Microprocessor design, on the massive scale on which Intel does it, is like an epic ocean voyage of discovery. The captain and his officers select the crew carefully, so as to represent in depth all the requisite skills. They pick the types, sizes, and numbers of ships. They stock the ships with whatever provisions they think they will need, and then they launch into the unknown. With a seasoned crew, good weather, and a fair amount of luck, most of the voyage will go as expected. But luck is capricious, and the crew must inevitably weather the storms, when everything goes awry at once and they must use all their experience to keep the ships afloat.

Being involved in a major microprocessor design project like the P6 feels exactly like that except for the passage of time. Most ocean voyages do not take the 4+ years we invested in the P6.

From the beginning, my publisher wanted to title this book “The Pentium Chronicles.” I was initially skeptical. When I hear “chronicle,” I immediately think of a history, and I am no historian. But the second definition of chronicle, “narrative,” fits exactly with what I envisioned for the book—a series of events as told by a P6 insider. So, reassured by this definition and the publisher’s prediction that Pentium Chronicles would have the majestic draw of Lord of the Rings, I acquiesced to the title (as long as Orlando Bloom plays me in the movie version).

WHAT YOU WILL FIND HERE

The substance of this book, its heart blood, are the events that taught me something about the craft of computer design in the context of a large company. Most of the narrative centers on events surrounding the P6 project, with some devoted to the Pentium 4 project as well. I had several reasons for focusing on P6, but the main one is that the project is far enough in the past that I can confidently avoid its intellectual property issues. The Pentium 4 project is still too recent; I cannot yet disentangle the general engineering lessons we learned there from the specific product context.

You will find real names in this book, but not in every instance. My policy on real names is simple: I refuse to purposely show any specific engineering colleague in a negative light. My intention is to use real incidents and events to illuminate the general engineering methods; it doesn’t matter who made a particular mistake. What matters are the conditions that led up to a particular design or project error, and how someone might prevent a similar occurrence in future designs. We all make mistakes—that’s one of the book’s major themes. Our designs have to work flawlessly despite us.

But I will use real names in other cases. One major reason is to avoid giving the impression that I was solely responsible for the outstanding engineering being described.
The P6 project had many more than its fair share of brilliant engineers, and credit should go where it’s due.

Parts of this book may come across as my personal treatise on how to do technical project management. It is not my intent to compete with the mountain of “how to do project management” tomes in existence. I learned more from my talented and dedicated design manager partner, Randy Steck, than I did from the dozen or so project management books I read along the way. My intent is simply to describe what we did in the P6 project and how it worked out—the bad along with the good.

I believe that in any well-intentioned, well-run engineering endeavor, choices will be made among multiple plausible options, some of which will turn out to be brilliant, others not. We should always learn from our mistakes, but if we are striving for excellence, we should also consider the triumphs and tribulations of others. I sincerely hope that by exploring lessons from the P6 and Pentium 4 projects, this book provides a way to do that.

HOW THE BOOK IS ORGANIZED

I wrestled with the question of how best to present these lessons. I could simply describe what happened in the project and let the reader draw inferences as needed. But then I would be guilty of breaking an important rule I learned as an engineering manager: Never let an engineer get away with simply presenting the data. Always insist that he or she lead off with the conclusion to which the data led. The engineer’s job in that presentation is to convince you that he reached the right conclusion. If he or she cannot draw a conclusion, or will not, then either the work or the engineer lacks the maturity to justify making the presentation at all.

I could also write a book that consists only of the lessons learned, with anecdotes provided to buttress the case where necessary. The trouble with that approach is that it might become boring, pedantic, quite possibly pretentious, and would push the book into the already overpopulated camp of how to manage design projects.

I decided to seek a middle ground, one that relies on actual events for its authenticity and immediacy, but is heavily seasoned with what I personally found important and educational about the events’ outcomes. My aim is to make that narrative compelling, and if it falls short in places, then at least it will be entertaining.

Readers who know me, who have heard me deliver technical talks, or who have read my columns in Computer, the IEEE Computer Society’s flagship magazine, know that I hold lots of opinions on pretty much everything, including other projects and product lines within Intel, other microprocessor design firms, the future of the industry, and so on. You won’t find me expounding on those topics at great length within these pages. Read the column and go to the talks. A book stretches my plausible deniability safety net just a little too far.

Having arrived at my middle ground, I was then faced with how to build on it. The most natural way was to present the P6 project chronologically, so the book is mostly organized in that sequence, although it is hardly linear. P6 was a big project that started with one person (me), grew over several years to 450+ people, and then eventually decreased to zero staff. In between were waves of concurrent activity. For the first year or two, the project was a handful of chip architects conceiving and documenting a general approach that would realize the project’s overall goals. Meanwhile, a design team was being recruited, organized, and trained so that it would be ready to hit the ground running when the architects were ready. Two validation teams, presilicon and postsilicon, were also recruiting, training, and writing tests for the day when they were needed. Other groups were
trying to anticipate the requisite manufacturing production challenges and what could be
done to ameliorate them. Marketing was preparing their campaign strategies, marketing
literature, and promotional materials, and setting up customer visits to validate the archi-
tects’ design feature choices and to inform potential customers of what lay ahead on In-
tel’s product roadmap. Try plotting all that on a single line.

