Search COMP2300

RESOURCES
FAQ
Online Labs/Lectures
Software setup
Books, links and other good things
Writing a design document
Project ideas
Exam Collection
Marking Statistics
Getting Jumper Wires
Final Exam 2020
Mid-Semester Exam 2020

RELATED SITES
CS Homepage
GitLab
Streams
Current students

You are about to attempt the multiple choice questions for this exam.

Make sure you read each question carefully as they ask you to make different kinds of choices.

All questions have multiple valid answers.

If you select an incorrect answer you will lose marks (just within that question). So only select answers that you are sure about.

---

1. condition codes

If the `xPSR` register is set so that Z=1, N=0, C=0, V=0, **which** of the following conditional branch commands will result in a **successful branch to the label** `hard_yakka`?

Select one or more of the following options:

- `beq hard_yakka`
- `bal hard_yakka`
- `ble hard_yakka`
- `bla hard_yakka`
- `blt hard_yakka`
- `bhi hard_yakka`
- `bcs hard_yakka`
- `bgt hard_yakka`

---

2. digital logic circuit question

The following digital logic circuit has three switched inputs (A, B, and C) which can be switched on (logical 1), and off (logical 0). The circuit has three output lights (red, yellow, and green), which light up when they receive a logical 1, and are off when they receive a logical 0.

Red = (A and B) nand ((A and B) or (not C)); Yellow = (A and B) or (not C); Green = ((A and B) or (not C)) xor (not C)

Which of the following options are true statements about this circuit? Select one or more.

- When A and B are switched on, only the yellow light is on.
- When all switches are on, the yellow and green lights are on and the red light is off.
- If B and C are switched off, then switching A on or off has no effect.
- If C is switched off, it's impossible for the green light to be on.
- It is possible to have all three lights turned on.
- Switching C on or off has no effect on any light.
- The yellow light will always be turned on.
- When B and C are switched on, the red light will be on.

---

3. interrupts

You're debugging a new discoboard program and have set a breakpoint at the start of one of your interrupt handlers (on the first instruction of the handler, i.e., just after the label). After running the program for a while, the debugger stops at your breakpoint.

Which of the following statements are most correct about the execution state of your discoboard?

Select one or more of the following options:

- The value in `lr` has nothing to do with the value of `pc` before the interrupt was triggered.
- Before this interrupt was triggered, your discoboard may have been executing a different interrupt handler.
- The interrupt connected to your handler has the numerically lowest valued priority of any currently pending interrupt, or interrupt being handled.
- The previous value of `r0` has been saved on the stack by the NVIC.
- Executing `bx lr` will return control to the `main` function of your program.
- The previous value of `r6` has been saved on the stack by the NVIC.
- After returning from an interrupt handler, your program will have to restore `r0`, `r5`, `r2`, and `r3`.
- `lr` contains the address of the code running when the interrupt was triggered.
- Your program was informed by the NVIC that an interrupt was about to occur.

---

4. networks

You're on a team developing a "discoboard to discoboard networking system" called *disconet*. You've been asked to come up with some arguments to justify particular design decisions at the next meeting and have brainstormed the following list.

Which of the following arguments are correct and should be presented at the meeting?

Select one or more of the following options:

- *disconet* should be a serial protocol because keeping one connection synchronised could be simpler and faster.
- *disconet* should use a packet switched network so that we can use data, rather than physical connections to address recipients.
- *disconet* should implement a network layer so that discoboards can pass data without being the original sender or final receiver.
- *disconet* shouldn't be a ring topology because if a discoboard runs out of batteries, the network will be broken.
- *disconet* should use TCP/IP because it implements the OSI standard.
- *disconet* only needs to define the application layer of the OSI standard as all the other layers are only related to hardware design.
- *disconet* shouldn't use a star topology, because there will be no way to determine which discoboard to send a message to.
- *disconet* should be parallel because sending multiple bits simultaneously will always be faster and more practical.
- *disconet* should be circuit-switched because the discoboards are circuit boards.

---

5. OS

Which of the following statements about Operating Systems are **true**?

Select one or more of the following options:

- System calls are used when a user program needs to ask the kernel to perform certain actions.
- Operating systems are normally in charge of allocating memory to each running program.
- Linux uses a hybrid monolithic and modular kernel.
- The Unix operating system uses files to represent data as well as devices.
- The Unix operating system uses a microkernel.
- System calls are used when the kernel needs to ask user programs to perform certain actions.
- First-come first-served scheduling means that the average waiting time for a process will be as low as possible.
- Once a process is running, it will remain in main memory until it has completed execution.

---

6. twiddle

Given that the following lines of code have **just been executed**:

```
ldr r0, =0xCAFEDB0O
ldr r1, =0x0
ldr r2, =0x1
ldr r3, =0xFF
```

Which of the following lines of code leaves a number in `r5` with bit 6 **set** (considering the bits to be zero-indexed, so bit 0 is the least-significant bit in `r5`).

Select one or more of the following options:

- `ror r5, r0, #6`
- `sub r5, r1, r2`
- `orr r5, r0, r2, lsl #6`
- `mov r5, r3,lsl #3`
- `mvn r5, r3`
- `orn r5, r0, r3`
- `adds r5, r0, r2`
- `and r5, r1, r0, lsl #6`

---

7. number of instructions

You're leading a team designing a new CPU instruction set architecture (ISA). One team member has suggested an ISA consisting of **2864 instructions** but another colleague has designed an ISA with only **178 instructions**.

At a recent meeting, both colleagues tried to convince you that their design was better and, together, they have given you a list of arguments (see below).

