VIEWPOINT

Mercy, Mercy, Merced

Schedule Slips? Performance Problems? It's the Same Old Story.

by Nick Tredennick & Brion Shimamoto

Despite Merced's recent tapeout, we continue to hear that the project's performance is falling below target, and that the official mid-2000 schedule for system shipments is unlikely to be met. No one should be surprised, however. Unrealistic targets are endemic to a highly visible project managed by executives disconnected from the engineers doing the real work.

Leading-Edge Projects: the Theory

Here's how leading-edge microprocessor projects are planned.

At month zero, executives and engineers meet to negotiate features, goals, and schedules. One input to this negotiation is the current competitive state of the industry. Another input is Moore's Law.

Moore's Law says that if they design a microprocessor on, say, a 36-month schedule, its performance has to be four times that of today's new microprocessors to be competitive. At their month-zero meeting, they plot their first performance target on a graph of performance versus time. Their performance target sits on the Moore's Law curve.

In theory, the engineers work 36 months to build a microprocessor that will satisfy Moore's Law when the part ships. While the engineers work on the implementation, the executives brief both each other and the company's customers on the new part.

But as someone said, "In theory, there's no difference between theory and practice. In practice, there is."

Leading-Edge Projects: the Practice

The competition might deliver surprisingly good performance, better than Moore's Law. The executives raise the performance target to allow for that. Also, projects can slip. If a project slips too much, it pushes the performance target below Moore's curve. If this happens, there's no market for the part, so the executives raise the performance target to allow for that, too. In practice, a project's target starts out looking like that in Figure 1.

At the beginning of a project to design a leading-edge microprocessor, the engineers are enthusiastic and confident. They sign up for a lofty performance target and for an aggressive schedule. But then the engineers begin the refinement process that leads to the real design.

Almost from the beginning, the actual performance of the design diverges from what is being "pitched." This happens because, as engineers decide on implementation details, they compromise performance to solve problems affordably. Figure 2 illustrates this situation.

Over time, the gap between "PowerPoint" Performance (what executives are telling each other and their customers) and Actual Performance (what the engineers can realistically achieve) grows. When the gap is too wide, there is turmoil in the project. Schedules and performance are renegotiated. As Figure 3 shows, a second target is set below and to the right of the first target (though possibly not as low as Actual Performance). The schedule slip and the new performance goals are announced, and the project begins a new phase.

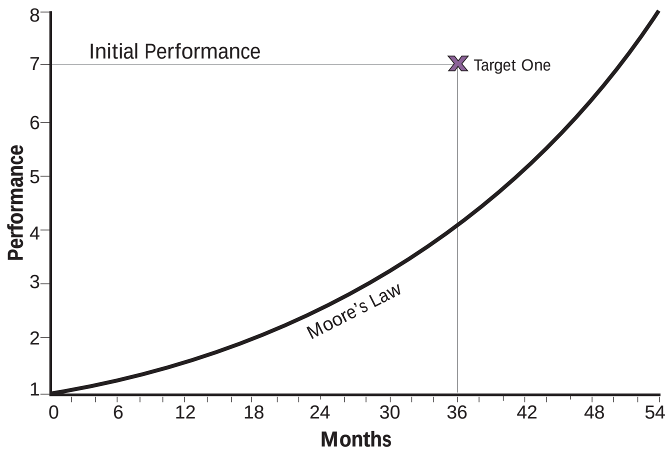

Figure 1. In the beginning of a microprocessor design project, estimated performance often exceeds Moore's Law projections.

