HY425 Lecture 06: Extracting more ILP: Multiple issue and speculation

Dimitrios S. Nikolopoulos

University of Crete and FORTH-ICS

October 21, 2011

Exploiting ILP in hardware

Dynamic scheduling

- Instruction scheduling critical for
  - Overlapping stalls due to hazards
  - Resolving hazards
- Two hardware dynamic scheduling schemes
  - Scoreboard: out-of-order execution, out-of-order completion, dependence tracking through register file, stall upon detection of hazards (RAW, WAW, WAR)
  - Tomasulo: out-of-order execution, out-of-order completion, renaming through FUs to resolve WAW, WAR hazards

Exploiting ILP in hardware

Branch prediction

- Reduce branch stall impact
- Dynamically predict next branch
  - Predict branch direction and return address
  - Use history of examined and possibly other branches
  - Adapt predictions as program executes

Getting CPI less than 1

CPI $<$ 1

- Prior techniques attempt to eliminate stalls and sustain high instruction graduation rate from the pipeline
  - At best one instruction can be graduate per cycle on average
- Attempts to graduate more than one instructions per cycle
  - Multiple-issue processors: Issue more than one instructions per cycle hoping to graduate more than one instructions per cycle
  - Two major categories: superscalar, VLIW
  - Goal: CPI $<$ 1, IPC $>$ 1

Getting CPI less than 1

Superscalar and VLIW processors

- Trade-off between static and dynamic instruction scheduling
  - Static scheduling places burden on software
  - Dynamic scheduling places burden on hardware
- Superscalar processors issue a variable number of instructions per cycle, up to a maximum, using static (compiler) or dynamic (hardware) scheduling
- VLIW processors issue a fixed number of instructions per cycle, seen as a packet (potentially with empty slots), and created and scheduled statically by the compiler

Superscalar execution

Characteristics

- Multiple instructions issued per cycle
- Dynamic (scoreboard, Tomasulo) or static (compiler) instruction scheduling
- Issue independent instructions, modulo constraints
  - Instructions dependent on pending instructions not issued
  - Instructions not issued due to hazards
  - FU constraints (e.g. one memory instruction per cycle, one INT/one FP instruction per cycle)
- Number of instructions issued per cycle variable
- CPI, IPC, limited by dependencies, hazards

Superscalar execution illustrated

- Assume example with one INT, one FP per cycle
- FP EX stage simplified to one cycle

Instruction type	Pipe stages							
INT instruction	IF	ID	EX	MEM	WB			
FP instruction	IF	ID	EX	MEM	WB			
INT instruction		IF	ID	EX	MEM	WB		
FP instruction		IF	ID	EX	MEM	WB		
INT instruction			IF	ID	EX	MEM	WB	
FP instruction			IF	ID	EX	MEM	WB	
INT instruction				IF	ID	EX	MEM	WB
FP instruction				IF	ID	EX	MEM	WB

Superscalar execution illustrated (cont.)

- Higher instruction fetch bandwidth
- INT instructions, loads and stores occupy one slot
- FP instructions occupy second slot
  - Need pipelined FP datapath, otherwise FP datapath becomes bottleneck

Instruction type	Pipe stages						
INT instruction	IF	ID	EX	MEM	WB		
FP instruction	IF	ID	EX	MEM	WB		
INT instruction		IF	ID	EX	MEM	WB	
FP instruction		IF	ID	EX	MEM	WB	
INT instruction			IF	ID	EX	MEM	WB
FP instruction			IF	ID	EX	MEM	WB

Superscalar vs. single-issue processors

- Similarities to dynamically scheduled processors
  - Multiple execution units (FP, INT)
  - Hazard detection and/or prevention logic
- FP loads and stores incur additional complexity
  - Contention for FP register file between load/store and arithmetic instructions
  - Potential RAW hazards between load and arithmetic instruction issued in same slot
  - Either prevent issue at performance loss or use more ports to register file

Superscalar vs. single-issue processors (cont.)

- Higher penalty for load-use hazard
  - Dependent arithmetic instruction following load requires 1 stall cycle
  - In superscalar dependent arithmetic instructions cannot issue in current or next cycle
  - 3 instead of 1 wasted instruction slots
- Higher penalty for branch stall
  - 3 instead of 1 wasted instruction slots during branch stall

Loops unrolling and instruction scheduling in superscalar

```
for (i=1; i<=1000; i=i+1;)
  x[i] = x[i] + s;
```

Assume latencies

Producer	Consumer	Latency
FP ALU op	Another FP ALU op	3
FP ALU op	store double	2
Load double	FP ALU op	1
Load double	store double	0

Loop unrolling and instruction scheduling in superscalar

Example in MIPS assembly

```
                                Clock cycle issued
Loop:   LD   F0, 0(R1)                1
        ADDD F4, F0, F2               3
        SD   F4, 0(R1)                6
        SUBI R1, R1, 8                7
        BNEZ R1, Loop                 8
```

- 9 cycles per loop iteration due to stalls

Loop unrolling in single-issue processor

Unroll 3 iterations with instruction scheduling

```
                              Clock cycle issued
Loop:  LD    F0, 0(R1)              1
       LD    F6, -8(R1)             2
       LD    F10, -16(R1)           3
       LD    F14, -24(R1)           4
       ADDD  F4, F0, F2             5
       ADDD  F8, F6, F2             6
       ADDD  F12, F10, F2           7
       ADDD  F16, F14, F2           8
       SD    F4, 0(R1)              9
       SD    F8, -8(R1)             10
       SD    F12, -16(R1)           11
       SUBI  R1, R1, 32             12
       BNEZ  R1, Loop               13
       SD    F16, 8(R1)              14
```

