	Write Result
1	2	3	4
5	6	7	8
6	9		
7	9		
8			

- Functional unit status

Time	Name	Busy	Op	D	S_1	S_2	FU_1	FU_2	R_1	R_2
	Integer	No								
10	Mult1	Yes	Mult	F0	F2	F4			Yes	Yes
	Mult2	No								
2	Add	Yes	Sub	F8	F6	F2			Yes	Yes
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
9	Mult1				Add	Divide			

Scoreboard Example: Cycle 10

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Read Oper	Exec Comp	Write Result
1	2	3	4
5	6	7	8
6	9		
7	9		
8			

- Functional unit status

Time	Name	Busy	Op	D	S_1	S_2	FU_1	FU_2	R_1	R_2
	Integer	No								
9	Mult1	Yes	Mult	F0	F2	F4			No	No
	Mult2	No								
1	Add	Yes	Sub	F8	F6	F2			No	No
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
10	Mult1				Add	Divide			

Scoreboard Example: Cycle 11

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Read Oper	Exec Comp	Write Result
1	2	3	4
5	6	7	8
6	9		
7	9	11	
8			

- Functional unit status

Time	Name	Busy	Op	D	S_1	S_2	FU_1	FU_2	R_1	R_2
	Integer	No								
8	Mult1	Yes	Mult	F0	F2	F4			No	No
	Mult2	No								
0	Add	Yes	Sub	F8	F6	F2			No	No
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
11	Mult1				Add	Divide			

Scoreboard Example: Cycle 12

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Read Oper	Exec Comp	Write Result
1	2	3	4
5	6	7	8
6	9		
7	9	11	12
8			

- Functional unit status

Time	Name	Busy	Op	D	S_1	S_2	FU_1	FU_2	R_1	R_2
	Integer	No								
7	Mult1	Yes	Mult	F0	F2	F4			No	No
	Mult2	No								
	Add	No								
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
12	Mult1					Divide			

Scoreboard Example: Cycle 13

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Read Oper	Exec Comp	Write Result
1	2	3	4
5	6	7	8
6	9		
7	9	11	12
8			
13			

- Functional unit status

Time	Name	Busy	Op	D	S_1	S_2	FU_1	FU_2	R_1	R_2
	Integer	No								
6	Mult1	Yes	Mult	F0	F2	F4			No	No
	Mult2	No								
	Add	Yes	Add	F6	F8	F2			Yes	Yes
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
13	Mult1			Add		Divide			

Scoreboard Example: Cycle 14

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Read Oper	Exec Comp	Write Result
1	2	3	4
5	6	7	8
6	9		
7	9	11	12
8			
13	14		

- Functional unit status

Time	Name	Busy	Op	D	S_1	S_2	FU_1	FU_2	R_1	R_2
	Integer	No								
5	Mult1	Yes	Mult	F0	F2	F4			No	No
	Mult2	No								
2	Add	Yes	Add	F6	F8	F2			Yes	Yes
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
14	Mult1			Add		Divide			

Scoreboard Example: Cycle 15

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Read Oper	Exec Comp	Write Result
1	2	3	4
5	6	7	8
6	9		
7	9	11	12
8			
13	14		

- Functional unit status

Time	Name	Busy	Op	D	S_1	S_2	FU_1	FU_2	R_1	R_2
	Integer	No								
4	Mult1	Yes	Mult	F0	F2	F4			No	No
	Mult2	No								
1	Add	Yes	Add	F6	F8	F2			No	No
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
15	Mult1			Add		Divide			

Scoreboard Example: Cycle 16

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Read Oper	Exec Comp	Write Result
1	2	3	4
5	6	7	8
6	9		
7	9	11	12
8			
13	14	16	

- Functional unit status

Time	Name	Busy	Op	D	S_1	S_2	FU_1	FU_2	R_1	R_2
	Integer	No								
3	Mult1	Yes	Mult	F0	F2	F4			No	No
	Mult2	No								
0	Add	Yes	Add	F6	F8	F2			No	No
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
16	Mult1			Add		Divide			

