Summary Exercise - Week 10

Due Dec 8 at 11:59pm   **Points** 14   **Questions** 9
Available Dec 1 at 12am - Dec 8 at 11:59pm 8 days   **Time Limit** 360 Minutes
Allowed Attempts 2

Attempt History

	Attempt	Time	Score
KEPT	**Attempt 2**	12 minutes	14 out of 14
LATEST	**Attempt 2**	12 minutes	14 out of 14
	Attempt 1	56 minutes	14 out of 14

Score for this attempt: **14** out of 14
Submitted Dec 3 at 1:47am
This attempt took 12 minutes.

Question 1   1 / 1 pts

The CPU clock cycle length is the only contributing factor to the speed of operations on a computer.

○ True

Correct! ◉ False

Question 2   2 / 2 pts

Suppose that you are working with a CISC machine using a 1.9 GHz clock (i.e., the clock ticks 1.9 billion times per second). This particular computer uses MASM-like instructions with the following timings:

```
add reg, mem    8 clock cycles (i.e., the ADD micro-program has 8 instructions)
add reg, immed  4 clock cycles
loop label      5 clock cycles
```

Suppose that the following code fragment is used to sum elements of a numeric array. For this problem, assume that memory limitations are non-existent and that there is no limit to the size of the array.

```
    mov bx,  0              ;initialize sum
    mov ecx, MAX_SIZE      ;initialize loop counter
    mov esi, OFFSET list   ;initialize array pointer
more:
    add bx,  [esi]          ;add current list element
    add esi, 2             ;move array pointer to next element
    loop    more           ;auto-decrement ecx, jump to more if ecx ≠ 0
```

After initialization, how many array elements can be processed in 2.2 ms? Round your answer to the nearest integer. Note that 1 ms. = 0.001 second.

Correct!

245,882

orrect Answer

245,882   margin of error +/- 1

Question 3   2 / 2 pts

Suppose you have a RISC machine with a 1.7 GHz clock (i.e., the clock ticks 1.7 billion times per second). This particular computer uses an instruction cache, a data cache, an operand fetch unit, and an operand store unit. The instruction set includes simple instructions with the following timings:

```
set  reg, immed  1 clock cycle
loop label       4 clock cycles
add reg, immed   2 clock cycle
add reg, reg     2 clock cycles
load reg, mem    2 clock cycles
```

Assume that the following code fragment is used to sum the element of a numeric array. If the initialization code has already executed (i.e. the SET

instructions have already finished execution) how many array elements can be processed in 2.6 ms? Round your answer to the nearest integer. Recall that 1 ms = 0.001 seconds. Also assume that there are no physical memory limitations, implying that the array can be as large as desired.

```
    set   r1, 0        ;initialize sum
    set   r2, MAX_SIZE ;initialize loop counter
    set   r3, @list    ;initialize array pointer
  more:
    load  r4, [r3]     ;fetch current list element
    add   r1, r4       ;add current list element
    add   r3, 4        ;move array pointer to next element
    loop  more         ;auto-decrement r2, jump to more if r2 != 0
```

Correct!

442,000

orrect Answer

442,000   margin of error +/- 1

Question 4

1 / 1 pts

Which of the following portions of a program can complicate the instruction-caching process? (Check all that apply)

Correct!

☑ Recursion

Correct!

☑ Decision Structures

☐ Sequential Execution

Correct!

☑ Repetition

Question 5

1 / 1 pts

Multiprocessor Parallelism (shared memory) usually has lower coordination overhead than Multicomputer Parallelism (distributed memory).

◉ True

◯ False

Question 6

1 / 1 pts

Software parallelism is currently much more developed than hardware parallelism.

◯ True

◉ False

Question 7

2 / 2 pts

An algorithm takes 7.5 seconds to execute on a single 3 GHz processor. 43% of the algorithm is sequential. Assume that there is zero latency and that the remaining code exhibits perfect parallelism.