At the P6 project outset, we were charged with designing a new microprocessor for In-
tel. But before we could start thinking about how transistors could be organized, we had
to consider how the project would be organized. P6’s first architecture manager, Fred Pol-
lack, and P6’s design manager, Randy Steck, came up with an initial division of labor and
made judgment calls about overall team sizes and effort allocations. It’s also clear in ret-
rospect that Fred, in particular, was a masterful corporate diplomat, keeping the existing
corporate antibodies from killing or damaging our project, back when we were a bunch of
unknowns proposing odd plans and slinging scary-sounding phrases like “out-of-order
microarchitectures” and “glueless multiprocessing.” Even early on, Fred gave us the free-
dom to roam, and trusted us to come back with something worth keeping.

The basic steps that the P6 project would have to accomplish were reasonably clear:
conceive a microarchitecture, do the detailed chip design, debug first silicon, and drive
the chip into production. You can map these steps informally into microarchitecture con-
ception, writing the structured register-transfer logic (SRTL) software model, and debug-
ging early silicon. But looking at the project with the benefit of hindsight, I believe it’s
more useful to view it as four phases: concept, refinement, realization, and production.
Consequently, these four phases constitute most of the book and can serve as a kind of
framework for any large chip-development project.

Naturally, not every lesson stems directly from the P6 project. Any undertaking is in-
fluenced by its environment, and the P6 project was no exception. The book is filled with
lessons about how teams, corporate policies, and even location can affect a project’s ef-
ectiveness.

As a final chapter, I couldn’t resist including some of the more interesting questions
that I’ve been asked over the years.

FOR THE RECORD

I dreaded writing the following section. I knew I couldn’t do an adequate job of acknowl-
edging the indispensable contributions of hundreds of people, and only a few words to
those I explicitly name isn’t nearly enough. I cannot do justice to all who contributed to
the microprocessor developments outlined here. Just listing all the names would make the
book twice as long and still would not be sufficient recognition for the dedication, creativ-
ity, enthusiasm, and unstoppable élan of a great design team.

But it would be an even greater injustice to say nothing. So with considerable humility
and trepidation, but profound gratitude, I offer the following thanks.

ACKNOWLEDGMENTS

The P6 project was an effort of many people, and, on a smaller scale, so was this book. In
a quasirandom order, here is a list of people I must single out for thanks.

Richard Calderwood, for being an outstanding patent attorney, an astute reviewer of
partially baked manuscripts, and a reliable friend.
Dave Papworth, Glenn Hinton, Michael Fetterman, and Andy Glew—with you guys on my team, I’d do it all again.

Wen-mei Hwu, for your encouragement, your perspective, your unselfishness, and your foreword to this book.

Readers of my IEEE Computer Magazine “At Random” column, for your interest and support. You convinced me there might be a book hiding somewhere in my memories of the P6 project. This book is all your fault.

Dadi Perlmutter, Mike Fister, Will Swope, and Pat Gelsinger, for the opportunities you gave me and the errors you forgave. In your own way, each of you gave me the maneuvering room to try new things and the support I needed when a few of them went awry.

Lew Paceley, Jerry Braun, Richard Dracott, and John Hyde—by your examples, you showed me where my biases were regarding marketing, and the lessons you taught me were among the most important that I learned at Intel. Frank Binns, thanks for proving that marketing people can do real technical work, and for having a twisted sense of humor—balm for tough days. Terry Nefcy, Adrian Carbine, Gary Brown, Jim Hunt, Tom Marchok, John Barton, Donald Parker, Keshavan Tiruvallur (“K7”), Bob Bentley, Ticky Thakkar, Rani Borkar, and Steve Pawlowski—thanks for helping make Intel a great place at which to work. Katie Abela-Gale, thanks for the candy and the smiles.

Dawn Kulesa, Human Resources representative extraordinaire—for me, you were the human face of what sometimes seemed a cold, impersonal corporate machine. Your initiatives on training and mentoring at Intel were exemplary and helped the team enormously.

Mary Kay Havens and Mary Killeen—it wasn’t your job, but you showed me the ropes of managing a large group and routinely went well beyond the call of duty. You can be on my design team any day.

Nancy Talbert, this book’s editor—thanks for your innumerable improvements to this manuscript. You helped turn a pile of ideas into something approaching coherency.

Joseph Malingowski, for your constant mentoring as I was growing up. It becomes clearer to me with each passing year how uniquely valuable your tutelage in technology really was. The details of how things work, I learned in college and industry; how to organize and think about them, I learned from you.

Scott Hamilton and Judi Prow, Computer Magazine’s dynamic duo: I wouldn’t have written this book if I hadn’t been writing the “At Random” column, and I wouldn’t have done that column without your help and encouragement.

Randy Steck, my partner in this enterprise—nobody does it better.

My parents, Bob and Agnes Colwell. I still aspire to the standards that you set, and thank you for that invaluable legacy.

Ellen Colwell, my wife, sounding board, editor-in-chief, and best friend. She wasn’t on the payroll, but in a very real sense she was also a member of the P6 team. I couldn’t have done that project, nor this book, without her unwavering support. Thanks also to my children Kelly, Ken, and Kristen, who grew up during the P6 and Willamette years and often wondered where Daddy was.

THE JOY OF PURPOSE

I realized early on that it would not be easy to capture the nuances of a project of this magnitude. P6 was a landmark design project and microarchitecture, both for Intel and for the industry. I felt that way during and after the project, and dozens of other project engi-
neers have told me they felt the same. For those very reasons, I felt compelled to record at least some of the vast experience that went into P6 development—insights and vision that might otherwise be lost.

George Bernard Shaw wrote, “This is the true joy in life, the being used for a purpose recognized by yourself as a mighty one . . . the being a force of nature instead of a feverish, selfish little clod of ailments and grievances complaining that the world will not devote itself to making you happy.”

Here’s to mighty purposes and vicarious educations.