EXTENDED MODEL ANSWER

Comparative Architectures 2001 Paper 7 Question 3 (IAP)

Instruction set architecture

Since this is an embedded device there is unlikely to be any need to be "PC compatible", or even instruction set compatible with other similar devices.

Since the device will have to deal with audio/video data, having integer SIMD extensions (similar to Intel's MMX) is likely to prove useful.

None of the above requirements appear to necessitate an FPU or MMU. However, the latter may be useful for debugging and fault isolation.

The device is likely to spend long periods of time in "standby mode". This suggests that architectural support for HALTing the processor until the next interrupt is important.

Micro architecture

Complex dynamic-execution schemes are likely to be too power hungry, and will consume too much chip area.

Simple statically-scheduled super-scalar may be worthwhile to offer improved performance over simple pipelined designed. Care can be taken to ensure that code in critical loops is optimised appropriately, even if the compiler is not normally up to the job.

It may even be good to go for a multi-threaded processor design, hiding memory accesses by interleaving two or more threads. There may be enough processes running on the device to exploit this parallelism.

Physical implementation

A System-On-a-Chip (SOC) implementation would be ideal here. Integrating as much as possible on the same chip would cut down the power required for inter-chip signalling. At the very least, all caches should be integrated onto the processor die.

Using as modern a CMOS process as is economically viable would be beneficial. Smaller transistors generally require lower voltage and are correspondingly more efficient. Static leakage may be higher though. Furthermore, a smaller process would enable integration of more components onto the SOC.

A system that enabled the processor clock frequency to be varied according to application work load would be beneficial, particularly if the core voltage could be reduced as the clock frequency was reduced. This could potentially provide quadratic power savings compared to the linear saving of just halting the CPU during idle times.

Benchmarking

Standard benchmarks like SPEC CPU2000, MIPS, whetstones etc. are pretty meaningless for such a device. The workload is just too different.

The best solution is probably to code up the CPU intensive applications and run them in a system simulator (or existing hardware system if one is available). Although the simulation may take many weeks to run it will provide a good degree of confidence.

An alternative approach is to try and identify the critical loops of these applications and simulate the execution of just these sections of code.