

The End of The Intel Age

by

Robert Swope Fleming

B.S.E. Electrical Engineering (2001)

Princeton University

Submitted to the System Design and Management Program
in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Engineering and Management

at the

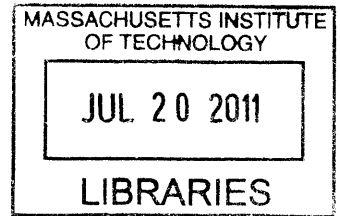
Massachusetts Institute of Technology

May 2011

[June 2011]

© 2011 Robert Swope Fleming

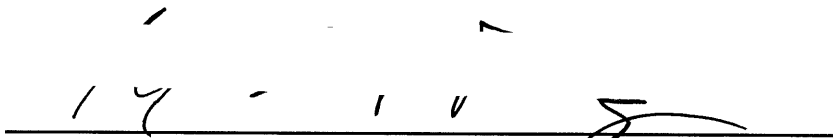
All rights reserved



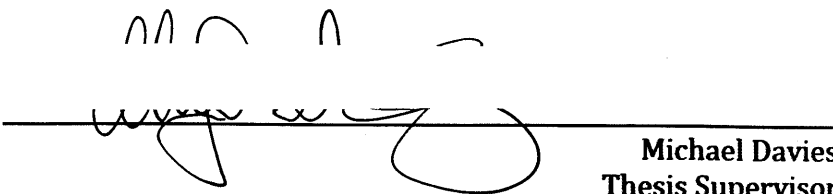
ARCHIVES

The author hereby grants to MIT permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole or in part in any medium now known or hereafter created.

Signature of Author


Robert Swope Fleming
System Design and Management Program
January 2010

Certified by


Michael Davies
Thesis Supervisor
Senior Lecturer, Engineer Systems Division

Accepted by

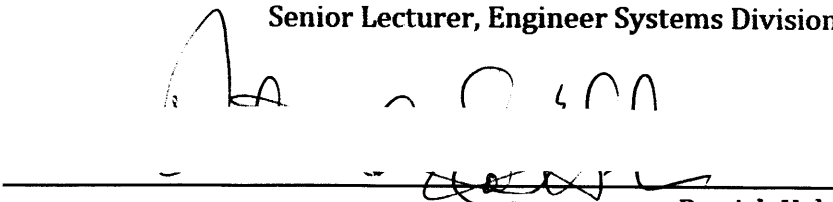

Patrick Hale
Director
System Design & Management Program

Table of Contents

<u>Executive Summary - The End of the Intel Era</u>	4
<u>Chapter 1 - Why Intel Must Change</u>	7
<u>Chapter 2 - Motivation and Terminology</u>	17
Categorization and Terminology.....	18
PART I	21
<u>Chapter 3 - Dynamics of Innovation</u>	23
The Abernathy-Utterback Model and Dominant Design.....	23
Christensen's Attack-From-Below Disruption	26
<u>Chapter 4 - Architecture, Modularity, and Innovation</u>	29
Architectural Innovation	29
Design Rules & Modularity	32
<u>Chapter 5 - Survival of the Incumbent</u>	36
PART II	40
<u>Chapter 6 - A Brief History of Innovation and the PC</u>	41
The IBM 5150	41
The Importance of Instruction Set Architectures	43
RISC Versus CISC	44
The Self-Fulfilling Prophecy of Moore's Law	46
<u>Chapter 7 - A PC on Every Desk</u>	48
The Beginnings of the PC Revolution	48
The PC Onslaught.....	49
The Commoditization of the PC	51
<u>Chapter 8 - The Semiconductor Ecosystem in the PC Era</u>	53
From Vertical to Horizontal.....	53
Increasing Performance, Decreasing Costs	54
The Advent of Multi-core.....	57
The Power-Wall	58
The Economics of Semiconductor Costs	62
The Economic Exhaustion of Intel's Business Model	65
The Rise of the Independent Wafer Foundries	67
The System on a Chip	68
The Advanced RISC Machine.....	71
PART III	75
<u>Chapter 9 - Computing Diversity</u>	76
A Myriad of Devices	76
Servers, Cloud Computing and Data Centers	79
Macro Level Forces Shaping Micro Level Architecture	81
After the Wall.....	82
<u>Chapter 10 - Where the Battle is Being Fought</u>	84
Smartphones	84
Data Centers and Servers	88
<u>Chapter 11 - ARM and Intel In the New Microprocessor Ecosystem</u>	91
The Challenges In Microprocessors	91

The Challenge In Modular Design.....	96
The Challenge With the Business Model.....	100
<u>Conclusion</u>	104
Bibliography	108

Executive Summary - *The End of the Intel Era*

Today, Intel is nearly synonymous with computers. In the past thirty years nearly all personal computers and the great majority of servers have shipped with a processor based on Intel's x86 architecture, of which Intel is the dominant vendor. Yet the past few years have seen a subtle yet remarkable convergence of different industry trends that very well may topple the semiconductor giant.

For the past three decades, computers have largely assumed the same shape and form, regardless of their task. Laptops, desktops, and servers have all been based on the same open modular architecture established by IBM. Yet this is not likely to be the case going forward. The past decade has seen the rise of embedded computing, perhaps best epitomized by smartphones and tablet computers.

Instead of the standard PC architecture where individual components can be easily exchanged, embedded devices are typically modular *designs* with highly integrated physical components. Independent functional units, all designed by independent companies, are integrated onto the same piece of silicon to achieve system cost and performance targets. Instead of a standard x86 processor, each device category likely has a chip optimized for its specific application.

At the same time that the form of computing is changing, we are witnessing a redistribution of where computing power resides with Cloud Computing and data centers. These have ordinarily been the province of Intel based machines, but data centers have moved from using standard off-the-shelf PCs to custom designed motherboards. Again, we are seeing a shift from the modular personal computer architecture to one that is customized for the task at hand.

Another concern for Intel is that the standard metrics by which products compete are in flux. For both embedded systems and data centers, the operational costs and constraints are starting to outweigh the initial outlay costs. An example is the industry shift from overall performance to system power efficiency. Intel has been a relentless driver of processor performance, and this is a significant change of focus for its R&D divisions.

Of all Intel's competitors, ARM best represents the magnitude of these challenges for Intel, and is well positioned to take advantage of all these trends. Their business model of licensing their design is well suited for a world with customized architectures, and their extensive experience in low power embedded devices has given them an advantage over Intel in processor power efficiency.

Intel is heavily invested in its existing vision of the market. They have always maintained a manufacturing process advantage through tremendous investments in new foundries, and have long championed the open PC modular architecture. Time will ultimately show if Intel is capable of meeting these growing challenges. Yet it is clear that in order to do so, it must make radical changes to itself. One may ask if it is even the same company that emerges.

Acknowledgements

As I come to the end of writing this thesis, it is clear to me that I am indebted to several people. First and foremost, my entire grad-school experiment would not have been possible without the love and support of my wife Dana. It is a testament to her support of me that she spent the afternoon proofreading this document on her very first Mother's Day as a Mother. I would also like to thank my son Benjamin. The first nine months of his life coincided with the writing of this thesis, and we spent many mornings discussing over breakfast why ARM is giving Intel fits.

I would like to thank my thesis advisor, Michael Davies, for his invaluable guidance. Michael's input was essential in helping me give structure to the jumble of ideas bouncing around in my head, and helping me to avoid unproductive dead ends. And I would not have been able to get this thesis across the goal line without his help these last few weeks. Finally, I would like Pat Hale, Professor James Utterback, my classmates, and everyone who is involved in making it such an excellent program.

Chapter 1 - Why Intel Must Change

Intel's dominance can be traced back to 1981, when IBM first introduced the personal computer. Since then Intel has dominated the processor segment of personal computing and achieved a success that few other companies have matched. In the past thirty years nearly all personal computers and the great majority of servers have shipped with a processor based on Intel's x86 architecture, of which Intel is the dominant vendor.

Yet the past few years have seen a subtle yet remarkable convergence of different industry trends that very well may topple the semiconductor giant. Not only does Intel find itself technologically behind an unexpected competitor, its very business model is threatened.

The challenges to Intel are threefold. First, the metric of competition has shifted from CPU performance to power efficiency, which is something Intel has not had to have as its primary focus in the past. Second, the *design* of chips is becoming more *modular* while the chips themselves are integrating a wider variety of functionality. This is an inversion of Intel's tradition of designing an integrated CPU and selling it into a modular system. Finally, Intel is at a significant disadvantage in a business ecosystem of licensed modular design and the commoditization of semiconductor fabrication, as its organizational structure and size is predicated on the sizable profit margins it makes from high performance processors and the co-specialization of design and manufacturing.

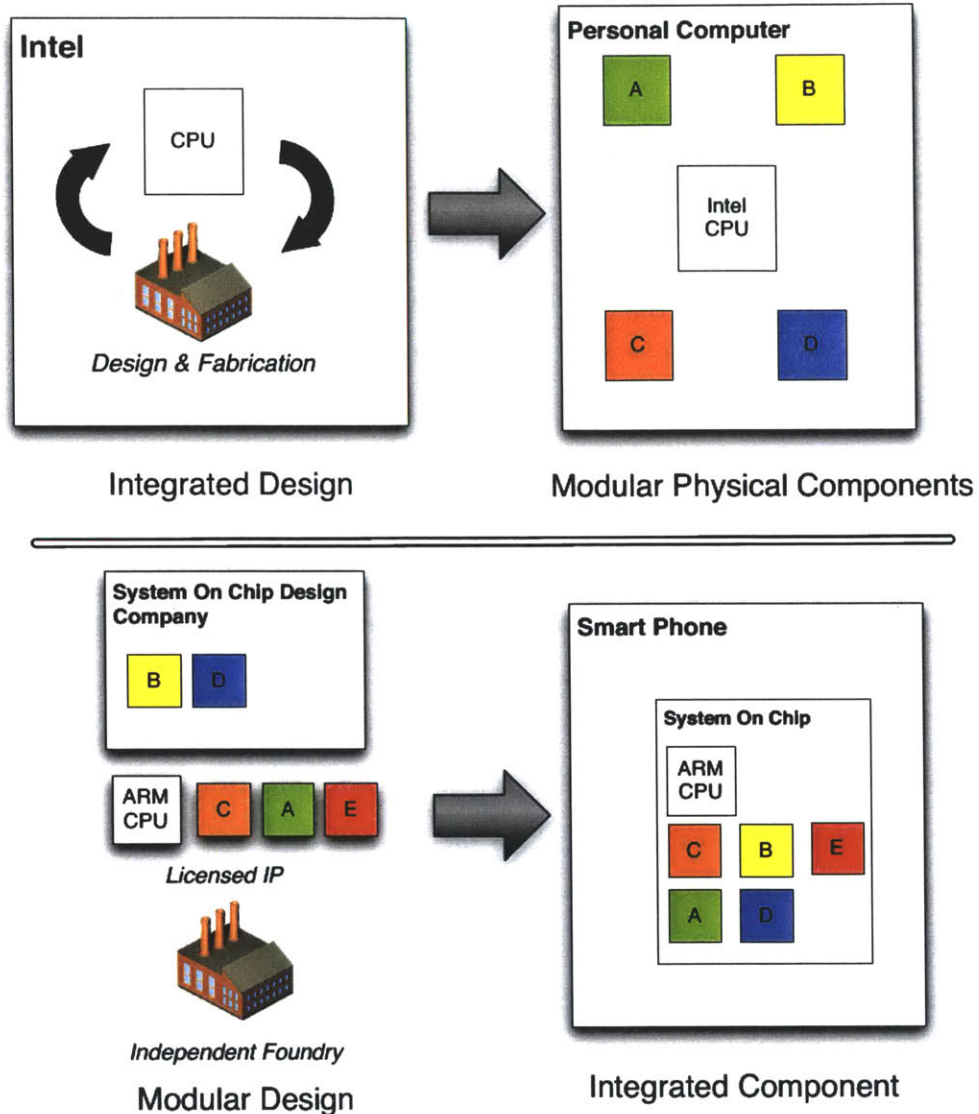


Figure 1 - Integrated Design and Modular Components Versus Modular Design and Integrated Components

In order to survive, Intel will likely need to shape itself into a fundamentally different company. In November 2010, Intel announced that it would allow Achronix to use its foundries¹. This marks the first time Intel has produced chips for another company, and is perhaps a sign that Intel recognizes that radical change is

¹ <http://www.eetimes.com/electronics-news/4210263/Intel-to-fab-FPGAs-for-startup-Achronix>, observed on March 7, 2010.

necessary. There are now also rumors afoot that it will compete to manufacture some high volume ARM chips for Apple².

Despite the myriad of tasks that computers are used for, the number of different form factors is remarkably small. Client and consumer devices can broadly be thought of as desktops and laptops (and arguably net-books as well). Servers can generally be categorized by processing power into small, medium, and large, determined by the number of processor sockets they have (Gillett, 2010).

Furthermore, these different form factors all have nearly identical internal system architectures, based on an open, modular design championed by Intel. The physical components of the system are interchangeable, and a PC firm can replace one vendor with another without significantly changing the design of their system. The only constant is an x86 processor and the Windows operating system, hence the term “Wintel.” In essence, we have a massive industry served by a handful of different product lines, all based on the same open modular system architecture.

This has resulted in Intel being able to serve an enormous market with only a handful of different processors. Instead of product diversity, they have focused on manufacturing excellence and the inherent performance advantages that can be found in smaller and smaller process geometries. If we measure CPU performance against cost (MIPS / \$), Intel has relentlessly improved this metric year over year, directing all their resources and innovation towards it. This is a strategy that has served them well and is perhaps best epitomized by Moore’s Law, the famous observation made by Intel co-founder Gordon Moore that the number of transistors on a chip doubles every two years.

