

October 6, 2011

Time carefully the adoption of your next-generation processor

Can you maximize first-to-market opportunities and minimize the extra risks of early adoption?

Robert Cravotta, Embedded Insights Inc -- EDN, October 6, 2011

At A Glance

- Suppliers use major releases to signal when new capabilities are built on top of proven products.
- A development team that adopts a new product or major update may experience the pain, cost, and risk of changes to their project schedules and engineering resources.
- Early adopters gain the largest first-to-market-window opportunity, but they may expend more engineering resources than developers who adopt the processor later.
- Working directly with an FAE (field-application engineer) usually provides the most reliable, accurate, and timely information about a new processor.
- Community-support forums provide avenues for finding timely information about using a new processor.



One of the first expressions I learned as a young embedded-system developer was “Smart money avoids Version 1.0.” Back then, there was little expectation that even a compiler would be available to developers soon after the availability of a new processor. Working with a new processor or development tool meant your own development team would experience some of the growing pains of determining what worked properly and what work-arounds were necessary to get the system to do what it needed to do.

Suppliers use major releases of available products to signal to users that the new features and capabilities are built on top of a proven product line; this helps mitigate some of the challenges of launching

completely new and unproven products. Suppliers often mark a major release of their product lines by increasing the major version number, such as to 2.0, or by launching a new family name within the product line. Major releases or new-product families can span and employ multiple already-available product lines. For example, Texas Instruments' recent launch of the Hercules safety-microcontroller family spans the TMS570 and RM4x product lines. TI based the new microcontrollers on proven processor families but packages new or different peripherals and features in a way that is appropriate to a vertical market—in this case, safety applications.

Today's newest processors are significantly more complicated than their predecessors. They may include new peripherals, application-specific accelerators, and additional processor cores that may not share the same architecture as the main processor core. New features can mean that the developer community may have little or no direct experience with the parts of the system and how to best use the new devices' features. To sufficiently minimize developers' learning curves for using the new processor and encapsulate the complexity of these systems, the supplier must provide appropriate development tools, drivers, boards, and support IP (intellectual property) when the company launches the processor.

Unfortunately, the wisdom contained in the smart-money expression also applies to major releases of products that have been in the market as production systems because customers have not yet robustly applied the new features and capabilities they contain to an exhaustive set of real-world use cases. A development team that adopts a new product or major update may experience the pain, cost, and risk of changes to their project schedules and engineering resources. As a result, some development teams

enforce a policy of avoiding using the latest one or two versions of a component in their designs.

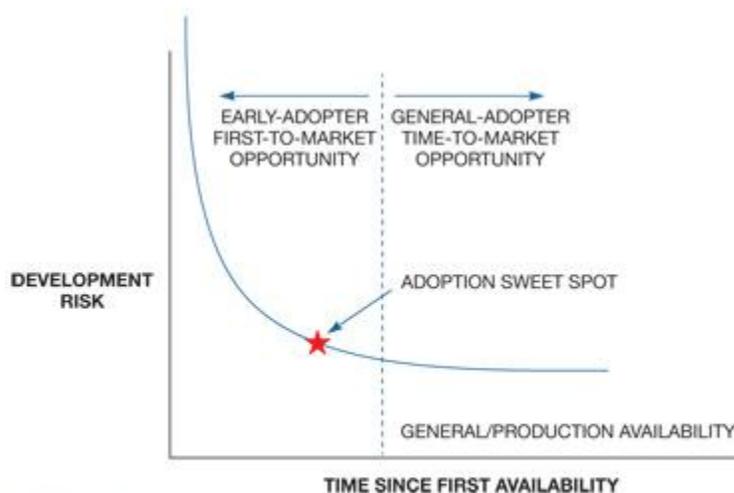


Figure 1 The adoption sweet spot provides the most opportunity for a first-to-market design and incurs the lowest additional development risk (courtesy Embedded Insights).

