IN1006 Systems Architecture (PRD1 A 2022/23)

Started on	Thursday, 15 December 2022, 5:29 PM
State	Finished
Completed on	Thursday, 15 December 2022, 5:49 PM
Time taken	20 mins 24 secs
Grade	10.00 out of 10.00 (100%)

Question 1

Correct

Mark 1.00 out of 1.00

In what way are compilers usually used to prevent pipeline hazards?

Select one:

- a. A method to avoid control hazards by predicting what way a conditional branch instruction will execute in advance.
- b. Preventing data hazards by releasing the results before they are written to the registers.
- c. Rearrange instructions such that an instruction that does not affect the decision is executed after the decision filling the bubble that would occur if the pipeline were to stall.
- d. Rearranging the instructions to avoid data hazards where possible. ✔
- e. Don't know/no answer
- f. Use of deep pipelines to maximise throughput.

Compliers can be used to rearrange the instructions to avoid data hazards.

The correct answer is: Rearranging the instructions to avoid data hazards where possible.

Question 2

Correct

Mark 1.00 out of 1.00

Consider the following MIPS code:

1. lw d1, 8($t0)
2. lw d2, 9($t0)
3. lw d3, 10($t0)
4. add e1, d1, d2
5. add e2, d1, d3
6. add e3, d2, d3

Which of the architectures in the Flynn taxonomy would be best suited for executing the above code?

Select one:

- a. Do not know the answer
- b. MIMD
- c. SISD
- d. SIMD ✔
- e. MISD
- f. SISD or SIMD

Your answer is correct.

Instructions 1-3 fit with SIMD as they involve the same instruction upon different data. Instructions 4-6 also fit with SIMD for the same reason.

Thus, SIMD is the best option for the above code.

The correct answer is: SIMD

Question 3

Correct

Mark 1.00 out of 1.00

How many superscalar execution units could be used for the MIPS pseudo-code below? Assume no register forwarding.

1. add a, b, c
2. add c, b ,c
3. add f, c ,d
4. add h, e, f
5. add f, h, b

Select one:

- a. Don't know/no answer
- b. 1 ✔
- c. 4
- d. 5
- e. 3
- f. 2

The number of execution units that can be used is limited to the largest sized block of non-dependent instructions. All instructions have some dependency with another.

The correct answer is: 1

Question 4

Correct

Mark 1.00 out of 1.00

Which of the following is *not* a design decision made in the MIPS architecture to support pipelining?

Select one:

- a. Instructions have the same format allowing the data to be fetched before the instruction is decoded.
- b. Operands are aligned hence no single data transfer instruction will require two memory accesses.
- c. Memory operands only appear in loads and stores allowing the execute stage to calculate addresses.
- d. Use of deep pipelines to maximise throughput. ✔
- e. Don't know/no answer
- f. All instructions have the same length and are therefore easier to decode in a pipeline.

The instruction pipelines in MIPS implementations are not especially deep, especially by modern standards.

The correct answer is: Use of deep pipelines to maximise throughput.

Question 5

Correct

Mark 1.00 out of 1.00

What is the maximum number of instructions that can be executed simultaneously for the MIPS pseudo-code below? Assume no register forwarding.

1. add a, b, c
2. add d, b ,c
3. add g, e ,c
4. add h, e, d
5. add f, a, b

Select one:

- a. 1
- b. 5
- c. 3 ✔
- d. 2
- e. Don't know/no answer
- f. 4

Instructions 1, 2 and 3 can be executed at the same time (they can share inputs as they have not changed). There is a dependency between 2 and 4 (i.e. d – the MIPS pipeline has two stages between execution and write to register). However 3, 4, 5 have no dependences. Therefore the largest size block of non-dependent instructions is 3.

The correct answer is: 3

Question 6

Correct

Mark 1.00 out of 1.00

Consider the following MIPS code:

1. add d1, b1, c1
2. add d2, b2 ,c2
3. add d3, b3 ,c3
4. add d4, b4, c4
5. add d5, c5, b5

Which of the four architectures in the Flynn taxonomy would be best suited for executing the above code?