² Barak, Sylvie. “Could Intel Churn out ARM chips for Apple?” *RCR Wireless*. May 4 2011 < <http://www.rcrwireless.com/article/20110504/CHIPS/110509966/0> > Observed May 6, 2011.

Yet the landscape of computing is changing. Instead of laptops, desktops, and servers we see the advent and ascent of smartphones and now tablet computers. Few will argue that these types of devices are not poised to grow significantly. The market research firm IDC estimates that smartphones outsold PCs for the first time in Q4 2010.³ But the change extends beyond just smartphones and tablets. Everyday devices are beginning to come equipped with significant processing power. The list is long and varied: automobiles, televisions, printers, routers, handheld gaming devices, e-books, cell-phones, digital cameras, etc . . .

While the personal computer and server market has a well-established and inherently modular architecture that has benefited Intel, the embedded devices market is populated by a myriad of different architectures. Each application has its own set of size, weight, and cost-performance constraints and challenges, which in turn impose tough tradeoffs between performance and other features. This has given rise to System on a Chip (SoC) based solutions, which allow embedded devices manufacturers to both reduce system costs and have a solution highly customized to their use-case.

These SoC designs pull multiple functions onto the same piece of silicon. Instead of reserving the entire die for the processor, SoCs incorporate other system support functions, such as a memory controller, as well as application specific logic, such as a wireless modem or an ink-jet print head controller. In essence, we have a trend of *modular designs* but highly integrated physical components.

Embedded devices also have different power considerations than traditional PCs. Many of these devices are portable, mobile devices that are disconnected from the power grid and have to run off a battery. Unfortunately, there is no Moore's Law for

³ 100.9M – 92.1M;
<http://www.idc.com/about/viewpressrelease.jsp?containerId=prUS22689111§ionId=null&elementId=null&pageType=SYNOPSIS> and
<http://www.idc.com/about/viewpressrelease.jsp?containerId=prUS22653511§ionId=null&elementId=null&pageType=SYNOPSIS> ; Observed on February 9.

batteries, and adding more processing power to a smartphone can come at the cost of a shortened operational period. While the processing power of a smartphone is certainly an important design parameter, the mobility requirements far outweigh it. Few people would use a mobile device that had to be recharged every hour. Across a wide variety of market segments, we can see the focus shifting from CPU performance to CPU energy efficiency.

At the same time that the form of computing is changing, we are witnessing a redistribution of where computing power resides with Cloud Computing. Cloud-based services can bring new functionality to consumers without requiring them to purchase new devices. Cloud computing has the potential to aggregate our computing power requirements and ultimately reduce the need for raw CPU horsepower in individual devices. Instead of a PC for every desktop, the future will likely be a smartphone or a tablet paired with cloud-based services for every person.

These changes also reinforce the aforementioned shift to CPU energy efficiency. Cloud computing is aggregating the processing needs of countless people into large data centers and server farms. And despite the role of data centers as providers of computing cycles, the cost of processor computing cycles has fallen to where these data centers are more concerned with the operating costs associated with processing, namely cooling and electricity requirements, than with the capital investment required to add more processing power.

Cloud computing has been around longer than the personal computer, but it is only recently that it has reached mass-market adoption as a result of better connectivity. Nearly every tablet computer sold has a network connection of some kind, and network connectivity is one of the defining features of smartphones. This has the dual effect of emphasizing the importance of network connectivity in devices while also deflating the need for processing power for the individual. In short, macro-

level changes in consumer behavior are inflicting change onto the micro-level architecture of devices.

Perhaps the company that best embodies this convergence of threats to Intel is ARM Holdings. ARM Holdings is the developer of the ARM architecture, which is the dominant processor in the embedded devices market. If the future of computing devices truly stems from embedded devices, then ARM stands to benefit simply because of its market share. But this is a threat that runs deeper than simple market positioning.

There are three reasons why ARM is of particular concern to Intel.

First is the rising concern with operational costs of a system over the upfront costs. This manifests itself as an industry shift from a performance-cost tradeoff (MIPS/\$), to a performance–*energy efficiency* tradeoff (MIPS/Watt). For several decades, Intel has created processors with higher and higher performance, and all the while driving down costs lower. Yet now that customers care about MIPS per Watt, it is ARM who has the advantage. In many ways, this is a classic Christensen disruption from below.

The second concern is the aforementioned trend of modular design but integrated physical components. Intel has made its fortune off serving the PC and server market with a handful of products. In a market where we have a plethora of customized and integrated components, Intel can no longer rely on its manufacturing volume. Moreover, when the design is modular, ARM's business model of licensing its processors to SoC designers allows it to be a critical part of any design, regardless of how customized or integrated it is. Firms can design the portion of their system that needs to be customized themselves, and then simply license an ARM processor to complete the system. Intel is not an option, because they neither offer a product that can address a particular niche, nor can they license their x86 processors into that design.

The Non-Recurring Engineering (NRE) costs associated with chip design have steadily risen year after year. For companies that would seek to make a custom chip for their product, it would at first appear that the NRE would be prohibitively high. There puts tremendous pressure on design teams to only to focus their engineering efforts on areas where they can cost effectively differentiate themselves from their competitors. In crude terms, there is good NRE, which is the creation of unique designs with value that attract customers, and there is bad NRE, which is engineering effort that creates something that is redundant in the market but is necessary. A modular design ecosystem allows companies to simply license IP for what would otherwise be 'bad' NRE. A modular design ecosystem, with ARM as the dominant processor core vendor, enables companies to entertain using customized chips.

Finally, we must question if the market of the future can support an integrated company of Intel's size. Intel is a company of \$100 Billion market cap because it has created a tremendous amount of value with its manufacturing capabilities. Intel's business model is predicated on processor design being tightly coupled with fabrication processes, which requires tremendous investments in process research and development and massive capital expenditures in fabrication equipment. Intel's business model is size. This is largely responsible for the performance improvements we have seen over the last three decades. Yet companies like TSMC are commoditizing the manufacturing of semiconductors and are willing to live off thinner margins. They generally lag behind Intel in process size, but it is not clear that this will continue to segment the market. While smaller process geometries have brought lower power consumption in the past, we are reaching a point where increased leakage current will attenuate these gains. If manufacturing excellence is no longer a competitive advantage, can Intel capture the requisite value to justify its size? Even if Intel is successful in matching ARM on energy efficiency and device integration, it may need to radically change itself in order to survive.

The differences between Intel and ARM Holdings go beyond simple market position. These are companies with two fundamentally different business models. Intel has 82,500 full-time employees and a Market Cap of \$119.66 Billion⁴ while ARM Holdings has 1,861 employees and a Market Cap of \$11.08 Billion.⁵ In 2009, Intel had revenues of approximately \$35.1 Billion and an operating income of \$5.7 Billion⁶. In the same year, ARM reported revenues of \$463.8 Million and an operating income of \$73.8 Million⁷. In engineering terms, Intel's x86 architecture is a CISC-based architecture (Complex Instruction Set Computer) while the ARM architecture is RISC-based (Reduced Instruction Set Computer). And finally the business models are radically different. Intel fabricates its own semiconductors, and is heavily invested in its operations. In contrast, ARM Holdings manufactures nothing and instead licenses its design out to other semiconductor companies.

This thesis will strive to answer the question – what will happen to Intel? - and is divided into three sections. Part I, spanning chapters 3, 4, and 5, is a survey of the academic frameworks used when writing this thesis. Chapter 3 covers the topics of radical innovation, dominant design, and disruptive innovation. In broad terms, it explores how industries have patterns of innovation that evolve as an industry matures, and the phenomenon of disruptions, where high performance technologies are disrupted by lower performing alternatives. These are useful frameworks to help dissect the conflict between Intel, an industry stalwart who played a crucial role in the birth of the PC industry, and ARM, whose expertise is in cell phones and the embedded space.

The nature of the differences between Intel and ARM are largely architectural differences. This is not the difference between cars and horse-drawn carts. Both companies are in the process of designing integrated circuits. Architectural

⁴ <http://finance.yahoo.com/q?s=INTC>, observed on January 25, 2011.

⁵ <http://finance.yahoo.com/q?s=ARMH>, observed on January 25, 2011.

⁶ Intel 2009 Annual Report.

⁷ ARM 2009 Annual Report, converted to dollars using historical exchange rate on December 31, 2009.

innovations have their own characteristics and can depart from much of the conventional wisdom. Chapter 4 investigates the concept of architectural innovation and the role of modularity in enabling rapid change.

Finally, after understanding the nature of the challenge to Intel, the logical extension is to explore the managerial consequences. The stakes are frighteningly high for companies faced with technological transitions, and history is littered with examples of organizations that failed to navigate the challenge. Chapter 5 is about the survival of incumbent firms when faced with dramatic change, and focuses on the highly opposite stories of IBM and the Polaroid Corporation.

Part II is a retrospective of the computing industry since the advent of the PC. I begin with the PC because the arrival of the PC was the last major revolution in the computing industry, and is responsible for much of the industry structure that have today. Chapter 6 begins Part II with a narrow focus and covers technological trajectories and significant innovations in processor architecture and design. In Chapter 7 the focus broadens to the PC market and traces the drivers behind the PC's spectacular success and its growth into new market after new market. The PC's success has been absolute, and what began as a hobbyists' toy now dominates nearly all forms of computing.

In Chapter 8, I look at the semiconductor industry ecosystem as a whole. This chapter touches on a variety of topics, including the industry structure, the dominant cost-performance metrics of the PC era. I spend some time exploring the massive challenge presented by the Power Wall, the escalating costs associated with building a foundry, and how Intel's business model is running out of runway, even without the challenge from ARM. Finally, Chapter 8 ends with a discussion of ARM, System on a Chip designs, and the rise of independent foundries.

Part III is a prediction of future trends. Chapter 9 focuses on technology and the trajectories of computing and how they will depart from past trends. The chapter

highlights moves towards diverse product architectures, as opposed to the PC monoculture and the emergence of cloud computing as a significant force. In Chapter 10, I investigate the design constraints in two intensely important markets laid out in Chapter 9, namely smartphones and data centers. Finally in Chapter 11, I lay out the challenges Intel faces in the present and future computing ecosystem at large and why ARM is positioned to thrive.

Chapter 2 – Motivation and Terminology

So why write a thesis on Intel? My fascination with processors dates back to my undergraduate education. (If we want to pinpoint when my interest with *computers* began, we would have to dig far deeper). My chosen Major was Electrical Engineering and within that my concentration was Computer Architecture. One of the benefits of my curriculum was that I emerged from college with an understanding of how a computer works from the operating system all the way down to the physics of a metal oxide transistor.

Bar none, my favorite classes were the two I took on processor design. In my junior year, my classmates and I recreated a PDP-8 processor using FPGAs. My senior year, the project was to turn a single-issue RISC core into a dual-issue superscalar processor. I was utterly taken with the simple elegance of RISC-based designs. At the time, it made a significant impression on me that something that I considered to be an example of sophisticated engineering was largely an afterthought in the general purpose computing market by the year 2000. It was my first lesson that there was more to processors than good engineering.

After I graduated college in 2001, I went to work for Sun Microsystems. I was part of a gigantic project designing an Ultra Sparc V mainframe. In particular, I was working on the chipset outside of the processor. It was my first introduction to ASICs and logic design that was tailored for specific tasks rather than running software. Unfortunately, my time at Sun coincided with the fallout from the dotcom bust, and I was introduced to another reality of life in high tech - layoffs.

In mid 2004 I joined a small fabless semiconductor company called Oasis Semiconductor, which made chips for ink-jet printers. Because Oasis was so small, I was able to see far more of the company than I was ever able to at Sun. At Oasis, I saw the power of System on a Chip designs, and the vibrancy. It was also my first

introduction to ARM processors. The realization that you could simply license a core rather than designing your own microcontroller was an epiphany for me. The elegant simplicity of ARM's business model struck me much in the same way that RISC processors did.

When I left oasis in 2010, I had an understanding that ARM was well established and would continue to do well. However, my studies at MIT have made me realize just how quickly dramatic change can sweep an industry. Violent change seems to come periodically, albeit with long periods of stability. This is similar to the evolutionary biology theory of punctuated equilibrium, where evolutionary change happens in short intense periods, followed by relative stability¹. Intel and x86 had been synonymous with computing as long as I could remember. When I left college, I could not conceive of them ever being displaced. Yet as I reflected on my undergraduate and professional experience through the lens of what I have studied in the last year and a half, it has become apparent to me that we are in the midst of a change as dramatic as the introduction of the personal computer.

This thesis is my attempt to document and outline the scope of this change, and show why we are witnessing the end of the Intel Age.

Categorization and Terminology

After completing my first draft of this thesis, it was pointed out to me that I use terms and phrases liberally that might be confusing to someone who does not have a background in semiconductors. Before I get too far into this thesis, I will attempt to provide some clarification for the reader. Let me begin with Semiconductors and the Semiconductor Industry. 'Semiconductors' is a shorthand way of referring to circuits, both analog and digital, integrated onto a single piece of silicon. I use 'Integrated Circuit' as a synonym for Semiconductor.

¹ Wikipedia. < http://en.wikipedia.org/wiki/Punctuated_equilibrium > Observed May 8, 2011.

Three other terms that I use interchangeably are Microprocessor, CPU (Central Processing Unit), and Processers. In the original commercial IBM mainframes, a processor was constituted of several individual components, but when Intel invented the microprocessor, the entirety of the processor was integrated onto a single piece of silicon. This is very much the norm now, so while it is technically incorrect, I use these terms interchangeably.

There is an important distinction to make between a chip and a processor. In my mind, a chip constitutes an integrated circuit, including its final packaging. Intel makes a variety of chips, which constitute nothing other than a processor. But chip and processor are not synonymous with each other. A chip can be any integrated circuit, and is not limited to processors. In addition, a processor can be integrated with other functional circuits on a single chip, which is known as a System on a Chip.