The intent of this type of policy is not to avoid using the new capabilities that the latest version supports but to avoid the risk of having to rework portions of a design because of changes in how to access and manage a new capability as the system undergoes final engineering tweaks and adjustments to deliver optimum performance and manufacturing yields before full production availability. This type of policy relies on the supplier's version nomenclature and production-

availability date to signal when the volatility of the system is stable enough that additional engineering tweaks are unlikely to cause rework by the development team. Although this policy is simple to follow, it can potentially cause a development team to miss out on substantial first-to-market benefits by adopting a new processor or tool months after the adoption sweet spot—that point in a new

processor's or development tool's early life cycle that provides the most opportunity for a design to reach the market first but incurring the lowest additional risks to the development effort (**Figure 1**).

Adoption sweet spot

The first users of a new processor, the early adopters, gain the largest first-to-market-window opportunity, but they may expend more engineering resources than developers who adopt the processor after it has reached general or production availability months later. As a processor proceeds through the normal life cycle after first availability, the number and severity of risks that users of the new processor are subject to decrease (see [sidebar](#) "Sources of risk").

General adopters adopt a new processor after the silicon and development tools have matured sufficiently and after the formation of a pool of people who have experience working with the new processor. General adopters expect that a stable set of development tools, IP blocks or software, and expert support staff is available to support the new processor so that they can focus their engineering resources on completing their design in the shortest time without necessarily requiring heroic engineering efforts outside their field of expertise. In other words, accessing the new features is fairly straightforward and well-defined, such as using available drivers, libraries, or API (application-programming interfaces).

Two previous hands-on projects illustrate some of the possible barriers between expectations and reality for early adopters ([references 1](#) and [2](#)). Both projects involved a recently launched system that comprised an ARM core and a DSP core within the same device. Some of the challenges included learning how to configure and use the system and the software at the same time that the engineering-support people were learning it, using drivers that were "hot off the press" to meet the project deadline, and having only one choice for the host-workstation configuration because the alternative configurations would not become available for several more months.

A key lesson from both of these projects is that early adopters must be aware that the ability to get to market before your competitors requires risk-taking in working with components that might be undergoing build and test cycles; building components in-house to meet schedule constraints, even though those components will be readily available to later adopters; or both of these scenarios. The payoff is that early adopters may be able to release their products with a new and important differentiating capability many months before their competitors.

Early adopters rarely shoulder all of the development risks, however ([Reference 3](#)). Rather, the processor vendor often enters into a strategic alliance with the early adopter that involves early access to the silicon, software, tools, engineering team, and dedicated FAEs (field-application engineers). Obvious examples of early adopters include tool providers that will support the new processor with their tools as well as strategic-partner companies that will be the first primary users of the new processor. The strategic-partner company is often key in specifying and testing the processor so that it best meets the requirements of the targeted application.

Find the knee

It is not feasible for everyone to enter an early-adopter partnership with a processor vendor. So, must smaller design houses resort to adopting a processor after it has reached general availability, or can they find the “knee,” or sweet spot, on the early-adopter side of the risk-and-adoption curve that gives them a competitive advantage? A number of engineering and marketing resources, such as FAEs, data sheets, and application notes, are available from the processor vendor to help the public engineering community determine when they should adopt a processor. It is important, however, to realize that each resource has a different timeliness and reliability, depending on the company’s procedures for capturing and disseminating engineering changes (**Figure 2**).