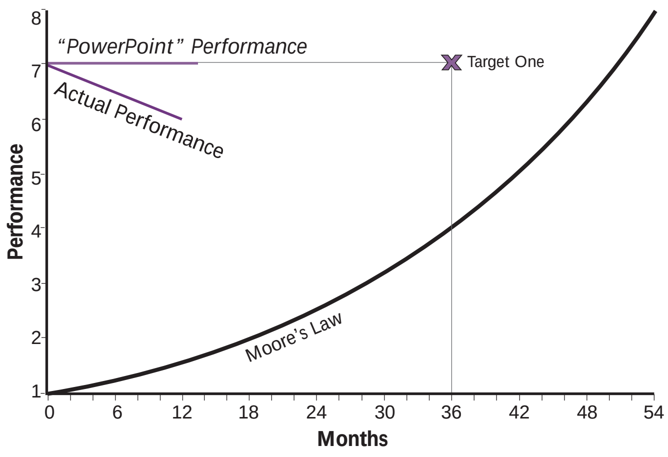

Figure 2. Later in the project, Actual Performance diverges from PowerPoint Performance.

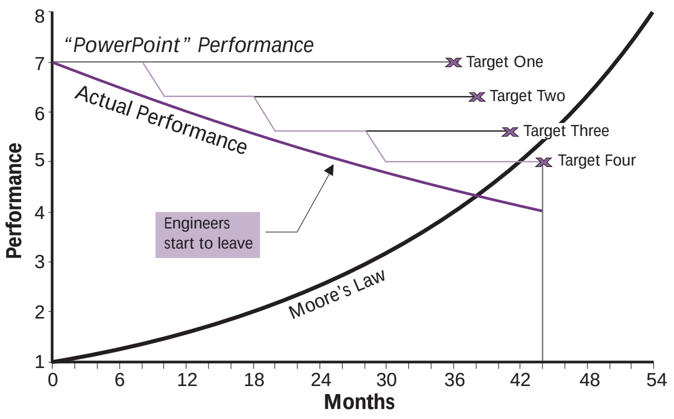

Figure 3. Bad news. As Actual Performance continues to decline, new targets must be set.

Engineers begin leaving the project once it is obvious to them that the Actual Performance will fall below Moore's Law before project completion.

The executives are the last to know that it's time to kill the project. This is because executives track PowerPoint Performance while engineers track Actual Performance. The executives do not become aware of missing the third target until the widening gap between the PowerPoint Performance and the Actual Performance precipitates the next round of performance negotiations. At this negotiation, the executives see that the fourth target will fall below Moore's curve. It might seem an obvious choice to kill the project once the performance target falls below Moore's curve, but this isn't necessarily so. Dealing with the consequences of Figure 3 is one of the core skills of an ambitious engineering manager.

Leading-Edge Projects: the Next Project

There is no fundamental relationship between the length of the design cycle and the need for new product introductions. A complex microprocessor design might take 36 months, but competitive pressures might require new product introductions every 24 months. The analog in microprocessor design is the difference between latency and throughput. The design cycle represents latency and the product introduction cycle represents throughput. To achieve higher throughput, a company must pipeline product design.

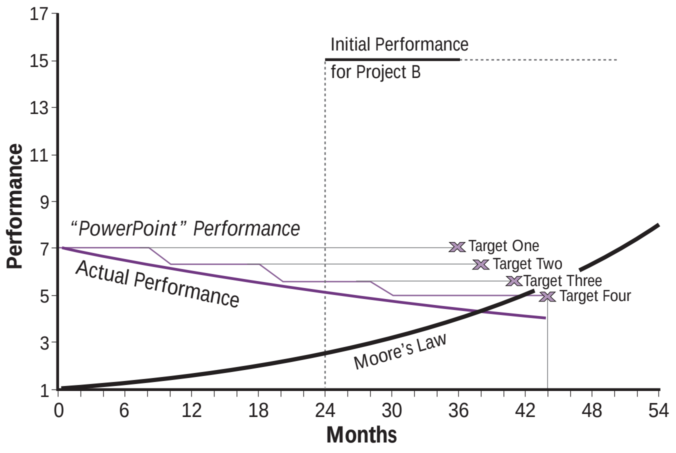

Figure 4. A follow-on project may have optimistic projections compared with the Actual Performance of the current project.