Improved instruction scheduling

- LD's up eliminate LD-ADDD 1-cycle stall
- ADDD's up eliminate ADDD-SD 2-cycle stalls
- SD down fills branch delay slot
- Need adjustment of SD indices
- 14 cycles for 4 iterations, or 3.5 cycles per iteration

Loop unrolling in superscalar processor

Unroll 3 iterations with dual-issue instruction scheduling

	INT	FP	Cycle
Loop:	LD F0,0(R1)		1
	LD F6,-8(R1)		2
	LD F10,-16(R1)	ADDD F4,F0,F2	3
		ADDD F8,F6,F2	4
		ADDD F12,F10,F2	5
	SD 0(R1),F4		6
	SD -8(R1),F8		7
	SUBI R1,R1,24		8
	BNEZ R1,Loop		9
	SD 8(R1),F12		10

- 10 cycles per 3 iterations, 3.3 cycles per iteration

Loop unrolling and static scheduling in superscalar

Unroll 5 iterations with dual-issue instruction scheduling

	INT		FP	Cycle
Loop:	LD	F0,0(R1)		1
	LD	F6,-8(R1)		2
	LD	F10,-16(R1)	ADDD F4,F0,F2	3
	LD	F14,-24(R1)	ADDD F8,F6,F2	4
	LD	F18,-32(R1)	ADDD F12,F10,F2	5
	SD	0(R1),F4	ADDD F16,F14,F2	6
	SD	-8(R1),F8	ADDD F20,F18,F2	7
	SD	-16(R1),F12		8
	SD	-24(R1),F16		9
	SUBI	R1,R1,40		9
	BNEZ	R1,Loop		11
	SD	8(R1),F20		12

- 12 cycles per 5 iterations, 2.4 cycles per iteration

Superscalar processor with dynamic scheduling

- Extend scoreboard or Tomasulo's algorithm to handle multiple issue
- Instructions issued in program order
- Can not issue dependent instruction in same cycle
  - Load-use hazard with FP instructions
  - Possible solutions:
    - Pipeline issue stage (split in halves) to issue dependent instructions in same cycle
    - Add logic to issue stage to process instructions and find dependencies

Dual-issue pipeline with Tomasulo

Assumptions

- Assumptions:
  - Change issue state to issue dependent instructions in same cycle
  - One INT, one FP instruction issued per cycle
  - One INT FU (loads, stores, arithmetic), as many FP FUs as the types of operations
  - Dynamic branch prediction with independent FU for computing branch condition
  - Latency of ALU INT ops is 1 cycle, loads is 2 cycles, and FP ops is 3 cycles (assuming FP ADDD execution takes 2 cycles)

Dual-issue pipeline with Tomasulo

Assume usual loop example

Iteration number	Instructions	Issues at	Starts execute	Memory access at	Writes CDB at	Comment
1	`LD F0,0(R1)`	1	2	3	4	First issue
1	`ADDD F4,F0,F2`	1	5		8	Wait for LD
1	`SD F4,0(R1)`	2	3	9		Wait for ADDD
1	`SUBI R1,R1,8`	3	4		5	Wait for INT FU
1	`BNEZ R1, Loop`	4	6			Wait for SUBI
2	`LD F0,0(R1)`	5	7	8	9	Wait for BNE
2	`ADDD F4,F0,F2`	5	10		13	Wait for LD
2	`SD F4,0(R1)`	6	8	14		Wait for ADDD
2	`SUBI R1,R1,8`	7	9		10	Wait for INT FU
2	`BNEZ R1, Loop`	8	11			Wait for SUBI
3	`LD F0,0(R1)`	9	12	13	14	Wait for BNE
3	`ADDD F4,F0,F2`	9	15		18	Wait for LD
3	`SD F4,0(R1)`	10	13	19		Wait for ADDD
3	`SUBI R1,R1,8`	11	14		15	Wait for INT FU
3	`BNEZ R1, Loop`	12	16			Wait for SUBI

- 19 cycles per 4 iterations, or 4.8 cycles per iteration

Dual-issue pipeline with Tomasulo
Assume separate FU for loads/stores

Iteration number	Instructions	Issues at	Starts execute	Memory access at	Writes CDB at	Comment
1	`LD F0,0(R1)`	1	2	3	4	First issue
1	`ADDD F4,F0,F2`	1	5		8	Wait for LD
1	`SD F4,0(R1)`	2	3	9		Wait for ADDD
1	`SUBI R1,R1,8`	2	3		4	Executes eariler
1	`BNEZ R1, Loop`	3	5			Wait for SUBI
2	`LD F0,0(R1)`	4	6	7	8	Wait for BNE
2	`ADDD F4,F0,F2`	4	9		12	Wait for LD
2	`SD F4,0(R1)`	5	7	13		Wait for ADDD
2	`SUBI R1,R1,8`	5	6		7	Executes earlier
2	`BNEZ R1, Loop`	6	8			Wait for SUBI
3	`LD F0,0(R1)`	7	9	10	11	Wait for BNE
3	`ADDD F4,F0,F2`	7	12		15	Wait for LD
3	`SD F4,0(R1)`	8	10	16		Wait for ADDD
3	`SUBI R1,R1,8`	8	9		10	Executes earlier
3	`BNEZ R1, Loop`	9	11			Wait for SUBI

