Exercise Session 4

Static BP+Complex, VLIW, Scoreboard

Advanced Computer Architectures

Politecnico di Milano April 2nd, 2025

Alessandro Verosimile <alessandro.verosimile@polimi.it>





Recall: Static Branch Prediction Techniques

Branch Always Not Taken (Predicted-Not-Taken)

Branch Always Taken (Predicted-Taken)

Backward Taken Forward Not Taken (BTFNT)

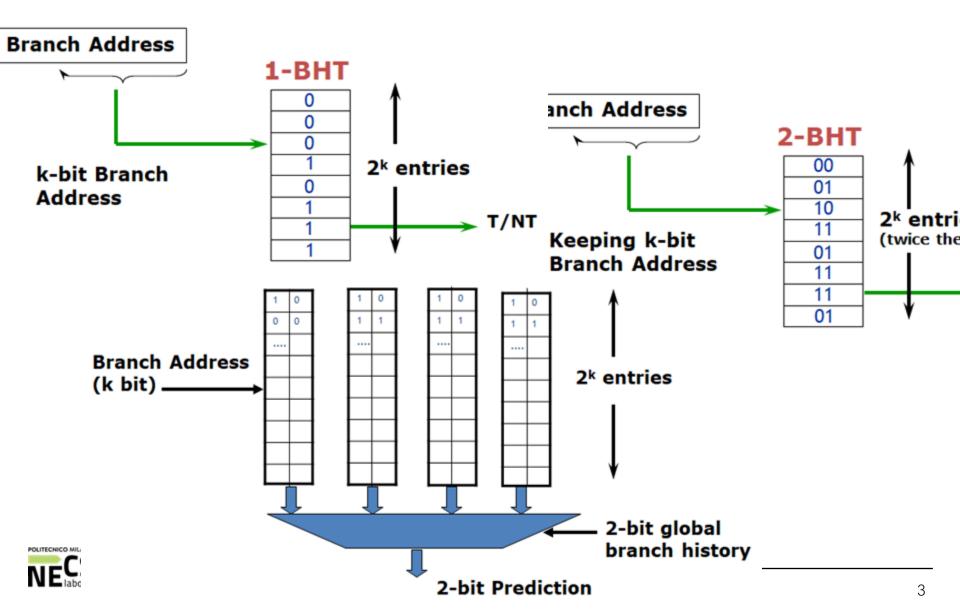
Profile-Driven Prediction

Delayed Branch

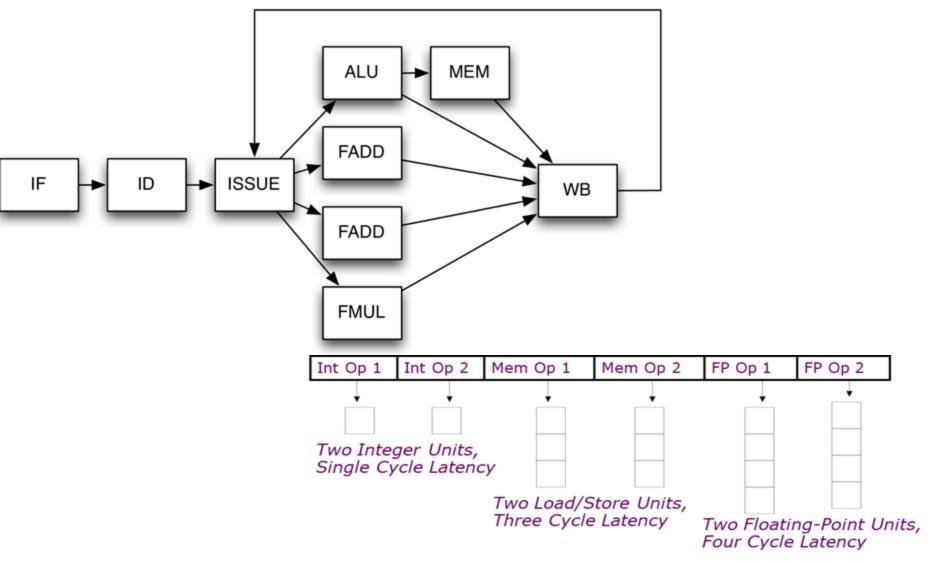




Recall: Dynamic Branch Prediction



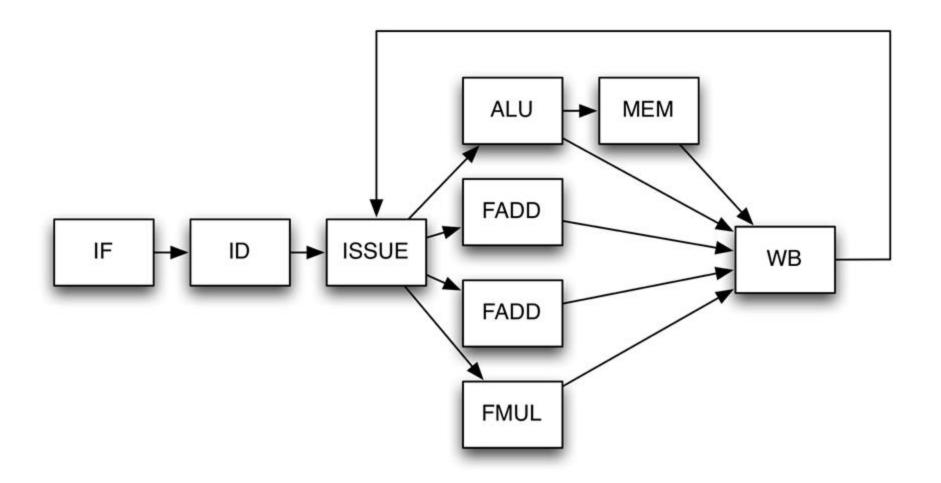
Recall: OoO and VLIW







Exe: Complex Pipeline

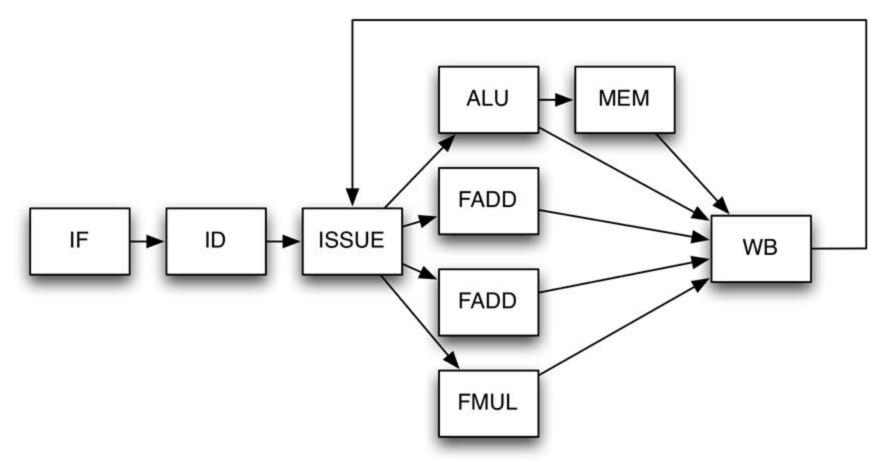






Exe: Complex Pipeline

In this problem we will examine the execution of a code segment on the following single-issue out-of-order processor:







You can assume that

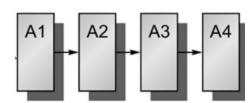
- All functional units are pipelined
- ALU operations take 1 cycle



- Floating-point add instructions take 3 cycles
- Floating-point multiply instructions take 5 cycles
- There is no register renaming. No forwarding
- Instructions are fetched, decoded and issued in order
- The ISSUE stage is a buffer of unlimited length that holds instructions waiting to start execution
- An instruction will only enter the issue stage if it does not cause a WAR or WAW hazard
- Only one instruction can be issued at a time, and in the case multiple instructions are ready, the oldest one will go first
- Program Counter calculation for branches and jumps has been anticipated in the ISSUE stage
- The target address for a branch is available in the FETCH stage

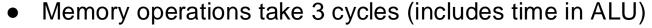






You can assume that

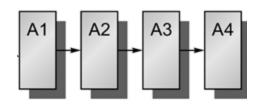
- All functional units are pipelined
- ALU operations take 1 cycle



- Floating-point add instructions take 3 cycles
- Floating-point multiply instructions take 5 cycles
- There is no register renaming. No forwarding
- Instructions are fetched, decoded and issued in order
- The ISSUE stage is a buffer of unlimited length that holds instructions waiting to start execution
- An instruction will only enter the issue stage if it does not cause a WAR or WAW hazard
- Only one instruction can be issued at a time, and in the case multiple instructions are ready, the oldest one will go first
- Program Counter calculation for branches and jumps has been anticipated in the ISSUE stage
- The target address for a branch is available in the FETCH stage







Code

Assembly Code:

I1: FOR: Id \$f2, VB(\$r6)

I2: fadd \$f3, \$f2, \$f6

I3: st \$f3, VA(\$r7)

14: Id \$f3, VC(\$r6)

15: st \$f3, VC(\$r7)

16: fadd \$f4,\$f4,\$f3

17: addi \$r6, \$r6, 4

18: addi \$r7, \$r7, 4

19: blt \$r7, \$r8, FOR





Code and Architecture

Assembly Code:

11: FOR: Id \$f2, VB(\$r6)

12: fadd \$f3, \$f2, \$f6

I3: st \$f3, VA(\$r7)

14: Id \$f3, VC(\$r6)

15: st \$f3, VC(\$r7)

16: fadd \$f4,\$f4,\$f3

17: addi \$r6, \$r6, 4

18: addi \$r7, \$r7, 4

19: blt \$r7, \$r8, FOR

ALU OP: 1 cycle

MEM OP: 3 cycles

FP ADD: 3 cycles

FP MULT: 5 cycles





ALU OP: 1 cycle

MEM OP: 3 cycles

FP ADD: 3 cycles

FP MULT: 5 cycles

Assembly Code:

I1: FOR: Id \$f2, VB(\$r6)

12: fadd \$f3, \$f2, \$f6

I3: st \$f3, VA(\$r7)

14: Id \$f3, VC(\$r6)

15: st \$f3, VC(\$r7)

16: fadd \$f4,\$f4,\$f3

17: addi \$r6, \$r6, 4

18: addi \$r7, \$r7, 4

19: blt \$r7, \$r8, FOR





ALU OP: 1 cycle

MEM OP: 3 cycles

FP ADD: 3 cycles

FP MULT: 5 cycles

Assembly Code:

11: FOR: Id \$f2 VB(\$r6)

12: fadd \$f3, \$f2 \$f6

I3: st \$f3, VA(\$r7)

14: Id(\$f3.)VC(\$r6)

15: st \$f3, VC(\$r7)

16: fadd \$f4,\$f4,\$f3

17: addi \$r6, \$r6, 4

18: add \$r7, \$r7, 4

19: blt \$r7, \$r8, FOR

RAW F2 I1-I2

RAW F3 I2-I3

RAW F3 I4-I5

RAW F3 I4-I6

RAW R7 I8-I9





ALU OP: 1 cycle

MEM OP: 3 cycles

FP ADD: 3 cycles

FP MULT: 5 cycles

Assembly Code:

11: FOR: Id \$f2 VB(\$r6)

12: fadd \$f3, \$f2 \$f6

I3: st \$f3, VA(\$r7)

14: ld(\$f3) VC(\$r6)

15: st \$f3, VC(\$r7)

16: fadd \$f4,\$f4,\$f3

17: addi \$r6, \$r6, 4

18: add \$r7, \$r7, 4

19: blt \$r7, \$r8, FOR

RAW F2 I1-I2

RAW F3 I2-I3

RAW F3 I4-I5

RAW F3 I4-I6

RAW R7 I8-I9

WAR R7 18-15

WAR R7 I8-I3





ALU OP: 1 cycle

MEM OP: 3 cycles

FP ADD: 3 cycles

FP MULT: 5 cycles

Assembly Code:

11: FOR: Id \$f2, VB(\$r6)

12: fadd \$f3, \$f2 \$f6

I3: st \$f3, VA(\$r7)

14: ld(\$f3, VC(\$r6))

15: st \$f3, VC(\$r7)

16: fadd \$f4,\$f4,\$f3

17: addi \$r6, \$r6, 4

18: add \$r7, \$r7, 4

19: blt \$r7, \$r8, FOR

RAW F2 I1-I2

RAW F3 I2-I3

RAW F3 I4-I5

RAW F3 I4-I6

RAW R7 I8-I9

WAR R7 I8-I5

WAR **R7** I8-I3

WAR R6 I7-I1

WAR R6 I7-I4





ALU OP: 1 cycle

MEM OP: 3 cycles

FP ADD: 3 cycles

FP MULT: 5 cycles

Assembly Code:

11: FOR: Id \$f2 VB(\$r6)

12: fadd \$f3, \$f2 \\$f6

I3: st \$f3, VA(\$r7)

14: ld(\$f3,)/C(\$r6)

I5: st \$f3, VC(\$r7)

16: fadd \$f4,\$f4,\$f3

17: addi \$r6, \$r6, 4

18: add \$r7, \$r7, 4

19: blt \$r7, \$r8, FOR

RAW F2 I1-I2

RAW F3 I2-I3

RAW F3 I4-I5

RAW F3 I4-I6

RAW R7 I8-I9

WAR R7 I8-I5

WAR **R7** I8-I3

WAR R6 I7-I1

WAR **R6** I7-I4

WAW F3 I2-I4





ALU OP: 1 cycle

MEM OP: 3 cycles

FP ADD: 3 cycles

FP MULT: 5 cycles

Assembly Code:

11: FOR: Id \$f2 VB(\$r6)

12: fadd \$f3, \$f2 \$f6

I3: st \$f3, VA(\$r7)

14: ld(\$t3,) VC(\$r6)

15: st \$f3, VC(\$r7)

16: fadd \$f4,\$f4,\$f3

17: addi \$r6, \$r6, 4

18: add \$r7, \$r7, 4

19: blt \$r7, \$r8, FOR

RAW F2 I1-I2

RAW F3 I2-I3

RAW F3 I4-I5

RAW F3 I4-I6

RAW R7 I8-I9

WAR R7 I8-I5

WAR R7 18-13

WAR R6 I7-I1

WAR R6 I7-I4

WAW F3 I2-I4

WAR F3 I3-I4





ALU OP: 1 cycle

MEM OP: 3 cycles

FP ADD: 3 cycles

FP MULT: 5 cycles

Assembly Code:

11: FOR: Id \$f2 VB(\$r6)

12: fadd \$f3, \$f2 \$f6

I3: st\$f3, VA(\$r7)

14: ld(\$t3,) VC(\$r6)

15: st \$f3, VC(\$r7)

16: fadd \$f4,\$f4,\$f3

17: addi \$r6, \$r6, 4

18: add \$r7, \$r7, 4

19 blt \$r7, \$r8, FOR

RAW F2 I1-I2

RAW F3 I2-I3

RAW F3 I4-I5

RAW F3 I4-I6

RAW R7 I8-I9

WAR R7 I8-I5

WAR R7 18-13

WAR R6 I7-I1

WAR R6 I7-I4

WAW F3 I2-I4

WAR **F3** I3-I4

CNTRL





ALU OP: 1 cycle MEM OP: 3 cycles FP ADD: 3 cycles FP MULT: 5 cycles

	Instruction	C1	C2	СЗ	C4	C5	C6	C 7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23	C24	C25	C26	C27	C28	Notes
1	FOR: ld \$f2,VB(\$r6)																													
2	fadd \$f3, \$f2, \$f6																													RAW I1 - I2 F2
3	st \$f3, VA(\$r7)																													RAW I2 - I3 F3
4	ld \$f3, VC(\$r6)																												1 1	WAR 13-14 F3 WAW 12-14 F3
5	st \$f3, VC(\$r7)																													RAW I4 - I5 F3
6	fadd \$f4,\$f4,\$f3																													RAW I4 - I6 F3
7	addi \$r6, \$r6, 4																												1 1	WAR I1-I7 r6 WAR I4-I7 r6,
8	addi \$r7, \$r7, 4																												1 1	WAR 13-18 r7, WAR 15-18 r7,
9	blt \$r7, \$r8, FOR																													RAW R7 18-19
10	(New Loop Iteration)																													CNTRL





ALU OP: 1 cycle MEM OP: 3 cycles FP ADD: 3 cycles FP MULT: 5 cycles

	Instruction	C1	C2	C 3	C4	C5	C6	C 7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23	C24	C25	C26	C27	C28	Notes
1	FOR: Id \$f2,VB(\$r6)	F	D	IS	E1	E2	E3	W																						
2	fadd \$f3, \$f2, \$f6																													RAW 11 - 12 F2
3	st \$f3, VA(\$r7)																													RAW 12 - 13 F3
4	ld \$f3, VC(\$r6)																													WAR 13-14 F3 WAW 12-14 F3
5	st \$f3, VC(\$r7)																													RAW I4 - I5 F3
6	fadd \$f4,\$f4,\$f3																													RAW I4 - I6 F3
7	addi \$r6, \$r6, 4																													WAR I1-I7 r6 WAR I4-I7 r6,
8	addi \$r7, \$r7, 4																													WAR 13-18 r7, WAR 15-18 r7,
9	blt \$r7, \$r8, FOR																													RAW R7 I8-I9
10	(New Loop Iteration)																													CNTRL





ALU OP: 1 cycle MEM OP: 3 cycles FP ADD: 3 cycles FP MULT: 5 cycles

	Instruction	C1	C2	СЗ	C4	C 5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23	C24	C25	C26	C27	C28	Notes
1	FOR: Id \$f2,VB(\$r6)	F	D	IS	E1	E2	E3	w																						
2	fadd \$f3, \$f2, \$f6		F	D	IS s	IS s	IS s	IS	E1	E2	E3	W																	•	RAW 11 - 12 F2
3	st \$f3, VA(\$r7)																													RAW 12 - 13 F3
4	ld \$f3, VC(\$r6)																												I I	WAR 13-14 F3 WAW 12-14 F3
5	st \$f3, VC(\$r7)																													RAW 14 - 15 F3
6	fadd \$f4,\$f4,\$f3																													RAW I4 - I6 F3
7	addi \$r6, \$r6, 4																												I I	WAR I1-I7 r6 WAR I4-I7 r6,
8	addi \$r7, \$r7, 4																													WAR 13-18 r7, WAR 15-18 r7,
9	blt \$r7, \$r8, FOR																													RAW R7 I8-I9
10	(New Loop Iteration)																													CNTRL





ALU OP: 1 cycle MEM OP: 3 cycles FP ADD: 3 cycles FP MULT: 5 cycles

	Instruction	C1	C2	СЗ	C4	C 5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23	C24	C25	C26	C27	C28	Notes
1	FOR: Id \$f2,VB(\$r6)	F	D	IS	E1	E2	E3	w																						
2	fadd \$f3, \$f2, \$f6		F	D	IS s	IS s	IS s	IS	E1	E2	E3	W																	•	RAW11 - 12 F2
3	st \$f3, VA(\$r7)			F	D	IS s	IS s	IS s	IS s	IS s	IS s	IS	E1	E2	E3	W													•	RAW 12 13 F3
4	ld \$f3, VC(\$r6)																													WAR 13-14 F3 WAW 12-14 F3
5	st \$f3, VC(\$r7)																													RAW I4 - I5 F3
6	fadd \$f4,\$f4,\$f3																													RAW I4 - I6 F3
7	addi \$r6, \$r6, 4																													WAR I1-I7 r6 WAR I4-I7 r6,
8	addi \$r7, \$r7, 4																												1 1	WAR 13-18 r7, WAR 15-18 r7,
9	blt \$r7, \$r8, FOR																													RAW R7 I8-I9
10	(New Loop Iteration)																													CNTRL





ALU OP: 1 cycle MEM OP: 3 cycles FP ADD: 3 cycles FP MULT: 5 cycles

	Instruction	C1	C2	СЗ	C4	C 5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23	C24	C25	C26	C27	C28	Notes
1	FOR: ld \$f2,VB(\$r6)	F	D	IS	E1	E2	E3	w																						
2	fadd \$f3, \$f2, \$f6		F	D	IS s	IS s	IS s	IS	E1	E2	E3	W																	•	RAW 11 - 12 F2
3	st \$f3, VA(\$r7)			F	D	IS s	IS s	IS s	IS s	IS s	IS s	IS	E1	E2	E3	W													•	RAW 12 13 F3
4	ld \$f3, VC(\$r6)				F	D s	D s	D s	D s	D s	D s	D s	D s	D s	D	IS	E1	E2	E3	W									<u> </u>	WAR I3-I4 F3 WAW I2-I4 F3
5	st \$f3, VC(\$r7)																													RAW I4 - I5 F3
6	fadd \$f4,\$f4,\$f3																													RAW I4 - I6 F3
7	addi \$r6, \$r6, 4																												1 1	WAR I1-I7 r6 WAR I4-I7 r6,
8	addi \$r7, \$r7, 4																												1 1	WAR 13-18 r7, WAR 15-18 r7,
9	blt \$r7, \$r8, FOR																													RAW R7 I8-I9
10	(New Loop Iteration)																													CNTRL





ALU OP: 1 cycle MEM OP: 3 cycles FP ADD: 3 cycles FP MULT: 5 cycles

	Instruction	C1	C2	С3	C4	C 5	C6	C7	C8	С9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23	C24	C25	C26	C27	C28	Notes
1	FOR: ld \$f2,VB(\$r6)	F	D	IS	E1	E2	E3	w																						
2	fadd \$f3, \$f2, \$f6		F	D	IS s	IS s	IS s	IS	E1	E2	E3	W																	•	RAW11 - 12 F2
3	st \$f3, VA(\$r7)			F	D	IS s	IS s	IS s	IS s	IS s	IS s	IS	E1	E2	E3	W													•	RAW 12 13 F3
4	ld \$f3, VC(\$r6)				F	o D	D s	D s	D s	D s	D s	D	ω D	s D	D	S	E1	E2	E3	8									<u> </u>	WAR 13-14 F3 WAW 12-14 F3
5	st \$f3, VC(\$r7)					F	F s	F s	F s	F s	F s	F s	F s	F s	F	D	IS s	IS s	IS s	IS	E1	E2	E3	W						RAW 14 15 F3
6	fadd \$f4,\$f4,\$f3																													RAW I4 - I6 F3
7	addi \$r6, \$r6, 4																												1 1	WAR I1-I7 r6 WAR I4-I7 r6,
8	addi \$r7, \$r7, 4																												1 1	WAR 13-18 r7, WAR 15-18 r7,
9	blt \$r7, \$r8, FOR																													RAW R7 18-19
10	(New Loop Iteration)																													CNTRL





ALU OP: 1 cycle MEM OP: 3 cycles FP ADD: 3 cycles FP MULT: 5 cycles

	Instruction	C1	C2	С3	C4	C 5	C6	С7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23	C24	C25	C26	C27	C28	Notes
1	FOR: ld \$f2,VB(\$r6)	F	D	IS	E1	E2	E3	w																						
2	fadd \$f3, \$f2, \$f6		F	D	IS s	ഗഗ	IS s	IS	E1	E2	E3	V																	•	RAW 11 - 12 F2
3	st \$f3, VA(\$r7)			F	D	IS s	IS s	IS s	IS s	IS s	IS s	IS	E1	E2	E3	W													•	RAW 12 13 F3
4	ld \$f3, VC(\$r6)				F	ω D	D s	D s	D s	D s	D s	D	ω D	D s	D	S	E1	E2	E3	8									<u> </u>	WAR I3-I4 F3 WAW I2-I4 F3
5	st \$f3, VC(\$r7)					F s	F s	F s	F s	F s	F s	F s	F s	F s	F	D	IS s	IS s	IS s	IS	E1	E2	E3	W					-	RAW 14 15 F3
6	fadd \$f4,\$f4,\$f3															F	D	IS s	IS s	IS s	IS	E1	E2	E3	W					RAW 14 16 F3
7	addi \$r6, \$r6, 4																													WAR I1-I7 r6 WAR I4-I7 r6,
8	addi \$r7, \$r7, 4																												1 1	WAR 13-18 r7, WAR 15-18 r7,
9	blt \$r7, \$r8, FOR																													RAW R7 I8-I9
10	(New Loop Iteration)																													CNTRL