Scoreboard Example: Cycle 17

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Read Oper	Exec Comp	Write Result
1	2	3	4
5	6	7	8
6	9		
7	9	11	12
8			
13	14	16	

- Functional unit status

Time	Name	Busy	Op	D	S_1	S_2	FU_1	FU_2	R_1	R_2
	Integer	No								
2	Mult1	Yes	Mult	F0	F2	F4			No	No
	Mult2	No								
	Add	Yes	Add	F6	F8	F2			No	No
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
17	Mult1			Add		Divide			

Scoreboard Example: Cycle 18

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Read Oper	Exec Comp	Write Result
1	2	3	4
5	6	7	8
6	9		
7	9	11	12
8			
13	14	16	

- Functional unit status

Time	Name	Busy	Op	D	S_1	S_2	FU_1	FU_2	R_1	R_2
	Integer	No								
1	Mult1	Yes	Mult	F0	F2	F4			No	No
	Mult2	No								
	Add	Yes	Add	F6	F8	F2			No	No
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
18	Mult1			Add		Divide			

Scoreboard Example: Cycle 19

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Read Oper	Exec Comp	Write Result
1	2	3	4
5	6	7	8
6	9	19	
7	9	11	12
8			
13	14	16	

- Functional unit status

Time	Name	Busy	Op	D	S_1	S_2	FU_1	FU_2	R_1	R_2
	Integer	No								
0	Mult1	Yes	Mult	F0	F2	F4			No	No
	Mult2	No								
	Add	Yes	Add	F6	F8	F2			No	No
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
19	Mult1			Add		Divide			

Scoreboard Example: Cycle 20

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Read Oper	Exec Comp	Write Result
1	2	3	4
5	6	7	8
6	9	19	20
7	9	11	12
8			
13	14	16	

- Functional unit status

Time	Name	Busy	Op	D	S_1	S_2	FU_1	FU_2	R_1	R_2
	Integer	No								
	Mult1	No								
	Mult2	No								
	Add	Yes	Add	F6	F8	F2			No	No
	Divide	Yes	Div	F10	F0	F6			Yes	Yes

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
20				Add		Divide			

Scoreboard Example: Cycle 21

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Read Oper	Exec Comp	Write Result
1	2	3	4
5	6	7	8
6	9	19	20
7	9	11	12
8	21		
13	14	16	

- Functional unit status

Time	Name	Busy	Op	D	S_1	S_2	FU_1	FU_2	R_1	R_2
	Integer	No								
	Mult1	No								
	Mult2	No								
	Add	Yes	Add	F6	F8	F2			No	No
40	Divide	Yes	Div	F10	F0	F6			Yes	Yes

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
21				Add		Divide			

Scoreboard Example: Cycle 22

- Instruction status

Instr	D	S_1	S_2	Issue	Read Oper	Exec Comp	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULTD	F0	F2	F4	6	9	19	20
SUBD	F8	F6	F2	7	9	11	12
DIVD	F10	F0	F6	8	21		
ADDD	F6	F8	F2	13	14	16	22

- Functional unit status

Time	Name	Busy	Op	D	S_1	S_2	FU_1	FU_2	R_1	R_2
	Integer	No								
	Mult1	No								
	Mult2	No								
	Add	No								
39	Divide	Yes	Div	F10	F0	F6			No	No

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
22						Divide			

Scoreboard Example: Cycle 61

- Instruction status

Instr	D	S_1	S_2	Issue	Read Oper	Exec Comp	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULTD	F0	F2	F4	6	9	19	20
SUBD	F8	F6	F2	7	9	11	12
DIVD	F10	F0	F6	8	21	61	
ADDD	F6	F8	F2	13	14	16	22

- Functional unit status

Time	Name	Busy	Op	D	S_1	S_2	FU_1	FU_2	R_1	R_2
	Integer	No								
	Mult1	No								
	Mult2	No								
	Add	No								
0	Divide	Yes	Div	F10	F0	F6			No	No

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
61						Divide			

Scoreboard Example: Cycle 62

- Instruction status

Instr	D	S_1	S_2	Issue	Read Oper	Exec Comp	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULTD	F0	F2	F4	6	9	19	20
SUBD	F8	F6	F2	7	9	11	12
DIVD	F10	F0	F6	8	21	61	62
ADDD	F6	F8	F2	13	14	16	22

