OBLIQUE PERSPECTIVE

The Irrelevance of Being Earnest

Good Intentions Pave the Road to Microprocessor Perdition

By John Wharton, Applications Research

It's springtime in Silicon Valley. The robins are back, there's new growth on the trees, and the technical conference season is in full swing.

At least half a dozen new microprocessors have been announced so far at ISSCC, CompCon, and assorted events thereafter. They include 32- and 64-bit RISC and CISC architectures, have conventional, superpipelined, or superscalar implementations, can run anywhere from 25 to 200 MHz, and thus deliver radically different levels of throughput. The *only* thing these parts have in common is that the people promoting each are convinced they'll soon revolutionize the computer business.

The proponents of every new microprocessor seem absolutely certain their chip will demolish its competition, and their arguments are usually convincing. After a two-hour briefing by product designers, it's hard *not* to go away thinking *this* time they got it right. Yet no matter how earnest the people involved, sales for new parts almost never achieve their developers' expectations.

Sometimes the gap between expectations and reality is absurd. An Intel marketeer bet me $100 that within 12 months of its introduction, the i860 would displace the x86 as the top-selling processor for desktop PCs. Instead, the i860 sold so poorly Intel abandoned desktop RISCs entirely. (I'm still waiting to collect on the bet.)

How can rational people harbor so clear a delusion? Partly it's myopic pride; new parents always see their kids as the most beautiful babies ever. Partly it's the subconscious mechanism by which people justify unsavory acts of the past. Salesmen *have* to believe that the products they sell are the best. Partly it's the tendency for engineers to feel most comfortable with the parts they know best. And partly it's because most engineers have a grossly disproportionate view of how technical nits affect a product's success, ignoring the importance of timing, marketing, and political issues.

But there's a lot more to it than that. Psychological blinders alone can't explain the absolute conviction of the architects and designers who promote each new part, or the utter bewilderment they feel when product sales fall flat. I blame the profound mismatch between vendor expectations and reality on a flaw in the process by which most microprocessors get designed.

Fundamental Flaws in the System

When I was growing up, each of the houses my family lived in had been designed by my father. He hired trained architects to formalize his blueprints and check building-code compliance, and contractors to handle construction, but in all the world there was no better expert than my dad on what he himself wanted in a home. And the houses he designed fit our needs perfectly, far better than anything else we could buy.

I think a similar phenomenon affects chip design. Once a company decides to build a new part, its design goes through many stages. An architecture is defined or selected, pipelines are planned, and cache sizes chosen, followed by logic design, floorplanning, layout, and so on. Each stage of the process is fairly well understood, and the work of each stage can be checked against earlier stages, so the resulting parts usually do pretty much what their designers envisioned.

So the designers feel justifiably smug about achieving their goals. Just as my father's houses exactly matched his needs, each new microprocessor exactly matches product requirements established years before. How could buyers not be impressed? How could they *not* beat a path to the vendor's door? But then why do the products still fail?

The weakest link of the design chain lies in product definition. This is the least rigorous step of the process, and seldom receives the attention it deserves. Target specifications have variously begun as one individual's notion of a chip he wants to build (the i860), as the groupthink consensus of a design committee (the 80186), to win a sale at one key customer (the 8096), or because a junior engineer wandered into a planning meeting in order to get a free lunch (the 8051).

Unfortunately, product definition is also the one design stage on which all others depend, and the one whose validity can not be tested until everything else is done. If the wrong target was picked, or the wrong goals were set, the product will die. The die was cast (quite literally) before the design process even began.

The Myth of Price and Performance

If there's one common failure in product definition, I feel, it's that most new microprocessors aspire to excel in some simple, easily measured way — to be the fastest, cheapest, or lowest-powered part on the market, for example, depending on perceived market needs. On the surface that makes sense, since competing processors often *are* compared solely on price and performance.