ALU OP: 1 cycle MEM OP: 3 cycles FP ADD: 3 cycles FP MULT: 5 cycles

	Instruction	C1	C2	C 3	C4	C 5	C6	С7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23	C24	C25	C26	C27	C28	Notes
1	FOR: ld \$f2,VB(\$r6)	F	D	IS	E1	E2	E3	w																						
2	fadd \$f3, \$f2, \$f6		F	D	IS s	IS s	IS s	IS	E1	E2	E3	W																	•	RAW 11 - 12 F2
3	st \$f3, VA(\$r7)			F	D	IS s	IS s	IS s	IS s	IS s	IS s	IS	E1	E2	E3	W														RAW 12 13 F3
4	ld \$f3, VC(\$r6)				F	D s	D s	D s	D s	D s	D s	D s	D s	D s	D	IS	E1	E2	E3	W									<u> </u>	WAR 13-14 F3 WAW 12-14 F3
5	st \$f3, VC(\$r7)					F s	F s	F s	F s	F s	F s	F s	F s	F s	F	D	IS s	IS s	IS s	IS	E1	E2	E3	W						RAW 14 15 F3
6	fadd \$f4,\$f4,\$f3															F	D	IS s	IS s	IS s	IS	E1	E2	E3	W					RAW 14 16 F3
7	addi \$r6, \$r6, 4																F	D s	D	IS s	IS s	IS	E1	E1 s	E1 s	W				WAR 11-17-r6 WAR 14-17-r6, Struct WB
8	addi \$r7, \$r7, 4																													WAR 13-18 r7, WAR 15-18 r7,
9	blt \$r7, \$r8, FOR																													RAW R7 I8-I9
10	(New Loop Iteration)																													CNTRL





ALU OP: 1 cycle MEM OP: 3 cycles FP ADD: 3 cycles FP MULT: 5 cycles

	Instruction	C1	C2	СЗ	C4	C5	C6	С7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23	C24	C25	C26	C27	C28	Notes
1	FOR: ld \$f2,VB(\$r6)	F	D	IS	E1	E2	E3	w																						
2	fadd \$f3, \$f2, \$f6		F	D	IS s	IS s	IS s	IS	E1	E2	E3	W																		RAW 11 - 12 F2
3	st \$f3, VA(\$r7)			F	D	IS s	IS s	IS s	IS s	IS s	IS s	IS	E1	E2	E3	W														RAW 12 13 F3
4	ld \$f3, VC(\$r6)				F	s D	D s	D ø	D s	o D	D s	D s	D s	D s	D	IS	E1	E2	E3	8									<u> </u>	WAR 13-14 F3 WAW 12-14 F3
5	st \$f3, VC(\$r7)					F s	F s	F s	F s	F s	F s	F s	F s	F s	F	D	IS s	IS s	IS s	IS	E1	E2	E3	W						RAW 14 15 F3
6	fadd \$f4,\$f4,\$f3															F	D	IS s	IS s	IS s	IS	E1	E2	E3	W					RAW 14 16 F3
7	addi \$r6, \$r6, 4																F	D s	D	s S	IS s	IS	E1	E1 s	E1 s	W				WAR I1-I7 r6 WAR I4-I7 r6, Struct WB
8	addi \$r7, \$r7, 4																	F s	F	D s	D s	D s	D	IS s	IS	E1	W			WAR I3-18 r7, WAR I5-18 r7, Struct ALU
9	blt \$r7, \$r8, FOR																													RAW R7 I8-I9
10	(New Loop Iteration)																													CNTRL





ALU OP: 1 cycle MEM OP: 3 cycles FP ADD: 3 cycles FP MULT: 5 cycles

	Instruction	C1	C2	СЗ	C4	C 5	C6	С7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23	C24	C25	C26	C27	C28	Notes
1	FOR: ld \$f2,VB(\$r6)	F	D	IS	E1	E2	E3	W																						
2	fadd \$f3, \$f2, \$f6		F	D	IS s	ω SI	IS s	IS	E1	E2	E3	W																		RAW 11 - 12 F2
3	st \$f3, VA(\$r7)			F	D	IS s	IS s	IS s	IS s	IS s	IS s	IS	E1	E2	E3	W													•	RAW 12 13 F3
4	ld \$f3, VC(\$r6)				F	D s	D s	D s	D s	D s	D s	D s	D s	D s	D	IS	E1	E2	E3	W										WAR 13-14 F3 WAW 12-14 F3
5	st \$f3, VC(\$r7)					F s	F s	F s	F s	F s	F s	F s	F s	F s	F	D	IS s	IS s	IS s	IS	E1	E2	E3	W						RAW 14 15 F3
6	fadd \$f4,\$f4,\$f3															F	D	IS s	IS s	IS s	IS	E1	E2	E3	W					RAW 14 16 F3
7	addi \$r6, \$r6, 4																F	D s	D	s S	IS s	IS	E1	E1 s	E1 s	W				WAR 11-17 r6 WAR 14-17 r6, Struct WB
8	addi \$r7, \$r7, 4																	F s	F	D s	D s	D s	D	IS s	IS	E1	W			WAR 13-18 r7, WAR 15-18 r7, Struct ALU
9	blt \$r7, \$r8, FOR																			F	F s	F s	F	D	IS s	IS s	IS	E1	W	RAW R7 18-19
10	(New Loop Iteration)																													CNTRL





ALU OP: 1 cycle MEM OP: 3 cycles FP ADD: 3 cycles FP MULT: 5 cycles

	Instruction	C1	C2	СЗ	C4	C 5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23	C24	C25	C26	C27	C28	Notes
1	FOR: ld \$f2,VB(\$r6)	F	D	IS	E1	E2	E3	W																						
2	fadd \$f3, \$f2, \$f6		F	D	IS s	IS s	IS s	IS	E1	E2	E3	W																		RAW 11 - 12 F2
3	st \$f3, VA(\$r7)			F	D	IS s	IS s	IS s	IS s	IS s	IS s	IS	E1	E2	E3	W													•	RAW 12 13 F3
4	ld \$f3, VC(\$r6)				H	S D	D s	D s	D s	D s	D s	D s	ω D	D s	О	IS	E1	E2	E3	W										WAR 13-14 F3 WAW 12-14 F3
5	st \$f3, VC(\$r7)					F s	F s	F s	F s	F s	F s	F s	F s	F s	F	D	IS s	IS s	IS s	IS	E1	E2	E3	W						RAW 14 15 F3
6	fadd \$f4,\$f4,\$f3															F	D	IS s	IS s	IS s	IS	E1	E2	E3	W					RAW 14 16 F3
7	addi \$r6, \$r6, 4																F	D s	D	IS s	IS s	IS	E1	E1 s	E1 s	W				WAR 11-17 r6 WAR 14-17 r6, Struct WB
8	addi \$r7, \$r7, 4																	F s	F	D s	D s	D s	D	IS s	IS	E1	W			WAR 13-18 r7, WAR 15-18 r7, Struct ALU
9	blt \$r7, \$r8, FOR																			F s	F s	F s	F	D	IS s	IS s	IS	E1	W	RAW R7 18-19
10	(New Loop Iteration)																							F s	F s	F s	F s	F	D.	CNTRL





ALU OP: 1 cycle MEM OP: 3 cycles FP ADD: 3 cycles FP MULT: 5 cycles

	1																													
	Instruction	C1	C2	СЗ	C4	C 5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23	C24	C25	C26	C27	C28	Notes
1	FOR: ld \$f2,VB(\$r6)	F	D	IS	E1	E2	E3	W																						
2	fadd \$f3, \$f2, \$f6		F	D	IS s	IS s	IS s	IS	E1	E2	E3	W																	•	RAW 11 - 12 F2
3	st \$f3, VA(\$r7)			F	D	S s	IS s	IS s	IS s	IS s	IS s	IS	E1	E2	E3	W														RAW 12 13 F3
4	ld \$f3, VC(\$r6)				F	D s	D s	D s		g	S	D) -	DS	D	IS	0 ■1		F 3	W										WAR 13-14 F3 WAW 12-14 F3
5	st \$f3, VC(\$r7)					F s	F s	F s	S	S	F s	S	, s	E S	F	D	S	s	IS s	IS	Q	E2	E3	W						RAW 14 15 F3
6	fadd \$f4,\$f4,\$f3															F	D	IS s	IS s	IS s	IS	E1	E2	E3	W					RAW 14 16 F3
	ad i \$r6, \$r6, 4									1	2						F	(o ll o	F	IS D s	IS S	IS s		s	E1	*				WAR I1 I7 r6, WAR I4 I7 r6, WAR I4 I7 r6, WAR I4 I7 r7, WAR I4 I7 r7, Struct ALU
9	blt \$r7, \$r8, FOR																			F s	F s	F s	F	D	IS s	IS s	IS	E1	W	RAW R7 18-19
10	(New Loop Iteration)																							F s	F s	F s	F s	F	D.	CNTRL





Recall: Static Branch Prediction Techniques

Branch Always Not Taken (Predicted-Not-Taken)

Branch Always Taken (Predicted-Taken)

Backward Taken Forward Not Taken (BTFNT)

Profile-Driven Prediction

Delayed Branch





No Branch Prediction

ALU OP: 1 cycle MEM OP: 3 cycles FP ADD: 3 cycles FP MULT: 5 cycles

	Instruction	C17	C18	C19	C20	C21	C22	C23	C24	C25	C26	C27	C28	Notes
1	FOR: ld \$f2,VB(\$r6)													
2	fadd \$f3, \$f2, \$f6													RAW I1 - I2 F2
3	st \$f3, VA(\$r7)													RAW I2 - I3 F3
4	ld \$f3, VC(\$r6)	E2	E3	W										WAR 13-14 F3 WAW 12-14 F3
5	st \$f3, VC(\$r7)	IS s	IS s	IS	E1	E2	E3	W						RAW 14 - 15 F3
6	fadd \$f4,\$f4,\$f3	<u>9</u> %	IS s	IS s	S	E1	E2	E3	W					RAW 14 - 16 F3
7	addi \$r6, \$r6, 4	D s	D	IS s	IS s	IS	E1	E1 s	E1 s	8				WAR I1-I7 r6 WAR I4-I7 r6, Struct WB
8	addi \$r7, \$r7, 4	F	D s	D s	D s	D s	D	IS s	IS	E1	W		1	WAR I3-I8 r7, WAR I5- I8 r7, Struct WB
9	blt \$r7 \$r8 FOR		F	F	F	F	F	Б	IS	IS	10	E 1	W	RAW R7 I8-I9
			S	S	S	S			S	S				111 15 15
10	(Following Instruction)							F	F s	F	F s	F	D	C TRL





Recall: Static Branch Prediction Techniques

Branch Always Not Taken (Predicted-Not-Taken)

Branch Always Taken (Predicted-Taken)

Backward Taken Forward Not Taken (BTFNT)