- Functional unit status

Time	Name	Busy	Op	D	S_1	S_2	FU_1	FU_2	R_1	R_2
	Integer	No								
	Mult1	No								
	Mult2	No								
	Add	No								
	Divide	No								

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
62									

CDC 6600 Scoreboard

- Speedup 1.7 from compiler; 2.5 by hand
  - But slow memory (no cache) limits benefit
- Limitations of 6600 scoreboard
  - No forwarding hardware
  - Limited to instructions in basic block (small window)
  - Small number of functional units (structural hazards), especially integer/load store units
  - Do not issue on structural hazards
  - Wait for WAR hazards
  - Prevent WAW hazards

Tomasulo Algorithm

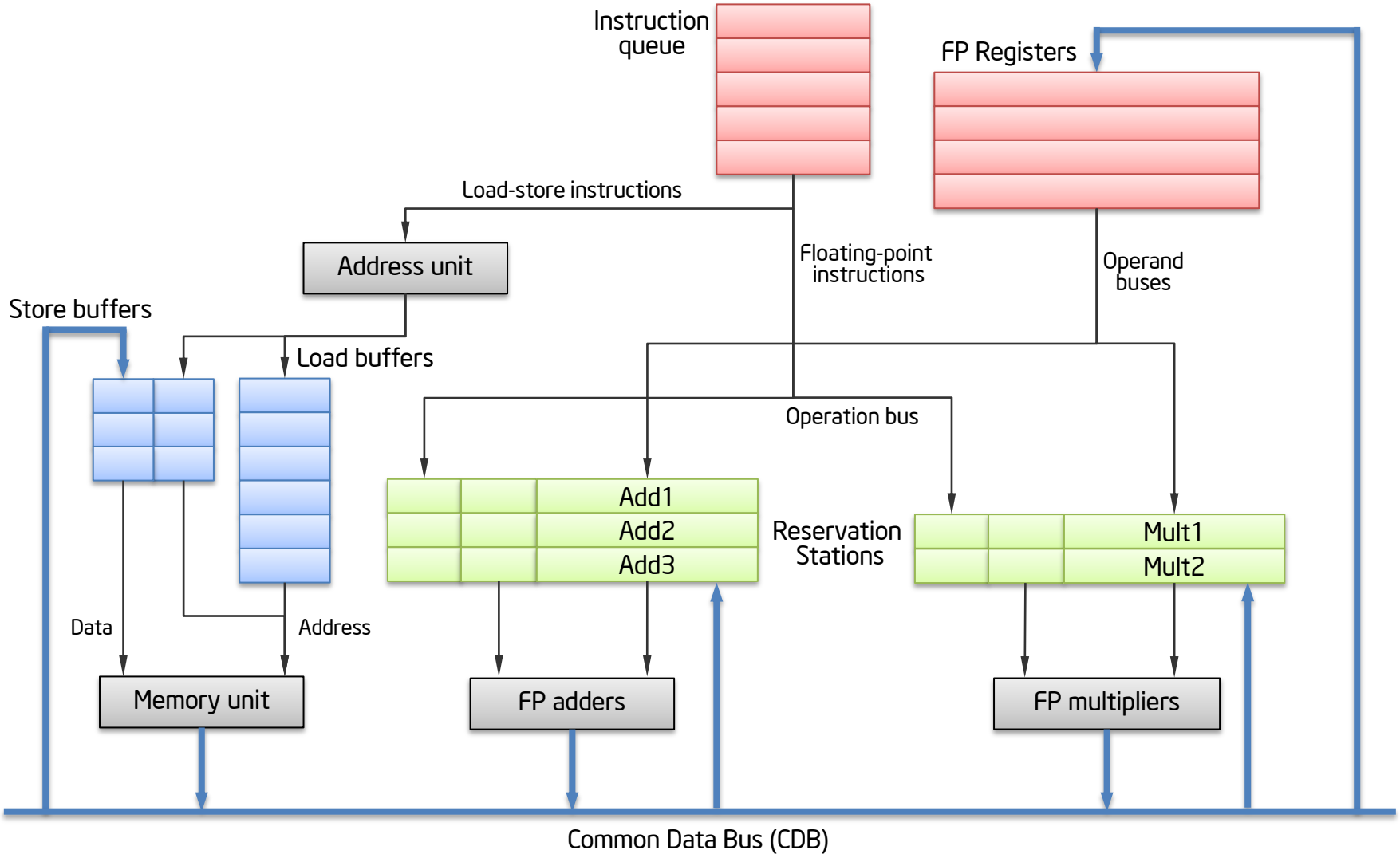
Tomasulo Algorithm

- Developed in 1967 by Robert Tomasulo
  - Implemented in IBM 360/91
  - About 3 years after CDC 6600
  - Received the Eckert-Mauchly Award in 1997 for this algorithm.
- Goal: High Performance without special compilers
- Differences between IBM 360 & CDC 6600 ISA
  - IBM has only 2 register specifiers/instr vs. 3 in CDC 6600
  - IBM has 4 FP registers vs. 8 in CDC 6600
  - IBM has memory-register ops
- Why study?
  - Lead to Alpha 21264, MIPS 10000, Pentium II, PowerPC 604, …

Tomasulo Algorithm vs. Scoreboard

- Control logic and buffers are distributed with Function Units
  - Buffers called reservation stations (RS)
  - Centralized in scoreboard
- Register renaming
  - Registers in instructions replaced by values or pointers to RS
  - Avoids WAR, WAW hazards
  - More reservation stations than registers
- Results to FU from RS, not through registers, over Common Data Bus that broadcasts results to all FUs
- Load and Stores treated as FUs with RSs as well

Tomasulo Organization



Reservation Station Components

- Op: Operation to perform in the unit (e.g., + or –)
- V_1, V_2: Value of Source operands
  - Store buffers has V field, result to be stored
- Q_1, Q_2: Reservation stations producing source registers
  - Value to be written
  - Note: No ready flags as in Scoreboard; Q_1, Q_2 = 0 $\Rightarrow$ Ready
  - Store buffers only have Q_i for RS producing result
- Busy: Indicates reservation station or FU is busy
- Register result status
