

ARM's Top Tips for IP providers

- **Don't try to do everything.** Automate whenever possible. Subcontract or partner where possible, even with competitors if necessary, because trust based relationships are important.
- **Work with the ecosystem.** Understand the value chain and find out how your success will help others. Look for the win-win. Build links (for example, between products or between tools). Connect people: exploit the 'network effect'.
- **Understand your customer's customer.** Understand their attitude to risk: are they visionary early adopters? Their success will reflect on you, but they may be reckless. Pragmatic customers may be risk averse.
- **Help others get to where they are going.** Seed what you have to, charge where you can. Be seen as a leader and ensure your place in the ecosystem is evident.
- **Look to the future.** Don't expect tomorrow to look after itself. Winners have a road map and invest in the future, so don't be a one trick wonder.

Practical advice for IP users

- Make sure you get access to the knowledge and technical data to integrate it efficiently. Check that the right models and other verification data and tools are available, either from the vendor or through the ecosystem.
- Don't even think about modifying third party IP. The impact on project schedules of modifying IP is often badly underestimated. The benefits of reuse (productivity gains, easier verification, lower cost, faster time to market) may well be negated.
- But the temptation to tinker is strong and may be necessary – especially for previously used IP – to gain a competitive edge. It is important to get the right alchemy between silicon proven IP, that can be easily reused unchanged, and differentiated or tweaked IP that may need more qualification and verification effort. Be prepared to take responsibility if you change things.
- Formal verification tools can be useful in some tasks, such as checking for protocol compliance, but are not necessary on many standards based IP blocks.

Embedded systems challenges

- Software productivity gap, particularly for functions such as software radio, where each waveform might require 75k lines of code.
- Multicore processors may solve thermal and performance problems, but programming principles need to exploit parallelism within the model driven development environment.

- Legacy software needs to be supported within a stable ecosystem and may need to be resynchronised to new hardware.
- Will need to deal with less reliable components due to process variability and greater susceptibility to soft errors.