Select one:

- a. SIMD ✔
- b. MIMD and MISD
- c. MIMD
- d. MISD
- e. Do not know the answer
- f. SISD

Your answer is correct.

Instructions 1-5 above are the same (add) and are executed upon different data. Thus, SIMD is the most suitable architecture.

The correct answer is: SIMD

Question 7

Correct

Mark 1.00 out of 1.00

How many superscalar execution units could be used for the MIPS pseudo-code below? Assume no register forwarding.

1. add a, b, c
2. sub d, b ,c
3. add g, e ,d
4. add h, e, f
5. sub a, a, b

Select one:

- a. 2
- b. 5
- c. 4
- d. Don't know/no answer
- e. 1
- f. 3 ✔

The number of execution units that can be used is limited to the largest sized block of non-dependent instructions. Instructions 1 and 2 can be executed at the same time (they can share inputs as they have not changed). There is a dependency between 2, 3 (i.e. d). However 3, 4, 5 have no dependences. Therefore the largest sized block of non-dependent instructions is 3. Mixing instruction types has no effect.

The correct answer is: 3

Question 8

Correct

Mark 1.00 out of 1.00

What is the maximum number of instructions that can be executed simultaneously for the MIPS pseudo-code below?

1. add a, b, c
2. add c, b ,c
3. add f, c ,d
4. add h, e, f
5. add f, h, b

Select one:

- a. 1 ✔
- b. 3
- c. 5
- d. Don't know/no answer
- e. 4
- f. 2

All instructions have some dependency with another.

The correct answer is: 1

Question 9

Correct

Mark 1.00 out of 1.00

How many superscalar execution units could be used for the MIPS pseudo-code below? Assume no register forwarding.

1. add a, b, c
2. add d, b ,c
3. add g, e ,d
4. add h, e, f
5. add f, a, b

Select one:

- a. 2 ✔
- b. Don't know/no answer
- c. 4
- d. 3
- e. 5
- f. 1

The number of execution units that can be used is limited to the largest sized block of non-dependent instructions. In the above block,

- Instructions 1, 2, do not depend on each other.
- Instruction 3 depends on instruction 2 (due to d)
- There is a dependency between 4 and 5 (due to f).

Therefore the largest size block of non-dependent instructions is 2.

The correct answer is: 2

Question 10

Correct

Mark 1.00 out of 1.00

Consider the following MIPS code:

1. add a, b, c
2. add c, b ,d
3. add f, c ,a
4. add k, l, m
5. add h, e, g
6. add f, h, b

Which of the following compiler optimisations could avoid data hazards in this code assuming a 4 stage pipeline (i.e., 1. fetch instruction; 2. Decode and load data; 3. Execute instruction; 4. Write to memory) with no forwarding?

- a. No optimisations are possible.
- b. Move instruction 4 and 5 before instruction 3 and swap the order of their execution. ✔
- c. Move instruction 4 and 5 before instruction 3.
- d. Move instruction 4 before instruction 3.
- e. Do not know the answer.

Your answer is correct.

The data hazards in this code arise due to instruction 3 (it depends on instructions 1 and 2) and instruction 6 (it depends on instruction 5).

Moving instruction 4 and 5 before instruction 3 and swapping the order of their execution would remove these hazards. This is because following these changes:

Instruction 3 would try to fetch data in the 6th cycle of the pipeline. In this cycle, instruction 2 will have completed the write to memory stage. Also instruction 6 which depends on instruction 5, would try to read data in cycle 7 by which instruction 5 will have written its output.

The correct answer is:
Move instruction 4 and 5 before instruction 3 and swap the order of their execution.

◄ Quiz 6 _ Weekly Assessed Quiz 2022

Jump to...

Quiz navigation

1 2 3 4 5 6 7 8 9 10

Show one page at a time

Finish review