- Eliminates structural hazards between SUBI and SD. LD not issued together with branch because of control hazard. 16 cycles per 4 iterations, or 4 cycles per iteration

Limitations of multiple-issue processors

Software and hardware implications

- Inherent limitations of ILP in programs (control and data dependencies)
  - Need as many independent instructions as pipeline depth
  - Deep unrolling, register pressure
- Hardware complexity increasing rapidly with instructions issued per cycle
  - Adding FUs scales complexity linearly
  - Higher memory and register-file bandwidth, more ports
  - Memory system implications, interleaving
  - Dynamic scheduling expensive
  - Issue logic of dynamic scheduled processors expensive
- Middle ground: combination of static (compiler) and dynamic scheduling

Motivation for speculative execution

Increasing available ILP

- Multiple-issue increases demand for independent instructions
- May need to predict one branch per cycle
  - Hard to maintain high clock rate
- Speculation attempts to overcome problem
  - Execute program as if all branches were predicted correctly
  - Dynamic scheduling waits for branch resolution

Motivation for speculative execution

Increasing available ILP

- Three elements of speculation
  - Dynamic branch prediction
  - Ability to undo speculative instructions from wrong path
  - Dynamic scheduling

Implementing speculative execution

Data flow details

- Instructions pass results to other instructions through CDB
- Speculative instructions pass results to following speculative instructions without committing to registers or memory, and without throwing exceptions
- Update register or memory only when we know that instruction is non-speculative
- Additional commit step in instruction execution
  - Execute out-of-order, commit in-order
  - Avoid irrevocable change of state or exception before checking speculation status of instruction

Implementing speculative execution

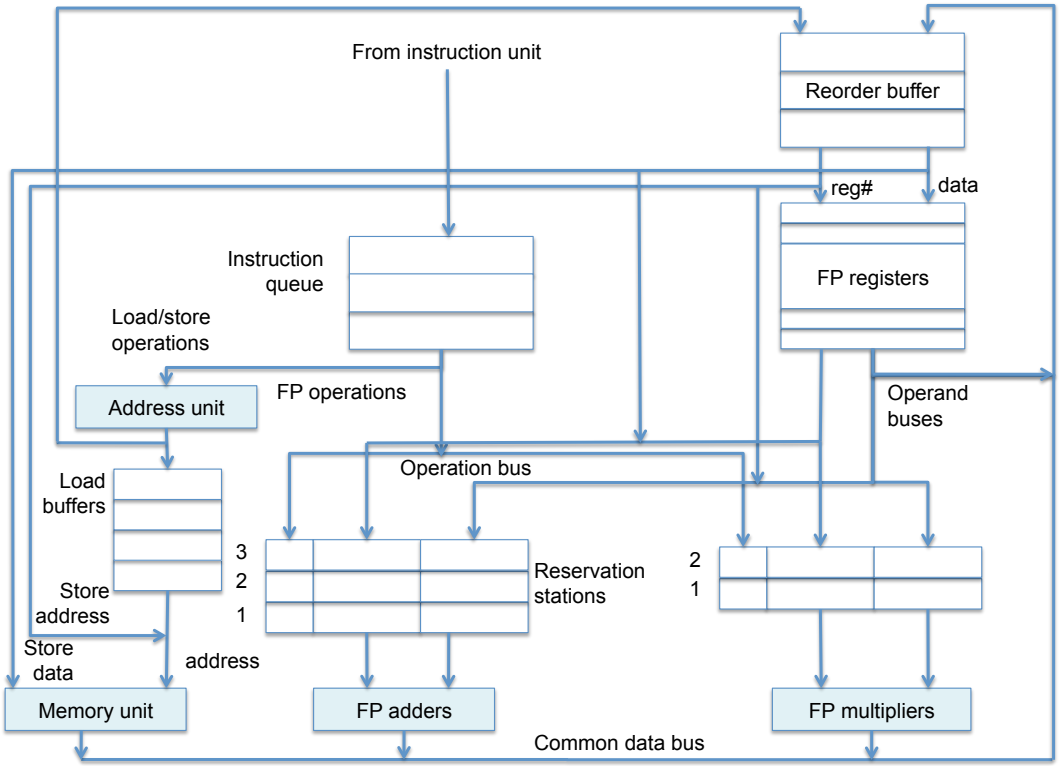
Reorder buffer (ROB)

- Separate instruction completion from commit of results
- Reorder buffer (ROB) used to hold temporarily results of speculative instructions
  - Results forwarded through ROB
  - ROB functions similarly to reservation stations or registers
  - Similar to Tomasulo's store buffer used to preserve memory dependencies
- ROB entry

Instruction type (branch,store,reg)	destination (reg/mem)	value	ready

Tomasulo's algorithm with speculation

Hardware extensions



Details

Reorder buffer holds instruction results until commit.
Stores checked against prior loads for memory RAW hazard

Example of speculation with dynamic scheduling

MIPS assembly

```
LD   F6, 34(R2)
LD   F2, 45(R3)
MULD F0, F2, F4
SUBD F8, F6, F2
DIVD F10, F0, F6
ADDD F6, F8, F2
```

- ADDD 2 cycles, MULD 10 cycles, DIVD 40 cycles

Tomasulo scheduling with speculation – Cycle 1

Instruction status:

Instruction		j	k	Issue	Exec Comp	Write Result	Commit
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				