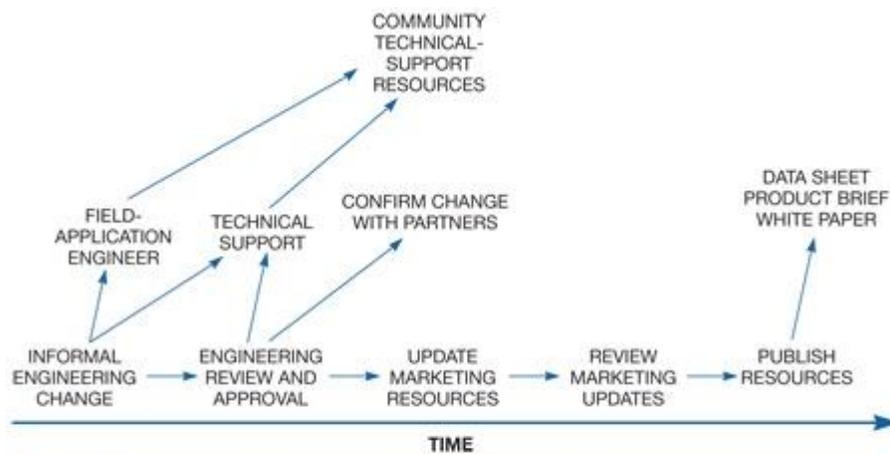


Figure 2 Engineering changes pass through a number of groups and cause a lag between when a change is made and when it is incorporated into the various information channels open to developers (courtesy Embedded Insights).

A substantial time interval usually occurs between when someone first identifies a bug and when an informal fix is available; more time elapses during the approval and implementation of a formal fix, the updating of the documentation, the release of the fix, the publication of the updated documents, and the awareness of interested developers that something about the new processor has changed.

Working directly with an FAE usually provides the most reliable, accurate, and timely information about a new processor because the FAE can obtain informal information directly from the engineering group and bypass the delays of days to weeks that can be involved in waiting for the results of engineering-change-review boards. The FAE provides the expert judgment to know when to share an engineering change that the vendor is considering but has not yet formally adopted.

Community-support forums provide avenues for finding timely information about using a new processor. Forums typically connect developer peers with each other and engineers from the processor

supplier. In some cases, the original developers of the processor may participate in the forums. Active forums can provide a place for developers to ask questions and to read or participate in conversations regarding how to address unexpected issues when operating in a scenario with a processor. Forums provide appropriate means of finding approved-formal-fix information; because they are static and persistent resources, however, they are unlikely to be good resources for finding informal-fix information. Examples of developer community forums include Atmel's AVR Freaks and Texas Instruments' e2e forums. Offering a community forum requires active management to ensure that the material in the forum is relevant and accurate. Without active monitoring, the "signal-to-noise ratio" can suffer from spam posts as well as posts that are incorrect or even misleading—intentionally or otherwise.

A knowledge base—a repository of relevant documents to disseminate information to the developer community—can complement the technical-support group. It may contain troubleshooting information, application notes, articles, white papers, data sheets, errata, bug lists, sample programs, user manuals, hardware-support documents, downloadable software, archived software, and answers to frequently asked questions. Running a knowledge base does not require the same level of active management that a forum needs because it is less interactive than a forum.

Participants sometimes use IRC (Internet-relay-chat) channels to engage in active conversations in active developer communities, such as at BeagleBoard. These channels provide a faster way to ask peers questions and engage in back-and-forth conversations; however, users of these channels should check the reliability of any information they see with their FAEs and suppliers because these conversations lack full moderation.

Many companies offer an e-mail service to notify developers whenever there is an update to the documentation pertaining to a processor family or a development tool. This service provides developers with reliable information, but relying on e-mail notification can mean that you find out about the latest information weeks later than you could have through other resources. E-mail notification is ideal for those developers who are interested in knowing what changes are happening but are not looking to immediately make a selection.

Adopting a processor too early can cause you to accept more risk than you are prepared to handle. Likewise, adopting a processor past the sweet spot means that you may miss out on a first-to-market opportunity. Hitting the adoption sweet spot for a new processor requires diligence and active effort. Numerous resources keep track of that effort, which together can provide you with a sense of where a processor is on the risk-and-adoption curve.

References

1. Cravotta, Robert, "[Forge ahead?](#)" *EDN*, March 6, 2003, pg 51.
2. Cravotta, Robert, "[Embedded-processing platforms: a second look,](#)" *EDN*, Nov 9, 2006, pg 54.

