

Panel: How to Choose Semiconductor IP? - Embedded Processors

Ian Phillips. Principal Staff Engineer, ARM Ltd, UK.

It is well recognised that the process of new-product development, introduction and marketing is fraught with difficulty. Indeed the probability of achieving plan timescale, costs and budget are so low, that some degree of failure is inevitable. So whilst the primary role of the Manager is to identify and minimise all major risks, and make sure the ones remaining are adequately resourced: A secondary role is to make sure that what failure does occur, does not damage his/her reputation!

The Virtual Component appears in the context of risk minimisation. CPU or UART, the motive is the same, get the right product from concept to customer as quickly as possible. The make or buy decision is a risk/cost trade-off, and as the cost of failure is normally so high, risk emerges as the dominant factor; Is it riskier to design, or buy-in?

Alas assessing risk is an imprecise science where the scope and nature of the product is evolving rapidly, and so great weight is attached to existence theorem. If somebody has used *it* before in a situation similar to yours, then the *risk* of it misbehaving in your *life-cycle and product* is low(er); minimised risk, but not zero! And risk is inversely proportional to Quality.

Traditionally this wisdom is captured in management-mantras, such as "You never get sacked for choosing IBM, Microsoft, ... and ARM". It is an encapsulation of risk in a way readily understood. But the career-limiting possibilities of alternative action is also on-show, and tend to give such mantras a bad reputation! It should be remembered that the main reason the technology is *preferred* is because of their implicit and/or explicit quality; Thereby, improving the probability of *success* of the *product*!

We are talking product life-cycle here! It is important to remember that only when product hits the shelves, and people start buying them, that funds flow back down to sustain the entire food-chain. Success is much more than just doing your bit correctly! It is also making sure that at every stage in the product introduction process, where 'your bit' has influence, that it does so *uneventfully*. For a simple

UART, its influence extends before physical, into the modelling, and after physical in the verification and documentation phases. For a more complicated device like a CPU (Embedded Processor), which transcends the boundary between the software and hardware worlds; It gains a whole new dimension, with its own modelling, verification and documentation aspects, as well as new consistency aspects. So whilst selecting and deploying an Embedded Processor is no different in concept than for a UART; The context in which it is used is much broader, and the consequential tool inter-play and interactions much wider.

I already noted that risk is inversely proportional to Quality; But Quality is also a *retrospective* measure of Suitability for Purpose. When you buy a VC, you are doing it because you want a *capability*, to produce a product; not just a few polygons!

... The capability you get with a UART is easy to comprehend. The documentation and models you need to describe its behaviour, minimal. The tools to exploit it, no more than those for hardware design.

... The physical manifestation of ARM1020 is ~6.5M transistors; But the simpler and less powerful ARM7 at ~50k, is not much more complicated than a sophisticated UART?! *The power of the CPU, is what it can be, not what it is!* The ARM1020 can be a sophisticated PDA/Communicator or a Tricorder; The ARM6, a Cell-Phone or a Printer. But their possibilities will *only be realised* if the models, tools, development support, OSs, documentation, verification, validation, reliability, availability (etc) ...*Quality*... exceeds requirements, when the need is encountered!

ARM's strength comes from its world-wide partnerships and inter-related products across the domain of deployment ... bound by a standard architecture and use model ... ARM/AMBA.

So choosing an Embedded Processor is not that difficult! Establish the value of your products success, then choose an appropriately scaled *Capability* to support its development...

The rest, as they say, are history!