QUESTION 2

When referring to a type of computer memory, *Dynamic Memory* is memory that:

- ○ Loses its contents when power is removed from it.
- ○ Loses its contents if it is not periodically "refreshed", i.e. periodically read and re-written.
- ○ Can update the data stored within it with new values whenever new data is written to it.
- ○ Is reserved for use with dynamic data structures like linked lists.

QUESTION 3

A *Memory-Mapped Interface* is:

- ○ Hardware that provides power to computer memory.
- ○ A means by which a library's API is accessed through memory locations.
- ○ A set of parameters for interacting with a program.
- ○ One or more locations in the computer's address space that are mapped to the interface registers of a peripheral.

QUESTION 4

In one sentence, explain the difference between a DCD statement and an EQU statement, e.g.

```
FOO EQU 271
FOO DCD 271
```

Path: p

Words:0

QUESTION 5

The term *Interrupt-Driven Input/Output* ("Interrupt-Driven I/O") means:

- ○ The I/O is done by interrupt handlers that are called in response to interrupt requests.
- ○ The computer generates interrupt requests to signal to the external devices that they should perform I/O.
- ○ The processor continually reads the status of peripheral devices and performs I/O when the status information indicates that the devices are ready.

QUESTION 6

Privileged Instructions are instructions on the ARM7TDMI-S:

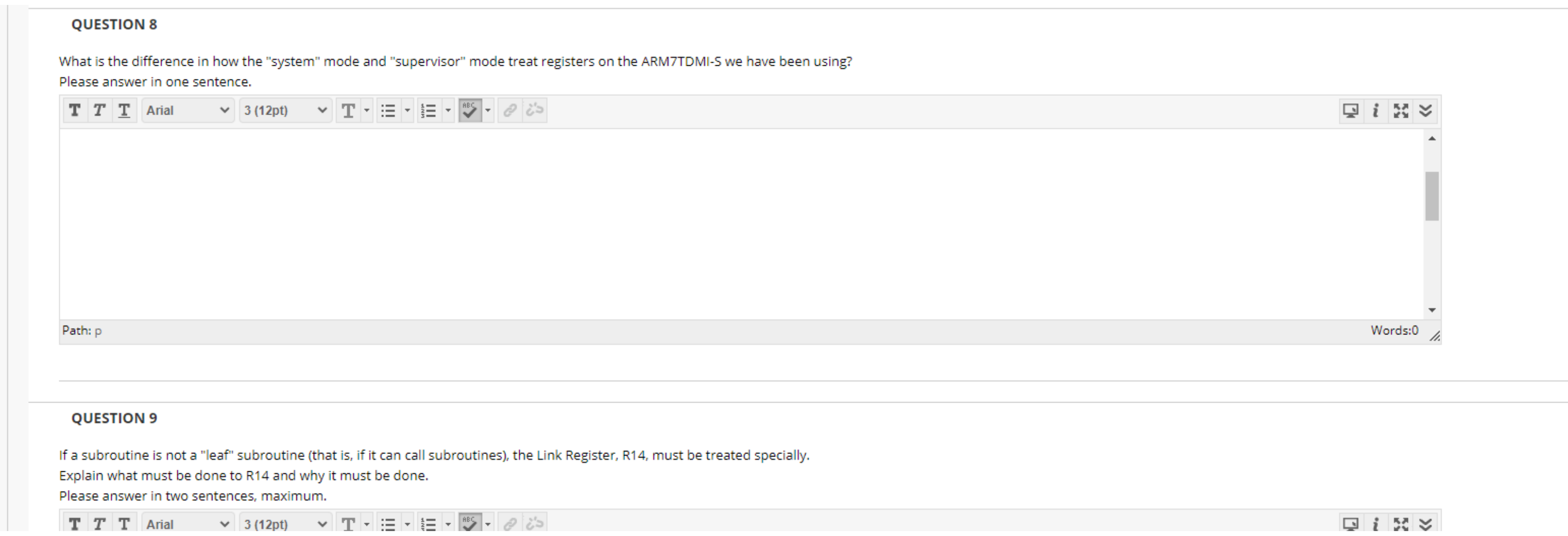
- ○ That modify the CPSR.
- ○ That only work when the system is in User mode.
- ○ That can only be fully executed when the system is not in User mode.
- ○ That promote the system from any of the Interrupt modes to Supervisor mode.