Profile-Driven Prediction

Delayed Branch





Static Branch Prediction: NT

ALU OP: 1 cycle MEM OP: 3 cycles FP ADD: 3 cycles FP MULT: 5 cycles

	Instruction	C17	C18	C19	C20	C21	C22	C23	C24	C25	C26	C27	C28	Notes
1	FOR: ld \$f2,VB(\$r6)													
2	fadd \$f3, \$f2, \$f6													RAW I1 - I2 F2
3	st \$f3, VA(\$r7)													RAW I2 - I3 F3
4	ld \$f3, VC(\$r6)	E2	E3	V										WAR 13-14 F3 WAW 12-14 F3
5	st \$f3, VC(\$r7)	IS s	IS s	IS	E1	E2	E3	W						RAW I4 - I5 F3
6	fadd \$f4,\$f4,\$f3	ഗഗ	s S	s S	IS	E1	E2	E3	W					RAW 14 - 16 F3
7	addi \$r6, \$r6, 4	s D	D	S s	IS s	S	E1	E1 s	E1 s	V				WAR I1-I7 r6 WAR I4-I7 r6, Struct WB
8	addi \$r7, \$r7, 4	F	D s	D s	D s	D s	D	IS s	IS	E1	W			WAR I3-I8 r7, WAR I5- I8 r7, Struct WB
9	blt \$r7, \$r8, FOR		F s	F s	F s	F s	F	D	IS	IS	IS	E1	W	RAW R7 I8-I9
10	(Following Instruction)				(F	D	IS s	IS s	IS flush?	flush	C TRL





Recall: Static Branch Prediction Techniques

Branch Always Not Taken (Predicted-Not-Taken)

Branch Always Taken (Predicted-Taken)

Backward Taken Forward Not Taken (BTFNT)

Profile-Driven Prediction

Delayed Branch



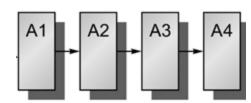


Recall: You can assume that

- All functional units are pipelined
- ALU operations take 1 cycle
- Memory operations take 3 cycles (includes time in ALU)
- Floating-point add instructions take 3 cycles
- Floating-point multiply instructions take 5 cycles
- There is no register renaming. No forwarding
- Instructions are fetched, decoded and issued in order
- The ISSUE stage is a buffer of unlimited length that holds instructions waiting to start execution
- An instruction will only enter the issue stage if it does not cause a WAR or WAW hazard
- Only one instruction can be issued at a time, and in the case multiple instructions are ready, the oldest one will go first
- Program Counter calculation for branches and jumps has been anticipated in the ISSUE stage
- The target address for a branch is available in the FETCH stage







Static Branch Prediction: T

ALU OP: 1 cycle MEM OP: 3 cycles FP ADD: 3 cycles FP MULT: 5 cycles

	Instruction	C17	C18	C19	C20	C21	C22	C23	C24	C25	C26	C27	C28	Notes
1	FOR: Id \$f2,VB(\$r6)													
2	fadd \$f3, \$f2, \$f6													RAW I1 - I2 F2
3	st \$f3, VA(\$r7)													RAW I2 - I3 F3
4	ld \$f3, VC(\$r6)	E2	E3	W										WAR 13-14 F3 WAW 12-14 F3
5	st \$f3, VC(\$r7)	IS s	IS s	IS	E1	E2	E3	W						RAW 14 - 15 F3
6	fadd \$f4,\$f4,\$f3	IS s	IS s	S s	S	E1	E2	E3	W					RAW 14 - 16 F3
7	addi \$r6, \$r6, 4	D s	D	IS s	IS s	S	E1	E1 s	E1 s	8				WAR I1-I7 r6 WAR I4-I7 r6, Struct WB
8	addi \$r7, \$r7, 4	F	D s	D s	D s	D s	D	IS s	IS	E1	W			WAR I3-I8 r7, WAR I5- I8 r7, Struct WB
9	blt \$r7, \$r8, FOR		F s	F s	F	F s	F	D	IS	IS	IS	E1	W	RAW R7 I8-I9
10	(I1 rexec)							F	D	s SI	IS s	IS flush?	flush	CNTRL, RAW I7 - I10 R6





Recall: Static Branch Prediction Techniques

Branch Always Not Taken (Predicted-Not-Taken)

Branch Always Taken (Predicted-Taken)

Backward Taken Forward Not Taken (BTFNT)

Profile-Driven Prediction

Delayed Branch





Static Branch Prediction: BTFNT

ALU OP: 1 cycle MEM OP: 3 cycles FP ADD: 3 cycles FP MULT: 5 cycles

CC 28

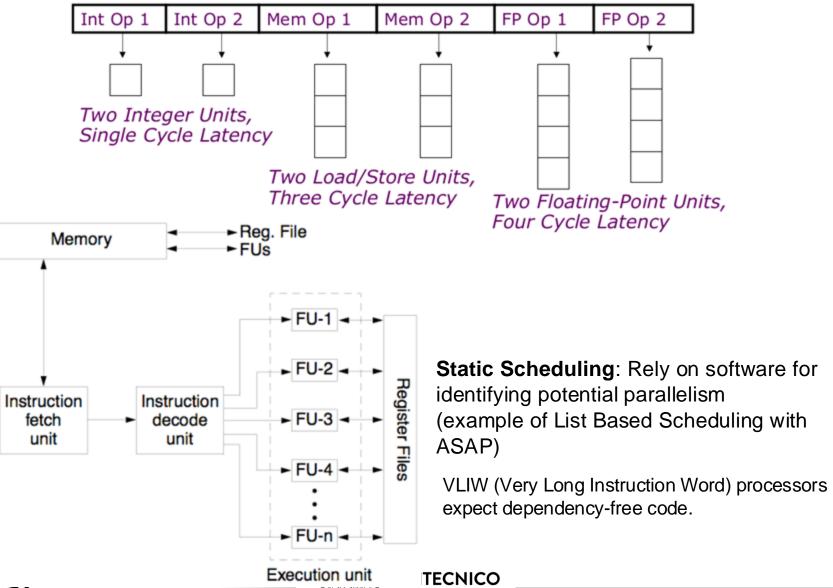
	Instruction	C17	C18	C19	C20	C21	C22	C23	C24	C25	C26	C27	C28	Notes
1	FOR: Id \$f2,VB(\$r6)													
2	fadd \$f3, \$f2, \$f6													RAW I1 - I2 F2
3	st \$f3, VA(\$r7)													RAW I2 - I3 F3
4	ld \$f3, VC(\$r6)	E2	E3	W										WAR 13-14 F3 WAW 12-14 F3
5	st \$f3, VC(\$r7)	IS s	IS s	IS	E1	E2	E3	W						RAW 14 - 15 F3
6	fadd \$f4,\$f4,\$f3	IS s	IS s	<u>S</u> 0	9	E1	E2	E3	V					RAW 14 - 16 F3
7	addi \$r6, \$r6, 4	D s	D	IS s	IS s	IS	E1	E1 s	E1 s	W				WAR I1-I7 r6 WAR I4-I7 r6, Struct WB
8	addi \$r7, \$r7, 4	F	D s	D s	D s	D s	D	IS s	IS	E1	W		l	WAR I3-I8 r7, WAR I5- I8 r7, Struct WB
9	blt \$r7, \$r8, FOR		F s	Fs	F	F s	F	D	IS	IS	IS	E1	W	RAW R7 I8-I9
10	(I1 rexec)							F	D	IS s	IS s	IS flush?	flush	CNTRL, RAW I7 - I10 R6







Recall VLIW and Static Scheduling



MILANO 1863





Exe 2 VLIW: Architecture

- Consider the program be executed on a 3-issue VLIW MIPS (Very Long Instruction Word) architecture with 3 fully pipelined functional units
- Integer ALU with 1 cycle latency to next Integer/FP and 2 cycle latency to next Branch
- Memory Unit with 3 cycle latency
- Floating Point Unit with 3 cycle latency (each FPU can complete one add or one multiply per clock cycle)
- Branch completed with 1 cycle delay slot (branch solved in ID stage)





Recall Delayed Branch

(Static Branch Prediction Techniques)

- The job of the compiler is to make the instruction placed in the branch delay slot valid and useful.
- There are three ways in which the branch delay slot can be scheduled:
 - From before
 - 2. From target
 - 3. From fall-through





- Considering one iteration of the loop
- schedule the assembly code for the 3-issue VLIW machine in the following table by using the listbased scheduling
- Do not use neither software pipelining nor loop unrolling nor modifying loop indexes.
- Please do not need to write in NOPs (can leave blank).





Exe 2 VLIW: the code

Assembly Code:

```
FOR: Id $f2, VB($r6) fadd $f3, $f2, $f6 st $f3, VA($r7) Id $f3, VC($r6) st $f3, VC($r7) fadd $f4,$f4,$f3 addi $r6, $r6, 4 addi $r7, $r7, 4 blt $r7, $r8, FOR
```





Recall: Conflicts

Assembly Code:

11: FOR: Id \$f2 VB (\$r6)

12: fadd \$f3, \$f2 \$f6

13: st \$f3 \ \ A(\$r7)

14: ld(\$t3,)VC(\$r6)

15: st \$f3, VC(\$r7)

16: fadd \$f4,\$f4,\$f3

17: addi \$r6, \$r6, 4

18: add \$r7, \$r7, 4

19 blt \$r7, \$r8, FOR

RAW F2 I1-I2

RAW F3 I2-I3

RAW F3 I4-I5

RAW F3 I4-I6

RAW R7 I8-I9

WAR R7 I8-I5

WAR R7 18-13

WAR R6 I7-I1

WAR R6 I7-I4

WAW F3 I2-I4

WAR **F3** I3-I4

CNTRL





FOR: Id \$f2, VB(\$r6)
fadd \$f3, \$f2, \$f6
st \$f3, VA(\$r7)
Id \$f3, VC(\$r6)
st \$f3, VC(\$r7)
fadd \$f4,\$f4,\$f3
addi \$r6, \$r6, 4
addi \$r7, \$r7, 4
blt \$r7, \$r8, FOR

ALU 1 cc Integer, 2 cc Branch

MU 3 cc

FPU 3 cc



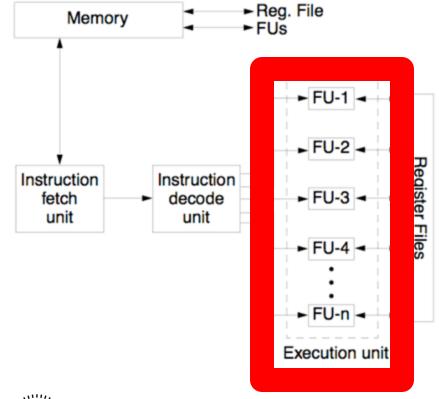


FOR: Id \$f2, VB(\$r6)
fadd \$f3, \$f2, \$f6
st \$f3, VA(\$r7)
Id \$f3, VC(\$r6)
st \$f3, VC(\$r7)
fadd \$f4,\$f4,\$f3
addi \$r6, \$r6, 4
addi \$r7, \$r7, 4
blt \$r7, \$r8, FOR

ALU 1 cc Integer, 2 cc Branch

MU 3 cc

FPU 3 cc







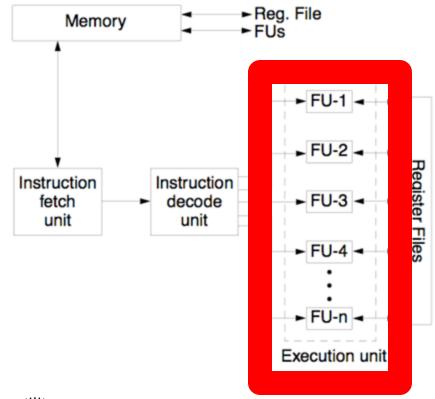
FOR: Id \$f2, VB(\$r6)
fadd \$f3, \$f2, \$f6
st \$f3, VA(\$r7)
Id \$f3, VC(\$r6)
st \$f3, VC(\$r7)
fadd \$f4,\$f4,\$f3
addi \$r6, \$r6, 4
addi \$r7, \$r7, 4
blt \$r7, \$r8, FOR

