IT5002 Computer Systems and Applications
2024/25 Semester II
Final Assessment

Question 1. (9 MARKS)

Consider the following program, where the base address of a 100 element array is in $s0. In all parts we assume that this program is executed to completion.

```
      addi $t0, $s0, 400
      add $s1, $zero, 0
L:    lw $t1, 0($s0)
      add $s1, $s1, $t1
      addi $s0, $s0, 4
      slt $t1, $s0, $t0
      bne $t1, $zero, L
```

a. How many instructions are executed by this program? (2 marks)

b. **Assuming that your answer to part a. is 637 instructions**, how long does it take in microseconds to execute this code on a single-cycle CPU with a clock rate of 100 MHz? Express your answer to two decimal places if it is not a whole number. (1 microsecond = 10^{-6} seconds). (2 marks)

c. **Again, assuming that your answer to part a. is 637 instructions**, How long does it take to execute this program on a CPU with an ideal **4-stage** pipeline? Assume the following timings for each stage (1 ns = 10^{-9} seconds)

Fetch	Decode	Execute	Writeback
10 ns	5 ns	5 ns	10 ns

Ignore dependencies between instructions and branching hazards (a branch hazard is caused by the pipeline not knowing whether to fetch the next instruction or the instruction from the branch target).

State your answer in ns, in two decimal places if your answer is not a whole number. (1 ns = 10^{-9} seconds) (2 marks)

d. Assuming a multicycle CPU where the stages take the following time in ns (1 ns = 10^{-9} seconds)

Fetch	Decode	Execute	Memory	Writeback
15 ns	5 ns	10 ns	15 ns	5 ns

Instructions take the following number of clock cycles each:

Instruction Type	**Number of Clock Cycles**
Arithmetic	4
Load/Store	5
Branching	3

If our program executed a total of 350 arithmetic, 180 load/store and 95 branching instructions, how long would it take to execute it?

Express your answer in ns, in two decimal places if it is not a whole number. (1 ns = 10^{-9} seconds) (3 marks)

Question 2. (16 MARKS)

We continue again with the same program from Question 1:

```
        addi $t0, $s0, 400      # i1
        add $s1, $zero, 0       # i2
L:      lw $t1, 0($s0)          # i3
        add $s1, $s1, $t1       # i4
        addi $s0, $s0, 4        # i5
        slt $t1, $s0, $t0       # i6
        bne $t1, $zero, L       # i7
```

All instructions and word sizes are 32-bits.

We consider first the instruction cache, which is a two-way set associative cache totalling 64 bytes, with a cache block size of 2 words.

a. How many sets would our instruction cache have? (2 marks)

b. If addresses are 32-bits long, how long in bits is the tag portion of the address? (2 marks)

The data cache is a direct-mapped cache totaling 128 bytes with a cache block size of 4 words:

c. How many blocks would our **data** cache have? (2 marks)

d. If addresses are 32-bits long, how long in bits is the tag portion of the address? (2 marks)

e. Assuming that the address of the first instruction (Marked "# i1") is at 0x2F1C4C, what is the hit-rate of our cache for the program given above after it has finished running? State your answer in percent (e.g. 15%), to two decimal places if the answer is not a whole number. (4 marks)

f. Assuming that register $s0 contains the address 0x1CFB2A40, what is the hit rate of our data cache in percent? Express your answer to two decimal places if it is not a whole number. (4 marks)

Question 3. (12 MARKS)

We consider 3 processes in a fixed-priority scheduling system with a single CPU but two independent I/O subsystems. **Reminder: There are no pre-emptions for I/O sub-systems. Additionally, if two processes want to start using the same piece of I/O at the same time, the higher priority process gets to go first.**

The processes have the following priority, start time and execution profiles:

Process	Priority	Start Time	Execution Profile
P0	Low	2	$C3 - IO_2 2 - C2 - IO_1 2 - C1$
P1	High	1	$C2 - IO_1 2 - C2 - IO_1 1 - C2$
P2	Medium	0	$C3 - IO_2 1 - C2 - IO_1 1 - C1$

Here Cn means the process uses the CPU for n cycles, $IO_x y$ means the the process uses IO unit x for y cycles. The "Start Time" of the process is not the time the process starts running on the CPU, but rather the time the process is first added to the ready queue.

a. What is the response time of each process? (This is the time the process waits until the first time it gets to execute) (3 marks)