3. Cravotta, Robert, “Welcome to the jungle: early adoption in embedded development,” *EDN*, Oct 30, 2003, pg 38.

Author’s biography

Robert Cravotta is principal analyst and co-founder of Embedded Insights. Previously, he worked as a technical editor at EDN, covering embedded processors. He also worked in aerospace on electronics and controls for pathfinding projects, such as fully autonomous vehicles, spacecraft and aircraft power-management systems, Space Shuttle payloads, and building-automation systems. Cravotta received a master’s degree in engineering management from California State University—Northridge and a bachelor’s degree in computer-science engineering from the University of California—Los Angeles.

For More Information

ARM	Atmel	BeagleBoard	Intel
LinuxWorks	Pentek	Texas Instruments	Xilinx

Sources of risk

As contemporary processor offerings pack more features and on-chip resources into single packages, it becomes increasingly more difficult and time-consuming for a developer to learn and understand how all of the parts of the entire system operate together. An effective way to improve the productivity of developers is to use operating systems, BIOS, drivers, software libraries, development kits, and feature-rich software-development-tool suites. However, these productivity tools take time—typically, a few months—to build and test against the actual processor when it first becomes available.

Early adopters might need to work with the processor before any of these productivity tools target the processor core they are using. One work-around for this problem is to use a simulator or an available processor that is similar to the new one; the corresponding risk is that the new processor implements some features in a subtly different way that will require the early adopter to rework a portion of the system to accommodate those differences once they are identified.

Development-tool providers are obvious candidates for early adoption of new processors because they supply the tools that general adopters must rely on. Development-tool providers support multiple versions of their tools for each processor target because their customer base does not automatically adopt the latest version of the tools. Moving to the latest revision may involve gaining access to a new capability that earlier versions do not support. In any case, a business-case scenario justifies the risk of moving to the latest version. Early-adopter issues may differ in the details, but they are similar whether you are an FPGA-, a board-level-, or a software-tool provider.

Rodger Hosking, vice president of Pentek, shares his company's experience with supporting new FPGAs. The FPGA-development tools may undergo modification because of hardware changes for as much as a year of production because the manufacturer is still "cooking" the silicon. During this time, the hardware vendor is still tweaking the system IP (intellectual property) to improve the performance of blocks in the system. Developers considering the purchase of a new FPGA frequently ask whether the vendor has tested it at a certain speed and what tools are available to support that level of performance. The answers to these questions are relevant because new silicon sometimes does not immediately support every function at the expected top clock rate. The details of the answers to these two questions provide prospective adopters with a sense of the maturity, stability, and engineering effort they will need to bring to a project if they adopt that FPGA at that moment.

Jason Kridner, a BeagleBoard expert at Texas Instruments, weighs in from the perspective of supporting a board-level resource for the BeagleBoard community. Important areas affecting the stability of a new board include interconnect and interprocessor communication. It takes weeks for the system to become stable after a change in ISA (instruction-set architecture) or in specialized processing engines in the target processor that affect the timing of the system. The support software, including the supplied kernel, drivers, and operating system, encapsulates such changes. Examples of such changes in the software could be to enable a driver to support a specific DMA (direct-memory-access) mode or to activate a vector processor. Another type of change that can occur and causes rework and delay is to substitute a component on the board with a similar part, such as a memory chip; the substitution can cause a ripple of changes in the system that can take a few weeks for testing and deployment.

Robert Day, vice president of marketing at LynuxWorks, sheds light on a different set of risks facing software-tool providers. It's easier to provide early support for some processors, such as those from Intel, because the embedded variants employ the PC or the server architectures that precede by several months the availability of the embedded parts. In contrast, a significantly longer lag occurs between when the IP core for ARM processors debuts and when actual silicon is first available. The development tools for the embedded processors come from a base that focuses on embedded-system constraints.

However, providing software-development tools for embedded processors involves validating what features the end device includes. This issue is important because each instantiation of an ARM processor by each licensee can expose dissimilarity between expected and missing features. An example with an ARM core involves a Cortex-A8 port that does not implement ARM's TrustZone technology, a systemwide approach to security. An example with an Intel processor involves an Atom core that lacks virtualization support. An instantiation of a processor core sometimes lacks all of the possible specified features.

http://www.edn.com/article/519497-Time_carefully_the_adoption_of_your_next_generation_processor.php