- Schedule the instructions
- Calculate the FP ops / cycle

ALU 1 cc Integer, 2 cc Branch

MU 3 cc

FPU 3 cc







FOR: Id \$f2, VB(\$r6)
fadd \$f3, \$f2, \$f6
st \$f3, VA(\$r7)
Id \$f3, VC(\$r6)
st \$f3, VC(\$r7)
fadd \$f4,\$f4,\$f3
addi \$r6, \$r6, 4
addi \$r7, \$r7, 4
blt \$r7, \$r8, FOR

		10 – 1 – 1	VVI OOIIOAAIO	
		Integer ALU(1/2 b)	Memory Unit(3cc)	FPU(3cc)
	C1			
	C2			
	C3			
	C4			
	C5			
R	C6			
	C 7			
	C8			
	C9			
	C10			
	C11			
	C12			
	C13			
	C14			
	C15			



Id \$f2, VB(\$r6)
fadd \$f3, \$f2, \$f6
st \$f3, VA(\$r7)
Id \$f3, VC(\$r6)
st \$f3, VC(\$r7)
fadd \$f4,\$f4,\$f3
addi \$r6, \$r6, 4
addi \$r7, \$r7, 4
blt \$r7, \$r8, FOF

		V L V LI	vv. oonoaare	
,		Integer ALU(1/2 b)	Memory Unit(3cc)	FPU(3cc)
)	C1		ld \$f2, VB(\$r6)	
)	C2			
)	C3			
'	C4			
	C5			
R	C6			
	C 7			
	C8			
	C 9			
	C10			
	C11			
	C12			
	C13			
	C14			
	C15			



FOR: Id \$f2, VB(\$r6)
fadd \$f3, \$f2, \$f6
st \$f3, VA(\$r7)
Id \$f3, VC(\$r6)
st \$f3, VC(\$r7)
fadd \$f4,\$f4,\$f3
addi \$r6, \$r6, 4
addi \$r7, \$r7, 4
blt \$r7, \$r8, FOR

			111 0011001	<u> </u>	
		Integer ALU(1/2 b)	Memory Unit(3cc)		FPU(3cc)
	C1		ld \$f2, VB(\$r6)	1cc	
	C2			2cc	
	C3			3cc	
	C4				
	C5				
R	C6				
	C 7				
	C8				
	C9				
	C10				
	C11				
	C12				
	C13				
	C14				
	C15				



FOR: Id \$f2, VB(\$r6) fadd \$f3, \$f2, \$f6 \$f3, VA(\$r7) st \$f3, VC(\$r6) ld \$f3, VC(\$r7) fadd \$f4,\$f4,\$f3 addi \$r6, \$r6, 4 addi \$r7, \$r7, 4

		NO Z V LI	vv. oorioa	GIO	
		Integer ALU(1/2 b)	Memory Unit(3cc)		FPU(3cc)
Id \$f2, VB(\$r6) fadd \$f3, \$f2, \$f6			ld \$f2, VB(\$r6)	1cc	
st \$f3, VA(\$r7)	C2			2cc	
Id \$f3, VC(\$r6) st \$f3, VC(\$r7)	1 1 2			3cc	
fadd \$f4,\$f4,\$f3	C4				fadd \$f3, \$f2, \$f6
addi \$r6, \$r6, 4	C 5				
addi \$r7, \$r7, 4 blt \$r7, \$r8, FO	R C6				
, , , ,	C 7				
	C8				
	С9				
	C10				
	C11				
	C12				
	C13				
NICO MILANO 1863	C14				
ECST -	C15				



FOR: Id \$f2, VB(\$r6)
fadd \$f3, \$f2, \$f6
st \$f3, VA(\$r7)
Id \$f3, VC(\$r6)
st \$f3, VC(\$r7)
fadd \$f4,\$f4,\$f3
addi \$r6, \$r6, 4
addi \$r7, \$r7, 4
blt \$r7, \$r8, FOR

		v. Scricadi	<u> </u>	_
	Integer ALU(1/2 b)	Memory Unit(3cc)	FPU(3cc)	
C1		ld \$f2, VB(\$r6)		
C2				
С3				
C4			fadd \$f3, \$f2, \$f6	1c
C5				2c
C6				3c
C7				
C8				
C9				
C10				
C11				
C12				
C13				
C14				
C15				-



5/

FOR: Id \$f2, VB(\$r6)
fadd \$f3, \$f2, \$f6
st \$f3, VA(\$r7)
Id \$f3, VC(\$r6)
st \$f3, VC(\$r7)
fadd \$f4,\$f4,\$f3
addi \$r6, \$r6, 4
addi \$r7, \$r7, 4
blt \$r7, \$r8, FOR

<u> </u>	V Z V LIV	v. Soncaai	<u> </u>	_
	Integer ALU(1/2 b)	Memory Unit(3cc)	FPU(3cc)	
C1		ld \$f2, VB(\$r6)		
C2				
С3				
C4			fadd \$f3, \$f2, \$f6	1c
C5				2c
C6				3c
C 7		st \$f3, VA(\$r7)		
C8				
С9				
C10				
C11				
C12				
C13				
C14				
C15				55



FOR: Id \$f2, VB(\$r6) fadd \$f3, \$f2, \$f6 \$f3, VA(\$r7) \$f3, VC(\$r6) \$f3, VC(\$r7) fadd \$f4,\$f4,\$f3 addi \$r6, \$r6, 4 addi \$r7, \$r7, 4

	<u> </u>	V Z V LI	vv. odriodai	<u> </u>
14		Integer ALU(1/2 b)	Memory Unit(3cc)	FPU(3cc)
Id \$f2, VB(\$r6) fadd \$f3, \$f2, \$f6	C1		ld \$f2, VB(\$r6)	
st \$f3, VA(\$r7)	C2			
Id \$f3, VC(\$r6) st \$f3, VC(\$r7)	С3			
fadd \$f4,\$f4,\$f3	C4			fadd \$f3, \$f2, \$f6
addi \$r6, \$r6, 4	C5			
addi \$r7, \$r7, 4 blt \$r7, \$r8, FOR	C6			
	C 7		st \$f3, VA(\$r7)	
	C8		ld \$f3, VC(\$r6)	
	C9			
	C10			
	C11			
	C12			
	C13			
RCO MILANO 1863	C14			
Laboratory	C15			



FOR: Id \$f2, VB(\$r6)
fadd \$f3, \$f2, \$f6
st \$f3, VA(\$r7)
Id \$f3, VC(\$r6)
st \$f3, VC(\$r7)
fadd \$f4,\$f4,\$f3
addi \$r6, \$r6, 4
addi \$r7, \$r7, 4
blt \$r7, \$r8, FOR

_			vvi oonoaan	
		Integer ALU(1/2 b)	Memory Unit(3cc)	FPU(3cc)
	C1		ld \$f2, VB(\$r6)	
	C2			
	C3			
	C4			fadd \$f3, \$f2, \$f6
	C 5			
R	C6			
	C 7		st \$f3, VA(\$r7)	
	C8		ld \$f3, VC(\$r6) 1cc	
	C9		2cc	
	C10		3cc	
	C11			
	C12			
	C13			
	C14			
	C15			



FOR: Id \$f2, VB(\$r6)
fadd \$f3, \$f2, \$f6
st \$f3, VA(\$r7)
Id \$f3, VC(\$r6)
st \$f3, VC(\$r7)
fadd \$f4,\$f4,\$f3
addi \$r6, \$r6, 4
addi \$r7, \$r7, 4
blt \$r7, \$r8, FOR

			vvi oonoaar	<u> </u>
		Integer ALU(1/2 b)	Memory Unit(3cc)	FPU(3cc)
	C1		ld \$f2, VB(\$r6)	
	C2			
	C3			
	C4			fadd \$f3, \$f2, \$f6
	C5			
R	C6			
	C 7		st \$f3, VA(\$r7)	
	C8		ld \$f3, VC(\$r6) 1cc	
	С9		2cc	
	C10		3cc	
	C11		st \$f3, VA(\$r7)	
	C12			
	C13			
	C14			
	C15			



FOR: Id \$f2, VB(\$r6) fadd \$f3, \$f2, \$f6 st \$f3, VA(\$r7) \$f3, VC(\$r6) \$f3, VC(\$r7) fadd \$f4,\$f4,\$f3 addi \$r6, \$r6, 4 addi \$r7, \$r7, 4

		C Z V L I V	v. Soncaai	<u> </u>
14		Integer ALU(1/2 b)	Memory Unit(3cc)	FPU(3cc)
Id \$f2, VB(\$r6) fadd \$f3, \$f2, \$f6	C1		ld \$f2, VB(\$r6)	
st \$f3, VA(\$r7)	C2			
Id \$f3, VC(\$r6) st \$f3, VC(\$r7)	C3			
fadd \$f4,\$f4,\$f3	C4			fadd \$f3, \$f2, \$f6
addi \$r6, \$r6, 4	C5			
addi \$r7, \$r7, 4 blt \$r7, \$r8, FOR	C6			
	C 7		st \$f3, VA(\$r7)	
	C8		ld \$f3, VC(\$r6)	
	C9			
	C10			
	C11		st \$f3, VA(\$r7)	
	C12			
	C13			
NICO MILANO 1863	C14			
Laboratory	C15			



FOR: Id \$f2, VB(\$r6) fadd \$f3, \$f2, \$f6 st \$f3, VA(\$r7) \$f3, VC(\$r6) \$f3, VC(\$r7) fadd \$f4,\$f4,\$f3 addi \$r6, \$r6, 4 addi \$r7, \$r7, 4

		CO Z V LIV	v. odriodai	<u> </u>
14		Integer ALU(1/2 b)	Memory Unit(3cc)	FPU(3cc)
Id \$f2, VB(\$r6) fadd \$f3, \$f2, \$f6	C1		ld \$f2, VB(\$r6)	
st \$f3, VA(\$r7)	C2			
Id \$f3, VC(\$r6) st \$f3, VC(\$r7)	C3			
fadd \$f4,\$f4,\$f3	C4			fadd \$f3, \$f2, \$f6
addi \$r6, \$r6, 4	C5			
addi \$r7, \$r7, 4 blt \$r7, \$r8, FOR	C6			
. , . ,	C 7		st \$f3, VA(\$r7)	
	C8		ld \$f3, VC(\$r6)	
	C9			
	C10			
	C11		st \$f3, VA(\$r7)	fadd \$f4, \$f4, \$f3
	C12			
	C13			
IICO MILANO 1863	C14			
Laboratory	C15			



FOR: Id \$f2, VB(\$r6) fadd \$f3, \$f2, \$f6 \$f3, VA(\$r7) \$f3, VC(\$r6) \$f3, VC(\$r7) fadd \$f4,\$f4,\$f3 addi \$r6, \$r6, 4 addi \$r7, \$r7, 4

blt

	10 – 1 – 1	111 001104410	
	Integer ALU(1/2 b)	Memory Unit(3cc)	FPU(3cc)
C1		ld \$f2, VB(\$r6)	
C2			
C3			
C4			fadd \$f3, \$f2, \$f6
C5			
C6			
C 7		st \$f3, VA(\$r7)	
C8		ld \$f3, VC(\$r6)	
C9			
C10			
C11		st \$f3, VA(\$r7)	fadd \$f4, \$f4 , \$f3
C12			
C13			
C14			
C15			
	C2 C3 C4 C5 C6 C7 C8 C9 C10 C11 C12 C13 C14	C1 C2 C3 C4 C5 C6 C7 C8 C9 C10 C11 C12 C13 C14	Integer ALU(1/2 b) Memory Unit(3cc) C1