To confuse things further, I often frequently refer to processor cores. This is a term that has risen with era of multi-core designs or System on a Chip designs, where the processor is only a part of the overall design. A processor core is a stand-alone processor, but is intended to be integrated with other circuits, be it additional processor cores or other functional circuits. A multi-core chip is an integrated circuit that has two or more processors inside of it.

There are a handful of other terms that are used throughout this thesis. 'Die' is another way of referring to the physical silicon that a circuit lives on. A wafer is a circular piece of silicon that fabrication processes are built around. They vary in size, but the current standard is 300 mm in diameter. Once fabrication is complete, the dies are cut from the wafer for individual chips. After being cut, the chips are then packaged into their ceramic packaging that most people would recognize when they look at the circuit boards inside their computers.

Finally, I need to clarify what I mean by Embedded Computing. An embedded computer is a computer system that is generally crafted to a specific task, and is only

one part of an overall product rather than the product itself. While a consumer may shop for a 'computer,' they do not go to the mall with the intent to buy an 'embedded computer.' Instead, they purchase a cell phone, a car, a printer, a microwave, a router, a television, an e-book, or any other number of devices. But an embedded computer is a critical component of each and every one of these devices. Most consumers do not realize the ubiquity of embedded computers, but our daily lives are surrounded by a myriad of them.

PART I

Academic Background

Chapter 3 - Dynamics of Innovation

The next three chapters will discuss existing academic work that I found particularly helpful in my analysis of Intel and ARM. As we move to a discussion of the PC industry in the next section, the Abernathy-Utterback model helps explain the differences between the Intel of 1981 and the Intel of 2011, and Christensen Disruption helps illustrate how a relatively small company like ARM can be so dangerous. In Chapter 4, I explore the concepts of Architectural Innovation and modularity in *Design Rules*, which I feel are the appropriate lenses to examine what ARM is doing differently. Chapter 5 discusses the managerial challenges beyond selecting the correct technology and strategy to compete. In short, failing to make the requisite managerial changes in the face of technological discontinuities can mean the end of an organization.

The Abernathy–Utterback Model and Dominant Design

The computer industry has a long history that is characterized by the rapid pace of change. It would be a daunting task to try to track and understand every single product or innovation introduced. Fortunately, there is an established body of academic work that can help us understand the inner dynamics of innovation in microprocessors.

The first model to discuss is the Abernathy – Utterback model, which characterizes innovation in an industry as going through three successive phases: the fluid phase, the transitional phase, and finally the specific phase (Abernathy, Utterback 1978). As a starting point, it is helpful to map these three phases onto the traditional S-curve often associated with performance trajectories.

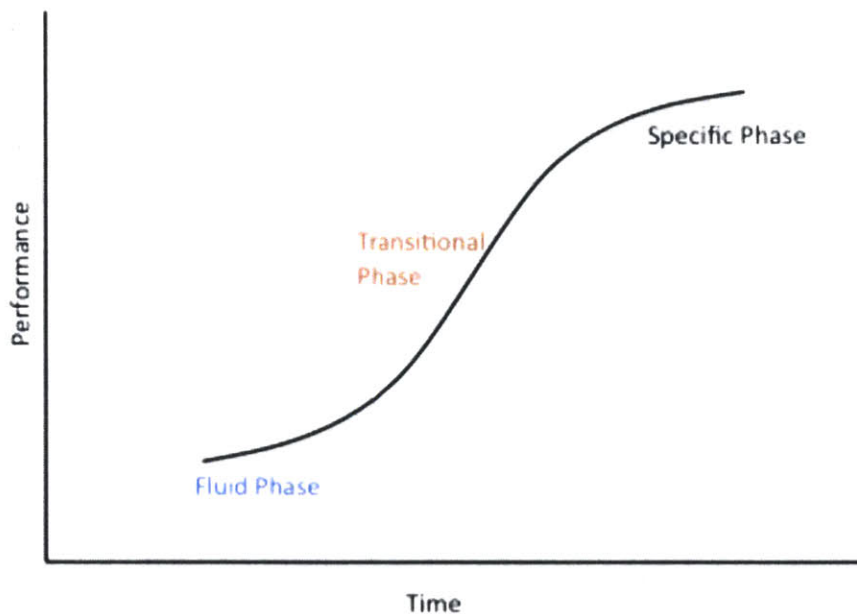


Figure 2 A hypothetical performance curve and the 3 phases of the Abernathy-Utterback Model

The fluid phase is characterized by the entry of both new and established firms into a market. The product category is likely new, and firms compete through product differentiation and radical innovation. A salient example would be the US automobile industry. From 1894 to 1918, 60 different firms entered the market, and the total number of firms peaked at 75 in 1923 (Utterback, 1994). This time period is populated by a number of radically different product designs, including electric cars, steam cars, as well as the internal combustion engine car.

In contrast, the transitional phase is often marked by consolidation within the industry. Using cars as the example again, between 1919 and 1941 a net of 44 firms left the automobile industry (Utterback, 1994). In addition to the declining number of competitors, the rate of product innovation typically drops off significantly. Instead, the industry begins to shift its focus from radical production innovation to *process* innovation and limits its R&D to a specific set of product features. As we can see, the shift from the fluid to the transitional phase is a dramatic change and begs the question of why does this occur?

Most often this change is attributable to the emergence of the dominant design. The dominant design is the result of all the product innovation that occurs in the fluid phase, and by definition “wins the allegiance of the marketplace.” Once a dominant design begins to emerge, it becomes increasingly difficult for firms to compete through product differentiation alone, as there are expectations in the marketplace of what the product should be. To use the automobile example again, it is clear that the dominant design is the internal combustion engine car, and all the features that are standard on cars today (windshield wipers, electric starters, seatbelts, etc. . . .) (Utterback, 1994)

It is important to note that the dominant design is not necessarily the product with the highest performance, or greatest amount of functionality. It is determined by a combination of technological and market forces, but can also be influenced by factors such as standards, regulation and government influence.

The final phase in the Abernathy-Utterback model is the specific phase, and in many ways extends the trends of the transitional phase. We see further consolidation, often resulting in an oligopoly of a handful of firms sharing the market. The dominant design is well defined, and the product category is heavily standardized. Whatever product innovation occurs is mostly incremental innovation, and even the manufacturing of products is fairly rigid, with well-defined supply chains and distribution channels.

It is also important to note that the Abernathy-Utterback model also recognizes that the different phases also have a strong influence on how an organization is structured. For example, entrepreneurship is common in the fluid phase, while project teams and task groups are more common in the transitional phase. Companies in the specific phase are characterized by highly structured and rigid organizational structures.

Christensen's Attack-From-Below Disruption

The stability of the specific phase leads directly into the next framework of interest, Clayton Christensen's attack-from-below disruptive innovation described in *The Innovator's Dilemma*. Christensen outlines a phenomenon where incumbent organizations, who are well into the specific phase, can be well run and make all the right decisions, yet still be overtaken in the market by lower performing technologies.

The concept of *disruption* is a complex and dynamic idea, but it begins with the recognition that technologies have performance trajectories. These trajectories are almost always upward over time, as products improve performance year after year. Some examples: with each generation of products hard drives add more storage, semiconductors add more transistors, printers print at faster page rates, bicycles get lighter and so on and so on.

Companies are rewarded for these trajectories when the market is segmented by performance. Increasing product performance allows companies to reach more lucrative segments of the market, and the market rewards companies that have performance advantages over their competitors. These trajectories and expectations of market behavior can embed themselves in an organization in a myriad of ways. Competitive strategies, relationships with the customers, organizational structure, market choices, and R&D capabilities are all optimized to race further ahead in the technological trajectory. Christensen uses the term *value-network* to describe this. The value-network drives companies to chase higher and higher product performance and makes it hard for a company to behave in any other way.

This sets the stage for one of the key ideas behind disruption: that the market's supply of performance can overshoot customers' demand. When a disruption occurs, a product attacks the incumbent firm "from below." That is to say that the

attacking product has lower performance on what has hitherto been the most important attribute, but still enough to satisfy the needs of a given segment. As both products meet the performance needs of a customer, the basis of competition begins to shift to other dimensions.

This leads to another central idea in a disruption: the attacking product often has superior performance in different dimension than the one along which the established technology trajectory was measured. This is a subtle concept, and is best demonstrated by the example of the hard drive industry used by Christensen. Within a generation of hard drives, the performance trajectory was for increasing storage space. Without fail, each successive market leader is disrupted by a product with smaller storage space. But each disruptive product is also *physically* smaller, allowing it to fit into smaller computers. The 14 inch hard drives were dominant in the mainframe market, but 8 inch hard drives were able to take the minicomputer market because while both technologies met the storage requires of minicomputers, the 8-inch technology offered a superior form-factor.

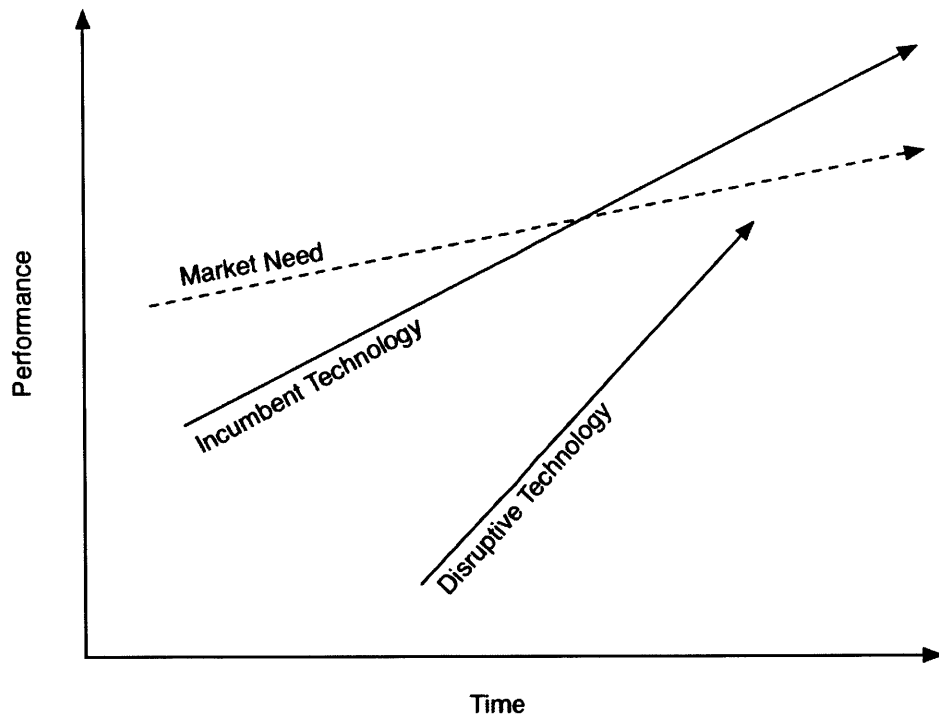


Figure 3 - Attack from below disruption

Disruptions sometimes catch companies by surprise, but they can be difficult to avoid even if they are seen coming. The reason for this is the aforementioned value-network, which establish overwhelming feedback mechanisms to keep companies focused along existing performance trajectories. It is for this reason that disruptions can be so dangerous to incumbent firms.

Chapter 4 - Architecture, Modularity, and Innovation

Architectural Innovation

While the previous chapter focused on dynamics of innovation, the discussion only differentiated between innovation in the product itself and innovation in the process to create the product. To be sure, innovation is a multi-dimensional phenomenon. In particular, this chapter will focus on innovations in the product *architecture* and the challenges faced by organizations when a product's architecture begins to change.

What is meant by architectural innovation? In the paper *Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms*, Rebecca Henderson and Kim Clark define architectural innovation as “innovations that change the way in which the components of a product are linked together, while leaving the core design concepts untouched” (Henderson and Clark, 1990). As that quote alludes, in this framework innovation can occur along two dimensions: the aforementioned architectural innovations and innovations in the core concepts.

In broad terms, this leaves us four types of innovation, shown in table 1 which is a replication of Henderson and Clark's work.

	Core Concepts Reinforced	Core Concepts Overturned
Architecture Unchanged	Incremental Innovation	Modular Innovation
Architecture Changed	Architectural Innovation	Radical Innovation

Table 1

Radical innovation and incremental innovation are compatible with the views of innovation discussed in the previous chapter. Radical innovation is diverse experimentation with all the aspects of a product, ultimately resulting in a dominant design. In the specific phase of an industry after a dominant design has been established the most common type of innovation is incremental innovation.

Modular Innovation is also a straightforward concept. Henderson and Clark use the example of replacing an analog phone with a digital one. The two phones are based on significantly different technologies, yet they both have keypads, antennas, speakers and microphones. In short, an analog phone and a digital phone are still both phones. A similar example would be the recent switch to HDTV. From a product standpoint, much of the technology has changed, yet the TVs all still have the same categories of components (signal inputs, a screen, speakers, a power cord, etc . . .).

Architectural innovation is more elusive concept. To recognize this type of innovation, we must realize that knowledge and expertise in a specific technology or component is altogether different than knowledge of how a technology or component interacts with other technologies and components. For example, a ceiling fan has many of the same components as a portable fan, such as an electric motor and the fan blades. Yet how these components are connected in very different ways, such as what the housing for the motor looks like. (Henderson and Clark, 1990).

The main thrust of Henderson and Clark's paper is that this type of architectural innovation presents a significant challenge to established firms. An established firm will likely have relevant expertise for the new architecture, but they may not understand how their knowledge is relevant. And in truth, their capabilities may also blind them to critical changes in the new architecture. RCA was an industry leader in transistors, radio circuits, and speakers, all critical parts of a transistor

radio, yet it was Sony who was the one who achieved market dominance. (Henderson and Clark, 1990).