So, two years after the original project (Project A) began, the company initiates Project B. The wily managers for Project B try to set their goals based on an extrapolation of Moore's Law from Target One. This makes Project A look bad, and it gives Project B more leeway. At this time, the original project may be on its way to missing Target Three. Project B hasn't slipped at all yet, and it has a performance target that makes Target Three look mundane. In its early stages, a project can tolerate a much wider gap between its PowerPoint Performance and its Actual Performance than it can in its later stages. This is because, as a design matures, its performance is locked in by the many details that are set.

Leading-Edge Projects: Merced?

We're only guessing (we have no inside information), but Merced seems to be a good example of this process. When the Merced project started, it must have had lofty performance goals. Achieving them has been an uphill battle.

Merced suffers from the Illiac Syndrome. Illiac was a research triumph. (A research triumph is easier to achieve than it may seem, because the researchers who do the work also do most of the reporting.) But Illiac was a commercial failure. So was Intel's iAPX432. So was MicroUnity. They all had goals that pushed too many frontiers (e.g., hardware, instruction set, software, compilers, complexity, design tools, and semiconductor process) at once. Producing x86-compatible products on the competitive treadmill (Moore's Law) is hard enough.

Intel could use Merced for the x86 market—and probably expected to. Here's the problem, however. When the Merced project started, it must have had an x86 performance target roughly twice the anticipated performance of the then-current IA-32 (x86) implementations. This would provide an incentive to convert to Merced from IA-32. (An "upgrade" should outperform the processor it replaces.)

Then the Merced project began slipping, while the IA-32 implementations forged ahead, spurred on by Intel and by Intel's competitors. As it stands, Willamette will arrive at the same time Merced is projected to reach the market. Willamette will be a native IA-32 processor running at 1.2 GHz, while Merced will run at 800 MHz or so. Willamette will be faster on x86 code. Willamette may even be faster running IA-32 code than Merced will be running IA-64 code. In other words, Merced's Actual Performance has sunk below Moore's Law for x86.

The IA-64 follow-on to Merced is McKinley. McKinley should be substantially faster than Merced and will be out only about a year later. Merced cannot hope to compete with contemporary versions of IA-32. And it will lead the market in native performance for only a short time—and probably at a performance disadvantage relative to x86 implementations. Merced's situation is shown in Figure 4, where Project B is McKinley, and Moore's Law (representing the x86 treadmill) sits above Merced's current performance target. Given this situation, what's Intel to do?

It might seem sensible to cancel the Merced project, but that's unlikely. At least one of the following reasons is sure to be compelling enough to ensure Merced's survival.

- Intel promised to deliver Merced and it's long overdue.
- Intel's developers need Merced (at least to develop software for McKinley).
- The sooner Intel ships Merced, the sooner Intel gets an objective evaluation.
- The strategies and fortunes of Intel's corporate partners depend on Merced's success.
- Intel can't learn all the possible lessons from Merced (and IA-64) unless it ships.
- Intel's engineers have killed themselves to get this far. If Intel c ancels Merced now, most will go elsewhere.
- If Intel doesn't build Merced, it can't generate revenue.
- Intel would rather be known for late delivery than for canceling projects (tardy is better than unreliable).
- Intel's hardware development teams will be more efficient with native machines.
- Intel needs real hardware to get real measurements.

Therefore we expect Intel will complete the Merced design and bring it to market. Canceling the project now would be a public-relations mess, given the number of public commitments Intel has made. But we do not expect end users to line up for Merced. Intel and the IA-64 system vendors will put on a happy face when discussing Merced. If buyers ask about performance, Intel will talk about McKinley. This positioning will probably work—at least until McKinley's engineers figure out the Actual Performance of their design. M

Nick Tredennick and Brion Shimamoto took turns managing the Micro/370 microprocessor project at IBM Research. They have worked with various startup companies since leaving IBM. You can reach them both at bozo@computer.org.