FOR: Id \$f2, VB(\$r6) fadd \$f3, \$f2, \$f6 st \$f3, VA(\$r7) \$f3, VC(\$r6) \$f3, VC(\$r7) fadd \$f4,\$f4,\$f3 addi \$r6, \$r6, 4 addi \$r7, \$r7, 4

blt

_		TO E V EI	vv. oonoaare	
¢f2 \/D/¢r6\		Integer ALU(1/2 b)	Memory Unit(3cc)	FPU(3cc)
\$f2, VB(\$r6) d \$f3, \$f2, \$f6	C1		ld \$f2, VB(\$r6)	
\$f3, VA(\$r7)	C2			
\$f3, VC(\$r6) \$f3, VC(\$r7)	C3			
d \$f4,\$f4,\$f3	C4			fadd \$f3, \$f2, \$f6
di \$r6, \$r6, 4	C5			
di \$r7, \$r7, 4 \$r7, \$r8, FOR	C6			
	C 7		st \$f3, VA(\$r7)	
	C8	addi \$r6, \$r6, 4	ld \$f3, VC(\$r6)	
	C 9			
	C10			
	C11		st \$f3, VA(\$r7)	fadd \$f4, \$f4, \$f3
	C12			
	C13			
O 1863	C14			
o 1863 atory	C15			



FOR: Id \$f2, VB(\$r6) fadd \$f3, \$f2, \$f6 st \$f3, VA(\$r7) \$f3, VC(\$r6) \$f3, VC(\$r7) fadd \$f4,\$f4,\$f3 addi \$r6, \$r6, 4 addi \$r7, \$r7, 4

blt

		V L	vv. oonoaare	
¢f2 \/D(¢r6\		Integer ALU(1/2 b)	Memory Unit(3cc)	FPU(3cc)
\$f2, VB(\$r6) 3 \$f3, \$f2, \$f6	C1		ld \$f2, VB(\$r6)	
\$f3, VA(\$r7)	C2			
\$f3, VC(\$r6) \$f3, VC(\$r7)	C3			
\$f4,\$f4,\$f3	C4			fadd \$f3, \$f2, \$f6
i \$r6, \$r6, 4	C5			
i \$r7, \$r7, 4 \$r7, \$r8, FOR	C6			
	C 7		st \$f3, VA(\$r7)	
	C8	addi \$r6, \$r6, 4	ld \$f3, VC(\$r6)	
	C9			
	C10			
	C11	addi \$r7, \$r7, 4	st \$f3, VA(\$r7)	fadd \$f4, \$f4, \$f3
	C12			
	C13			
1863	C14			
1863 Ory	C15			



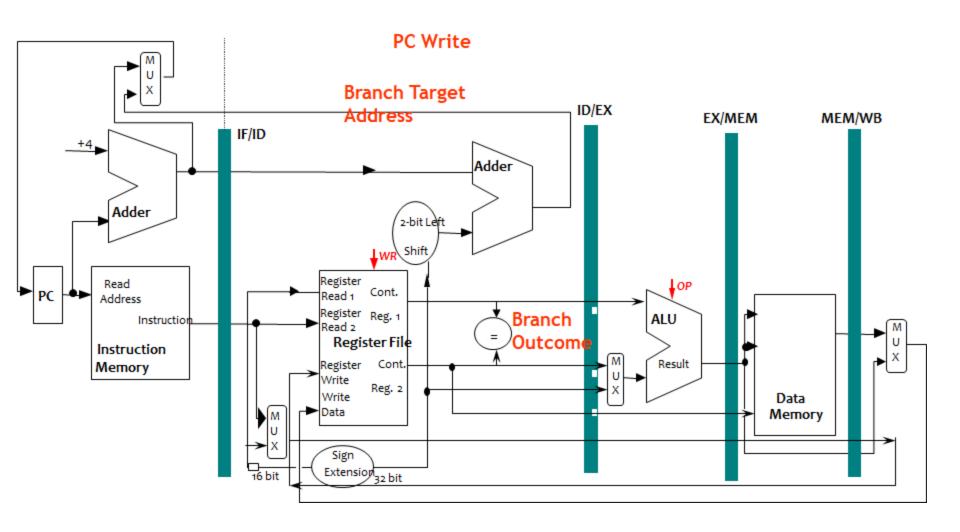
FOR: Id \$f2, VB(\$r6)
fadd \$f3, \$f2, \$f6
st \$f3, VA(\$r7)
Id \$f3, VC(\$r6)
st \$f3, VC(\$r7)
fadd \$f4,\$f4,\$f3
addi \$r6, \$r6, 4
addi \$r7, \$r7, 4
blt \$r7, \$r8, FOR

	Integer ALU(1/2 b)	Memory Unit(3cc)	FPU(3cc)		
C1		ld \$f2, VB(\$r6)			
C2					
C3					
C4			fadd \$f3, \$f2, \$f6		
C5					
C6					
C7		st \$f3, VA(\$r7)			
C8	addi \$r6, \$r6, 4	ld \$f3, VC(\$r6)			
С9					
C10					
C11	addi \$r7, \$r7, 4	st \$f3, VA(\$r7)	fadd \$f4, \$f4, \$f3		
C12					
C13					
C14					
C15					



64

Recall: Early-evaluation PC







FOR: Id \$f2, VB(\$r6)
fadd \$f3, \$f2, \$f6
st \$f3, VA(\$r7)
Id \$f3, VC(\$r6)
st \$f3, VC(\$r7)
fadd \$f4,\$f4,\$f3
addi \$r6, \$r6, 4
addi \$r7, \$r7, 4
blt \$r7, \$r8, FOR

		NO E V EI	vv. oonoaare	
		Integer ALU(1/2 b)	Memory Unit(3cc)	FPU(3cc)
	C1		ld \$f2, VB(\$r6)	
	C2			
	C3			
	C4			fadd \$f3, \$f2, \$f6
	C 5			
R	C6			
	C 7		st \$f3, VA(\$r7)	
	C8	addi \$r6, \$r6, 4	ld \$f3, VC(\$r6)	
	C9			
	C10			
	C11	addi \$r7, \$r7, 4	st \$f3, VA(\$r7)	fadd \$f4, \$f4 , \$f3
	C12			
	C13	blt \$r7, \$r8, FOR		
	C14			
	C15			



66

FOR: Id \$f2, VB(\$r6)
fadd \$f3, \$f2, \$f6
st \$f3, VA(\$r7)
Id \$f3, VC(\$r6)
st \$f3, VC(\$r7)
fadd \$f4,\$f4,\$f3
addi \$r6, \$r6, 4
addi \$r7, \$r7, 4
blt \$r7, \$r8, FOR

	\ <u>\</u>	111 001104410	
	Integer ALU(1/2 b)	Memory Unit(3cc)	FPU(3cc)
C1		ld \$f2, VB(\$r6)	
C2			
C3			
C4			fadd \$f3, \$f2, \$f6
C5			
C6			
C 7		st \$f3, VA(\$r7)	
C8	addi \$r6, \$r6, 4	ld \$f3, VC(\$r6)	
C9			
C10			
C11	addi \$r7, \$r7, 4	st \$f3, VA(\$r7)	fadd \$f4, \$f4 , \$f3
C12			
C13	blt \$r7, \$r8, FOR		
C14	(br delay slot)		
C15			
	C2 C3 C4 C5 C6 C7 C8 C9 C10 C11 C12 C13 C14	C1 C2 C3 C4 C5 C6 C7 C8 addi \$r6, \$r6, 4 C9 C10 C11 addi \$r7, \$r7, 4 C12 C13 blt \$r7, \$r8, FOR C14 (br delay slot)	C1



67

Exe 2 VLIW: The Resulting Code

FOR: Id \$f2, VB(\$r6)
fadd \$f3, \$f2, \$f6
st \$f3, VA(\$r7)
Id \$f3, VC(\$r6)
st \$f3, VC(\$r7)
fadd \$f4,\$f4,\$f3
addi \$r6, \$r6, 4
addi \$r7, \$r7, 4

blt \$r7, \$r8, FOR

	Integer ALU(1/2 b)	Memory Unit(3cc)	FPU(3cc)
I1	NOP	ld \$f2, VB(\$r6)	NOP
I2	NOP	NOP	NOP
13	NOP	NOP	NOP
14	NOP	NOP	fadd \$f3, \$f2, \$f6
15	NOP	NOP	NOP
I 6	NOP	NOP	NOP
17	NOP	st \$f3, VA(\$r7)	NOP
18	addi \$r6, \$r6, 4	ld \$f3, VC(\$r6)	NOP
19	NOP	NOP	NOP
l10	NOP	NOP	NOP
I11	addi \$r7, \$r7, 4	st \$f3, VA(\$r7)	fadd \$f4, \$f4 , \$f3
l12	NOP	NOP	NOP
l13	blt \$r7, \$r8, FOR	NOP	NOP
l14	(br delay slot)	NOP	NOP
	- / (12/0) > -		-





FOR: Id \$f2, VB(\$r6)
fadd \$f3, \$f2, \$f6
st \$f3, VA(\$r7)
Id \$f3, VC(\$r6)
st \$f3, VC(\$r7)
fadd \$f4,\$f4,\$f3
addi \$r6, \$r6, 4
addi \$r7, \$r7, 4
blt \$r7, \$r8, FOR

			· · · · · · · · · · · · · · · · · · ·	
		Integer ALU(1/2 b)	Memory Unit(3cc)	FPU(3cc)
	C1		ld \$f2, VB(\$r6)	
	C2			
	C3			
	C4			fadd \$f3, \$f2, \$f6
	C5			
R	C6			
	C 7		st \$f3, VA(\$r7)	
	C8	addi \$r6, \$r6, 4	ld \$f3, VC(\$r6)	
	C9			
	C10			
	C11	addi \$r7, \$r7, 4	st \$f3, VA(\$r7)	fadd \$f4, \$f4, \$f3
	C12			
	C13	blt \$r7, \$r8, FOR		
	C14	(br delay slot)		
	C15			



60

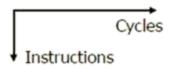
EOD: Id		Integer ALU(1/2 b)	Memory Unit(3cc)	FPU(3cc)
FOR: Id	C1		ld \$f2, VB(\$r6)	
st \$f3, VA(\$r7)	C2			
ld \$f3, VC(\$r6) st \$f3, VC(\$r7)	C3			
fadd \$f4,\$f4,\$f3	C4			fadd \$f3, \$f2, \$f6
addi \$r6, \$r6, 4 addi \$r7, \$r7, 4	C5			
blt \$r7, \$r8, FOR	C6			
	C 7		st \$f3, VA(\$r7)	
	C8	addi \$r6, \$r6, 4	ld \$f3, VC(\$r6)	
FPops/cycle=	С9			
=2 fadds / 14 cycles =	C10			
= 0.143	C11	addi \$r7, \$r7, 4	st \$f3, VA(\$r7)	fadd \$f4, \$f4 , \$f3
- 6.1 16	C12			
	C13	blt \$r7, \$r8, FOR		
POLITECNICO MILANO 1863	C14	(br delay slot)		
NECST	C15			

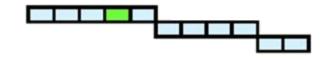
70



Recall: The ILP Architecture Journey

Steps towards exploiting more ILP





Sequential (non pipelined) ____ IDEAL CPI > 1

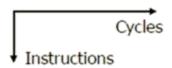


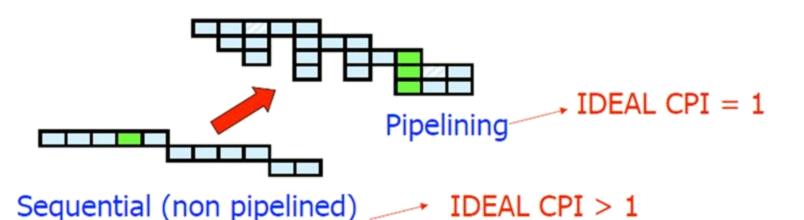




Recall: The ILP Architecture Journey

Steps towards exploiting more ILP



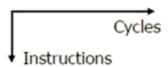


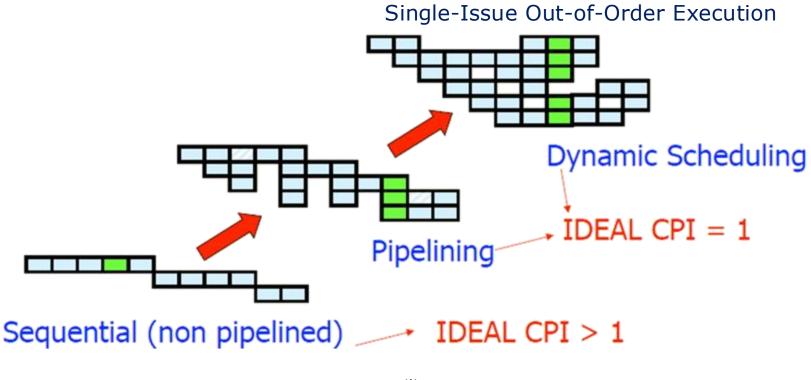




Recall: The ILP Architecture Journey

Steps towards exploiting more ILP









Problem:

data dependences that cannot be hidden with bypassing or forwarding cause hardware stalls of the pipeline





Problem:

data dependences that cannot be hidden with bypassing or forwarding cause hardware stalls of the pipeline

Solution: allow instructions behind a stall to proceed

HW rearranges the instruction execution to reduce stalls





Problem:

data dependences that cannot be hidden with bypassing or forwarding cause hardware stalls of the pipeline

Solution: allow instructions behind a stall to proceed

- HW rearranges the instruction execution to reduce stalls
 Enables out-of-order execution and completion (commit)
- Out-of order execution introduces possibility of WAR, WAW data hazards.