Why is this a challenge for established firms? In a market in the specific phase, where the dominant design has been established, firms must be highly efficient to survive. As the architecture of a product is stable, it can become “embedded” in the organization. For example, a designer of televisions will likely have a screen design team and a control electronics design team. This specialization is an effective way to execute incremental innovation, and contributes to deep domain knowledge.

The way these different groups interact is also likely to be standardized and optimized around the product design process for televisions. This creates “information filters” within an organization. The screen design team and control team will likely share information such as bandwidth and signaling requirements, but not share with each other information such as thermals or size and weight. These information filters are often necessary for high performing teams involved in incremental innovation, as it helps to block out unnecessary distractions. In summary, an organization can become like a mirror of the product it is designing, both in its physical organization and its methodologies.

However as architectural innovations change the way components are connected, they will likely be incompatible with these structures. The aforementioned information filters can cause teams to not fully understand new critical interactions in a new architecture. The internal processes that companies organizations rely on to create high performing products can become a liability in the face of a change in architecture. And even if an architectural change is fully understood by an incumbent, it still has a significant handicap as organizational structure is painful to uproot and rebuild while simultaneously trying to design a fundamentally different product.

Design Rules & Modularity

One potent form of architectural innovation is the introduction of modularity in a design. This is not to be confused with the aforementioned modular innovation. Modular innovation refers to innovation within the components of an already modular design. In *Design Rules*, Carliss Baldwin and Kim Clark discuss how introducing modularity into a design, thereby innovating in its architecture, can have profound effects on a products value and performance. These changes can be so powerful as to reshape the entire competitive landscape.

Modularity can be used to make a system more flexible and adaptable, and allow a market to rapidly find a system with the most value. Instead of a single monolithic design, a modular design consists of separate components whose interactions are defined by the systems “design rules.” The core of *Design Rules* built around identification of six operators that designers could use to create and modify a modular system.

The first operator is *splitting*. Splitting is the first operation that must occur to create a modular system. It is the act of separating functions that were previously integrated together and having them both adhere to the same design rules which allows them to interact. The second operation of *substitution* is simply replacing one module with another. For example, a designer can replace a hard drive on computer with one with entirely different performance specifications without incurring significant economic costs. As long as the modules adhere to the same design rules, substitution allows market forces to begin operating within a design. Without splitting and substitution, market choice can only act on the system as a whole.

The next operator is *augmentation*. Augmentation adds new modules to a system and introduces new functionality. *Exclusion* is the inversion of augmentation, and is the removal of a module from a system. Logically, the exclusion operator reduces

the functional range of a system. Again, all modules in a system must adhere to the same design rules. A simple example would be the Swiss-Army knife. Adding a new corkscrew or blade to the knife is augmentation, while removal of the can opener would be exclusion.

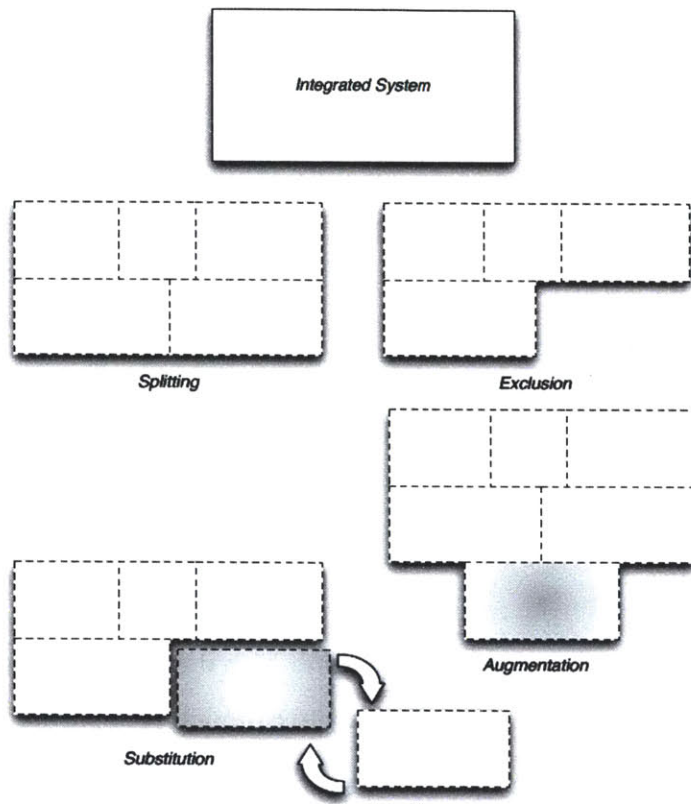


Figure 4 Four of the Six Modular Operators

Inversion can be used when several different modules end up creating solutions to the same problem repeatedly. Inversion creates a standard module to solve a problem. For example, it is common in software to create a data-structure called a linked list. Now most programming languages come with built in linked-list modules for programmers to use. The term inversion comes from the act of a designer bringing something from within a module up to a higher level for other modules to use.

The final operator is *porting*, and in many ways is a logical extension *inversion*. While inversion brought solutions the inner workings of a module to a higher level, porting allows a module to be used in other systems. Using the words of Baldwin and Clark, porting lets a module “breaks free” of the system. Tesla Motors first designed its battery packs for use in the Tesla Roadster, but it ported its battery for use in the Daimler Smart ED car¹.

These six operators can lead to a number of different outcomes dizzying in their variety. In complex systems, it is rare that a company understands fully how the final system will perform - let alone how a market will receive its product. A modular system allows designers to create a large number of different systems at a relatively low cost. This higher rate of “experimentation” results in a higher probability of a company will create a design that is of greater economic value.

Of particular relevance to this document is how introducing a modular design can lead to modular clusters, which is a term *Design Rules* uses to describe a type of industry structure. As a design shifts from an integrated monolithic design to a modular based design, design efforts become decentralized. This decentralization can evolve from separate design teams within a company to completely separate organizations. In this type of industry structure, the design of a system is spread over multiple independent firms, who sometimes are actively competing with each other.

Carliss Baldwin and Kim Clark explore this idea further in the working paper “Architectural Innovation and Dynamic Competition: The Smaller ‘Footprint’ Strategy.” By leveraging modularity, companies can focus themselves on the portion of the design where they believe they can create the most value. Sun Microsystems focused their design efforts on the interface they believed to be the performance bottleneck, namely the interface between the CPU and memory and used standard

¹ Autoblog.com. January 13, 2009 < <http://green.autoblog.com/2009/01/13/tesla-confirms-smart-ed-battery-supply-deal/> > Observed April 26, 2011.

modules for the rest of the design. With this approach they were able to outperform systems designed by Apollo Computer, who designed nearly all aspects of their product. Sun's smaller organizational "footprint" allowed them to not only build a better product, but also be much more capital efficient while doing so. (Baldwin and Clark, 2006) It is easy to see how modular clusters can create feedback loops and allows small, networked organizations to create powerful systems quickly and efficiently.

Modular clusters share a great deal of overlap with "business ecosystems." While a review of the relevant literature of business ecosystems is outside the scope of this thesis, a great discussion of the dynamics and powers of ecosystems can be found in the articles "Strategy as Ecology" (Iansiti and Levien, 2004) and "Predators and Prey: A New Ecology of Competition." (Moore, 1993)

The authors of *Design Rules* posit that this evolution is beyond the control of an initial firm, and modular clusters can emerge despite the active efforts of the originating firm to maintain control of the design. The modern computer industry, itself a modular cluster, is a consequence of the decision by IBM to make System/360 a modular design. It is truly striking when engineering decisions made at the micro-architectural level can resonate to the point where a giant like IBM was knocked from their perch.

Chapter 5 - Survival of the Incumbent

There is nothing so captivating as the fall of the giant. There is no shortage of analysis of why companies fail, and yet there is no widely accepted wisdom as to if established incumbents have long-term advantages or if they are doomed to eventual failure. At this point, much of the work is anecdotal in scope, and only analyzes the success or failure of individual companies rather than developing a common framework. But if anything is clear, it is that companies can fail in a spectacular variety of ways.

In all of this analysis, by far the most common question is “Was failure avoidable?” Some of the academic work explored in chapters 3 and 4 would seem to suggest that certain types of challenges are particularly difficult for companies to overcome: architectural innovations may not be understood by the incumbent as their internal structure filters out critical information flow; Clayton Christensen’s book suggests that an attack-from-below based disruption is nearly impossible for an incumbent to beat back. But perhaps we need to separate the question of a company’s failure in a market from its overall failure.

The first example to investigate is perhaps the most famous turnaround story in business today: Lou Gerstner’s miraculous turnaround of IBM in the 1990s. In the early 1990’s, IBM’s earnings were falling off a cliff. In 1991, IBM reported a loss of \$2.82 Billion, a dramatic swing from its earnings of over \$6 billion the previous year. In 1992 the losses nearly doubled to \$4.96 Billion. And in 1993, the losses ballooned to a staggering \$8.1 Billion.¹ While it dominated the mainframe market, IBM’s market dominance was being attacked by mini-computers and PCs. At the same time, it was facing a critical technology change in its flagship product line, as

¹ IBM Archives: 1990s < http://www-03.ibm.com/ibm/history/history/decade_1990.html > Observed on March 28, 2011.

mainframes needed for transition from bipolar designs to CMOS based designs (Wladwasky-Berger, 2009)

A decades of avoiding layoffs, IBM cut 125,000 employees between the years of 1986 and 1993. It was at this point that the existing CEO John Akers resigned and John Gerstner was selected to be his replacement. Throughout the 1990s Gerstner transformed IBM from a hardware technology vendor to a broad-based solutions and services provider (Harreld et al., 2007). There is no shortage of books and articles on what IBM did to execute its turnaround, and Gerstner overhauled nearly aspect of the business, from its internal accounting principles, to its advertising strategies, to its product development. But it was the role of IBM's culture in its turnaround that is perhaps the most broadly applicable.

Lou Gerstner was certainly a capable executive before he came to IBM but perhaps the most notable qualification was the fact he was the first CEO to come from outside IBM. In fact, this was a critical requirement of the job. When searching for a new executive, the executive only considered candidates from outside the company as the board "felt strongly that what was wrong with IBM couldn't be fixed by an IBMer." (Garr, 1999, pg. 21) Why make this a requirement of the job? Perhaps it was a tacit acknowledgement by IBM's board that its problems ran deeper than poor strategic choices, and that the culture of the organization needed to change as well.

In his first year on the job, Gerstner had to struggle with "malicious obedience," where the existing senior and mid-level managers would agree to anything he suggested but would continue to operate in the manner they were used to. (Garr, 1999, pg. 71) To combat this, Gerstner made a series of moves and changes that sent a clear and undeniable signal to IBM that the culture had to change. He forced out senior managers who demonstrated that they were incapable of changing their ways. He wrested power from IBM's successful foreign subsidiaries, and ousted their senior executives who had long and successful track records. (Garr, 1999,

pg.72). Finally, he changed the bonus and incentive plan for the rank-and-file employees, all of which was Gerstner's way of saying things must change and that IBM would have standards accountability. (Garr, 1999, pg. 135)

Another potent example of the role of culture in an organizations ability to adapt is the story of Polaroid. Unlike IBM, Polaroid was unable to overcome its challenges and filed for bankruptcy in October of 2001.² Polaroid's failure is often attributed to its inability to transition from analog to digital photography. What is tragic about Polaroid is that they were an early technology leader in digital photography. Their Electronic Imaging Group was founded in 1981 and in 1992 they had a working prototype of their PDC-2000 digital camera. However, the belief that Polaroid was a film company was deeply ingrained in the management at Polaroid, and a product that did not conform to the razor & blades business model, such as a digital camera, struggled for executive support. As a result, the PDC-2000 was not released until 1996. What is remarkable is that even with the long delay, the PDC-2000 was still a best in class digital camera, but at that point there were over 40 competitors in the market and it failed to take off. (Tripsas and Gavetti, 2000)

So why did Polaroid fail while IBM was able reverse its fortunes? There are numerous differences between these companies, but perhaps it was the pace of failure that is the critical difference. That is to say, IBM's earnings had plummeted dramatically in the span of two years, which perhaps engendered a sense of crisis that may have been lacking at Polaroid, who's failure played out over a decade. The old anecdote of the frog in a boiling pot of water is an apt metaphor for this.

Finally, we must consider if failure of an organization is inevitable. The creation of new firms and the failure of existing companies is an established tenet of business today. With each challenge a company faces, it must ask if it could be the one that

² CNN Money, "Polaroid files Chapter 11" < <http://money.cnn.com/2001/10/12/companies/polaroid/> > Observed on March 29, 2011.

puts them out of business. In *Organizational Ecology*, Michael Hannan and John Freeman use biological theory to explore the evolution of organizations. One consequence of their work is the realization that just like in biological populations, the death of an organization is highly correlated with its age. (Hannan and Freeman, 1989, pg. 245) There is no fountain of youth for companies. While we cannot say with any certainty which organizations will survive and which will fail, it is clear that giants like Polaroid, IBM, or even Intel must take any challenge seriously.

PART II

The Past

Chapter 6 - A Brief History of Innovation and the PC

In this second part, I begin with the IBM 5150, the machine that signified the arrival of the personal computer. Why start with the PC? In short, the advent of the PC was the last major revolution in the computing industry, and is responsible for the industry ecosystem we currently have. The revolution we are witnessing today in many ways mirrors what started in 1981. As this is a thesis about Intel, this chapter will focus in on technological trajectories and innovation in processors. In Chapter 7, I will trace the PC onslaught as it overtakes higher performing machines with frighteningly regularity, and how much of that growth is due to innovations and advances in processor performance. Chapter 8 will outline the challenges Intel faces, and the Herculean efforts it must undertake to keep Moore's Law chugging along. It will then pivot to ARM and System on a Chip semiconductor design, which sets up the final third of this thesis.