  - Indicates which functional unit will write each register, if one exists.
  - Blank when no pending instructions that will write that register

Three Stages of Tomasulo Algorithm

- Issue — get instruction from FP Op Queue
  - If reservation station free (no structural hazard), control issues instruction & sends operands (renames registers).
- Execution (EX) — operate on operands
  - When both operands ready then execute
  - If not ready, watch Common Data Bus for result
- Write-back result (WB) — finish execution
  - Write on Common Data Bus to all awaiting units; mark reservation station available
- Common data bus: data + source ("come from" bus)
  - 64 bits of data + 4 bits of Functional Unit source address
  - Write if matches expected Functional Unit (produces result)
  - Compare to normal data bus: data + destination ("go to" bus)

Tomasulo Example

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Exec Comp	Write Result

Load	Busy	Address
Load1		
Load2		
Load3		

- Reservation stations

Time	Name	Busy	Op	S_1	S_2	RS	RS
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
	Mult2	No					

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
0									

Tomasulo Example: Cycle 1

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Exec Comp	Write Result
1		

Load	Busy	Address
Load1	Yes	34+R2
Load2	No	
Load3	No	

- Reservation stations

Time	Name	Busy	Op	S_1	S_2	RS	RS
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
	Mult2	No					

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
1				Load1					

Tomasulo Example: Cycle 2

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Exec Comp	Write Result
1		
2		

Load	Busy	Address
Load1	Yes	34+R2
Load2	Yes	45+R3
Load3	No	

- Reservation stations

Time	Name	Busy	Op	S_1	S_2	RS	RS
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
	Mult2	No					

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
2		Load2		Load1					

Tomasulo Example: Cycle 3

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Exec Comp	Write Result
1	3	
2		
3		

Load	Busy	Address
Load1	Yes	34+R2
Load2	Yes	45+R3
Load3	No	

- Reservation stations

Time	Name	Busy	Op	S_1	S_2	RS	RS
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	Yes	MULTD		R(F4)	Load2	
	Mult2	No					