Problem:

data dependences that cannot be hidden with bypassing or forwarding cause hardware stalls of the pipeline

Solution: allow instructions behind a stall to proceed

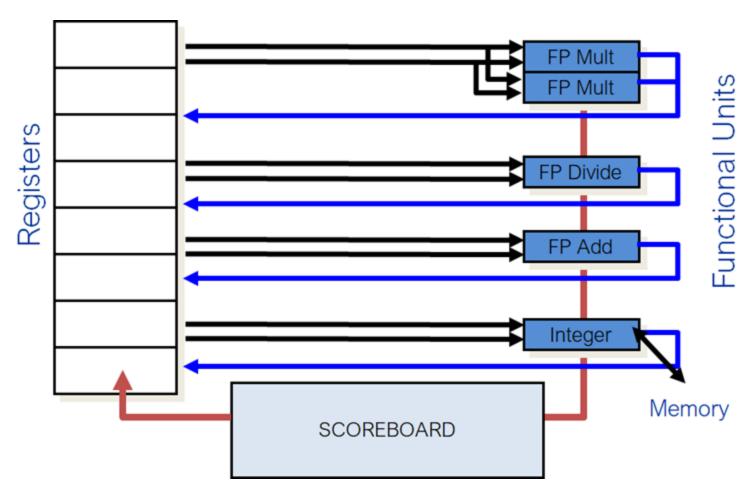
- HW rearranges the instruction execution to reduce stalls
 Enables out-of-order execution and completion (commit)
- Out-of order execution introduces possibility of WAR, WAW data hazards.

First implemented in CDC6600 (1963)





Exe 1 Scoreboard



Parallel operation in the control data 6600





Recall: the Scoreboard pipeline

ISSUE	READ OPERAND	EXE COMPLETE	WB
Decode instruction;	Read operands;	Operate on operands;	Finish exec;
Structural FUs check; WAW checks	RAW check; WAR if need to read	Notify Scoreboard on completion;	WAR &Struct check (FUs will hold results); Can overlap issue/read&write 4 Structural Hazard;





Exe 1 Scoreboard: the Code

I1: LD F6 32+ R2

I2: ADDD F2 F6 F4

I3: MULTD F0 F4 F2

I4: SUBD F12 F2 F6





I1: LD F6 32+ R2

I2: ADDD F2 F6 F4

I3: MULTD F0 F4 F2

I4: SUBD F12 F2 F6





RAW F6 I1-I2

I1: LD F6 32+ R2

12: ADDD F2 F6 F4

I3: MULTD F0 F4 F2

I4: SUBD F12 F2 F6





RAW F6 I1-I2

RAW F6 I1-I4

I1: LD(F6)32+ R2

I2: ADDD F2 F6 F4

I3: MULTD F0 F4 F2

I4: SUBD F12 F2 F6





I1: LD(F6)32+ R2

12: ADDD F2 F6 F4

I3: MULTD F0 F4 F2

I4: SUBD F12 F2(F6)

I5: ADDD F0 F12 F2

RAW F6 I1-I2

RAW F6 I1-I4

RAW F2 I2-I3





I1: LD(F6)32+ R2

I2: ADDD E20F6 F4

I3: MULTD F0 F4 F2

I4: SUBD F12(F2)F6

I5: ADDD F0 F12 F2

RAW F6 I1-I2

RAW F6 I1-I4

RAW F2 I2-I3

RAW F2 I2-I4





I1: LD(F6)32+ R2

I2: ADDD **F2**F6 F4

I3: MULTD F0 F4 F2

I4: SUBD F12(F2)F6

I5: ADDD F0 F12 F2

RAW F6 I1-I2

RAW F6 I1-I4

RAW F2 I2-I3

RAW F2 I2-I4

RAW F2 I2-I5





I1: LD(F6)32+ R2

I2: ADDD **F2** F6 F4

I3: MULTD F0 F4 F2

I4: SUBD (F12) F2 (F6)

I5: ADDD F0 F12 F2

RAW F6 I1-I2

RAW F6 I1-I4

RAW F2 I2-I3

RAW F2 I2-I4

RAW F2 I2-I5

RAW F12 I4-I5





I1: LD(F6)32+ R2

I2: ADDD **E2** F6 F4

13: MULTO F0 F4 F2

I4: SUBD (F12) F2 (F6)

15: ADDD F0 F120 F2

RAW F6 I1-I2

RAW F6 I1-I4

RAW F2 I2-I3

RAW F2 I2-I4

RAW F2 I2-I5

RAW F12 I4-I5





	Issue	Read Op	Exec Co.	Write R.
I1: LD F6 32+ R2	1	2	7	8
12: ADDD F2 F6 F4	2	9	11	12
13: MULTD F0 F4 F2	4	13	43	44
I4: SUBD F12 F2 F6	3	9	11	12
I5: ADDD F0 F12 F2	13	17	19	20

- Is there a "configuration" that can respect the shown execution?
- How many units? Which kind? What latency?





	Issue	Read Op	FALC C	o. Write R.
I1: LD F6 32+ R2	1	2	7	8
I2: ADDD F2 F6 F4	2	9	11	12
I3: <u>MULTD</u> F0 F4 F2	4	13	43	44
I4: SUBD F12 F2 F6	3	9	11	12
I5: ADDD F0 F12 F2	13	17	19	20
			V	

- Is there a "configuration" that can respect the shown execution?
- How many units? Which kind? What latency?





	Issue	Read Op	FAC C	o. Vri 🤋	R.
I1: LD F6 32+ R2	1	2	7	8	
I2: ADDD F2 F6 F4	2	9	11	12	
I3: MULTD F0 F4 F2	4	13	43	44	
I4: SUBD F12 F2 F6	3	9	11	12	
I5: ADDD F0 F12 F2	13	17	19	20	
			V	V	

- Is there a "configuration" that can respect the shown execution?
- How many units? Which kind? What latency?





	Issue	Read Op	FAC CO.	Vri\ ₹ R.
I1: LD F6 32+ R2	1	2	7	8
I2: ADDD F2 F6 F4	2	9	11	12
I3: MULTD F0 F4 F2	4	13	43	44
I4: SUBD F12 F2 F6	3	9	11	12
I5: ADDD F0 F12 F2	13	17	19	20
				V

- Is there a "configuration" that can respect the shown execution?
- How many units? Which kind? What latency?





	Issue	Read Op	FAC Co.	Vri ₹ R.
I1: LD F6 32+ R2	1	2	7	8
I2: ADDD F2 F6 F4	2	9	11	12
I3: MULTD F0 F4 F2	4	13	43	44
I4: SUBD F12 F2 F6	3	9	11	12
I5: ADDD F0 F12 F2	13	17	19	20
				V

- Is there a "configuration" that can respect the shown execution?
- How many units? Which kind? What latency?





	iss le		Read Op	FACC	Co.	Vri\ ?	R.
I1: LD F6 32+ R2	1		2	7		8	
I2: ADDD F2 F6 F4	2		9	11		12	
I3: MULTD F0 F4 F2	4		13	43		44	
I4: SUBD F12 F2 F6	3		9	11		12	
I5: ADDD F0 F12 F2	13		17	19		20	
	V					V	

- Is there a "configuration" that can respect the shown execution?
- How many units? Which kind? What latency?





	ISS VE	•	Read Op	FAL C	Co.	//ri\ ?	R.
I1: LD F6 32+ R2	1		2	7		8	
I2: ADDD F2 F6 F4	2		9	11		12	
I3: MULTD F0 F4 F2	4		13	43		44	
I4: SUBD F12 F2 F6	3		9	11		12	
I5: <u>ADDD</u> F0 F12 F2	13		17	19		20	
	V					V	1

- Is there a "configuration" that can respect the shown execution?
- How many units? Which kind? What latency?





	Instruction	ISSUE	READ OPERAND	EXE COMPLETE	WB	Hazards	Unit
11	LD F6 32+ R2						
12	ADDD F2 F6 F4						
13	MULTD F0 F4 F2						
14	SUBD F12 F2 F6						
15	ADDD F0 F12 F2						

If the previous table was not correct, please, write the right one and specify the number, kind and latency for each unit.





	Instruction	ISSUE	READ OPERAND	EXE COMPLETE	WB	Hazards	Unit
11	LD F6 32+ R2						
12	ADDD F2 F6 F4						
13	MULTD F0 F4 F2						
14	SUBD F12 F2 F6						
15	ADDD F0 F12 F2						

If the previous table was not correct, please, write the right one and specify the number,

kind and latency for each unit.

4 FPALU 3 cc latency, single write port for the pool

1 MEM 2 cc latency



RAW F6 I1-I2

WAW **FO** I3-I5





	Instruction	ISSUE	READ OPERAND	EXE COMPLETE	WB	Hazards	Unit
11	LD F6 32+ R2	1					MU
12	ADDD F2 F6 F4						
13	MULTD F0 F4 F2						
14	SUBD F12 F2 F6						
15	ADDD F0 F12 F2						

If the previous table was not correct, please, write the right one and specify the number,

kind and latency for each unit.

4 FPALU 3 cc latency, single write port for the pool

1 MEM 2 cc latency



RAW **F6** I1-I2 RAW **F6** I1-I4



	Instruction	ISSUE	READ OPERAND	EXE COMPLETE	WB	Hazards	Unit
I1	LD F6 32+ R2	1	2				MU
12	ADDD F2 F6 F4	2					FPU1
13	MULTD F0 F4 F2						
14	SUBD F12 F2 F6						
15	ADDD F0 F12 F2						

If the previous table was not correct, please, write the right one and specify the number,

kind and latency for each unit.

4 FPALU 3 cc latency, single write port for the pool

1 MEM 2 cc latency



RAW F6 I1-I2 RAW F6 I1-I4





	Instruction	ISSUE	READ OPERAND	EXE COMPLETE	WB	Hazards	Unit
11	LD F6 32+ R2	1	2				MU
12	ADDD F2 F6 F4	2				RAW F6	FPU1
13	MULTD F0 F4 F2	3					FPU2
14	SUBD F12 F2 F6						
15	ADDD F0 F12 F2						

If the previous table was not correct, please, write the right one and specify the number, **RAW F6 I1-I2**

kind and latency for each unit.