The IBM 5150

While technically not the first Personal Computer, the release of the IBM 5150 in 1981 is widely recognized as the event that kick-started the PC industry¹. In a move highly atypical of IBM, the 5150 used technology developed by outside companies, shipping with the 8086 processor designed by Intel and an operating system developed by Microsoft. In a strange twist of fate, Intel may owe its place the PC business to Motorola. Before the 5150, IBM had a favored internal project that was stumbling because Motorola was late in delivering the processor. In order to satisfy a corporate deal with Sears, IBM created the 5150 as a stopgap and chose Intel to supply the processor. (Jackson, 1997, pg. 203)

The 5150 project was a crash project through and through. To save time and money, the IBM design team elected to use off the shelf parts and software rather than

¹ IBM Archives: 1981. < http://www-03.ibm.com/ibm/history/history/year_1981.html > Observed on March 30, 2011.

develop internally, as evidenced by the inclusion of Intel and Microsoft. ("Getting Personal", Economist 2006) In turn, the Intel 8086 chip IBM selected was itself a stopgap for Intel's iAPX432, which was a much delayed and troubled project. And Microsoft purchased MS-DOS from a third company, where it was originally named Q DOS, which stands for Quick and Dirty Operating System. (Jackson, 1997, pg 162, 205) Finally, IBM elected to make the specifications open to facilitate the development of outside software. IBM had modest sales goals for the 5150, and wanted to keep the development as low cost as possible.



Figure 5 - The IBM 5150²

IBM's decision to build an open architecture for the PC and its "Big Blue" reputation in the market turned out to be a fateful pairing. Although there were many firms trying to grab a piece of the nascent PC market, IBM's entry was a legitimization of a market that up until that point many people considered to be merely the province of hobbyists. The founders of Compaq recognized that whenever IBM entered a market, whatever it released became the standard. (Wilcox, 1998) As the 5150's architecture was open, Compaq could simply buy the same off the shelf parts and

² Image Source: Self-reliance-works.com < <http://www.self-reliance-works.com/wp-content/uploads/2011/01/IBM-5150-PC.jpg> > Observed May 8, 2011.

software and was able to create a software compatible machine within a year. After this, the history of the PC one of a steady stream of IBM PC clones and diminishing IBM fortunes. With a single product release, IBM had both defined the modern PC and gave away the keys to the kingdom.

The Importance of Instruction Set Architectures

While much of the literature focuses on the significance of IBM outsourcing the operating system of the 5150 to Microsoft, IBM's decision to use Intel's 8086 processor was equally important. In short, it allowed Intel to define the Instruction Set Architecture (ISA) of the PC market. The ISA is a critical element of computer architecture. It defines the interface between hardware and software. An ISA defines a set of instructions and operations that a processor can interpret and execute. All software is a string of these instructions, which when executed in order create higher-level functions.

IBM first established the importance of the ISA with its System/360 mainframe. (Lee, 2011) Without a stable ISA, software is not guaranteed to continue operating correctly as it transitions from one generation of products to the next. And as the body of software written grows for a particular product, so do the costs of switching to a new product. IBM realized that a lack of software compatibility drove up costs both for itself and for its customers, making it harder to sell them new products. This is tremendously significant, and the System/360 represents the first time that software was not exclusively co-specialized with the hardware. The concept of backwards-compatibility had arrived.

The ISA frees up processor designers to introduce whatever innovations they like without fear of affecting software compatibility, as long as their innovations don't modify the ISA. An ISA is important because decouples the hardware-software interface from the implantation. It is a contract between hardware and software that all parties must adhere to.

When IBM selected Intel to supply the processor for the 5150, they didn't ask them to design an implementation of an IBM's ISA. They selected a processor that was an implementation of Intel's own x86 ISA. An ISA is not something chosen lightly, as it only becomes harder to make changes to it with time. But given the crash nature of the 5150, IBM's actions are understandable. But this simple choice codified the x86 ISA, which IBM neither owned nor controlled, into the standard architecture of PCs.

RISC Versus CISC

Although an ISA is rarely changed, they can be discarded if the benefits outweigh the costs. While IBM dealt them an extraordinarily strong hand, Intel's dominance was hardly *fait accompli*. One of the significant challenges the x86 faced was from processors with a RISC-based ISA. RISC stands for Reduced Instruction Set Computing, and it represents a school of thought and design that was a significant departure from the existing practices. By contrast, the x86 ISA is often referred to as a CISC (Complex Instruction Set Computing) based design.

The key difference between RISC and CISC is the number of instructions included in the ISA as well as complexity of the instructions themselves. When CISC processors were first designed, the amount of memory space software occupied was critical. (Lee, 2011) By expanding the number of supported instructions, software could be written with fewer instructions. A good analogy is the difference between the Chinese written-language, where a single character represents a syllable, versus the Latin alphabet, where multiple characters must be used to construct a syllable. If paper is extraordinarily valuable, writing in Chinese is preferable, as it will require less paper. CISC also allowed instructions to be variable in length (i.e. 4 bytes versus 10 bytes), which allowed the software instructions to not take up any more room than was necessary.

However, as memories became larger and larger, the memory footprint of software became less and less of a concern and instruction throughput, how many instructions a processor can process in a second, became an industry focus. RISC is very much aligned with this shift. The design philosophy of RISC was to reduce and standardize the instructions in the ISA. For example, instead of a single instruction that can read data from memory, add two numbers, and write the result back to memory, a RISC processor would require a load instruction, an add instruction, and finally a write instruction. But by reducing the number of instructions supported, the circuits needed to decode the instruction became simpler. And by standardizing instructions, such as disallowing variable length instructions, designers can introduce innovations such as pipelines. Processor pipelines are analogous to assembly lines, and allow multiple instructions to be operated on simultaneously. In short, RISC designers believe a simpler ISA will enable a faster processor.

Intel faced many challenges from RISC based designs: Sun's SPARC chips, Hewlett-Packard's PA-RISC series, Motorola's PowerPC, etc . . . The RISC versus CISC debate also played out in academia, with much of the literature touted the performance advantages of RISC based designs. In an industry driven by performance, this would seem to be a decisive blow to Intel.

So why do all PCs today ship with Intel processors instead of PowerPC chips? While there was much academic debate about the benefits of RISC versus CISC, it ultimately came down to a business decision. Microsoft and Intel were always careful to ensure backwards software compatibility, both in the ISA and the operating system. There was just too much industry investment in the x86 based PC to justify switching. (Lee, 2011)

In fact, Intel was able to close much of the performance gap through mimicking many of the innovations first introduced in RISC based designs. For example, without changing the x86 instruction set Intel broke up the long complicated CISC based instructions into simpler micro-ops, allowing them to pipeline their

processors. By the late 1990's, x86 chips were meeting or beating the performance of many of their RISC counterparts. (Mann, 1997) (Lee, 2011)

The Self-Fulfilling Prophecy of Moore's Law

Intel's pursuit of higher performance did not stop after its victory over RISC. Gordon Moore, an Intel co-founder, famously recognized that the number of transistors in an integrated circuit was doubling roughly every two years (Moore, 1965). This observation is now referred to as Moore's Law. While Moore's Law was originally in reference to the number of transistors on a chip, it soon became interpreted as predicting that processor *performance* will double. This helped stoke a tremendous amount of focus on increasing chip performance and became something of a self-fulfilling prophecy (Lee, 2011).

In the hunt for higher and higher performance, Intel introduced many innovations that, while they increased performance, achieved only declining incremental benefits. In turn these changes lead to highly complex, power-hungry designs. For example, the initial RISC pipelines were only 5 or 6 stages long, but Intel soon reached pipeline depths of 15 or 20 stages (Lee, 2011). This complexity comes with a cost.

Using the pipeline example again, processors sometimes have to throw away the instructions they are processing and start over. A longer pipeline means more instructions queued up in their pipeline, and hence a higher penalty if the pipeline needs to be flushed. As pipeline flushes have a higher penalty, designers will add more logic to try to detect and avoid the hazards that trigger them. This leads to bigger and bigger circuits, which consume more and more power all for incremental performance gains. But the market was demonstrating an appetite for higher performance, and this lead to tremendous pressure on engineers to keep the performance gains coming (Lee, 2011).

Around 2003, Intel hit what is known in engineering circles as the 'power-wall'. The power-wall refers to the amount of heat that a processor core can dissipate. Higher performance means more heat that must be dissipated to maintain the operating performance of the chips. As silicon circuits get hotter, their performance begins to degrade dramatically. As Intel was running out of ways to increase processor performance, it elected to begin designing chips with multiple processor cores on them. With each core below the power-wall, Intel found they could keep increasing the theoretical performance of their products (Lee, 2011) (Patterson, 2010)

Multi-core chips are predicated on the ability of software to take advantage of the parallel processing capacity. Theoretically, software should be able to break up tasks into operations that can be conducted simultaneously on different processor cores. However, effective use of parallelism has proved notoriously difficult for compilers and software writers. Parallel computing has been around for nearly 50 years, and there are countless also-rans who tried to capitalize on parallel processing. There are only a handful of success stories, where the applications map very well to parallel processing, such as the rendering of computer animated movies or weather simulations. Yet, as a whole it would seem that software developers are not prepared. (Patterson, 2010) Moore's Law marches on, but Intel may have finally surpassed software's ability to use the additional performance.

Chapter 7 - A PC on Every Desk

The Beginnings of the PC Revolution

While the last chapter points to the IBM 5150 as the advent of the modern PC, the roots of personal computers go back to the 1960s. On December 9th, 1968 Douglas Engelbart gave a demonstration of a project that is now known as “The Mother of All Demos.”¹ In this demonstration, Engelbart presented a vision of what a personal computer could look like, and marked the debut of the mouse, “what you see is what you get” text editing, hyperlinks, text and graphics on the same screen, and even a program that looks remarkably like PowerPoint.

Though much of what Engelbart demonstrated looks very familiar to a modern PC user, it was radically different from what computing looked like in the 1970s. Mainframes, what IBM called “Big Iron”, typically did not have interactive interfaces and would take up entire rooms. Minicomputers, such as the famous DEC PDP-8, reduced the size of machines considerably, but were still primarily used for computationally intense work, not for day-to-day office work. The closest product to Engelbart’s vision would be the standalone word processors, such as the Wang 1200, but even they did not encompass the breadth of functionality envisioned in Engelbart’s demonstration.

But work continued apace on the personal computer. The late 1970s is littered with kit computers, such as the Altair computer kit and the RadioShack TRS-80, which were popular with technicians and hobbyists. Apple was founded in this tradition, with Steve Wozniak and Steve Jobs building computers in and selling them out of their garage (Jackson, 1997, pg. 202).

¹ Tweney, Dylan “Dec. 9, 1968: The Mother of All Demos.” *Wired*. December 9, 2008. < http://www.wired.com/science/discoveries/news/2008/12/dayintech_1209 > Observed on April 4, 2011.

When IBM tiptoed into personal computers, it initially targeted sales into the consumer market. As IBM's strength otherwise lay in the B2B market, the 5150 personal computer was distributed through a deal with Sears (Jackson, 1997, pg 203). Yet precisely because IBM was known for its business products, the IBM PC found an enthusiastic market in office sales. After a sluggish start, IBM's sales took off in 1982 (Reimer, 2005).

Though PCs first found their footing in the business world, the low price point allowed a rapid expansion into the home market. IBM made several attempts to regain control of the IBM-PC market, such as their release of the PC jr in 1984², but by that point the PC market was clearly in control of the clone manufacturers. Much of this success was due to the popular Lotus 1-2-3 application. Compaq realized the importance of the office market, and made a strategic partnership with Lotus 1-2-3 creator Mitch Kapor to sell their machines with the famous spreadsheet program (Wilcox, 1998). By 1986, IBM and the various clone manufacturers who conformed to the X86 ISA collectively crossed the 50% market share threshold (over proprietary alternatives offered by Commodore, Amiga, and Apple) and never looked back. (Reimer, 2005)

The PC Onslaught

Dating back to the founding of Microsoft, Bill Gates and Paul Allen had a vision of computers becoming ubiquitous, perhaps best epitomized by their slogan "A PC on every desk."³ Much of the growth PCs experienced in the early days was in this spirit, with computers finding their way into contexts and environments where computers had never been used before. Yet the PC also proved to be something of an invasive species, displacing industry incumbents such as IBM itself (which later

² IBM Archives: 1980s < http://www-03.ibm.com/ibm/history/history/decade_1980.html > Observed on March 28, 2011.

³ Microsoft's Tradition of Innovation < <http://www.microsoft.com/about/companyinformation/ourbusinesses/profile.mspx> > Observed on April 4, 2011.

diverted this business to Lenovo), Digital Equipment Company (DEC), and Wang Laboratories. While these companies were correct that the initial personal computer could not compete with word processors, minicomputers and mainframes, they failed to recognize the rapidly growing threat. PCs were benefited from Moore's law more than any other device, and it wasn't long before the PC was in the same performance class as its beefier cousins. (Haynes, 1994)

The word processor was perhaps the first victim of the PCs success, effectively killing the market within eight years (Haynes, 1994). The PC even chased Wang from its lucrative hi-end markets, such as law firms who replaced their sizeable investments in word processing systems with cheap, networked PCs (Nash, 1993). With its core business eroded, Wang Laboratories was forced to file for bankruptcy in August of 1992.⁴ The minicomputer was the next victim of the PC, and not one of the major minicomputer manufacturers, such as Data General, Apollo Computer, and Prime Computer, was successful in the personal computer business (Christensen and Overdorf, 2000). No event captures this like the sale of DEC to Compaq in January of 1998⁵. DEC, who gave birth to the minicomputer with the PDP-8 and became a giant, was now owned by the company that created the first IBM-PC clone.

