SoC Chapter 4: System Design

Modern SoC Design on Arm
End-of-Chapter Exercises

Q1 Speedup

If an accelerator multiplies the performance of one quarter of a task by a factor of four, what is the overall speedup?

Q2 Queuing Theory

The server for a queue has a deterministic response time of 1 μs. If arrivals are random and the server is loaded to 70% utilisation, what is the average time spent waiting in the queue?

Q3 Queuing Theory — Two Queues

If the server is still loaded to 70% but now has two queues, with one being served in preference to the other, and 10% of the traffic is in the high-priority queue, how much faster is the higher-priority work served than the previous design where it shared its queue with all forms of traffic?

Q4 Energy Saving

If a switched-on region of logic has an average static to dynamic power use of 1 to 4 and a clock gating can save 85% of the dynamic power, discuss whether there is a further benefit to power gating.

Q5 Debug Trace

What is the minimum information that needs to be stored in a processor trace buffer to capture all aspects of the behaviour of a program model given that the machine code image is also available?

Q6 Fault Redundancy

A 100-kbit SRAM mitigates against a manufacturing fault using redundancy. Compute the percentage overhead for a specific design approach of your own choosing. Assuming at most one fault per die, which may or may not lie in an SRAM region, how do the advantages of your approach vary according to the percentage of the die that is an SRAM protected in this way?

Q7 High-level Energy Modelling

Assuming an embarrassingly parallel problem, in which all data can be held close to the processing element that operates on it, use Pollack's rule and other equations to derive a formula for approximate total power and energy use with a varying number of cores and various clock frequencies within a given silicon area.

Q8 Static Scheduling

Consider a succession of matrix multiplications, as performed by *convolutional neural networks (CNNs)* and similar applications in which the output of one stage is the input to the next. Is FIFO storage needed between stages and if so, could a region of scratchpad RAM be sensibly used or would it be better to have

a full hardware FIFO buffer?