Which of the arguments are valid and should be taken into account when finalising your CPU's ISA? Select one or more of the following arguments:

- Having more instructions is better as it will allow programmers to optimise code for special situations, resulting in faster programs.
- Having fewer instructions is better as it means programmers will have to use simpler instructions that are faster to execute.
- Having more instructions is worse because op-codes will have to be longer, and having longer instructions means that programs take up more memory.
- Having fewer instructions is worse because programmers will have to use more instructions to accomplish the same task, so programs will take up more memory.
- Having fewer instructions is better because it will take the CPU less time to search for each instruction, so execution will be faster.
- Having more instructions is worse because it will make high-level programming languages more complex for programmers.
- Having more instructions is better because the CPU will be able to express more algorithms.
- Having fewer instructions is worse because transistors on the CPU will be wasted.
- The number of instructions needs to be close to a power of two (e.g., 256) matching the bus-width of the CPU, so only the smaller ISA will work.

---

8. pipeline hazard question

You're working on a development team for a new discoboard with a pipelined CPU and come across the following function:

```
down_under:
    ldr r0, [sp]
    ldr r7, [sp, 4]
    push {r4-r12, lr}
    mov r4, 32
L1:
    bl plunder
    cmp r0, r4
    bge L2
    bl thunder
    b L1
L2:
    ldr r5, =vegemite_sandwich
    ldr r2, [r5]
    add r4, r0, r0
    add r1, r1, r0
    sub r3, r2, r1
    bl take_cover
    mov r0, r3
    pop {r4-r12, lr}
    str r0, [sp]
    str r1, [sp, 4]
    bx lr
```

Right after the function, you can see some notes made in comments by another developer. **Which of these comments is correct?** (You can assume your team members have followed correct calling convention).

Select one or more of the following comments:

- The line `bge L2` presents a control hazard.
- The lines `add r1, r1, r0` and `sub r3, r2, r1` are a data hazard.
- The line `ldr r2, [r5]` can be moved down in the function to resolve a data hazard.
- The line `bl take_cover` presents a data hazard.
- The line `bl thunder` is a structural hazard.
- Loading data from the stack (`ldr r0, [sp]`) is a data hazard.
- The line `ldr r2, [r5]` can be moved up in the function to resolve a data hazard.
- The line `cmp r0, r4` is a structural hazard as `r4` could have been changed by the function `plunder`.

You are about to attempt the code questions for this exam.

Make sure you read the question carefully so that you understand what to do.

You will not lose marks for minor syntax errors.

Remember to include comments to explain your code. Clear and concise answers are preferable. Unclear code may lose marks.

---

9. Even or not in memory

Write down an ARM assembly function starting at the label `even_or_not` that performs the following operation on a 32-bit parameter *x* which is found in memory at location `0x19000000`.

If *x* is even, then return $2 * x$; Otherwise, return $2 + x$.

The output value should be written to the same memory location where *x* was found.

---

10. Iterate over array

You're part of the team designing a discoboard-powered calculator which can store 32 32-bit signed integers at a time in an array (the array is stored in packed format at 4-byte boundaries).

Given that the first entry in the array is stored in statically-allocated memory at the label `discovector`, **write a function** to calculate the average (rounded down to the nearest integer) of the 32 numbers in the array.

Add a comment to explain where the return value is stored at the end of the function call.

---

11. fibonotchi series

The consider the following definition for a series of numbers:

$F_0 = 2,$

$F_1 = 3,$

$$F_n = F_{n-1} + F_{n-2}$$

Write a recursive function, starting from the label `series_n`, which takes one argument, n, and calculates the number F_n.

---

12. networks sync async

Serial connections can be **synchronous**, where a clock line connects both ends of the connection, or **asynchronous**, where there is no clock line.

Explain **why** a clock line is useful for transmitting data between two computers and **how** an asynchronous serial connection can work without a clock line. Explain the **advantages** and **disadvantages** of both approaches.

---

13. OS question privilege levels

The ARMv7-M architecture supports two privilege levels for code execution: `privileged` and `unprivileged`.

Regular code can run at either level, while interrupt handlers always run in the `privileged` level.

What are the **purposes** of these privilege levels? **Explain how** they would be useful for implementing an operating system.

---

14. why ldrex strex

A programmer working on software for a multi-tasking operating system is editing the following function called `strike_one` which is designed to subtract one from a value held in memory:

```
strike_one:
    ldr r2, =taskcount
    ldrex r3, [r2]
    sub r3, r3, #1 # subtract one
    strex r4, r3, [r2]
    cmp r4, 0
    bne strike_one
    bx lr
```

The programmer has used the `ldrex` and `strex` instructions. **Explain why** the programmer has used these instructions. **What situations** might make these instructions necessary and **how** these instructions help.

---

15. Flynn Taxonomy

Flynn's taxonomy of CPU architectures has four categories:

```
SISD (single instruction stream, single data stream),
MISD (multiple instruction stream, single data stream),
SIMD (single instruction stream, multiple data stream), and
MIMD (multiple instruction stream, multiple data stream).
```

Explain each category and discuss how such a CPU would be useful (or not useful) to a programmer using examples.

You are about to attempt the hurdle question.

This question is intended to test your understanding of a topic from COMP2300.

There are two questions below (Q16 and Q17) and you can choose which one you would like to answer.

Only answer one question!

Type "chose not to answer" in the answer field for the question you choose not to answer.

This question is worth 20 marks so you should spend around 30 minutes on this question and write a substantial answer (while remembering to be clear and precise).

Copy/paste is disabled, so write your answer directly into the text box. If you change to another question page your answer will be saved.

(Students received individual, randomized hurdle questions)