Not content with pushing out word processors, PCs have steadily encroached on the territory of mainframes. Again Compaq was a pioneer: in 1989 it released Systempro, the first PC built specifically to target the mainframe market (Martin, 1995). Initially Systempro was slow to take off, as it lagged most mainframes in performance. Historically there has always been a performance gap between PCs and mainframes, but PC manufacturers made serious inroads against mainframes through a combination of lower price and increasing PC server performance (Vijayan, 1995). Today "Big Iron" mainframes still exist, but they compete directly

⁴ New York Times. "COMPANY NEWS; WANG STOCK AND BOND TRADING TO BE HALTED." September 18, 1993.

⁵ Wood, Bob "Digital Stock Price Jumps In Wake of Compaq Deal." *Newsbytes*. January 26, 1998.

with machines whose lineage is in PCs. Intel's recently announced Xeon 7500 processor is specifically tailored for the enterprise computing market. Ironically, the biggest victim of the Xeon 7500 may be Intel's own Itanium line, which was a clean-slate architecture co-designed with HP that targeted servers and mainframes. (Clark, 2010)

While PCs have traditionally used Microsoft's operating system, the open source operating system Linux has been a key enabler for the PCs in the high-end server markets. For many years, the mainframe market was populated by several variants of the Unix operating system, such as Sun's Solaris, HP's HP-Unix, or IBM's AIX. Linux is a free open-source operating system that while it was originally developed on the PC, was very similar to Unix. The combination of Unix-like features and an unbeatable price point led many customers to adopt Linux for server applications. IBM famously embraced Linux in the early 2000s, committing to support Linux on its systems. This allowed IBM to save on development costs while still having enterprise level performance (Ante, 2001). The combination of Linux and PC servers proved to be such a potent combination that Sun Microsystems was forced to start giving away Solaris for free to preserve its market share. (Schofield, 2004).

The Commoditization of the PC

Looking back on the past thirty years, the story of the PC has been a story of importance of the price-performance curve. Higher performance at a lower cost is a winning combination. In each case of word processors, minicomputers, and mainframes, the incumbents believed that their products provided superior performance and the personal computer did not threaten their business. But Moore's Law has been unrelenting, and as the PCs performance has increased customers have consistently switched to the lower priced PC based solutions.

One has to look no further than Dell for importance of cost in the PC market. By cutting out the middleman and through efficient supply-chain control, Michael Dell

was able to take his company from his dorm room to the top of the PC market. By 2004, Dell was dominant PC manufacturer, with rivals struggling to match their production system (Grennell and Muise, 2010). With companies like Dell relentless driving down prices and the PC conquering all the established markets, has it become a commodity?

There is some evidence to support this. The past decade has seen mergers of former giants, such as the merger of HP and Compaq, and other players divesting themselves of their consumer PC divisions, as when IBM sold IBM PC to Lenovo⁶. Perhaps the best evidence for the communization of PCs is that many of the large PC vendors now look to services, rather than market share or sales, to fuel their corporate growth. IBM is the most prominent example of this, and their emphasis on services was central to their turnaround, but it is also a critical also a critical part of the strategies at consumer focused companies like Dell and Gateway (Burrows, 1999) (Grennell and Muise, 2010).

Now that the PC is ubiquitous, it is understandable that companies are looking to new opportunities peripheral to the personal computer. But what does the future hold for a company like Intel that has been at the core of the PC ever since its introduction 30 years ago?

⁶ "Lenovo Buys IBM PC For US \$1.25B." *China Daily*. December 9, 2004.

Chapter 8 - The Semiconductor Ecosystem in the PC Era

From Vertical to Horizontal

One of the more dramatic consequences of IBM's choice to use off the shelf parts and the subsequent spectacular growth was the segmentation of the computer industry. Before the PC, vertically integrated companies populated the computer industry. Companies like IBM and Wang Laboratories designed and manufactured all the components in the computer, designed the operating system and software, and finally were responsible for sales. This is shown in the following image.

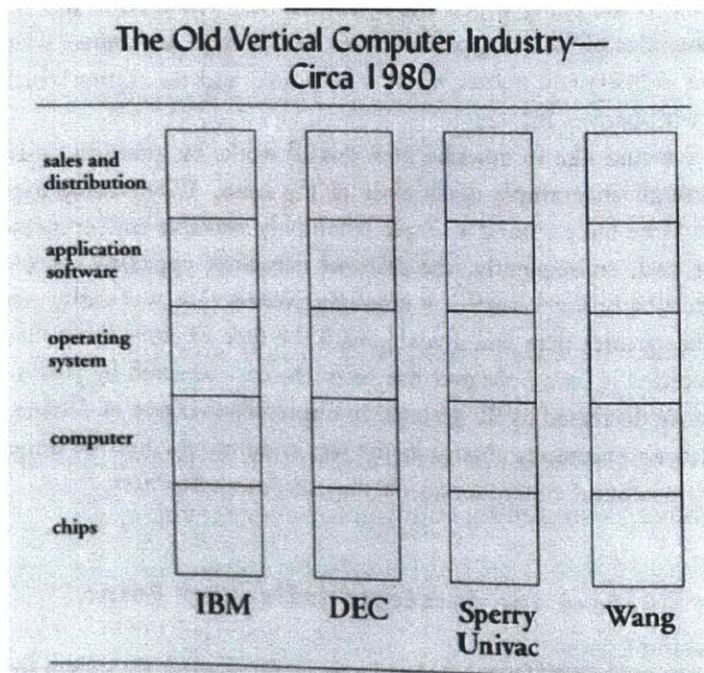


Figure 6 - The Vertically Integrated Computer Industry¹

Over time as the PC became more and more successful, the industry coalesced around a very different structure. Companies like Intel and Microsoft were able to carve out huge chunks of the value chain for themselves, and companies like Compaq and Dell concerned themselves primarily with the assembly of machines. The formally vertically integrated companies either fragmented from the

¹ Image Source: Grove, Andrew. *Only the Paranoid Survive* 1996 Page 40

competitive pressure or left the business altogether. Today, the industry has what many call a horizontal structure, as depicted in the following image.

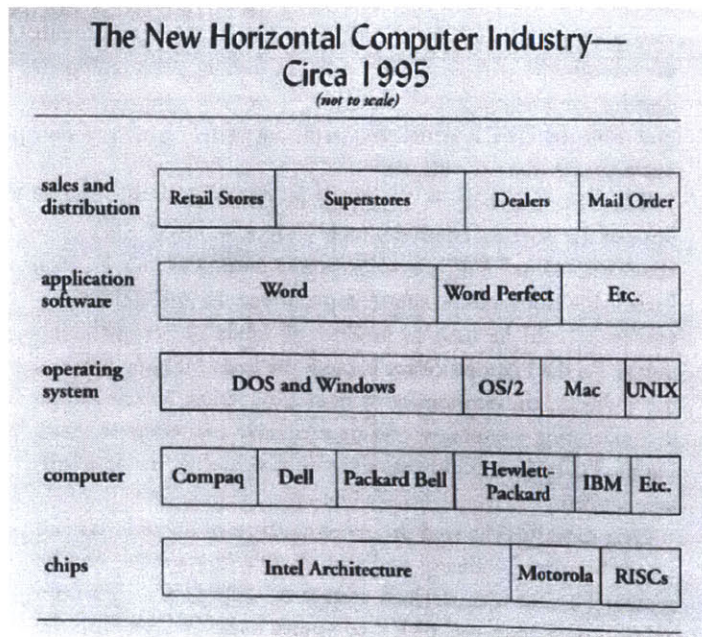


Figure 7 - The Horizontal Industry²

For many companies, including Intel, this horizontal solution was beneficial. It allowed firms to focus on the areas where they could differentiate from their competition in the same horizontal band, while also allowing them to choose the best in class solution from their suppliers. In the vertically integrated days, in order to compete, a company had to have a compelling solution for every level.

Increasing Performance, Decreasing Costs

Intel found it could differentiate through increasing processor performance and driving the performance-cost curve down. Intel has pursued both of these relentlessly. Figure 8 depicts the relative performance of Intel processors overtime. Please note that it is a *logarithmic* scale, and processor performance is increasing exponentially.

² Image Source: Grove, Andrew. *Only the Paranoid Survive* 1996 Page 42.

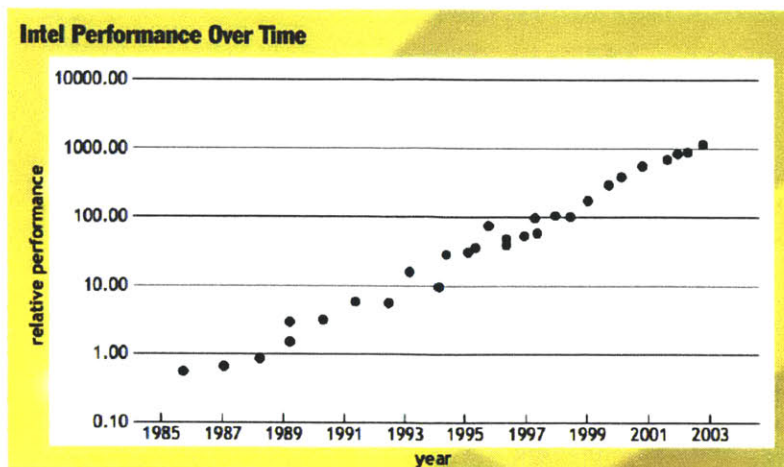


Figure 8 - Relative CPU Performance³

This trend is truly remarkable. Every decade, Intel has been able to increase performance by a factor of 100. And as noted in previous chapters, this has not come with the same cost increases. If you track the performance-cost ratio, Intel's efforts have steadily driven down the cost-per-MIPS.⁴ Remarkably, it has the same factor of 100 over a decade.

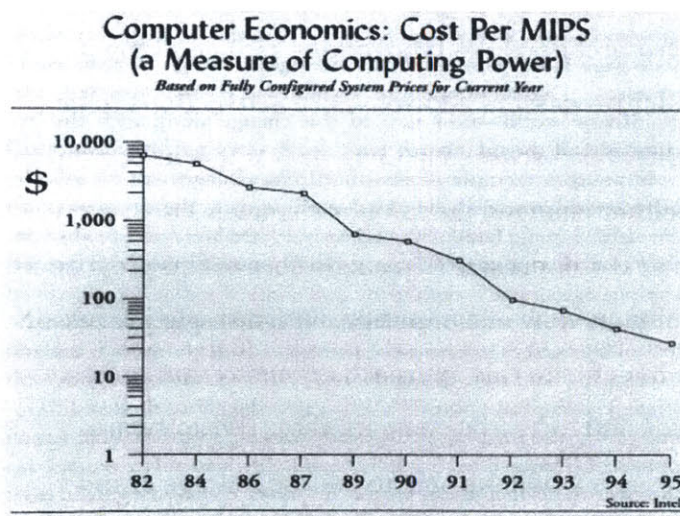


Figure 9 - The Price-Performance cost trend⁵

³ Image Source: Olukotun, Kunle and Lance Hammond. "The Future of Microprocessors." *ACM Queue*, September 2005. Page 28.

⁴ MIPS is a standard metric of CPU performance. It stands for Millions of Instructions Per Second. However, as CPUs may have a different set of instructions, it is an imperfect way to compare CPUs that implement different ISAs.

⁵ Image Source: Grove, Andrew. *Only the Paranoid Survive* 1996 Page 63.

This dramatic price-performance curve is the engine behind the PC's dramatic growth and success over the past 30 years. In product category after product category, PCs were able to demonstrate similar or better performance for a lower price.

The reasons behind the performance gains of processors are twofold. First is the trend to smaller and smaller geometries: the physics of semiconductors means that transistors can be switched on and off quicker as they are made smaller. If you take a digital circuit where the minimum transistor feature size is 130 nanometers and then port it to a process where the minimum transistor feature size is 90 nanometers, you can run the circuit at a faster frequency, increasing its performance. This is known as CMOS scaling. (Nowak, 2002) (Haensch et al, 2006)

The second way Intel has increased processor performance is by introducing and exploiting parallelism in the micro-architecture. Innovations as such as pipelining, register renaming, and superscalar processors allowed more instructions to be processed in parallel (much of this was discussed in Chapter 6). It is important to note is that these innovations neither changed the ISA nor changed the programming paradigm. As the CPU still presented a single core to the programmer, Intel complied with the Von Neumann view of computing, with a single processor and a monolithic memory. The benefits to this approach are many, but most importantly it ensured backwards and forwards compatibility. Old software originally written on older processors would be able to take advantage of the performance increases in newer processors, and computer manufacturers could switch to newer processors without fear of obsolescing existing software.

However, this micro-architecture approach to parallelism is running out of steam. Parallelism does not scale indefinitely, as almost all programs require that some number instructions be executed in order. Adding the ability to process more instructions in parallel does not mean that software will be able to exploit the

capacity. Figure 10 plots the relative performance per clock cycle, which is one way to capture how much added benefit parallelism is introducing.

