

ECE 468 & 573

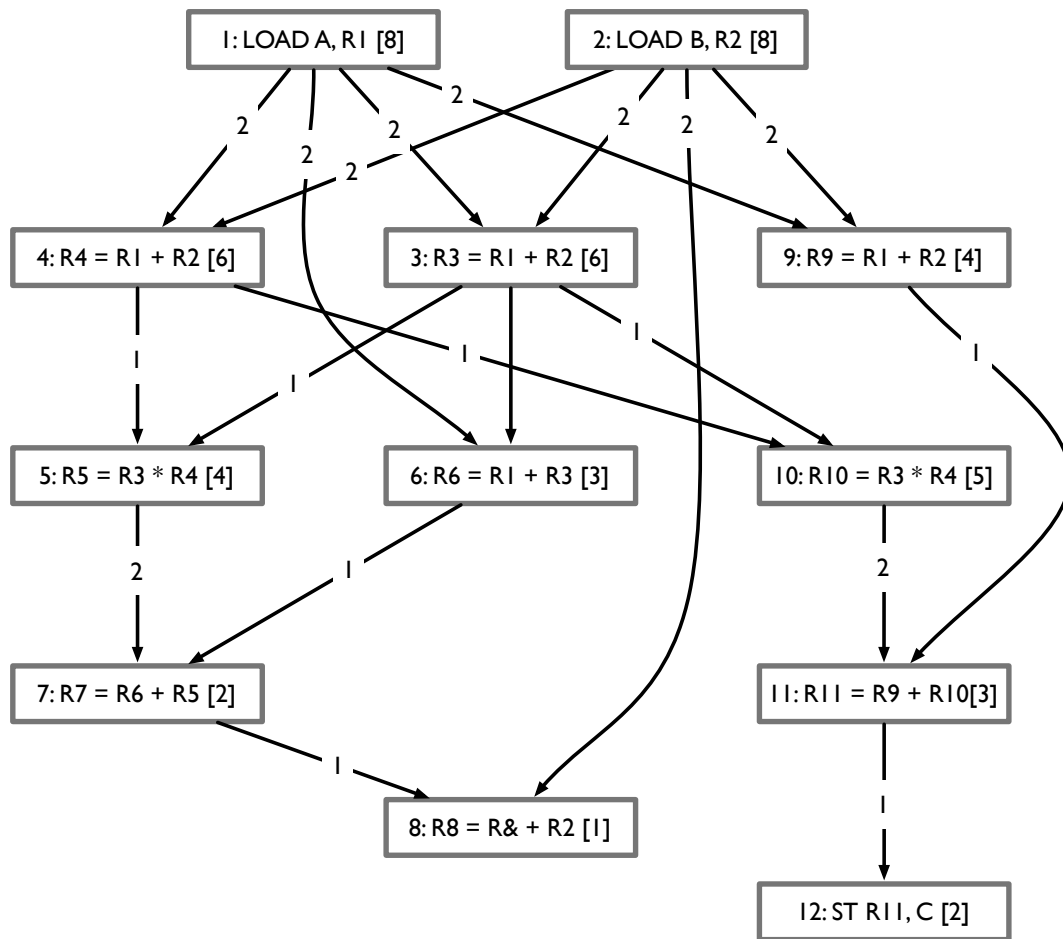
Problem Set 4: Instruction scheduling

For the following problems, consider the following piece of assembly code

1. LD A, R1
2. LD B, R2
3. R3 = R1 + R2
4. R4 = R1 + R2
5. R5 = R3 * R4
6. R6 = R1 + R3
7. R7 = R6 + R5
8. R8 = R7 + R2
9. R9 = R1 + R2
10. R10 = R3 * R4
11. R11 = R9 + R10
12. ST R11, C

1. Assume that the latency of ADDs is 1 cycle, the latency of MULs is 2 cycles, the latency of LOADs is 2 cycles and the latency of STs is 2 cycles [NOTE: the original problem set mistakenly said 1 cycle!]. Draw the data dependence graph for the above code, including heights.

Answer:



2. Assume that the target architecture has an *unlimited* amount of functional units. Give the schedule of execution for this code (for each instruction, list which cycle it would execute in).

Answer:

Cycle	Instructions scheduled
0	1, 2
1	1 (cont), 2 (cont)
2	3, 4, 9
3	5, 6, 10
4	5 (cont), 10 (cont)
5	7, 11
6	8, 12
7	12 (cont)

3. Assume that the architecture has 4 functional units: 2 ALUs, 1 LD/ST unit, and one (fully pipelined) MUL unit. ADDs take up either ALU for 1 cycle, LOADs and STOREs take up either ALU for 1 cycle and then the LD/ST unit for 1 cycle, while MULs can run only on the MUL unit. Draw the reservation tables for this architecture.

Answer:

ADD1:

ALU0	ALU1	MUL	LD/ST
x			

ADD2:

ALU0	ALU1	MUL	LD/ST
	x		

MUL: (Note that the MUL unit is fully pipelined—even though the latency of a multiply is 2 cycles, it only occupies the functional unit for 1)

ALU0	ALU1	MUL	LD/ST
		x	

LD1:

ALU0	ALU1	MUL	LD/ST
x			x

LD@:

ALU0	ALU1	MUL	LD/ST
	x		x

ST1 and ST2 look just like LD1 and LD2.

4. Show the results of performing height-based list scheduling for the architecture in the previous question.

Answer:

Cycle	ALU0	ALU1	MUL	LD/ST	Instructions scheduled
0	1				1
1	2			1	1 (cont), 2
2				2	2 (cont)
3	3	4			3, 4
4	6	9	10		6, 9, 10
5			5		5, 10 (cont)
6	11				5 (cont), 11
7	7	12			7, 12
8	8			12	8, 12 (cont)