4 FPALU 3 cc latency, single write port for the pool

1 MEM 2 cc latency



RAW F2 I2-I3

RAW F2 I2-I4 RAW F2 I2-I5

RAW F12 I4-I5



	Instruction	ISSUE	READ OPERAND	EXE COMPLETE	WB	Hazards	Unit
11	LD F6 32+ R2	1	2	4			MU
12	ADDD F2 F6 F4	2				RAW F6	FPU1
13	MULTD F0 F4 F2	3				RAW F2	FPU2
14	SUBD F12 F2 F6	4					FPU3
15	ADDD F0 F12 F2						

If the previous table was not correct, please, write the right one and specify the number, kind and latency for each unit.

RAW F6 I1-I2

4 FPALU 3 cc latency, <u>single write</u> port for the pool 1 MEM 2 cc latency





PAW F2 I2-I3 RAW F2 I2-I4 RAW F2 I2-I5

RAW F12 I4-I5

	Instruction	ISSUE	READ OPERAND	EXE COMPLETE	WB	Hazards	Unit
11	LD F6 32+ R2	1	2	4	5		MU
12	ADDD F2 F6 F4	2				RAW F6	FPU1
13	MULTD F0 F4 F2	3				RAW F2	FPU2
14	SUBD F12 F2 F6	4					FPU3
15	ADDD F0 F12 F2					WAW F0	

If the previous table was not correct, please, write the right one and specify the number,

kind and latency for each unit.

4 FPALU 3 cc latency, <u>single write</u> port for the pool

1 MEM 2 cc latency





RAW F6 I1-I4

RAW F2 I2-I3

RAW F2 I2-I4

RAW F2 I2-I5

D A 147 F49 14 15



	Instruction	ISSUE	READ OPERAND	EXE COMPLETE	WB	Hazards	Unit
11	LD F6 32+ R2	1	2	4	5		MU
12	ADDD F2 F6 F4	2				RAW F6	FPU1
13	MULTD F0 F4 F2	3				RAW F2	FPU2
14	SUBD F12 F2 F6	4				RAW F2	FPU3
15	ADDD F0 F12 F2					WAW F0	

If the previous table was not correct, please, write the right one and specify the number, kind and latency for each unit.

RAW F6 I1-I2

4 FPALU 3 cc latency, <u>single write</u> port for the pool 1 MEM 2 cc latency





RAW F2 I2-I3 RAW F2 I2-I4 RAW F2 I2-I5 RAW F12 I4-I5 WAW F0 I3-I5

RAW F6 I1-I4

	Instruction	ISSUE	READ OPERAND	EXE COMPLETE	WB	Hazards	Unit
11	LD F6 32+ R2	1	2	4	5		MU
12	ADDD F2 F6 F4	2	6			RAW F6	FPU1
13	MULTD F0 F4 F2	3				RAW F2	FPU2
14	SUBD F12 F2 F6	4				RAW F2	FPU3
15	ADDD F0 F12 F2					WAW F0	

4 FPALU 3 cc latency, single write port for the pool

1 MEM 2 cc latency



RAW F6 I1-I4
RAW F2 I2-I3
RAW F2 I2-I4
RAW F2 I2-I5

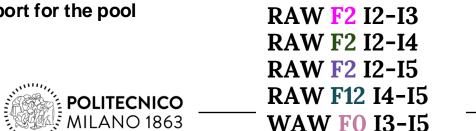
RAW F12 I4-I5 WAW F0 I3-I5



	Instruction	ISSUE	READ OPERAND	EXE COMPLETE	WB	Hazards	Unit
11	LD F6 32+ R2	1	2	4	5		MU
12	ADDD F2 F6 F4	2	6	9		RAW F6	FPU1
13	MULTD F0 F4 F2	3				RAW F2	FPU2
14	SUBD F12 F2 F6	4				RAW F2	FPU3
15	ADDD F0 F12 F2					WAW F0	

If the previous table was not correct, please, write the right one and specify the number, kind and latency for each unit.

4 FPALU 3 cc latency, single write port for the pool







	Instruction	ISSUE	READ OPERAND	EXE COMPLETE	WB	Hazards	Unit
11	LD F6 32+ R2	1	2	4	5		MU
12	ADDD F2 F6 F4	2	6	9	10	RAW F6	FPU1
13	MULTD F0 F4 F2	3				RAW F2	FPU2
14	SUBD F12 F2 F6	4				RAW F2	FPU3
15	ADDD F0 F12 F2					WAW F0	

4 EDALLI 2 on latency, single write part for the pack

4 FPALU 3 cc latency, <u>single write</u> port for the pool 1 MEM 2 cc latency

RAW F2 I2-I3

RAW F2 I2-I4

RAW F2 I2-I5





	Instruction	ISSUE	READ OPERAND	EXE COMPLETE	WB	Hazards	Unit
11	LD F6 32+ R2	1	2	4	5		MU
12	ADDD F2 F6 F4	2	6	9	10	RAW F6	FPU1
13	MULTD F0 F4 F2	3	11			RAW F2	FPU2
14	SUBD F12 F2 F6	4	11			RAW F2	FPU3
15	ADDD F0 F12 F2					WAW F0	

If the previous table was not correct, please, write the right one and specify the number,

kind and latency for each unit.

PAY

4 FPALU 3 cc latency, single write port for the pool

1 MEM 2 cc latency

RAW F6 I1-I4

RAW F2 I2-I3

RAW F2 I2-I4

DAW F2 I2-I5

RAW F12 I4-I5

WAW F0 I3-I5





	Instruction	ISSUE	READ OPERAND	EXE COMPLETE	WB	Hazards	Unit
11	LD F6 32+ R2	1	2	4	5		MU
12	ADDD F2 F6 F4	2	6	9	10	RAW F6	FPU1
13	MULTD F0 F4 F2	3	11	14		RAW F2	FPU2
14	SUBD F12 F2 F6	4	11	14		RAW F2	FPU3
15	ADDD F0 F12 F2					WAW F0	

If the previous table was not correct, please, write the right one and specify the number,

kind and latency for each unit.

4 FPALU 3 cc latency, single write port for the pool

1 MEM 2 cc latency



RAW F2 I2-I5

RAW F12 I4-I5

WAW F0 I3-I5





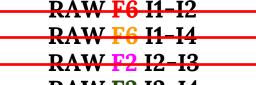
	Instruction	ISSUE	READ OPERAND	EXE COMPLETE	WB	Hazards	Unit
I1	LD F6 32+ R2	1	2	4	5		MU
12	ADDD F2 F6 F4	2	6	9	10	RAW F6	FPU1
13 14 15	MALFO F0 F4 F2 SJND F12 F2 F6 ADDD F0 F12 F2	/A / A / A	11	14	RT	RAV F2	FPU3

If the previous table was not correct, please, write the right one and specify the number,

kind and latency for each unit.

4 FPALU 3 cc latency, single write port for the pool

1 MEM 2 cc latency



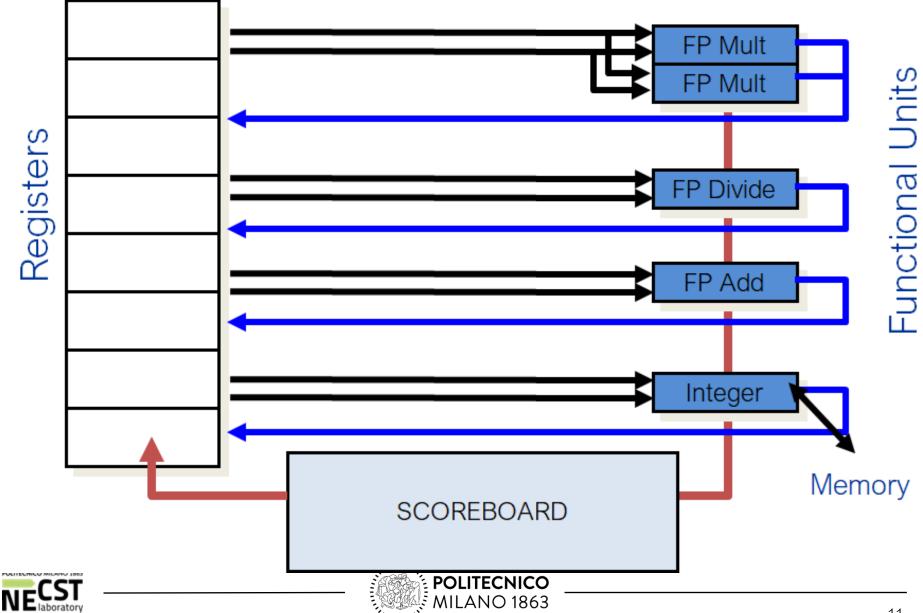
DAW E2 12-14

RAW F12 I4-I5

WAW F0 I3-I5







	Instruction	ISSUE	READ OPERAND	EXE COMPLETE	WB	Hazards	Unit
11	LD F6 32+ R2	1	2	4	5		MU
12	ADDD F2 F6 F4	2	6	9	10	RAW F6	FPU1
13	MULTD F0 F4 F2	3	11	14	15	RAW F2	FPU2
14	SUBD F12 F2 F6	4	11	14		RAW F2 + Struct RF	FPU3
15	ADDD F0 F12 F2					WAW F0	

If the previous table was not correct, please, write the right one and specify the number,

kind and latency for each unit.

4 FPALU 3 cc latency, single write port for the pool

1 MEM 2 cc latency

RAW F6 I1-I4

RAW F2 I2-I3

RAW F2 I2-I4

RAW F2 I2-I5

RAW F12 I4-I5 WAW F0 I3-I5





	Instruction	ISSUE	READ OPERAND	EXE COMPLETE	WB	Hazards	Unit
11	LD F6 32+ R2	1	2	4	5		MU
12	ADDD F2 F6 F4	2	6	9	10	RAW F6	FPU1
13	MULTD F0 F4 F2	3	11	14	15	RAW F2	FPU2
14	SUBD F12 F2 F6	4	11	14	16	RAW F2 + Struct RF	FPU3
15	ADDD F0 F12 F2	16				WAW F0	FPU4

If the previous table was not correct, please, write the right one and specify the number,

kind and latency for each unit.

4 FPALU 3 cc latency, single write port for the pool







	Instruction	ISSUE	READ OPERAND	EXE COMPLETE	WB	Hazards	Unit
11	LD F6 32+ R2	1	2	4	5		MU
12	ADDD F2 F6 F4	2	6	9	10	RAW F6	FPU1
13	MULTD F0 F4 F2	3	11	14	15	RAW F2	FPU2
14	SUBD F12 F2 F6	4	11	14	16	RAW F2 + Struct RF	FPU3
15	ADDD F0 F12 F2	16	17			WAW F0	FPU4

If the previous table was not correct, please, write the right one and specify the number,

kind and latency for each unit.

4 FPALU 3 cc latency, single write port for the pool







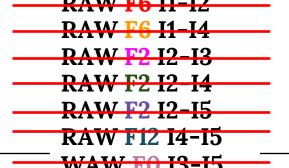
	Instruction	ISSUE	READ OPERAND	EXE COMPLETE	WB	Hazards	Unit
11	LD F6 32+ R2	1	2	4	5		MU
12	ADDD F2 F6 F4	2	6	9	10	RAW F6	FPU1
13	MULTD F0 F4 F2	3	11	14	15	RAW F2	FPU2
14	SUBD F12 F2 F6	4	11	14	16	RAW F2 + Struct RF	FPU3
15	ADDD F0 F12 F2	16	17	20	21	WAW F0	FPU4

If the previous table was not correct, please, write the right one and specify the number,

kind and latency for each unit.

4 FPALU 3 cc latency, single write port for the pool









Thank you for your attention Questions?

Alessandro Verosimile <alessandro.verosimile@polimi.it>

Acknowledgements

Davide Conficconi, E. Del Sozzo, Marco D. Santambrogio, D. Sciuto Part of this material comes from:

- "Computer Organization and Design" and "Computer Architecture A Quantitative Approach" Patterson and Hennessy books
- News and paper cited throughout the lecture

and are properties of their respective owners