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
3	Mult1	Load2		Load1					

Tomasulo Example: Cycle 4

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Exec Comp	Write Result
1	3	4
2	4	
3		
4		

Load	Busy	Address
Load1	No	
Load2	Yes	45+R3
Load3	No	

- Reservation stations

Time	Name	Busy	Op	S_1	S_2	RS	RS
	Add1	Yes	SUBD	M(A1)			Load2
	Add2	No					
	Add3	No					
	Mult1	Yes	MULTD		R(F4)	Load2	
	Mult2	No					

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
4	Mult1	Load2		M(A1)	Add1				

Tomasulo Example: Cycle 5

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Exec Comp	Write Result
1	3	4
2	4	5
3		
4		
5		

Load	Busy	Address
Load1	No	
Load2	No	
Load3	No	

- Reservation stations

Time	Name	Busy	Op	S_1	S_2	RS	RS
2	Add1	Yes	SUBD	M(A1)	M(A2)		
	Add2	No					
	Add3	No					
10	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
5	Mult1	M(A2)		M(A1)	Add1	Mult2			

Tomasulo Example: Cycle 6

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Exec Comp	Write Result
1	3	4
2	4	5
3		
4		
5		
6		

Load	Busy	Address
Load1	No	
Load2	No	
Load3	No	

- Reservation stations

Time	Name	Busy	Op	S_1	S_2	RS	RS
1	Add1	Yes	SUBD	M(A1)	M(A2)		
	Add2	Yes	ADDD		M(A2)	Add1	
	Add3	No					
9	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
6	Mult1	M(A2)		Add2	Add1	Mult2			

Tomasulo Example: Cycle 7

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Exec Comp	Write Result
1	3	4
2	4	5
3		
4	7	
5		
6		

Load	Busy	Address
Load1	No	
Load2	No	
Load3	No	

- Reservation stations

Time	Name	Busy	Op	S_1	S_2	RS	RS
0	Add1	Yes	SUBD	M(A1)	M(A2)		
	Add2	Yes	ADDD		M(A2)	Add1	
	Add3	No					
8	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
7	Mult1	M(A2)		Add2	Add1	Mult2			

Tomasulo Example: Cycle 8

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Exec Comp	Write Result
1	3	4
2	4	5
3		
4	7	8
5		
6		

Load	Busy	Address
Load1	No	
Load2	No	
Load3	No	

- Reservation stations

Time	Name	Busy	Op	S_1	S_2	RS	RS
	Add1	No					
2	Add2	Yes	ADDD	(M-M)	M(A2)		
	Add3	No					
7	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
8	Mult1	M(A2)		Add2	(M-M)	Mult2			

Tomasulo Example: Cycle 9

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Exec Comp	Write Result
1	3	4
2	4	5
3		
4	7	8
5		
6		

Load	Busy	Address
Load1	No	
Load2	No	
Load3	No	

- Reservation stations

Time	Name	Busy	Op	S_1	S_2	RS	RS
	Add1	No					
1	Add2	Yes	ADDD	(M-M)	M(A2)		
	Add3	No					
6	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
9	Mult1	M(A2)		Add2	(M-M)	Mult2			

Tomasulo Example: Cycle 10

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Exec Comp	Write Result
1	3	4
2	4	5
3		
4	7	8
5		
6	10	

Load	Busy	Address
Load1	No	
Load2	No	
Load3	No	

- Reservation stations

Time	Name	Busy	Op	S_1	S_2	RS	RS
	Add1	No					
0	Add2	Yes	ADDD	(M-M)	M(A2)		
	Add3	No					
5	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
10	Mult1	M(A2)		Add2	(M-M)	Mult2			

Tomasulo Example: Cycle 11

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Exec Comp	Write Result
1	3	4
2	4	5
3		
4	7	8
5		
6	10	11

Load	Busy	Address
Load1	No	
Load2	No	
Load3	No	

- Reservation stations

Time	Name	Busy	Op	S_1	S_2	RS	RS
	Add1	No					
	Add2	No					
	Add3	No					
4	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
11	Mult1	M(A2)		(M-M+M)	(M-M)	Mult2			