	Busy	Addr.	Dest
Load1	Yes	34+R2	1
Load2	No		
Load3	No		

Reorder buffer:

Entry	Busy	Instruction	State	Destination	Value
1	Yes	LD F6,34(R2)	Issue	F6	

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk	Dest
	Add1	No						
	Add2	No						
	Add3	No						
	Mult1	No						
	Mult2	No						

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
1	ROB#				1					
	Busy	No	No	No	Yes	No	No	No	No	No

Tomasulo scheduling with speculation – Cycle 2

Instruction status:

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

	Busy	Addr.	Dest
Load1	Yes	34+R2	1
Load2	Yes	45+R3	2
Load3	No		

Reorder buffer:

Entry	Busy	Instruction	State	Destination	Value
1	Yes	LD F6,34(R2)	Execute	F6	
2	Yes	LD F2,45(R3)	Issue	F2	

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk	Dest
	Add1	No						
	Add2	No						
	Add3	No						
	Mult1	No						
	Mult2	No						

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
2	ROB#		2		1					
	Busy	No	Yes	No	Yes	No	No	No	No	No

Tomasulo scheduling with speculation – Cycle 3

Instruction status:

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

	Busy	Addr.	Dest
Load1	Yes	34+R2	1
Load2	Yes	45+R3	2
Load3	No		

Reorder buffer:

Entry	Busy	Instruction	State	Destination	Value
1	Yes	LD F6,34(R2)	Execute	F6	
2	Yes	LD F2,45(R3)	Execute	F2	
3	Yes	MULTD F0,F2,F4	Issue	F0	

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk	Dest
	Add1	No						
	Add2	No						
	Add3	No						
	Mult1	Yes	MULTD		R(F4)	2		3
	Mult2	No						

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
3	ROB#	3	2		1					
	Busy	Yes	Yes	No	Yes	No	No	No	No	No

Tomasulo scheduling with speculation – Cycle 4

Instruction status:

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

	Busy	Addr	Dest
Load1	No		
Load2	Yes	45+R3	2
Load3	No		

Reorder buffer:

Entry	Busy	Instruction	State	Destination	Value
1	Yes	LD F6,34(R2)	Write	F6	M[R2+34]
2	Yes	LD F2,45(R3)	Execute	F2	
3	Yes	MULTD F0,F2,F4	Issue	F0	
4	Yes	SUBD F8,F6,F2	Issue	F8	

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk	Dest
	Add1	Yes	SUBD	M[R2+34]			2	4
	Add2	No						
	Add3	No						
	Mult1	Yes	MULTD		R(F4)	2		3
	Mult2	No						

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
4	ROB#	3	2		1	4				
	Busy	Yes	Yes	No	Yes	Yes	No	No	No	No

Tomasulo scheduling with speculation – Cycle 5

Instruction status:

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

	Busy	Addr	Dest
Load1	No		
Load2	No		
Load3	No		

Reorder buffer:

Entry	Busy	Instruction	State	Destination	Value
1	No	LD F6,34(R2)	Commit	F6	M[R2+34]
2	Yes	LD F2,45(R3)	Write	F2	M[R3+45]
3	Yes	MULTD F0,F2,F4	Issue	F0	
4	Yes	SUBD F8,F6,F2	Issue	F8	
5	Yes	DIVD F10,F0,F6	Issue	F10	

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk	Dest
	Add1	Yes	SUBD	M[R2+34]	M[R3+45]			4
	Add2	No						
	Add3	No						
	Mult1	Yes	MULTD	M[R3+45]	R(F4)			3
	Mult2	Yes	DIVD		R(F6)	3		5

Register result status:

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
5	3	2			4	5			
Busy	Yes	Yes	No	No	Yes	Yes	No	No	No

Tomasulo scheduling with speculation – Cycle 6

Instruction status:

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

	Busy	Addr	Dest
Load1	No		
Load2	No		
Load3	No		

Reorder buffer:

Entry	Busy	Instruction	State	Destination	Value
1	No	LD F6,34(R2)	Commit	F6	M[R2+34]
2	No	LD F2,45(R3)	Commit	F2	M[R3+45]
3	Yes	MULTD F0,F2,F4	Execute	F0	
4	Yes	SUBD F8,F6,F2	Execute	F8	
5	Yes	DIVD F10,F0,F6	Issue	F10	
6	Yes	ADDD F6,F8,F2	Issue	F6	

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk	Dest
2	Add1	Yes	SUBD	M[R2+34]	M[R3+45]			4
	Add2	Yes	ADDD		R(F2)	4		6
	Add3	No						
10	Mult1	Yes	MULTD	M[R3+45]	R(F4)			3
	Mult2	Yes	DIVD		R(F6)	3		5

Register result status:

Clock **6**

	F0	F2	F4	F6	F8	F10	F12	...	F30
ROB#	3			6	4	5			
Busy	Yes	No	No	Yes	Yes	Yes	No	No	No

Tomasulo scheduling with speculation – Cycle 7

Instruction status:

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

	Busy	Addr	Dest
Load1	No		
Load2	No		
Load3	No		

Reorder buffer:

Entry	Busy	Instruction	State	Destination	Value
1	No	LD F6,34(R2)	Commit	F6	M[R2+34]
2	No	LD F2,45(R3)	Commit	F2	M[R3+45]
3	Yes	MULTD F0,F2,F4	Execute	F0	
4	Yes	SUBD F8,F6,F2	Execute	F8	
5	Yes	DIVD F10,F0,F6	Issue	F10	
6	Yes	ADDD F6,F8,F2	Issue	F6	