How long (in seconds) should the algorithm take to execute on a parallel machine made of 4 3 GHz processors? Round answers to one decimal place.
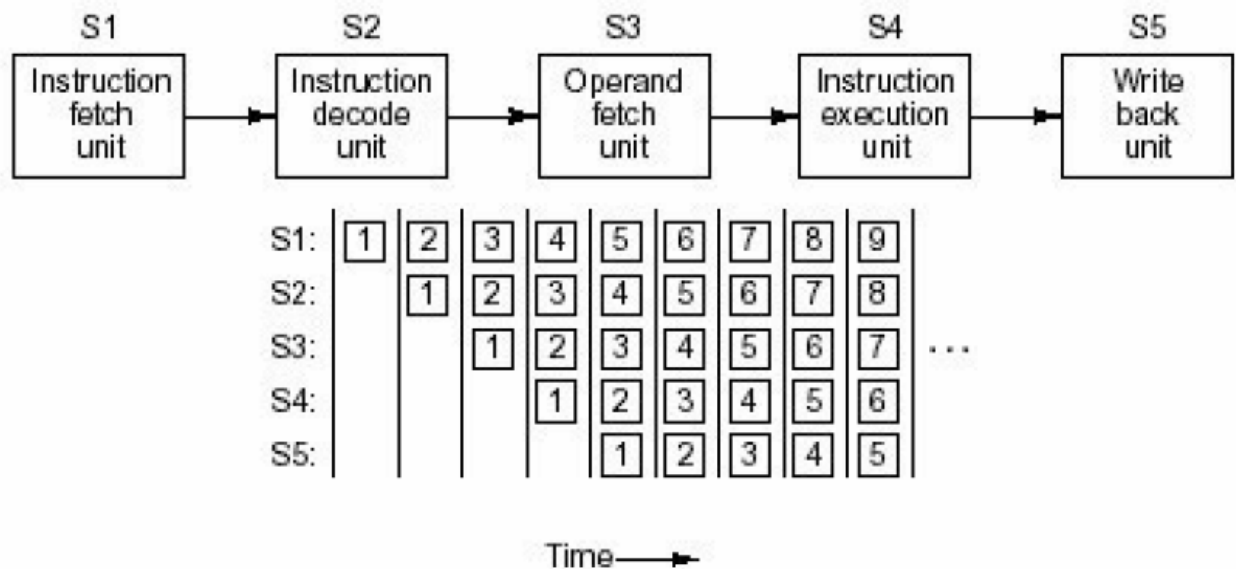
4.3

4.3 margin of error +/- 0.1

Question 8

2 / 2 pts

Assume that you are working with the five-stage pipeline shown in the diagram.

Suppose that each stage requires 3.5 nanoseconds to complete its task. How many nanoseconds would it take to complete 57 instructions *without* pipelining? Round your answer to the nearest integer.

997

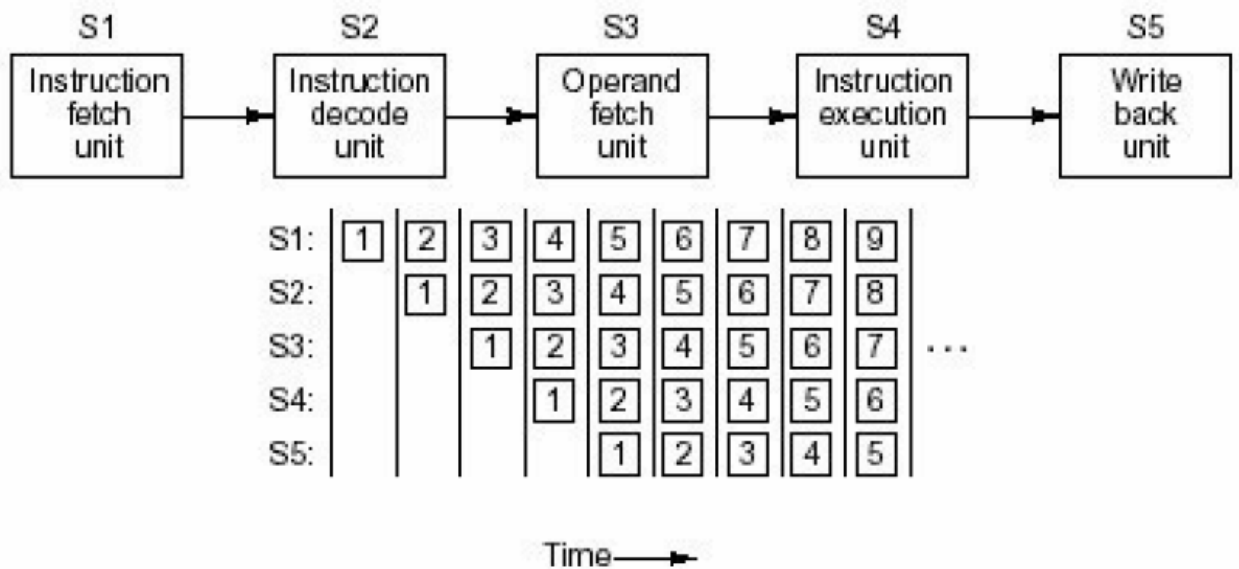998   margin of error +/- 1

Question 9

2 / 2 pts

Assume that you are working with the five-stage pipeline shown in the diagram.

	S1		S2		S3		S4		S5

S1: Instruction fetch unit → S2: Instruction decode unit → S3: Operand fetch unit → S4: Instruction execution unit → S5: Write back unit

S1: 1 2 3 4 5 6 7 8 9
S2: 1 2 3 4 5 6 7 8
S3: 1 2 3 4 5 6 7 ...
S4: 1 2 3 4 5 6
S5: 1 2 3 4 5

Time ⟶

Suppose that each stage requires 3.8 nanoseconds to complete its task. How many nanoseconds will it take to complete 87 instructions *with* pipelining? Round your answer to the nearest integer.

Correct!

345

Correct Answer

346   margin of error +/- 1

Quiz Score: **14** out of 14