IN1006 Systems Architecture (PRD1 A 2022/23)

| My Moodle | IN1006_PRD1_A_2022-23 | COURSEWORK 1: Weekly Assessed Quiz | Quiz 7 _ Weekly Assessed Quiz 2022

Started on	Thursday, 15 December 2022, 5:53 PM
State	Finished
Completed on	Thursday, 15 December 2022, 6:02 PM
Time taken	8 mins 29 secs
Grade	10.00 out of 10.00 (100%)

Question 1

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 ,b
4. add h, g, k
5. add f, a, h

Select one:

○ a. Don't know/no answer

● b. 3 ✓

○ c. 2

○ d. 1

○ e. 4

○ f. 5

Your answer is correct.

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 and 3 have no dependency.
- There is a dependency between 4 and 3 (due to g).
- There is a dependency between 4 and 5 (due to h).

Therefore the largest size block of non-dependent instructions is 3.

The correct answer is: 3

Question 2

Correct

Mark 1.00 out of 1.00

Which of the following is the best description of a ‘structural hazard’?

Select one:

○ a. A gap in the pipeline where some or all of the stages are not processing an instruction.

○ b. Waiting for the next instruction to be fetched after a branch is taken.

○ c. Instruction depends upon the results of a previous instruction still in the pipeline, e.g. compound math expressions.

● d. Hardware cannot support a combination of instructions in the same clock cycle, e.g. two simultaneous memory accesses. ✓

○ e. Don't know/no answer

○ f. Loading instructions into pipeline before the result of a decision is known, e.g. loading instructions after a branch

A structural hazard occurs when hardware cannot support a combination of instructions in the same clock cycle, e.g. two simultaneous memory accesses.

The correct answer is: Hardware cannot support a combination of instructions in the same clock cycle, e.g. two simultaneous memory accesses.

Question 3

Correct

Mark 1.00 out of 1.00

Consider the following MIPS code:

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

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. Move instruction 5 before instruction 1.

○ b. No compiler optimisation is possible.

● c. Move instructions 4 and 5 before instruction 3. ✓

○ d. Move instruction 4 before instruction 3.

○ e. Do not know the answer.

Your answer is correct.

Instruction 3 depends on instruction 2 (due to b). Instructions 4 and 5 can be moved before it in the program without affecting the overall program outcome. This move would make instruction 3 to load b in the 7th cycle in a pipeline execution (assuming that instruction 1 starts in cycle 1). In the 7th cycle however instruction 2 will have written b to memory (this would happen in cycle 5 of the pipeline). So the data hazard would no longer be present.

The correct answer is:
Move instructions 4 and 5 before instruction 3.

Question 4

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. 1

○ b. 2

○ c. 4

○ d. Don't know/no answer

● e. 3 ✓

○ f. 5

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 5

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. Move instruction 4 before instruction 3.

○ b. Move instruction 4 and 5 before instruction 3.

○ c. No optimisations are possible.

● d. Move instruction 4 and 5 before instruction 3 and swap the order of their execution. ✓

○ 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.

Question 6

Correct

Mark 1.00 out of 1.00

Which of the following describes best the concept of "out-of-order" execution?

○ a. Instructions are loaded first to cache before brought to the CPU for execution.

● b. An instruction and a data queue are used to regulate the execution of instructions by the CPU. ✓

○ c. Instructions are executed in two separate CPUs operating in parallel.

○ d. Instructions are executed in pipelines.

○ e. Do not know the answer

Your answer is correct.

Instructions are not given to the CPU for execution immediately when they are due. They are placed in a queue and are moved from it to the CPU only when all the data that they need are available. Also the data D produced by an instruction I are placed in a queue and only when the data of all the instructions that were earlier than I are also written in the queue data, D are written to memory.

The correct answer is:
An instruction and a data queue are used to regulate the execution of instructions by the CPU.

Question 7

Correct

Mark 1.00 out of 1.00

Which of the following is the best description of a ‘stall’?

Select one:

○ a. A gap in the pipeline where some or all of the stages are not processing an instruction.

○ b. Don't know/no answer

● c. Waiting for the next instruction to be fetched after a branch is taken. ✓

○ d. Hardware cannot support a combination of instructions in the same clock cycle, e.g. two simultaneous memory accesses.

○ e. Instruction depends upon the results of a previous instruction still in the pipeline, e.g. compound math expressions.

○ f. Loading instructions into pipeline before the result of a decision is known, e.g. loading instructions after a branch

A stall occurs in response to a control hazard. It involves waiting for the next instruction to be fetched after a branch is taken.

The correct answer is: Waiting for the next instruction to be fetched after a branch is taken.

Question 8

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. SISD

○ b. MISD

○ c. Do not know the answer

○ d. MIMD and MISD

○ e. MIMD

● f. SIMD ✓

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 9

Correct

Mark 1.00 out of 1.00

Which of the following is the best description of a ‘data hazard’?

Select one:

○ a. Loading instructions into pipeline before the result of a decision is known, e.g. loading instructions after a branch

○ b. Don't know/no answer

○ c. A gap in the pipeline where some or all of the stages are not processing an instruction.

● d. Instruction depends upon the results of a previous instruction still in the pipeline, e.g. compound math expressions. ✓

○ e. Waiting for the next instruction to be fetched after a branch is taken.

○ f. Hardware cannot support a combination of instructions in the same clock cycle, e.g. two simultaneous memory accesses.

A data hazard occurs when an instruction depends upon the results of a previous instruction still in the pipeline, e.g. compound math expressions.

The correct answer is: Instruction depends upon the results of a previous instruction still in the pipeline, e.g. compound math expressions.

Question 10

Correct

Mark 1.00 out of 1.00

What does the acronym ILP stand for?

Select one:

● a. Instruction Level Parallelism ✓

○ b. Integer Logic Parallelism

○ c. Information Leak Prevention

○ d. Don't know/no answer

○ e. Inductive Logic programming

○ f. Integer Linear programming

ILP stands for Instruction Level Parallelism in the context of systems architecture. See: http://en.wikipedia.org/wiki/Instruction_level_parallelism

The correct answer is: Instruction Level Parallelism

◄ 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