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk	Dest
1	Add1	Yes	SUBD	M[R2+34]	M[R3+45]			4
	Add2	Yes	ADDD		R(F2)	4		6
	Add3	No						
9	Mult1	Yes	MULTD	M[R3+45]	R(F4)			3
	Mult2	Yes	DIVD		R(F6)	3		5

Register result status:

Clock 7

	F0	F2	F4	F6	F8	F10	F12	...	F30
ROB#	3			6	4	5			
Busy	Yes	No	No	Yes	Yes	Yes	No	No	No

Tomasulo scheduling with speculation – Cycle 8

Instruction status:

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

	Busy	Addr	Dest
Load1	No		
Load2	No		
Load3	No		

Reorder buffer:

Entry	Busy	Instruction	State	Destination	Value
1	No	LD F6,34(R2)	Commit	F6	M[R2+34]
2	No	LD F2,45(R3)	Commit	F2	M[R3+45]
3	Yes	MULTD F0,F2,F4	Execute	F0	
4	Yes	SUBD F8,F6,F2	Execute	F8	
5	Yes	DIVD F10,F0,F6	Issue	F10	
6	Yes	ADDD F6,F8,F2	Issue	F6	

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk	Dest
0	Add1	Yes	SUBD	M[R2+34]	M[R3+45]			4
	Add2	Yes	ADDD		R(F2)	4		6
	Add3	No						
8	Mult1	Yes	MULTD	M[R3+45]	R(F4)			3
	Mult2	Yes	DIVD		R(F6)	3		5

Register result status:

Clock **8**

	F0	F2	F4	F6	F8	F10	F12	...	F30
ROB#	3			6	4	5			
Busy	Yes	No	No	Yes	Yes	Yes	No	No	No

Tomasulo scheduling with speculation – Cycle 9

Instruction status:

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

	Busy	Addr	Dest
Load1	No		
Load2	No		
Load3	No		

Reorder buffer:

Entry	Busy	Instruction	State	Destination	Value
1	No	LD F6,34(R2)	Commit	F6	M[R2+34]
2	No	LD F2,45(R3)	Commit	F2	M[R3+45]
3	Yes	MULTD F0,F2,F4	Execute	F0	
4	Yes	SUBD F8,F6,F2	Write	F8	M-M
5	Yes	DIVD F10,F0,F6	Issue	F10	
6	Yes	ADDD F6,F8,F2	Issue	F6	

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk	Dest
	Add1	No						4
	Add2	Yes	ADDD	M-M	R(F2)	4		6
	Add3	No						
7	Mult1	Yes	MULTD	M[R3+45]	R(F4)			3
	Mult2	Yes	DIVD		R(F6)	3		5

Register result status:

Clock **9**

	F0	F2	F4	F6	F8	F10	F12	...	F30
ROB#	3			6	4	5			
Busy	Yes	No	No	Yes	Yes	Yes	No	No	No

Tomasulo scheduling with speculation – Cycle 10

Instruction status:

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

	Busy	Addr	Dest
Load1	No		
Load2	No		
Load3	No		

Reorder buffer:

Entry	Busy	Instruction	State	Destination	Value
1	No	LD F6,34(R2)	Commit	F6	M[R2+34]
2	No	LD F2,45(R3)	Commit	F2	M[R3+45]
3	Yes	MULTD F0,F2,F4	Execute	F0	
4	Yes	SUBD F8,F6,F2	Write	F8	M-M
5	Yes	DIVD F10,F0,F6	Issue	F10	
6	Yes	ADDD F6,F8,F2	Execute	F6	

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk	Dest
	Add1	No						
2	Add2	Yes	ADDD	M-M	R(F2)	4		6
	Add3	No						
6	Mult1	Yes	MULTD	M[R3+45]	R(F4)			3
	Mult2	Yes	DIVD		R(F6)	3		5

Register result status:

Clock **10**

	F0	F2	F4	F6	F8	F10	F12	...	F30
ROB#	3			6	4	5			
Busy	Yes	No	No	Yes	Yes	Yes	No	No	No

Tomasulo scheduling with speculation – Cycle 11

Instruction status:

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

	Busy	Addr	Dest
Load1	No		
Load2	No		
Load3	No		

Reorder buffer:

Entry	Busy	Instruction	State	Destination	Value
1	No	LD F6,34(R2)	Commit	F6	M[R2+34]
2	No	LD F2,45(R3)	Commit	F2	M[R3+45]
3	Yes	MULTD F0,F2,F4	Execute	F0	
4	Yes	SUBD F8,F6,F2	Write	F8	M-M
5	Yes	DIVD F10,F0,F6	Issue	F10	
6	Yes	ADDD F6,F8,F2	Execute	F6	

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk	Dest
	Add1	No						
1	Add2	Yes	ADDD	M-M	R(F2)	4		6
	Add3	No						
5	Mult1	Yes	MULTD	M[R3+45]	R(F4)			3
	Mult2	Yes	DIVD		R(F6)	3		5

Register result status:

Clock **11**

	F0	F2	F4	F6	F8	F10	F12	...	F30
ROB#	3			6	4	5			
Busy	Yes	No	No	Yes	Yes	Yes	No	No	No

Tomasulo scheduling with speculation – Cycle 12

tion status:

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

	Busy	Addr	Dest
Load1	No		
Load2	No		
Load3	No		

er buffer:

Entry	Busy	Instruction	State	Destination	Value
1	No	LD F6,34(R2)	Commit	F6	M[R2+34]
2	No	LD F2,45(R3)	Commit	F2	M[R3+45]
3	Yes	MULTD F0,F2,F4	Execute	F0	
4	Yes	SUBD F8,F6,F2	Write	F8	M-M
5	Yes	DIVD F10,F0,F6	Issue	F10	
6	Yes	ADDD F6,F8,F2	Execute	F6	

vation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk	Dest
	Add1	No						
0	Add2	Yes	ADDD	M-M	R(F2)	4		6
	Add3	No						
4	Mult1	Yes	MULTD	M[R3+45]	R(F4)			3
	Mult2	Yes	DIVD		R(F6)	3		5

er result status:

Clock
12

	F0	F2	F4	F6	F8	F10	F12	...	F30
ROB#	3			6	4	5			
Busy	Yes	No	No	Yes	Yes	Yes	No	No	No

Tomasulo scheduling with speculation – Cycle 13

Instruction status:

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

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reorder buffer:

Entry	Busy	Instruction	State	Destination	Value
1	No	LD F6,34(R2)	Commit	F6	M[R2+34]
2	No	LD F2,45(R3)	Commit	F2	M[R3+45]
3	Yes	MULTD F0,F2,F4	Execute	F0	
4	Yes	SUBD F8,F6,F2	Write	F8	M-M
5	Yes	DIVD F10,F0,F6	Issue	F10	
6	Yes	ADDD F6,F8,F2	Write	F6	M-M+M

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk	Dest
	Add1	No						
	Add2	No						
	Add3	No						
3	Mult1	Yes	MULTD	M[R3+45]	R(F4)			3
	Mult2	Yes	DIVD		R(F6)	3		5

Register result status:

Clock **13**

	F0	F2	F4	F6	F8	F10	F12	...	F30
ROB#	3			6	4	5			
Busy	Yes	No	No	Yes	Yes	Yes	No	No	No

Tomasulo scheduling with speculation – Cycle 14

Instruction status:

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

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reorder buffer:

Entry	Busy	Instruction	State	Destination	Value
1	No	LD F6,34(R2)	Commit	F6	M[R2+34]
2	No	LD F2,45(R3)	Commit	F2	M[R3+45]
3	Yes	MULTD F0,F2,F4	Execute	F0	
4	Yes	SUBD F8,F6,F2	Write	F8	M-M
5	Yes	DIVD F10,F0,F6	Issue	F10	
6	Yes	ADDD F6,F8,F2	Write	F6	M-M+M

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk	Dest
	Add1	No						
	Add2	No						
	Add3	No						
2	Mult1	Yes	MULTD	M[R3+45]	R(F4)			3
	Mult2	Yes	DIVD		R(F6)	3		5

Register result status:

Clock **14**

	F0	F2	F4	F6	F8	F10	F12	...	F30
ROB#	3			6	4	5			
Busy	Yes	No	No	Yes	Yes	Yes	No	No	No

Tomasulo scheduling with speculation – Cycle 15

Instruction status:

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

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reorder buffer:

Entry	Busy	Instruction	State	Destination	Value
1	No	LD F6,34(R2)	Commit	F6	M[R2+34]
2	No	LD F2,45(R3)	Commit	F2	M[R3+45]
3	Yes	MULTD F0,F2,F4	Execute	F0	
4	Yes	SUBD F8,F6,F2	Write	F8	M-M
5	Yes	DIVD F10,F0,F6	Issue	F10	
6	Yes	ADDD F6,F8,F2	Write	F6	M-M+M

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk	Dest
	Add1	No						
	Add2	No						
	Add3	No						
1	Mult1	Yes	MULTD	M[R3+45]	R(F4)			3
	Mult2	Yes	DIVD		R(F6)	3		5

Register result status:

Clock **15**

	F0	F2	F4	F6	F8	F10	F12	...	F30
ROB#	3			6	4	5			
Busy	Yes	No	No	Yes	Yes	Yes	No	No	No

Tomasulo scheduling with speculation – Cycle 16

Instruction status:

Instruction		j	k	Issue	Exec Comp	Write Result	Commit
LD	F6	34+	R2	1	3	4	5
LD	F2	45+	R3	2	4	5	6
MULTD	F0	F2	F4	3	16		
SUBD	F8	F6	F2	4	8	9	
DIVD	F10	F0	F6	5			
ADDD	F6	F8	F2	6	12	13	

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reorder buffer:

Entry	Busy	Instruction	State	Destination	Value
1	No	LD F6,34(R2)	Commit	F6	M[R2+34]
2	No	LD F2,45(R3)	Commit	F2	M[R3+45]
3	Yes	MULTD F0,F2,F4	Execute	F0	
4	Yes	SUBD F8,F6,F2	Write	F8	M-M
5	Yes	DIVD F10,F0,F6	Issue	F10	
6	Yes	ADDD F6,F8,F2	Write	F6	M-M+M

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk	Dest
	Add1	No						
	Add2	No						
	Add3	No						
0	Mult1	Yes	MULTD	M[R3+45]	R(F4)			3
	Mult2	Yes	DIVD		R(F6)	3		5

Register result status:

Clock **16**

	F0	F2	F4	F6	F8	F10	F12	...	F30
ROB#	3			6	4	5			
Busy	Yes	No	No	Yes	Yes	Yes	No	No	No