Overly simplistic objectives, though, produce overly simplistic designs. A house designed with no windows or closets may well minimize cost or maximize floor space, but would likely not sell even so. And those few application areas that *do* choose a processor based solely on price, performance, or power should probably be avoided anyway. Their profit potential is low, and sales may vanish entirely once a cheaper, faster, or lower-powered part comes along.

In the real world, price and performance are just two of the factors that affect processor selection. Equally important are CPU integration level and package types, the cost and complexity of support circuitry, whether commodity memories may be used, statistical vs. deterministic performance, vendor reputation, how many alternate sources exist, product-line upgrade and downgrade paths, mil-spec qualification, power and cooling requirements, ease of programming, compatibility with existing home-grown and third-party software, hardware and software development tools, hooks for real-time debugging, and so on, to say nothing of office politics, vendor relationships, and how well the company founder likes Intel.

Instead of a two-dimensional price-performance graph, then, it might be better to think of real-world selection criteria as a complex, multi-dimensional hyperspace in which each processor feature is evaluated along a different, mutually orthogonal axis. The ideal CPU for a given application would occupy a different, n-dimensional point in space, depending on the requirements and importance of each factor. Alas, no two customers are likely even to define the same evaluation space, much less agree on the target.

Microprocessors designed solely to excel in one area too often do poorly in others. The faster the clock, for example, the worse the power and cooling demands. The wider the buses, the more expensive the package and external glue. The more single-minded the design, as a rule, the tinier the market niche it fits. At the things it was good at, the i860 was superb. As a general-purpose computer, though, it left much to be desired.

Strategies for Success

The microprocessor selection process is often one of elimination. Hundreds of microprocessors exist from which system designers can choose. Some are too expensive for a particular budget, and must be ruled out. Others may be rejected for failing to deliver adequate performance, or for requiring too much power, or because they have inadequate tools. Once you've eliminated the unsuitable options, Sherlock Holmes might say, the solution must be found among what's left. The best part for a design need not be the cheapest, fastest, or easiest to design with, but neither can it be fatally flawed on any count.

So what's a microprocessor vendor to do, to improve its odds of success? One ploy is to build on the popularity of an established product line. The 8051 was essentially a new, faster 8048, with more registers, instructions, ports, memory, and peripherals. The Intel P5 is the sixth-generation upgrade of the 8008, which began as a Datapoint terminal controller.

On certain occasions it does make sense to target a a single, well-defined configuration — preferably one you control. When IBM, HP, or DEC design a new CPU for a specific system product, the resulting performance can be very good indeed. Thus the RS/6000, Snakes, and Alpha each leapfrogged the competition of the day. In all the world there's no better expert than DEC, for example, on what DEC values most in a micro.

But can the mainstream workstation market afford Alpha? Not in its current form. Would cost-competitive, lower-power versions of Alpha outperform today's SPARC and MIPS workstations? I don't think so.

More widespread success in the market requires more careful planning up front. It's not enough just to understand market needs, you must anticipate how they'll change by the time a design reaches production. And while it's good to excel at the things you do well, you dare not ignore the things you do poorly.

But no matter how well you plan, market needs keep changing. The fastest way to respond to new markets is then to reposition existing parts. The i960 and 29000 families initially targeted workstation-class systems. By introduction, though, each had been repositioned for embedded control.

Or, you can subset existing parts for a new market. The i960 K-series includes three different products, all with the same fully-tested die. The 29000 and 29005 sell equivalent die in different market segments, as do the 376 and 386SX, and the 486SX, 487SX, and 486DX. By staking out different regions of the price/capability continuum, customers can be made to feel they're paying only for the capabilities they need.

Summary

It takes more than good intentions to make a product succeed in the market. Meet all the real needs of all of your customers, and you stand a good chance of success. Leave too many holes in the spec, though, and customers will buy something else. If the wrong goals were set, the product will die, and the most earnest explanations in the world won't help. ♦