Multiple issue processors

E. Sanchez, M. Sonza Reorda

Politecnico di Torino

Dipartimento di Automatica e Informatica (DAUIN)

Torino - Italy

INTRODUCTION

- The techniques for reducing the effects of data and control dependencies can be further exploited to obtain a CPI less than one: this requires issuing more than one instruction per clock cycle.
- There are two kinds of processors able to do so:
  - *superscalar* processors, either statically or dynamically scheduling instructions
  - *Very Long Instruction Word* (VLIW) processors.
- In both groups the architecture obviously includes multiple functional units.

MULTIPLE-ISSUE STATIC SCHEDULING

Implementing a superscalar processor able to possibly issue several instructions according to the static order defined by the compiler is rather easy and inexpensive.

Statically scheduled superscalar MIPS version

- Two instructions can be issued per clock cycle if:
  - one is a load, store, branch, or integer ALU operation
  - the other is any FP operation (but load and store, which belong to the first category).
- Two instructions (64 bits) are fetched and decoded at every clock cycle.
- The two instructions are aligned on a 64-bit boundary and constitute an *issue packet*.

Instruction order within the issue packet

- Old superscalar processors required a fixed structure for issue packets (e.g., that the integer instruction is always the first one).
- Current processors normally do not have this limitation.

Ideal situation

FP instructions are all assumed to last for 3 clock cycles

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

Instruction fetching

- At each clock cycle 2 instructions (64 bits) must be read from the instruction memory (i.e., from the cache).
- If the two instructions belong to different cache blocks, several processors fetch one instruction, only.
- In most of the cases, the issue packet may only contain one branch instruction.

FP Units

- In order to obtain a real benefit, the floating point units should be either pipelined, or multiple and independent.

FP Register contention

- When the first instruction is a FP load, store, or move, there is a possible contention for a FP register port.
- Possible solutions are:
  - forcing the first instruction to be executed by itself
  - giving the FP register file an additional port.

Possible RAW Hazard

- When the first instruction is a FP load, store, or move, and the second reads its result, a RAW hazard is also possible.
- In this case the second instruction must then be delayed by one clock cycle.

Data and Branch Delay

- In the MIPS pipeline, load has a latency of one clock cycle, which means that in the superscalar pipeline the result of a load instruction can not be used on the same clock cycle but on the next one.
- Therefore, in the static MIPS superscalar version the load delay slot (as well as the branch delay slot) becomes equal to *three* instructions.

Data and Branch Delay

Load instruction

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

Conclusions

- Static multiple-issue scheduling is mainly adopted by processors for the embedded market.

MULTIPLE-ISSUE DYNAMIC SCHEDULING

- It can be obtained by adopting a scheme similar to the Tomasulo one.
- To make the implementation easier, instructions are never issued to the reservation stations out-of-order.

CDB criticality

- In some clock cycles, more than one instruction may be ready to write on the CDB, that can only service one instruction at a time.
- For this reason, duplicating the CDB can provide higher performance, at the cost of some area overhead.

Example

Consider the following code

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

Let us suppose that up to two instructions can issue and commit per clock cycle and consider the two cases:

- Without speculation
- With speculation.

Case 1

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

Case 2

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

Performance evaluation

- In the first case the first 3 iterations require more than 19 clock cycles.
- In the second one, they require 14 clock cycles.