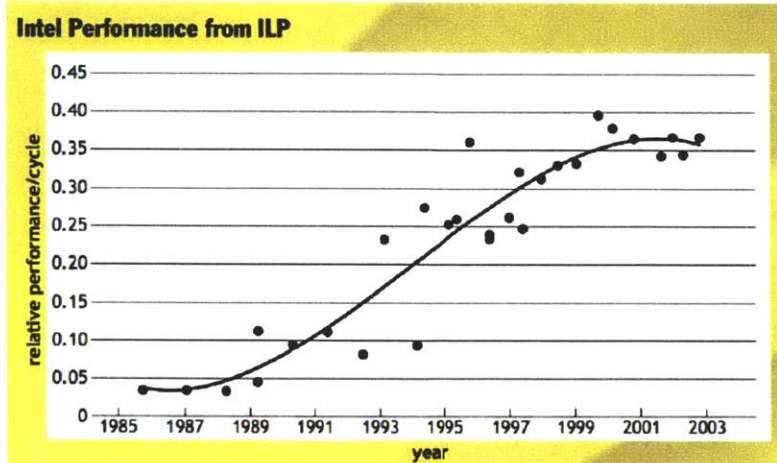


Figure 10 Relative Performance per clock cycle⁶

The incremental benefits of each new innovation are decreasing, and Intel may have fully exploited the benefits of parallelism in the micro-architecture.

The Advent of Multi-core

In addition, Intel also has to contend with the power wall, which is a way of saying that Intel cannot increase processor core performance without exceeding the ability to cool the core economically. In short, adding more parallelism or adding more complexity and transistors may mean that the processor core stops working as its temperature rises to untenable levels. The logic here is inescapable, the higher performance of a chip, the higher its power needs, as can be seen in the trend of the

⁶ Image Source: Olukotun, Kunle and Lance Hammond. "The Future of Microprocessors." *ACM Queue*, September 2005. Page 29.

power used by Intel's products.

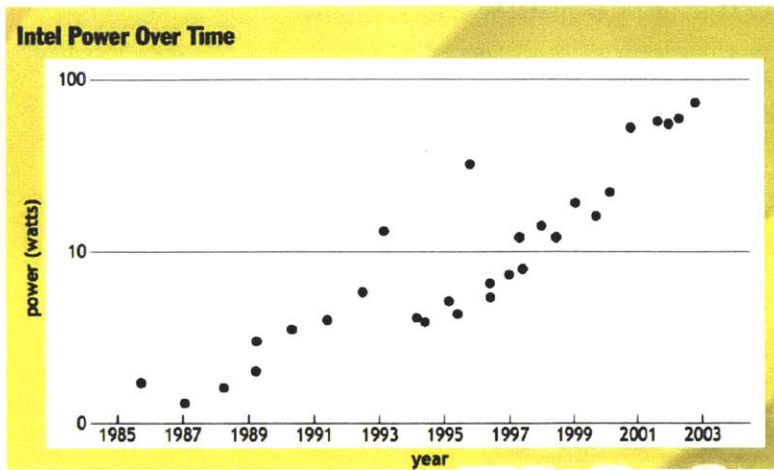


Figure 11 - Intel Power Over Time⁷

Traditionally, Intel could have worked around the power wall by relying on smaller process geometry to reduce the power consumption of an individual transistor, however in the past decade Intel has shrunk transistor dimensions to the point where passive and leakage currents now dominant the power consumption of a processor core. This will be discussed in greater depth in the following section.

Far from letting the power wall slow their pursuit of superior chip performance, Intel has responded by introducing multi-core designs. This allows them to claim greater theoretical chip performance. But even though the ISA is preserved, it is a fundamental shift from the Von Neumann model of a computer. It is not clear at this point if software will be able to crack the multi-core nut, so to speak, and take advantage of the available processing horsepower.

The Power-Wall

Power a driving concern in modern semiconductor design. It has already driven Intel to switch to multi-core, and power-performance trade offs drive much of the innovation in microprocessors today. It is impossible to overstate how important

⁷ Image Source: Olukotun, Kunle and Lance Hammond. "The Future of Microprocessors." *ACM Queue*, September 2005. Page 30.

power consumption is and we also must understand that power is a multi-dimensional problem.

One way to slice the problem is to look at the power performance per operation and the power density (power consumed per unit area of silicon). For many years Intel and other semiconductor manufacturers were able to improve power performance while keeping power density relatively constant. (Nowak, 2002) (Haensch, 2006) While as we saw in a previous chapter that Intel's chips have had increasing power consumption, this relationship allowed them to pack in tremendous performance benefits with only trivial costs to overall power consumption.

Much of this was due to the benefits CMOS scaling. If you shrank a CMOS circuit by a factor of α , the power performance per operation would be improved by α^3 with constant power density. However, when CMOS manufacturers were no longer able to lower the threshold voltage with each generation of scaling (the reasons for this are beyond the scope of this thesis), the math of power density changed significantly. Today if you shrink a CMOS circuit by a factor of α , the power performance per operation improves by only a factor of α , with power density rising by a factor of α^2 . In the following figure, notice how the power performance curve has bent, providing less diminishing benefit with each successive generation, and the power density trend has dramatically flipped directions after passing the 130 nm gate length node.

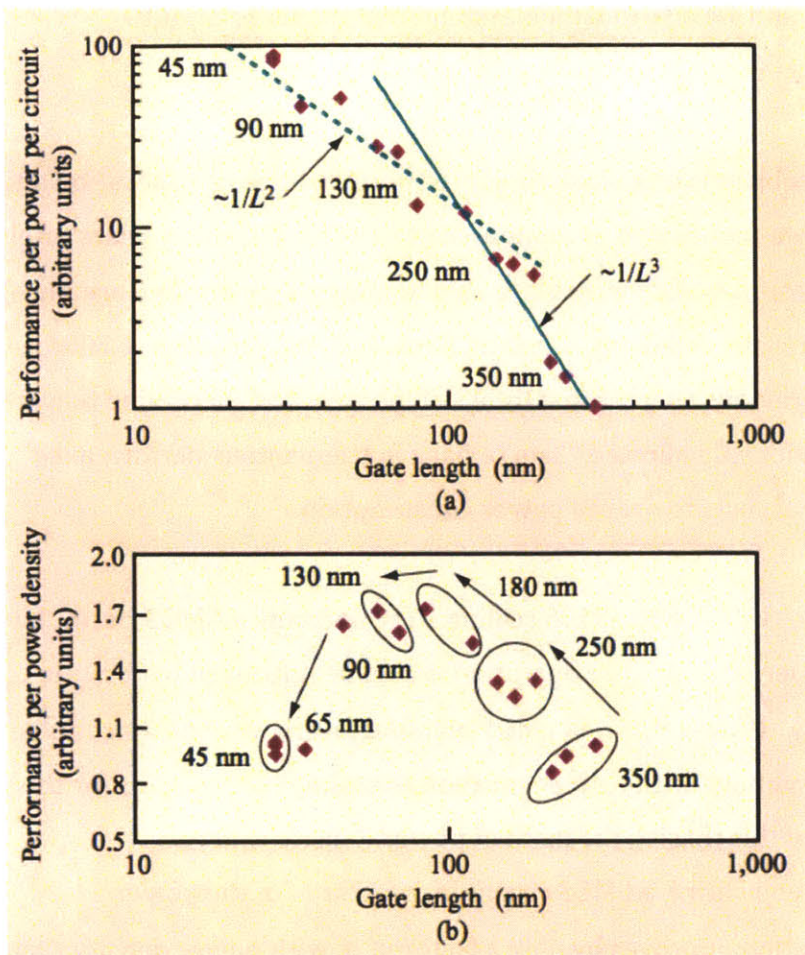


Figure 12 - Power Performance and Power Density Trends⁸

With power density, an important distinction to make is between active power and passive power in semiconductors. Active power refers to the power consumed by a circuit in its normal operation, with transistors turning on and off. Passive power refers to the power consumed by a circuit even if it is idle and not active. It is the increase in passive power that is ultimately responsible for the dramatic turnaround we see in overall power density. Even when a CMOS transistor is in the off state, a small amount of current can leak through. As the transistor dimensions get smaller and smaller, the leakage current can grow to a greater proportional amount of total current. And finally, when the sub-threshold voltage is not reduced as the process

⁸ Image Source: W Haensch, E J Nowak, R H Dennard, P M Solomon, and et al. "Silicon CMOS devices beyond scaling." *IBM Journal of Research and Development* 50.4/5 (2006) Page 342.

geometry is shrunk, the leakage current consumes more and more of the total power⁹. Ultimately, this means that we have reached the point where passive power, which used to be so low that it was literally negligible, now rivals active power for total power consumption. Figure 13 plots active power versus passive power density for various process geometries and projects that the two trends will cross somewhere around 20 nanometers (Nowak, 2002). As a comparison, Intel announced that two of their fabs are being upgraded to their next generation process of 22 nanometers.¹⁰

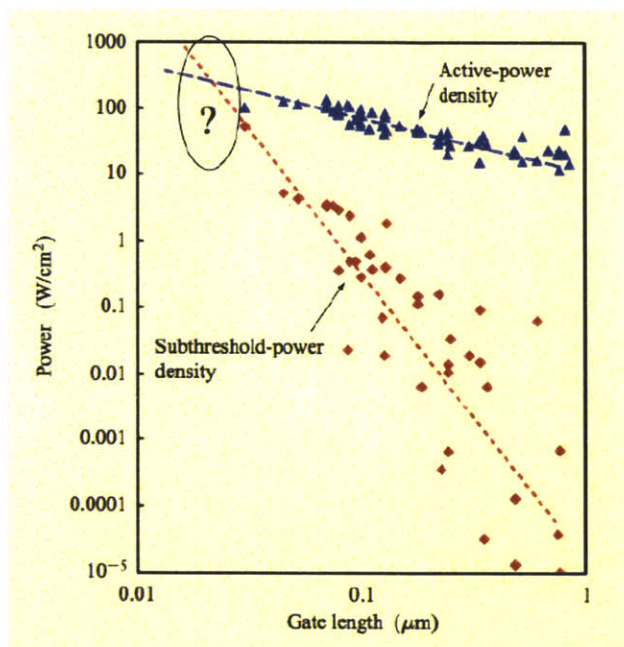


Figure 13 - Active Power and Passive Power (referred to here as Subthreshold power)¹¹

What this all means is that CMOS scaling, as we know it, is done. The power consumption in chips is rising too fast to cool the chips economically. The semiconductor industry has hit the power-wall.

⁹ Power = Current * Voltage

¹⁰ Intel Newsroom. <

http://newsroom.intel.com/community/intel_newsroom/free_press/blog/2010/10/21/moores-law-around-the-world-in-bricks-and-mortar/ > Observed April 14, 2011.

¹¹ Image Source: E J Nowak. "Maintaining the benefits of CMOS scaling when scaling bogs down." *IBM Journal of Research and Development* 46.2/3 (2002). Page 173.

The Economics of Semiconductor Costs

Anyone in semiconductor industry, whether an IDM (Integrated Device Manufacturer) like Intel or a Fabless design company¹², is subject to some powerful economic forces. To begin, the cost of building a new foundry, which is necessary every time a company wants to move to a new smaller process geometry, is growing exponentially. The graph in Figure 10 projects that a new plant built today will cost nearly \$10 Billion. These high costs are exacerbated if a company is building a cutting edge process plant, as the R&D costs associated with developing a brand new process are exponentially higher than if one chooses to be a “follower” (Kumar, 2008). With the required capital expenditures, it is no wonder that more companies are choosing to shed themselves of their fabrication facilities.

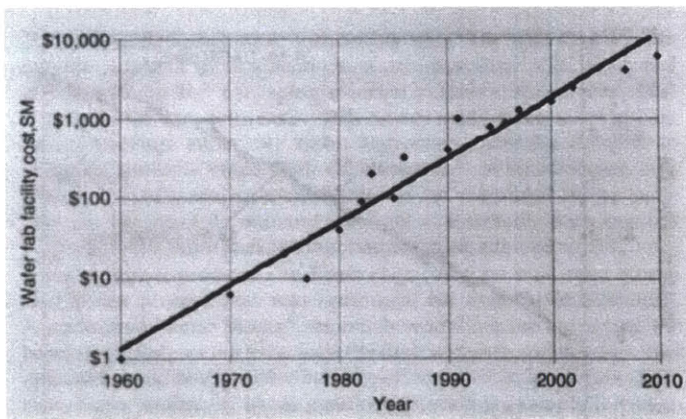


Figure 10 - Cost trend of fabrication plants¹³

In addition to capital expenditures, the size of a semiconductor circuit design has a tremendous effect on its cost structure. All semiconductor chips are cut from standard sized silicon wafers, which the fabrication process is designed around. When creating a design, you can get more chips per wafer by making the design

¹² “Fabless” is an industry colloquialism. A plant where semiconductor chips are fabricated is often referred to as a “fab”. Therefore a company who does not own such a plant is known as “fabless”.

¹³ Image Source: Kumar, Rakesh. *Fabless Semiconductor Implementation*. 2008 Pg 16.

smaller, or you can pack more transistors (and therefore functionality) at the cost of fewer chips per wafer. Figure 14 is a representation of this choice.

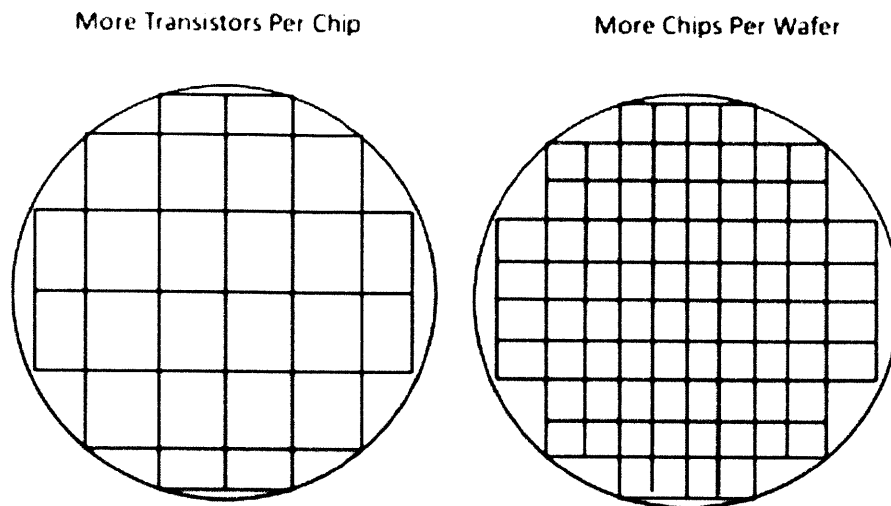


Figure 14 - A smaller die size yields more chip per wafer

Initially it is more cost effective to pack more transistors into a chip. You get proportionally more performance and functionality in each chip, and you have less of the variable costs associated with each individual chip, such as assembling, packaging, and testing. However, this has a limit. Each wafer has a certain number of defects introduced throughout the fabrication process. The larger your chip, the more likely it is that a defect will land within the boundaries of a chip and will proportionally affect a greater number of your chips. This proportion good chips that come out of a fabrication process is known as the “yield”, and having too large of a chip can adversely affect the yield.