Tomasulo Example: Cycle 12

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Exec Comp	Write Result
1	3	4
2	4	5
3		
4	7	8
5		
6	10	11

Load	Busy	Address
Load1	No	
Load2	No	
Load3	No	

- Reservation stations

Time	Name	Busy	Op	S_1	S_2	RS	RS
	Add1	No					
	Add2	No					
	Add3	No					
3	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
12	Mult1	M(A2)		(M-M+M)	(M-M)	Mult2			

Tomasulo Example: Cycle 13

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Exec Comp	Write Result
1	3	4
2	4	5
3		
4	7	8
5		
6	10	11

Load	Busy	Address
Load1	No	
Load2	No	
Load3	No	

- Reservation stations

Time	Name	Busy	Op	S_1	S_2	RS	RS
	Add1	No					
	Add2	No					
	Add3	No					
2	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
13	Mult1	M(A2)		(M-M+M)	(M-M)	Mult2			

Tomasulo Example: Cycle 14

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Exec Comp	Write Result
1	3	4
2	4	5
3		
4	7	8
5		
6	10	11

Load	Busy	Address
Load1	No	
Load2	No	
Load3	No	

- Reservation stations

Time	Name	Busy	Op	S_1	S_2	RS	RS
	Add1	No					
	Add2	No					
	Add3	No					
1	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
14	Mult1	M(A2)		(M-M+M)	(M-M)	Mult2			

Tomasulo Example: Cycle 15

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Exec Comp	Write Result
1	3	4
2	4	5
3	15	
4	7	8
5		
6	10	11

Load	Busy	Address
Load1	No	
Load2	No	
Load3	No	

- Reservation stations

Time	Name	Busy	Op	S_1	S_2	RS	RS
	Add1	No					
	Add2	No					
	Add3	No					
0	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
15	Mult1	M(A2)		(M-M+M)	(M-M)	Mult2			

Tomasulo Example: Cycle 16

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Exec Comp	Write Result
1	3	4
2	4	5
3	15	16
4	7	8
5		
6	10	11

Load	Busy	Address
Load1	No	
Load2	No	
Load3	No	

- Reservation stations

Time	Name	Busy	Op	S_1	S_2	RS	RS
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
40	Mult2	Yes	DIVD	M*F4	M(A1)		

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
16	M*F4	M(A2)		(M-M+M)	(M-M)	Mult2			

Tomasulo Example: Cycle 55

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Exec Comp	Write Result
1	3	4
2	4	5
3	15	16
4	7	8
5		
6	10	11

Load	Busy	Address
Load1	No	
Load2	No	
Load3	No	

- Reservation stations

Time	Name	Busy	Op	S_1	S_2	RS	RS
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
1	Mult2	Yes	DIVD	M*F4	M(A1)		

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
55	M*F4	M(A2)		(M-M+M)	(M-M)	Mult2			

Tomasulo Example: Cycle 56

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Exec Comp	Write Result
1	3	4
2	4	5
3	15	16
4	7	8
5	56	
6	10	11

Load	Busy	Address
Load1	No	
Load2	No	
Load3	No	

- Reservation stations

Time	Name	Busy	Op	S_1	S_2	RS	RS
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
0	Mult2	Yes	DIVD	M*F4	M(A1)		

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
56	M*F4	M(A2)		(M-M+M)	(M-M)	Mult2			

Tomasulo Example: Cycle 57

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Exec Comp	Write Result
1	3	4
2	4	5
3	15	16
4	7	8
5	56	57
6	10	11

Load	Busy	Address
Load1	No	
Load2	No	
Load3	No	

- Reservation stations

Time	Name	Busy	Op	S_1	S_2	RS	RS
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
	Mult2	No					

- Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
57	M*F4	M(A2)		(M-M+M)	(M-M)	Result			

Compare to Scoreboard: Cycle 62

- Instruction status

Instr	D	S_1	S_2
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issue	Write Result
1	4
2	5
3	16
4	8
5	57
6	11