QUESTION 7

Why is using repeated addition to implement multiplication such a bad idea?
Please answer in one sentence.

Arial 3 (12pt)

Path: p Words:0



QUESTION 8

What is the difference in how the "system" mode and "supervisor" mode treat registers on the ARM7TDMI-S we have been using?
Please answer in one sentence.

QUESTION 9

If a subroutine is not a "leaf" subroutine (that is, if it can call subroutines), the Link Register, R14, must be treated specially.
Explain what must be done to R14 and why it must be done.
Please answer in two sentences, maximum.

QUESTION 10

The SWI instruction (also known as the SVC instruction) and the BL instruction are similar in that they both cause a separate piece of code to be executed. But they are also quite different.
List two major differences between the SWI (aka SVC) instruction and the BL instruction.
Please answer in two sentences, maximum.

Arial 3 (12pt)

Path: p Words:0

QUESTION 11

Consider the following sequence of instructions that are executed on an an imaginary non-pipelined CPU with a cache. We assume that writeback caching is enabled.

An uncached instruction normally takes 10 nanoseconds to run. A cached instruction normally takes 1 nanosecond.

Then, if the instruction is a load (LDR) instruction, the instruction will take an extra 10 nanoseconds if the location to be loaded from is uncached, and no extra time if the location is cached.

If the instruction is a memory store (STR) instruction, since writeback caching is enabled, the data has only to be written to the cache, which will take 1 extra nanosecond.

Assume that the each of the following instructions and data items are uncached when first executed or accessed and cached afterwards. Further, assume that r1 contains a valid memory address.

Taking the following sequence of instructions:

```
   mov r0,#2
l1  ldr r2,[r1]
   add r2,#1
   str r2,[r1]
   subs r0,#1
   bne l1
```

How long will this sequence of instruction take to complete?

- ○ 77 nanoseconds.
- ○ 80 nanoseconds
- ○ 150 nanoseconds
- ○ 132 nanoseconds

QUESTION 12

Consider a hypothetical one-byte interface at location 0xFE808018 where four push buttons are connected to bits 7, 6, 5 and 4 respectively. When a button is pressed, the bit it is connected to goes to 0, and when the button is not pressed or when it has been released, the connected bit goes to 1.

Part 1: Write a subroutine that will poll the interface and return when exactly one of the buttons has been pressed, returning the button's bit number (7, 6, 5 or 4) in R0. You may assume that no switch bounce occurs; in other words you may ignore the possibility of "noise" due to switch bounce, which can cause the switch to open and close very rapidly for a short period.

[4 marks]

(This should take no more than about 20 lines of code.)

Part 2: Write a subroutine that will return when exactly two buttons have been pressed, and which returns the button bit numbers in registers R0 and R1, with the higher-numbered button's bit number in R0. For example, if the buttons associated with pins 6 and 4 are pressed, R0 should return with 6 and R1 with 4.

[6 marks]

(This should take no more than about 20 lines of code.)

Arial 3 (12pt)

QUESTION 13

Write a main program and an associated interrupt handler to (a) output a logic 1 for an exact number of milliseconds specified by the value of the label HIGHTIME on a interface pin, then (b) output a logic 0 on the same interface pin for an exact period of milliseconds specified by the label LOWTIME. It should then repeat the process, going back to step (a), then step (b), back to step (a), then step (b), and so on, continuing indefinitely...

The interface pin is Bit 0 at location 0xE0003000.

HIGHTIME and LOWTIME are defined, for example, as follows:

```
HIGHTIME EQU 34
LOWTIME EQU 66
```

Notes:

1. You do not have to set up the timer interrupt -- assume it's been set up for you.
2. Do not write the code to acknowledge the interrupt -- assume that this is handled automatically.
3. Assume the interface is set up, and to write a 1 or a 0, simply write a byte containing the appropriate value in bit 0 to 0xE0003000.

Make sure your interrupt handler is well behaved in the sense that we have discussed in class. Do not reserve any registers for the use of the interrupt handler.
Suggestion: keep the interrupt handler really simple, and put any complexity needed into the main program.
(This should take no more than about 20 lines of code.)

Arial 3 (12pt)