Started on	Friday, 12 April 2024, 3:48 PM
State	Finished
Completed on	Friday, 12 April 2024, 4:09 PM
Time taken	21 mins 9 secs
Grade	**1.00** out of 13.00 (**7.69**%)

Question **1**

Not answered

Marked out of 0.50

An atomic instruction...

Select one:

- a. ... is formed by multiple non-atomic instructions.
- b. ... is executed in exactly 1 clock cycle.
- c. ... is executed indivisibly.
- d. ... can be interrupted only by hardware interrupts.

Your answer is incorrect.

The correct answer is: ... is executed indivisibly.

Question **2**

Not answered

Marked out of 1.00

Consider a simple core with a single-level of cache that achieves 1 CPI when every memory operation hits in the cache. The miss penalty to the main memory is 50 cycles. For a specific workload, the miss rate is 1 miss per every 50 instructions.

What is the CPI (rounded to the nearest integer) of this chip for this workload?

Answer:

✖

The correct answer is: 2 $^{+}/_{-}$ 0

Question **3**

Not answered

Marked out of 1.00

Consider adding a second-level cache in between the first-level cache and memory. For this workload, the miss rate in the second-level cache is 1 miss per 300 instructions and has the same latency as accessing memory as above (50 cycles). The first-level cache is unchanged with a miss rate of 1 miss per 50 instructions. In order to make sure that adding this cache reaches a performance of CPI = 2, calculate the maximum number of cycles (rounded to the nearest integer) that are allowed for the access latency of this second-level cache from the first-level cache.

Answer:

✖

The correct answer is: 42 +/- 0

Question **4**

Not answered

Marked out of 1.00

The importance of having a good branch predictor depends on how often conditional branches are executed. Together with branch predictor accuracy, this will determine how much time is spent stalling due to mispredicted branches. Assume that the breakdown of dynamic instructions into various instruction categories is as follows:

R-Type	BEQ	JMP	LW	SW
40%	25%	5%	25%	5%

Also, assume the following branch predictor accuracies:

Always-Taken	Always-Not-Taken	2-Bit
45%	50%	85%

Stall cycles due to mispredicted branches increase the CPI. Assuming the 5-stage MIPS processor from the textbook, what is the extra CPI due to mispredicted branches with the "2-bit" predictor? Assume that branch outcomes are determined in the EX stage, that there are no data hazards, and that no delay slots are used.

Give your answer with an accuracy of at least two decimal places.

Answer:

✖

Each branch that is not correctly predicted by the 2-bit predictor will cause 3 stall cycles, so we have:

Extra CPI = 3* (1 – (2-bit accuracy)) * 0.25

The correct answer is: 0.1125 +/- 0.05

Question **5**

Not answered

Marked out of 1.00

With the 2-bit predictor, what speedup would be achieved if we could convert half of the branch instructions in a way that replaces a branch instruction with an ALU instruction? Assume that correctly and incorrectly predicted instructions have the same chance of being replaced.

Give your answer with an accuracy of at least two decimal places.

Answer:

✖

Correctly predicted branches had CPI of 1 and now they become ALU instructions whose CPI is also 1. Incorrectly predicted instructions that are converted also become ALU instructions with a CPI of 1, so we have:

CPI without conversion = 1 + 3 * (1 – (2-bit accuracy)) * 0.25

CPI with conversion = 1 + 3 * (1 – (2-bit accuracy)) * 0.25 * 0.5

Speedup from conversion = CPI without conversion / CPI with conversion

The correct answer is: 1.0532544378698 +/- 0.05

Question **6**

Not answered

Marked out of 1.00

Some branch instructions are much more predictable than others. If we know that 80% of all executed branch instructions are easy-to-predict loop-back branches that are always predicted correctly, what is the accuracy, in percentage, of the 2-bit predictor on the remaining 20% of the branch instructions?

Important Note : Only answer with percentage value without using "%" sign. E.g: If half of the remaining branches are correctly predicted, answer with "50".

Answer:

✖

Let the total number of branch instructions executed in the program be B. Then we have:

Correctly predicted: B * (2-bit accuracy)

Correctly predicted non-loop-back: B * (2-bit accuracy - 0.80)

Accuracy on non-loop-back branches: Correctly predicted non-loop-back / (B * 0.20)

The correct answer is: 25 +/- 0.05

Question **7**

Not answered

Marked out of 1.00

Consider the following MIPS loop:

```
LOOP: slt  $t2, $0, $t1
      beq  $t2, $0, DONE
      subi $t1, $t1, 1
      addi $s2, $s2, 2
      j    LOOP
DONE:
```

Assume that the register t1 is initialized to the value 17. What is the value in register s2 assuming s2 is initially zero?