Issue	Write Result
1	4
5	8
6	20
7	12
8	62
13	22

- Why take longer on scoreboard/6600?
  - Structural hazards
  - Lack of forwarding

Tomasulo Alogrithm vs. Scoreboard
(IBM 360/91 vs. CDC 6600)

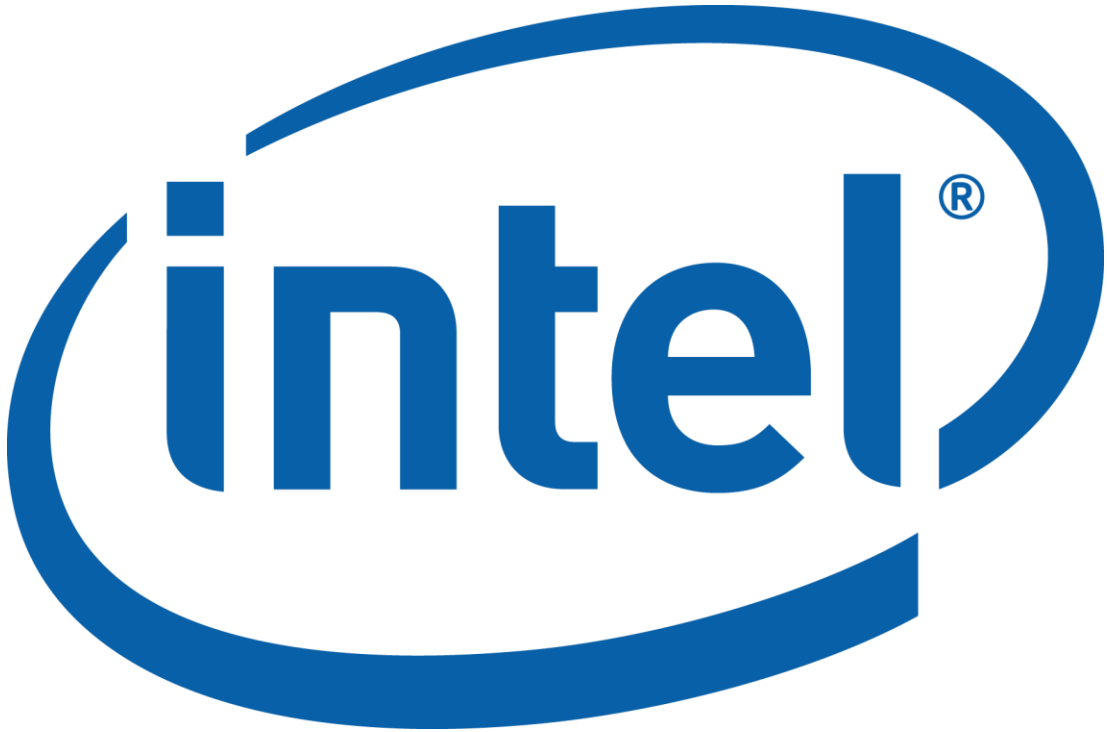
Feature	Tomasulo algorithm	Scoreboard
Functional units	Multiple, pipelined	Multiple, unpipelined
	6 load, 3 store, 3+, 2×/÷	1 load/store, 1+ , 2×, 1÷
Window size	≤14 instructions	≤5 instructions
On structural hazard	No issue	No issue
WAR	Renaming avoids	Stall completion
WAW	Renaming avoids	Stall issue
Data exchange	Broadcast results from FU	Write/read registers
Control	Reservation stations	Central scoreboard

Explicit Register Renaming

- Tomasulo provides implicit register renaming
  - User registers renamed to reservation station tags
- Explicit register renaming
  - Use physical register file larger than specified by ISA
  - Keep a translation table
    - When register is written, replace table entry with new register
    - Physical register becomes free when not being used by any instructions
  - Pipeline can be exactly like "standard" pipeline
- Advantages
  - Removes all WAR and WAW hazards
  - Allows data to be fetched from a single register file
  - Like Tomasulo, good for allowing full out-of-order retirement

Acknowledgements

- These slides contain material developed and copyright by:
  - Arvind (MIT)
  - Joel Emer (Intel/MIT)


intel