   P0: _____
   P1: _____
   P2: _____

b. What is the turn-around-time for each process? The turn-around-time is the time between the process's start time and the time it ends. (3 marks)

   P0: _____
   P1: _____
   P2: _____

c. What is the total waiting time of each process? (This is the total time the process spends waiting for the CPU) (3 marks)

P0: _____
P1: _____
P2: _____

d. What is the CPU utilization of the 3 processes in percent (e.g. 10%)? State your answer in two decimal places if it is not a whole number. (3 marks)

Question 4. (5 MARKS)

We have the following three concurrent processes sharing semaphores binary S1, S2 and S3:

<table>
<tr><td colspan="3">Shared Semaphores with Initial Values

Semaphore S1 = 0;
Semaphore S2 = 1
Semaphore S3 = 0</td></tr>
<tr><td>Process P1</td><td>Process P2</td><td>Process P3</td></tr>
<tr><td>P1-1: up(S1);
P1-2: down(S1);
P1-3: up(S2);
P1-4: up(S1);</td><td>P2-1: down(S1);
P2-2: up(S3);
P2-3: up(S1);
P2-4: down(S1);
P2-5: up(S3);</td><td>while(1) {
  P3-1: down(S2);
  P3-2:printf(“Hello world!\n”);
  P3-3: down(S3);
}</td></tr>
</table>

(Note: P1-1, P2-1, etc are just statement labels and can be ignored)

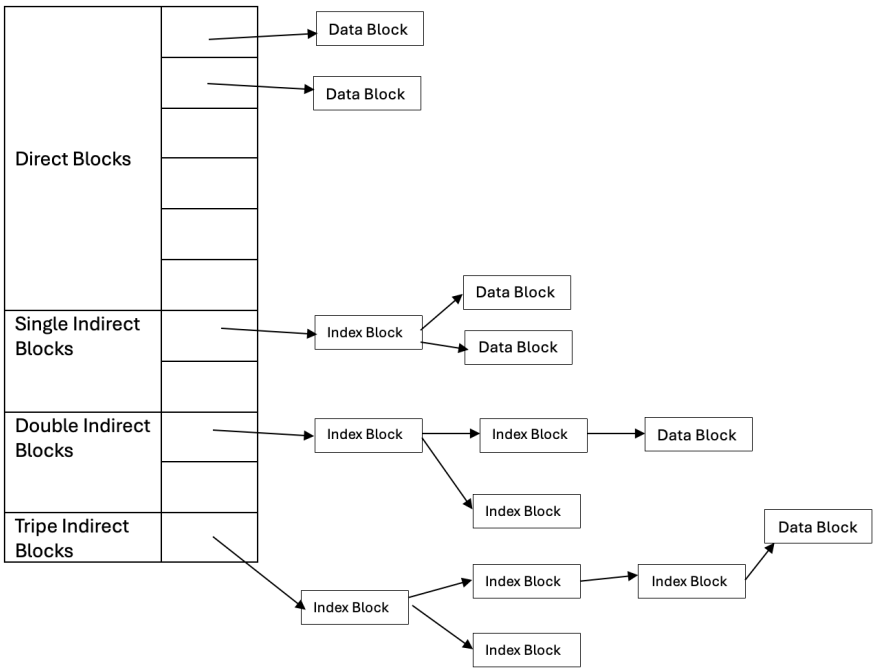
What is the maximum number of times will the phrase “Hello world!” appear when the three processes are run concurrently? Assume that semaphores unblock processes in first-in-first-out (FIFO) order. I.e. if P1 blocks on S, then P2 blocks on S, and P3 does up(S), P1 will always be unblocked first. (4 marks)

Question 5. (8 MARKS)

We consider an inode based file system with the following specifications:

# of direct blocks:	6
# of single-indirect blocks	2
# of double-indirect blocks	2
# of single-indirect blocks	1

This is partially shown below:



All block pointers are 8 bytes long, and data blocks are 4096 bytes long.

a. How many block pointers can each data block hold? (4 marks)

b. Assuming that the answer to part a is 256 file pointers (blocks are still 4096 bytes long), what is the maximum size of a file in this file sytem? Express your answers in gigabytes (1 gigabyte = 2^{30} bytes) to two decimal places if it is not a whole number. (4 marks)

~~~ END OF PAPER ~~~
~~~