Quiz 1: Solutions

EEE4084F

2015-03-05

Question 1: Processing Architectures **[20 Total]**

#1 Any 4 of:

- Improved hyper-threading
- Faster front-side bus clock
- Better branch prediction
- Out-of-order execution
- Instruction-level parallelism
- Larger cache
- Cleaver cache that pre-empts fetches and writes with better prediction

[4]

#2 Each of the four options should have one pro and one con (8 marks). The other two marks are for clarity, relevance to embedded systems, small explanations, examples, more pro's and con's, etc. Pro's (left) and con's (right) include (among others):

ASIC:

- Fast
- Fully customisable during design
- Inexpensive when mass-produced
- Heavily optimised for the task to be performed
- Good for parallelisable applications

- Expensive to prototype
- Long development cycles
- Fixed-function
- Difficult to design
- Static (when a bug is discovered, the IC must be re-manufactured)

FPGA:

- Reconfigurable on both hardware and software level
- Fully customisable hardware (and software)
- Relatively inexpensive prototyping
- Heavily optimised for the task to be performed
- Good for parallelisable applications

- Little cost gain when mass-producing the final product
- Slow clock-rate (compared to ASIC and processor)
- Challenging to develop code for
- Often over-kill for the target application

Microcontroller:

- Very inexpensive
- Low power consumption
- Convenient peripherals come on-chip
- Good for small embedded application (such as garage-door openers)
- Easy to develop code for
- Easy to use

- Low performance
- Single-thread serial execution
- Small and light-weight applications only

Processor:

- Multiple very powerful cores
- Good library support
- Good operating system support
- Good for multitasking

- Limited number of cores
- Large power consumption
- Inefficient for course-grained parallel applications

[10]

#3 Local execution on the CPU does not involve costly memory transfers over the PCI-Express bus. A CPU generally contains a small number of powerful cores and is good for execution of a small number of vastly different tasks (multitasking operating system applications, for instance). It is inefficient for course-grained parallel tasks due to the limited number of threads.

Coprocessor execution involves costly memory transfers between the host (CPU) and the coprocessor memory. It generally contains a large number of light-weight cores, which are inefficient at executing serial-like code. It is therefore suited mostly for course-grained parallel tasks with little cross-thread communication. It also leaves the CPU free to do other things while it is busy, which is often useful.

The phrase “small is beautiful” refers to the idea of using many small cores, rather than a few powerful ones. The idea is that one can obtain many more instructions per second (and instructions per unit of power) by using many small cores instead of few large cores. This approach is not well suited to all tasks, however. Not all applications can be parallelised. The coprocessor follows the ”small is beautiful” approach, whereas the general-purpose CPU does not.

[6]

Question 2: Memory Architectures **[15 Total]**

#1 Shared memory architectures allow easier and faster inter-process communication, easier memory coherence and easier source distribution (as only a single copy is required). Problems include serial memory access, where multiple processes fight over memory bandwidth limitations (this is partially solved by caching), and the need for mutually exclusive locks when modifying shared memory areas, effectively blocking the other threads from execution.

Distributed memory requires expensive inter-process communication schemes, there is no automatic coherence between copies of the data and multiple copies of source data is required. Advantages include isolation (each process works independently from the others and therefore does not need to wait for them or share memory bandwidth), and there is no need for mutually exclusive locks. **[5]**

#2 Memory locks are used to ensure memory validity when multiple processes can potentially read and modify the same memory location at the same time. Mutually exclusive locks ensure that only one thread can enter a so-called critical region and modify that memory location.

Potential dangers include dead-lock, especially where more than one lock is used. Thread A could, for instance, lock memory location X, while thread B locks memory location Y. If thread B is then dependent on also locking memory location X, and thread A is also dependent on locking memory location Y, both threads wait for each other. Careful thought must therefore be given to locking schemes and how mutually exclusive locks are used. **[5]**

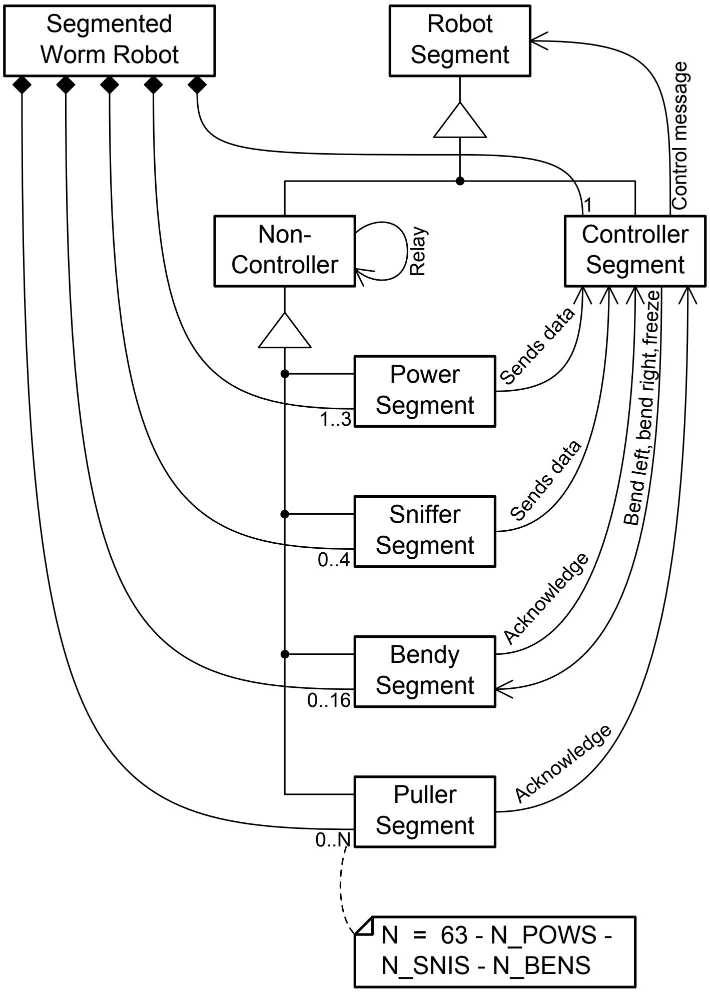
#3 The memory wall refers to a limitation in memory bandwith and latency, in conjunction with many processor cores accessing the memory over the same bus. The processors can process data much faster than the memory can provide (or store) it.

Caching schemes often include a caching hierarchy. Each processor IC has an on-chip cache that attempts to reduce memory latency by storing data that will most likely be used next in memory that is physically closer to the processor cores. This cache often has higher bandwidth as well.

The next level in the hierarchy involves a separate cache for each core within the processor. This attempts to mimic a distributed memory architecture, so that each core can access the memory concurrently, without waiting for neighbouring cores to finish using the memory. Cleverly implemented caching schemes also perform prediction, in which data is pre-fetched while the processing core is busy processing other data. This greatly reduce memory latency and bandwidth issues. **[5]**

Question 3: UML **[15 Total]**

#1 **[12]**



#2 The whole system runs off the same voltage, so we only need to work with current. The total current drawn by the given configuration is be given by:

POWS:		50 mA
BENS	:	2× 200 mA
PULS	:	2× 500 mA
CONS:		100 mA
SNIS	:	100 mA
Total	:	1 650 mA

The power supply provides 2 Ah. If the device needs to last 2 hours, it may not draw more than 1 A. The given configuration will therefore not last 2 hours. **[3]**