Answer:

✖

The correct answer is: 34 +/- 0

Question **8**

Not answered

Marked out of 1.00

Assume that the register t1 is initialized to the value 86. How many MIPS instructions are executed before the instruction at the 'DONE:' label if the branch delay slot is not used?

Answer:

✖

The correct answer is: 432 +/- 0

Question **9**

Incorrect

Mark 0.00 out of 1.00

Consider the 5-stage pipelined MIPS processor, as described in the book, running the following code (9 instructions):

```
lw  $t0, 64($zero)
lw  $t1, 0($t0)
lw  $t2, 4($t0)
add $t5, $t5, 8
add $t3, $t1, $t2
sw  $t3, 12($t0)
lw  $t4, 8($t0)
add $t5, $t1, $t4
sw  $t5, 16($t0)
```

How many cycles are required for the execution of this code when **no forwarding** circuitry has been implemented? Assume **3 stall cycles** in case of RaW dependencies. Omit the 4 cycles required to fill the pipeline.

Answer: 24 ✖

```
lw  t0,64(zero)
( 3 cycle stall )
lw  t1,0(t0)
lw  t2,4(t0)
add t5,t5, 8
( 2 cycle stall )
add t3,t1, $t2
( 3 cycle stall )
sw  t3,12(t0)
lw  t4,8(t0)
( 3 cycle stall )
add t5,t1, $t4
( 3 cycle stall )
sw  t5,16(t0)
```

9+14 = 23

The correct answer is: 23

Question **10**

Correct

Mark 1.00 out of 1.00

Consider the 5-stage pipelined MIPS processor, as described in the book, running the following code (8 instructions):

```
 lw  $t1, 0($t0)
 lw  $t2, 4($t0)
add  $t3, $t1, $t2
add  $t3, $t2, $t3
 sw  $t3, 12($t0)
 lw  $t4, 8($t0)
add  $t5, $t1, $t4
 sw  $t5, 16($t0)
```

How many cycles are required for the execution of this code **with all possible forwarding (bypassing) and remaining hazard detection** circuitry implemented? You may omit the 4 cycles required to fill the pipeline. (Hint: Carefully check all the RaW dependencies).

Answer: 10 ✔

```
 lw  t1,0(t0)
 lw  t2,4(t0)
( 1 cycle stall, $t2 is available to be forwarded after its data is loaded from memory, which occurs
at the MEM stage )
add  t3,t1, $t2
add  t3,t2, $t3
 sw  t3,12(t0)
 lw  t4,8(t0)
( 1 cycle stall )
add  t5,t1, $t4
 sw  t5,16(t0)
```

8+2 = 10 cycles

The correct answer is: 10

Question **11**

Incorrect

Mark 0.00 out of 0.50

In MIPS pipeline, what are the measures that can be taken to reduce the impact of **data** hazards?

Select one or more:

- a. Allow split register write and read during the two halves of the same clock cycle
- b. Splitting the memory into separate instruction and data memories. ✖ Reduces the impact of structural hazards
- c. Replicate the register bank
- d. Implement data forwarding in the datapath

Your answer is incorrect.

The correct answers are: Implement data forwarding in the datapath, Allow split register write and read during the two halves of the same clock cycle

Question **12**

Not answered

Marked out of 0.50

What is the primary reason that a translation lookaside buffer (TLB) is used?

Select one:

- a. None of the above
- b. A TLB makes translating virtual addresses to physical addresses possible
- c. A TLB makes translating virtual addresses to physical addresses faster
- d. A TLB allows multiple processes to share the L1 cache
- e. A TLB ensures that a process does not access memory outside of its address space

Your answer is incorrect.

