IN1006 Systems Architecture (PRD1 A 2022/23)

Started on	Thursday, 15 December 2022, 6:12 PM
State	Finished
Completed on	Thursday, 15 December 2022, 6:36 PM
Time taken	23 mins 35 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. Loading instructions into pipeline which try to access memory at the same stage.
- b. Don't know/no answer
- c. Executing an instruction in the pipeline whose execution depends upon a yet unknown condition check. ✔
- 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. 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 **2**

Correct

Mark 1.00 out of 1.00

Match the description for the correct MIPS notation for MIPS Pipeline.

Instruction fetch from memory	IF ✔
Instruction decode and register read	ID ✔
Write result back to register	WB ✔
Access memory operand	MEM ✔
Execute operation or calculate address	EX ✔

Your answer is correct.

The correct answer is: Instruction fetch from memory → IF,
Instruction decode and register read
→ ID,
Write result back to register
→ WB,
Access memory operand
→ MEM,
Execute operation or calculate address
→ EX

Question **3**

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

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 d3, b2 ,c2
4. sub d4, b2, c2

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

Select one:

- a. SISD or SIMD
- b. SIMD or MISD ✔
- c. MIMD or MISD
- 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.

Instructions 1, 3 fit with SIMD as they involve the same instruction upon different datasets. Instructions 2-4 also fit with SIMD for the same reason.

Thus, MISD or SIMD would be the best options for the above code.

The correct answer is: SIMD or MISD

Question **5**

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

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

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

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

```
add a, b, c
add c, b ,c
add f, c ,d
add h, e, f
add f, h, b
```

Select one:

- a. 3
- b. Don't know/no answer
- c. 2
- d. 4
- e. 1 ✔
- 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. All instructions have some dependency with another.

The correct answer is: 1

Question **9**

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. Do not know the answer.
c. No compiler optimisation is possible.
d. Move instruction 4 before instruction 3.
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 **10**

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

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

◄ Quiz 6 _ Weekly Assessed Quiz 2022

Jump to...

Quiz navigation

10

Show one page at a time

Finish review