IN1006 Systems Architecture (PRD1 A 2022/23)

Started on	Thursday, 15 December 2022, 5:32 PM
State	Finished
Completed on	Thursday, 15 December 2022, 5:50 PM
Time taken	17 mins 25 secs
Grade	**10.00** out of 10.00 (**100**%)

Question **1**

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

Select one:

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

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 size block of non-dependent instructions is 3. Mixing instruction types has no effect.

The correct answer is: 3

Question **2**

Correct

Mark 1.00 out of 1.00

Consider the following MIPS code:

1. add d1, b1, c1
2. sub d2, b1 ,c1
3. add e1, d1, d2
4. sub e2, d1, d2

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

Select one:

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

Your answer is correct.

Instructions 1-2 fit with MISD as they involve different instructions upon the same data. Instructions 3-4 also fit with MISD for the same reason.

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

The correct answer is: MISD

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. Move instruction 4 before instruction 3.
c. Do not know the answer.
d. No compiler optimisation is possible.
e. Move instructions 4 and 5 before instruction 3. ✔

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

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

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 **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?

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

All instructions have some dependency with another.

The correct answer is: 1

Question **6**

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

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 **7**

Correct

Mark 1.00 out of 1.00

Which of the following is the best description of a 'stall'?

Select one:

- a. Waiting for the next instruction to be fetched after a branch is taken. ✔
- b. Don't know/no answer
- c. Instruction depends upon the results of a previous instruction still in the pipeline, e.g. compound math expressions.
- d. Loading instructions into pipeline before the result of a decision is known, e.g. loading instructions after a branch
- e. A gap in the pipeline where some or all of the stages are not processing an instruction.
- f. Hardware cannot support a combination of instructions in the same clock cycle, e.g. two simultaneous memory accesses.

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

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

- a. Instructions are executed in two separate CPUs operating in parallel.
- b. Instructions are executed in pipelines.
- c. An instruction and a data queue are used to regulate the execution of instructions by the CPU. ✔
- d. Instructions are loaded first to cache before brought to the CPU for execution.
- 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 **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 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. 2
- 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. All instructions have some dependency with another.

The correct answer is: 1

Question **10**

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

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

◄ 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