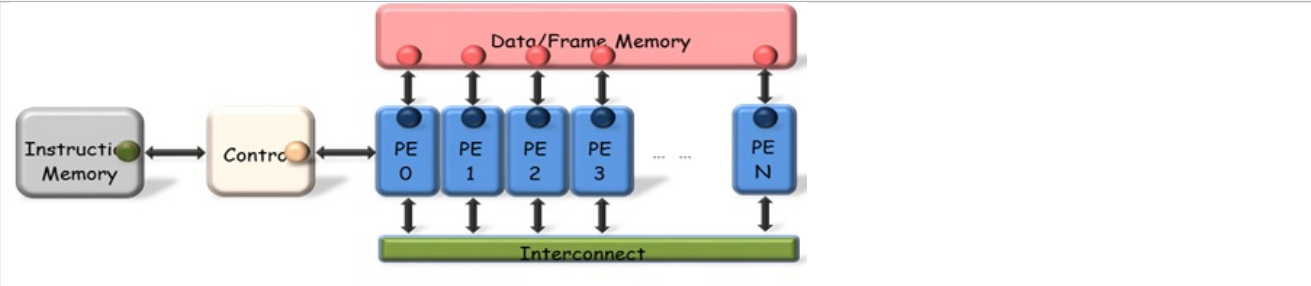
The correct answer is: A TLB makes translating virtual addresses to physical addresses faster

Question **13**

Not answered

Marked out of 1.00

Look at the block diagram of an SIMD processor and answer the questions.



On an SIMD every PE executes the same operation.	Choose...
A 4 PE SIMD needs less energy than a 4 core MIMD.	Choose...
On an SIMD If some lanes go a different direction after a branch, then both the then- and else-parts have to be executed after each other for all PEs which increases efficiency.	Choose...
Vector parallelism is exploited in an SIMD processor.	Choose...
A 4 PE SIMD needs to fetch and decode 4 instruction per cycle.	Choose...
On an MIMD different cores can execute different branches simultaneously.	Choose...

Your answer is incorrect.

The correct answer is: On an SIMD every PE executes the same operation. → True, A 4 PE SIMD needs less energy than a 4 core MIMD. → True, On an SIMD If some lanes go a different direction after a branch, then both the then- and else-parts have to be executed after each other for all PEs which increases efficiency. → False, Vector parallelism is exploited in an SIMD processor. → True, A 4 PE SIMD needs to fetch and decode 4 instruction per cycle. → False, On an MIMD different cores can execute different branches simultaneously. → True

Question **14**

Not answered

Marked out of 0.50

The time required to create a new thread in an existing process is:

Select one:

- a. less than the time required to create a new process
- b. none of the mentioned
- c. equal to the time required to create a new process
- d. greater than the time required to create a new process

Your answer is incorrect.

The correct answer is: less than the time required to create a new process

Question **15**

Not answered

Marked out of 1.00

Suppose you want to achieve a speed-up of 60 times faster with 146 processors.

What percentage of the original computation must be parallelizable? **Use 4 significant figures in your answer (2 becomes 2.000, 0.015151 becomes 0.01515).**

Answer:

✖

From the book at page 505:

Speed-up Challenge

EXAMPLE

Suppose you want to achieve a speed-up of 90 times faster with 100 processors. What percentage of the original computation can be sequential?

ANSWER

Amdahl's Law (Chapter 1) says

$$\text{Execution time after improvement} = \frac{\text{Execution time affected by improvement}}{\text{Amount of improvement}} + \text{Execution time unaffected}$$

We can reformulate Amdahl's Law in terms of speed-up versus the original execution time:

$$\text{Speed-up} = \frac{\text{Execution time before}}{(\text{Execution time before} - \text{Execution time affected}) + \frac{\text{Execution time affected}}{\text{Amount of improvement}}}$$

This formula is usually rewritten assuming that the execution time before is 1 for some unit of time, and the execution time affected by improvement is considered the fraction of the original execution time:

$$\text{Speed-up} = \frac{1}{(1 - \text{Fraction time affected}) + \frac{\text{Fraction time affected}}{\text{Amount of improvement}}}$$

Substituting 90 for speed-up and 100 for amount of improvement into the formula above:

$$90 = \frac{1}{(1 - \text{Fraction time affected}) + \frac{\text{Fraction time affected}}{100}}$$

The correct answer is: 0.99011494252874 +/- 0.0099011494252874

Information

WARNING!

Submitting the quiz is final. Your attempt cannot be reopened. The quiz will be submitted automatically at the end of the exam.

You do not need to submit the quiz to access the hardware design questions.

You must submit the quiz if you intend to leave the exam room before the end of the exam.