IN1006 Systems Architecture (PRD1 A 2022/23)

Started on	Thursday, 15 December 2022, 5:28 PM
State	Finished
Completed on	Thursday, 15 December 2022, 5:49 PM
Time taken	21 mins 47 secs
Grade	**10.00** out of 10.00 (**100**%)

Question **1**

Correct

Mark 1.00 out of 1.00

Consider the following MIPS code:

```
1. lw  $t1, 0($t0)
2. lw  $t2, 1($t0)
3. add  $t3, $t1, $t2
4. sw  $t3, 0($t1)
5. lw  $t4, 2($t0)
6. add  $t5, $t1, $t4
7. sw  $t5, 1($t1)
8. lw  $t6, 3($t0)
9. add  $t5, $t1, $t6
10. sw  $t5, 2($t1)
```

Which of the following would offer the maximum protection against data hazards in this code without affecting the results of its execution, if a 5-stage pipeline is assumed?

- a. Move instruction 5 before instruction 3.
- b. Move instructions 3 and 4 after instruction 5.
- c. Move instruction 5 and instruction 7 before instruction 3.
- d. Move instructions 3 and 6 after instruction 10.
- e. Move instruction 5 and instruction 8 before instruction 3. ✔

Your answer is correct.

Data hazards in this code extract arise due to loading data immediately before the computations that use them as will make computations stall whilst waiting loading instructions to complete their execution.

Such hazards can be removed by moving instruction 5 before instruction 3 (this will avoid the hazard between instructions 2 and 3 and the hazard between instructions 5 and 6), and instruction 8 before instruction 3 (this will avoid the hazard between instructions 8 and 9).

The correct answer is:
Move instruction 5 and instruction 8 before instruction 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. Don't know/no answer
- d. Loading instructions into pipeline before the result of a decision is known, e.g. loading instructions after a branch
- e. Hardware cannot support a combination of instructions in the same clock cycle, e.g. two simultaneous memory accesses. ✔
- f. Instruction depends upon the results of a previous instruction still in the pipeline, e.g. compound math expressions.

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

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

Select one:

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

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 **4**

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

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 **5**

Correct

Mark 1.00 out of 1.00

Consider the following MIPS code:

1. add a, b, c
2. add b, b ,d
3. add f, e , a
4. add h, k, g
5. add m, a, 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. Do not know the answer.
b. No optimisations are possible.
c. Move instructions 5 before instruction 4.
d. Move instruction 4 before instruction 3. ✔
e. Move instruction 5 before instruction 3.

Your answer is correct.

The data hazards in this code arise due to instruction 3 (it depends on instruction 1).

Moving instruction 4 before instruction 3 will resolve the hazard as following the move instruction 3 will read data in cycle 5 and by this cycle instruction 1 will have stored its output.

The correct answer is:
Move instruction 4 before instruction 3.

Question **6**

Correct

Mark 1.00 out of 1.00

Which of the following is the best description of a 'control hazard'?

Select one:

a. Loading instructions into pipeline which try to access memory at the same stage.
b. Instruction depends upon the results of a previous instruction still in the pipeline, e.g. compound math expressions.
c. Hardware cannot support a combination of instructions in the same clock cycle, e.g. two simultaneous memory accesses.
d. Executing an instruction in the pipeline whose execution depends upon a yet unknown condition check. ✔
e. Don't know/no answer
f. A gap in the pipeline where some or all of the stages are not processing an instruction.

Your answer is correct.

A control hazard occurs when an instruction I has been loaded to a pipeline although due to some condition check which is already being executed in the pipeline it turns out that I should not be executed.

The correct answer is: Executing an instruction in the pipeline whose execution depends upon a yet unknown condition check.

Question **7**

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

All instructions have some dependency with another.

The correct answer is: 1

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. MIMD and MISD
- b. Do not know the answer
- c. MISD
- d. MIMD
- e. SISD
- 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 describes best the concept of "out-of-order" execution?

- a. Instructions are loaded first to cache before brought to the CPU for execution.
- b. Instructions are executed in two separate CPUs operating in parallel.
- c. Do not know the answer
- d. An instruction and a data queue are used to regulate the execution of instructions by the CPU. ✔
- e. Instructions are executed in pipelines.

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 **10**

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

Select one:

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

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 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).
- There is a dependency between 1 and 2 (due to d).

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

The correct answer is: 2

Jump to...

Quiz navigation

1 2 3 4 5 6 7 8 9 10

Show one page at a time

Finish review