SoC Chapter 6: Design Exploration

Modern SoC Design on Arm
End-of-Chapter Exercises

Q1 Product Design

Consider the design of a high-quality digital movie camera. Sketch a feasible top-level block diagram, remembering to include a viewfinder and audio subsystem, but you may ignore auto-focusing. How many circuit boards, SoCs and processors should it have? To mitigate against the camera shaking, what are the relative costs of implementing a vision stabiliser using voice-coil prism hardware compared with an electronic/software-only implementation.

Q2 Memory Bandwidth

An algorithm performs a task that is essentially the same as completing a jigsaw puzzle. Input values and output results are to be held in DRAM. Describe input and output data formats that might be suitable for a jigsaw with a plain image but no mating edge that can falsely mate with the wrong edge. The input data set is approximately 1 Gbyte. By considering how many DRAM row activations and data transfers are needed, estimate how fast this problem can be solved by a uniprocessor, a multi-core PRAM model and a hardware accelerator. State any assumptions.

Q3 Parallel Processing

Two processes that run largely independently occasionally have to access a stateless function that is best implemented using about 1 mm^2 of silicon. Two instances could be put down or one instance could be shared. What considerations affect whether sharing or replication is best? If shared, what sharing mechanisms might be appropriate and what would they look like at the hardware level?

Q4 Custom Instructions/Accelerators

Consider the following kernel, which tallies the set bit count in each word. Such bit-level operations are inefficient using general-purpose CPU instruction sets. If hardware support is to be added, what might be the best way of making the functionality available to low-level software?

```
for (int xx=0; xx<1024; xx++)
{
   unsigned int d = Data[xx];
   int count = 0;
   while (d > 0) { if (d&1) count ++;    d >>= 1; }
   if (!xx || count > maxcount) { maxcount = count; where = xx; }
}
```

Q5 Flow Control

Data loss can be avoided during a transfer between adjacent synchronous components by using a bi-directional handshake or performance guarantees. Explain these principles. What would be needed for such components to be imported into an IP-XACT-based system integrator tool if the tool allows easy interconnection but can also ensure that connections are always lossless?

Q6 Pipelined FUs

What is a fully pipelined component? What is the principal problem with an RTL logic synthesizer automatically instantiating pipelined ALUs? A fully pipelined multiplier has a latency of three clock cycles. What is its throughput in terms of multiplications per clock cycle?

Q7 Modulo Scheduling

A bus carries data values, one per clock cycle, forming a sequence $X(t) = X_t$ as illustrated in Figure 1. Also shown is a circuit that supposedly computes a value Y_{t-3}. It uses two adders that have a pipeline latency of 2 and an initiation interval of 1. The circuit was designed to compute the running sum of bus values. Check that it does this or else design an equivalent circuit that works but uses the same adder components.

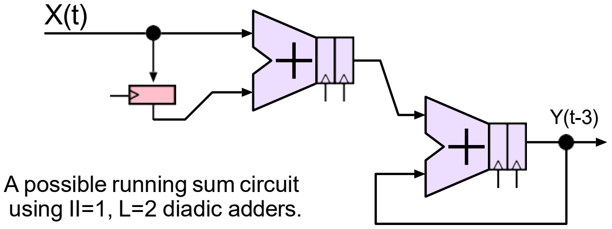
X(t)
Y(t-3)
A possible running sum circuit
using II=1, L=2 diadic adders.

Figure 1: A circuit intended to compute the running sum of streaming data values.

Q8 Operator Identities

Does *strength reduction* help save area, energy or both? Give an expression that can benefit from three different strength reduction rules.

Q9 Cut-through and Deadlock

Is a NoC that uses store-and-forward elements with cut-through routing a multi-access NoC? Does multi-access lead to more or fewer chances of deadlocking?

Q10 Loop-carried dependencies.

Does either of the following two loops have dependencies or anti-dependencies between iterations? How can they be parallelised [48]?

```
loop1:  for (i=0; i<N; i++) A[i] := (A[i] + A[N-1-i])/2
loop2:  for (i=0; i<N; i++)  A[2*i] = A[i] + 0.5f;
```