Tomasulo scheduling with speculation – Cycle 17

Instruction status:

Instruction		j	k	Issue	Exec Comp	Write Result	Commit
LD	F6	34+	R2	1	3	4	5
LD	F2	45+	R3	2	4	5	6
MULTD	F0	F2	F4	3	16	17	
SUBD	F8	F6	F2	4	8	9	
DIVD	F10	F0	F6	5			
ADDD	F6	F8	F2	6	12	13	

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reorder buffer:

Entry	Busy	Instruction	State	Destination	Value
1	No	LD F6,34(R2)	Commit	F6	M[R2+34]
2	No	LD F2,45(R3)	Commit	F2	M[R3+45]
3	Yes	MULTD F0,F2,F4	Write	F0	M*F4
4	Yes	SUBD F8,F6,F2	Write	F8	M-M
5	Yes	DIVD F10,F0,F6	Issue	F10	
6	Yes	ADDD F6,F8,F2	Write	F6	M-M+M

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk	Dest
	Add1	No						
	Add2	No						
	Add3	No						
	Mult1	No						
	Mult2	Yes	DIVD	M*F4	R(F6)			5

Register result status:

Clock **17**

	F0	F2	F4	F6	F8	F10	F12	...	F30
ROB#	3			6	4	5			
Busy	Yes	No	No	Yes	Yes	Yes	No	No	No

Tomasulo scheduling with speculation – Cycle 18

Instruction status:

Instruction		j	k	Issue	Exec Comp	Write Result	Commit
LD	F6	34+	R2	1	3	4	5
LD	F2	45+	R3	2	4	5	6
MULTD	F0	F2	F4	3	16	17	18
SUBD	F8	F6	F2	4	8	9	
DIVD	F10	F0	F6	5			
ADDD	F6	F8	F2	6	12	13	

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reorder buffer:

Entry	Busy	Instruction	State	Destination	Value
1	No	LD F6,34(R2)	Commit	F6	M[R2+34]
2	No	LD F2,45(R3)	Commit	F2	M[R3+45]
3	No	MULTD F0,F2,F4	Commit	F0	M*F4
4	Yes	SUBD F8,F6,F2	Write	F8	M-M
5	Yes	DIVD F10,F0,F6	Execute	F10	
6	Yes	ADDD F6,F8,F2	Write	F6	M-M+M

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk	Dest
	Add1	No						
	Add2	No						
	Add3	No						
40	Mult1	No						
	Mult2	Yes	DIVD	M*F4	R(F6)			5

Register result status:

Clock **18**

	F0	F2	F4	F6	F8	F10	F12	...	F30
ROB#				6	4	5			
Busy	No	No	No	Yes	Yes	Yes	No	No	No

Tomasulo scheduling with speculation – Cycle 19

Instruction status:

Instruction		j	k	Issue	Exec Comp	Write Result	Commit
LD	F6	34+	R2	1	3	4	5
LD	F2	45+	R3	2	4	5	6
MULTD	F0	F2	F4	3	16	17	18
SUBD	F8	F6	F2	4	8	9	19
DIVD	F10	F0	F6	5			
ADDD	F6	F8	F2	6	12	13	

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reorder buffer:

Entry	Busy	Instruction	State	Destination	Value
1	No	LD F6,34(R2)	Commit	F6	M[R2+34]
2	No	LD F2,45(R3)	Commit	F2	M[R3+45]
3	No	MULTD F0,F2,F4	Commit	F0	M*F4
4	No	SUBD F8,F6,F2	Commit	F8	M-M
5	Yes	DIVD F10,F0,F6	Execute	F10	
6	Yes	ADDD F6,F8,F2	Write	F6	M-M+M

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk	Dest
	Add1	No						
	Add2	No						
	Add3	No						
39	Mult1	No						
	Mult2	Yes	DIVD	M*F4	R(F6)			5

Register result status:

Clock **19**

	F0	F2	F4	F6	F8	F10	F12	...	F30
ROB#				6		5			
Busy	No	No	No	Yes	No	Yes	No	No	No

Tomasulo scheduling with speculation – Cycle 20

Instruction status:

Instruction		j	k	Issue	Exec Comp	Write Result	Commit
LD	F6	34+	R2	1	3	4	5
LD	F2	45+	R3	2	4	5	6
MULTD	F0	F2	F4	3	16	17	18
SUBD	F8	F6	F2	4	8	9	19
DIVD	F10	F0	F6	5			
ADDD	F6	F8	F2	6	12	13	

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reorder buffer:

Entry	Busy	Instruction	State	Destination	Value
1	No	LD F6,34(R2)	Commit	F6	M[R2+34]
2	No	LD F2,45(R3)	Commit	F2	M[R3+45]
3	No	MULTD F0,F2,F4	Commit	F0	M*F4
4	No	SUBD F8,F6,F2	Commit	F8	M-M
5	Yes	DIVD F10,F0,F6	Execute	F10	
6	Yes	ADDD F6,F8,F2	Write	F6	M-M+M

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk	Dest
	Add1	No						
	Add2	No						
	Add3	No						
38	Mult1	No						
	Mult2	Yes	DIVD	M*F4	R(F6)			5

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
20	ROB#				6		5			
	Busy	No	No	No	Yes	No	Yes	No	No	No

Tomasulo scheduling with speculation – Cycle 58

Instruction status:

