IN1006 Systems Architecture (PRD1 A 2022/23)

Started on	Thursday, 15 December 2022, 5:21 PM
State	Finished
Completed on	Thursday, 15 December 2022, 5:43 PM
Time taken	22 mins 14 secs
Grade	**10.00** out of 10.00 (**100**%)

Question **1**

Correct

Mark 1.00 out of 1.00

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

Select one:

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

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 **2**

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. 1
- c. 5
- d. 2 ✔
- e. 3
- f. 4

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

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. 3
- b. 2
- c. 5
- d. 4
- e. Don't know/no answer
- 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 **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. SIMD
- b. MIMD
- c. MISD ✔
- d. MIMD or SISD
- e. SISD or SIMD
- 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 **5**

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

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 **6**

Correct

Mark 1.00 out of 1.00

What does the acronym ILP stand for?

Select one:

- a. Integer Logic Parallelism
- b. Integer Linear programming
- c. Don't know/no answer
- d. Instruction Level Parallelism ✔
- e. Inductive Logic programming
- f. Information Leak Prevention

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

Question **7**

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

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 **8**

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

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

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. Use of deep pipelines to maximise throughput. ✔
- e. Operands are aligned hence no single data transfer instruction will require two memory accesses.
- 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.

◄ Quiz 6 _ Weekly Assessed Quiz 2022

Jump to...