Instruction		j	k	Issue	Exec Comp	Write Result	Commit
LD	F6	34+	R2	1	3	4	5
LD	F2	45+	R3	2	4	5	6
MULTD	F0	F2	F4	3	16	17	18
SUBD	F8	F6	F2	4	8	9	19
DIVD	F10	F0	F6	5	58		
ADDD	F6	F8	F2	6	12	13	

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reorder buffer:

Entry	Busy	Instruction	State	Destination	Value
1	No	LD F6,34(R2)	Commit	F6	M[R2+34]
2	No	LD F2,45(R3)	Commit	F2	M[R3+45]
3	No	MULTD F0,F2,F4	Commit	F0	M*F4
4	No	SUBD F8,F6,F2	Commit	F8	M-M
5	Yes	DIVD F10,F0,F6	Execute	F10	
6	Yes	ADDD F6,F8,F2	Write	F6	M-M+M

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk	Dest
	Add1	No						
	Add2	No						
	Add3	No						
0	Mult1	No						
	Mult2	Yes	DIVD	M*F4	R(F6)			5

Register result status:

Clock **58**

	F0	F2	F4	F6	F8	F10	F12	...	F30
ROB#				6		5			
Busy	No	No	No	Yes	No	Yes	No	No	No

Tomasulo scheduling with speculation – Cycle 59

Instruction status:

Instruction		j	k	Issue	Exec Comp	Write Result	Commit
LD	F6	34+	R2	1	3	4	5
LD	F2	45+	R3	2	4	5	6
MULTD	F0	F2	F4	3	16	17	18
SUBD	F8	F6	F2	4	8	9	19
DIVD	F10	F0	F6	5	58	59	
ADDD	F6	F8	F2	6	12	13	

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reorder buffer:

Entry	Busy	Instruction	State	Destination	Value
1	No	LD F6,34(R2)	Commit	F6	M[R2+34]
2	No	LD F2,45(R3)	Commit	F2	M[R3+45]
3	No	MULTD F0,F2,F4	Commit	F0	M*F4
4	No	SUBD F8,F6,F2	Commit	F8	M-M
5	Yes	DIVD F10,F0,F6	Write	F10	(M*F4)/M
6	Yes	ADDD F6,F8,F2	Write	F6	M-M+M

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk	Dest
	Add1	No						
	Add2	No						
	Add3	No						
0	Mult1	No						
	Mult2	No						

Register result status:

Clock **59**

	F0	F2	F4	F6	F8	F10	F12	...	F30
ROB#				6		5			
Busy	No	No	No	Yes	No	Yes	No	No	No

Tomasulo scheduling with speculation – Cycle 60

Instruction status:

Instruction		j	k	Issue	Exec Comp	Write Result	Commit
LD	F6	34+	R2	1	3	4	5
LD	F2	45+	R3	2	4	5	6
MULTD	F0	F2	F4	3	16	17	18
SUBD	F8	F6	F2	4	8	9	19
DIVD	F10	F0	F6	5	58	59	60
ADDD	F6	F8	F2	6	12	13	

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reorder buffer:

Entry	Busy	Instruction	State	Destination	Value
1	No	LD F6,34(R2)	Commit	F6	M[R2+34]
2	No	LD F2,45(R3)	Commit	F2	M[R3+45]
3	No	MULTD F0,F2,F4	Commit	F0	M*F4
4	No	SUBD F8,F6,F2	Commit	F8	M-M
5	No	DIVD F10,F0,F6	Commit	F10	(M*F4)/M
6	Yes	ADDD F6,F8,F2	Write	F6	M-M+M

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk	Dest
	Add1	No						
	Add2	No						
	Add3	No						
0	Mult1	No						
	Mult2	No						

Register result status:

Clock **60**

	F0	F2	F4	F6	F8	F10	F12	...	F30
ROB#				6					
Busy	No	No	No	Yes	No	No	No	No	No

Tomasulo scheduling with speculation – Cycle 61

Instruction status:

Instruction		j	k	Issue	Exec Comp	Write Result	Commit
LD	F6	34+	R2	1	3	4	5
LD	F2	45+	R3	2	4	5	6
MULTD	F0	F2	F4	3	16	17	18
SUBD	F8	F6	F2	4	8	9	19
DIVD	F10	F0	F6	5	58	59	60
ADDD	F6	F8	F2	6	12	13	61

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reorder buffer:

Entry	Busy	Instruction	State	Destination	Value
1	No	LD F6,34(R2)	Commit	F6	M[R2+34]
2	No	LD F2,45(R3)	Commit	F2	M[R3+45]
3	No	MULTD F0,F2,F4	Commit	F0	M*F4
4	No	SUBD F8,F6,F2	Commit	F8	M-M
5	No	DIVD F10,F0,F6	Commit	F10	(M*F4)/M
6	No	ADDD F6,F8,F2	Commit	F6	M-M+M

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk	Dest
	Add1	No						
	Add2	No						
	Add3	No						
0	Mult1	No						
	Mult2	No						

Register result status:

Clock **61**

	F0	F2	F4	F6	F8	F10	F12	...	F30
ROB#									
Busy	No	No	No	No	No	No	No	No	No

Extracting more ILP

- Multiple instructions issued per cycle
  - Natural parallelism in loops
  - Increased hardware complexity
    - More ports to register files, memory, other resources
    - Branch predictors, potentially for multiple branches simultaneously
    - Complex issue logic and control logic
- Speculation
  - Look for parallelism beyond branches
  - Need easy roll-back – ROB
  - In-order commit limits performance
  - Combined with multiple-issue regains performance loss