The result of these two opposing forces is a u-shaped cost curve. There is a significant economic incentive to size your chips to be at the knee of the curve. As processes mature, these costs curves will trend downwards, but they will not lose their shape. Figures 15 and 16 show the cost curves over time for a 65-nanometer process and a 45-nanometer process.

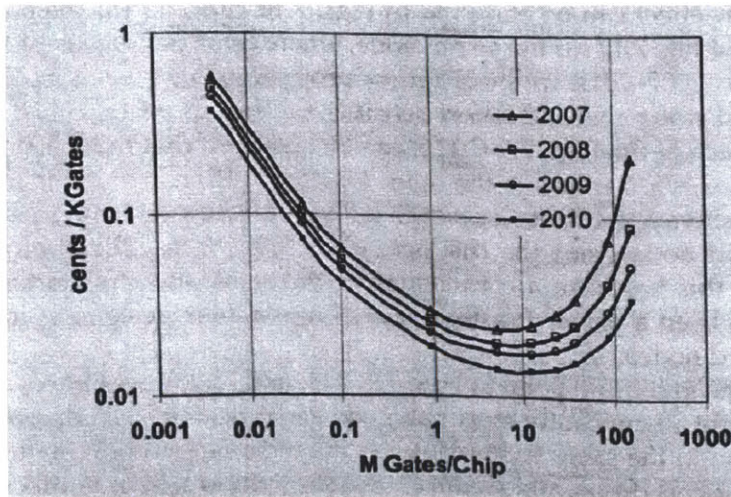


Figure 15 - 65nm Process Cost Curve¹⁴

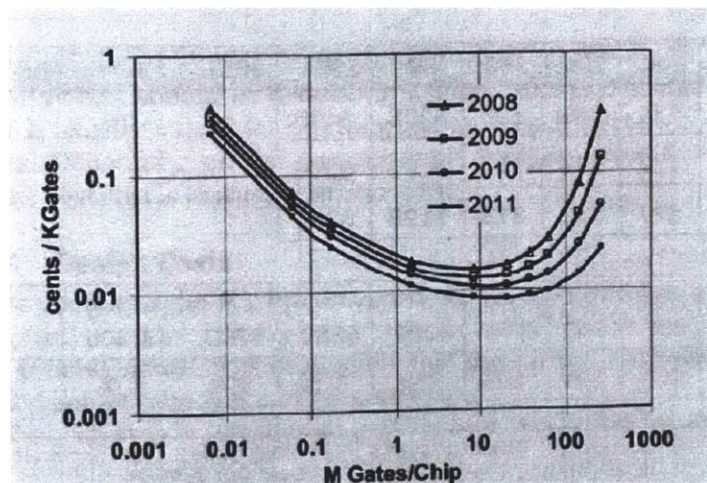


Figure 16 - 45nm Process Cost Curve¹⁵

It is interesting to note that the knee occurs roughly around the same place, 10 million gates (roughly the equivalent of 40 million transistors). For reference, the Intel Pentium D Processor 900 was a product of a 65 nm process with 376 million

¹⁴ Image Source: Kumar, Rakesh. *Fabless Semiconductor Implementation*. 2008 Pg 232.

¹⁵ Image Source: Kumar, Rakesh. *Fabless Semiconductor Implementation*. 2008 Pg 233.

transistors and the Intel Core 2 Duo E8300 was made with a 45 nm process and has 410 million transistors.¹⁶

The staggering size of Intel's chips is not a surprise. It is a direct consequence of their relentless pursuit of raw chip performance. A goal of increased serial speed leads to longer pipelines, look-ahead logic, and out of order operation. Incidentally the more complex the pipelines became, the larger the penalties became for a pipeline drained, which required sophisticated algorithms for branch prediction. In addition, processor speeds have long been diverging from the speed of memory which has not been improving at the same rate, which incurs a huge performance cost for cache misses, so Intel began adding progressively larger and deeper caches to their chips. All of this adds progressively more complexity and exponentially more transistors. It is no surprise then that Intel finds itself an order of magnitude away from the optimal die size. This has consequences in the forms of additional die fabrication and processor operational costs.

The Economic Exhaustion of Intel's Business Model

Before continuing this chapter's exploration of the semiconductor ecosystem, I want to highlight that these trends taken together indicate that Intel's historical trajectory is no longer sustainable. Intel's success over the past thirty years has been predicated on a tight coupling between processor design and wafer fabrication process technology on a massive scale. Intel's relentless driving of Moore's law forward has achieved incredible performance gains while simultaneously dramatically lowering the cost-performance curve. These trends can be seen clearly in Figures 8 and 9, and they are the primary reason that the PC took over computing. Year after year, the PC was capable of more and more computing applications at a superior cost. The performance gains from changes in the processor architecture

¹⁶ Intel Microprocessor Quick Reference Guide, <
<http://www.intel.com/pressroom/kits/quickreffam.htm>>. Observed on April 8, 2011.

eventually petered out (Figure 10) but Intel was still able to achieve gains through process improvement and CMOS scaling.

However processor throughput (i.e. MIPS) is only one axis of technological performance, and Cost per MIPS is only one way to do to a cost-performance analysis. And now these other dimensions have caught up with Intel. While Intel is focused in improving MIPS, Figure 11 shows us 18 years of consistent increases in power consumption. While CMOS scaling still had runway, Intel could effectively ignore power consumption while it provided customers with higher and higher processor throughput. But Figures 12 and 13 illustrate the power wall dilemma quite dramatically. Intel will now have to find radical new innovations and expend an exponentially increasing amount of engineering effort to keep process line-width shrinking. And even if Intel does manage this feat, it is unclear if it will provide the same benefits like CMOS scaling use to with each successive generation.

If Intel cannot create higher and higher performance processors, can it continue to maintain its processor prices? Figures 15 and 16 demonstrate the knee-shaped cost curve of circuit size. There is tremendous economic pressure to be at the knee of the curve and the size of Intel's premiere products are well beyond this point. Intel found that customers were willing to pay a premium for more MIPS, but if Intel has run afoul of the power-wall, we should expect prices to erode. Finally, we must consider the explosive growth in costs to develop a new process, which Figure 10 depicts so dramatically. Can Intel justify spending \$10 Billion on a new process with eroding processor prices?

In short, Intel's business model has run out of steam, regardless of the threats it faces from the embedded space, ARM, and low power computing. What is so concerning for Intel is that these additional challenges are coming precisely at a time when it is most vulnerable.

The Rise of the Independent Wafer Foundries

During the past thirty years the semiconductor industry went through a transition similar to the PC industry's vertical to horizontal shift. The seeds for this transition were laid in the late 1970s. It first began with a recognition that the design of a semiconductor and the fabrication of a semiconductor could be decoupled (Baldwin and Clark, 2000), and subsequently the respective responsibilities were divided between different teams in an organization. At the same time, large vertically integrated companies set up factories in the Far East for assembly and packaging operations for semiconductors. These plants served as a "second source," which is to say that they provided both cost and operational flexibility. A natural follow-on was to move up the value chain and establish fabrication in the Far East, which the Taiwanese government and Philips did with the founding of the Taiwanese Semiconductor Manufacturing Company (TSMC) in 1987 (Kumar, 2008). Since then, several other independent foundries have sprung up.

The independent foundries initially served as a "second source" for the vertically integrated companies, but they enabled the growth of a new type of company, the fabless semiconductor company. Instead of being a "second source" for companies like IBM that had their own fabrication capabilities, independent foundries like TSMC are the primary manufacturer for designs created by fabless semiconductor companies. Since their advent in the early 90s, the fabless segment has grown at an impressive clip. In 2006 fabless semiconductors claimed 20% of total revenues for the semiconductor industry, and since 1994 they have an aggregate CAGR of 26% as compared to 6% for integrated semiconductor companies (Kumar, 2008).

Furthermore, the trend towards vertical segmentation has found its way into the design itself. Today it is common for fabless semiconductor company to only design a portion of a chip, and license the rest of the design from 3rd party vendors. This has come to be known as the "design ecosystem" (Kumar, 2008). One of these vendors, ARM Holdings, will be discussed in further detail later in this chapter.

Overall, this evolution is a remarkable change that mirrors the dramatic change in the PC industry. Figure 17 provides a timeline of the segmentation of the semiconductor industry.

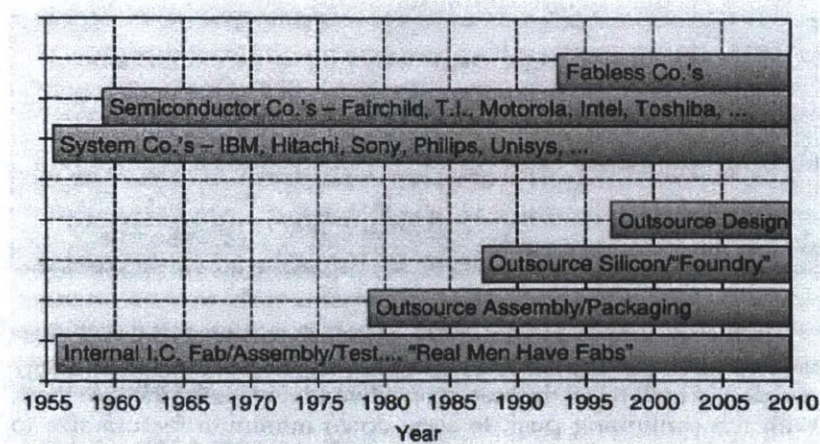


Figure 17 Evolution of the Industry¹⁷

The System on a Chip

Most fabless semiconductor companies make chips tailored for specific consumer applications, often referred to as Application Specific Integrated Circuits (ASIC), rather than a general-purpose processor like Intel does.

In addition, many modern ASICs try to pull functions that might have previously been scattered across multiple chips on to the same piece of silicon. For example, most PCs ship with an x86 processor and a memory controller on a separate chip. An ASIC inside of a smartphone would have these two functions integrated onto the same chip. ASIC designs that pull system functions onto the same chip are known as System-on-a-Chip designs (SoC).

The motivation behind this integration was originally to reduce system costs. These ASICs were initially very small and designers could make more cost effective designs through integration. However as we discussed above, there is cost ceiling to this

¹⁷ Image Source: Kumar, Rakesh. *Fabless Semiconductor Implementation*. 2008 Pg 19.

integration. There is tremendous motivation to pull in as much functionality onto a SoC, but designers also must use their gates very judiciously.

When designing a circuit, it is helpful to think of a spectrum where on one end you have general-purpose circuits, like a microprocessor, in the middle you have circuits that are well suited for a certain class of applications, such as a digital signal processor (DSP), and on the other extreme you have custom circuits designed for a niche application. Custom circuits are hands down the cheapest way to tackle certain applications and often have the best performance, but are limited in what they can do. Rather than have a custom logic block for every problem that an ASIC may be asked to solve, it may be more gate cost-effective to have a DSP, but at the same time building a custom circuit to handle a commonly occurring task may also be more cost-effective. As SoC integrates several circuit blocks, it can allow a processor, a DSP, and custom logic blocks to co-exist on the same die. A SoC approach enables ASIC designers to pick blend general-purpose and custom circuits to help them optimize their chip to be high performance while also at the most cost-effective size.

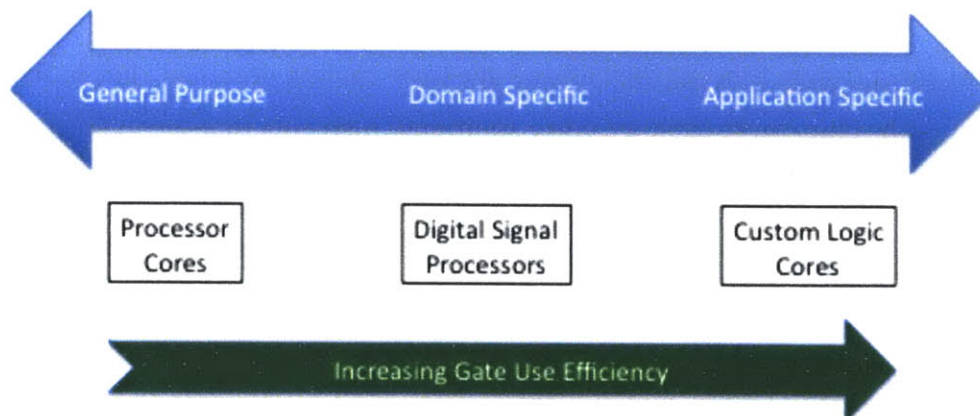


Figure 18

While the integrated memory controller was used as an example above, it is actually one of the fundamental attributes of a System on a Chip. On an Intel Processor,

memory accesses must leave the silicon for the processor, travel to the memory controller chip on a circuit board, win arbitration, and then finally reach the DRAM chip. Suffice to say that memory accesses are expensive operations in terms of time and incur a significant cost on performance. Intel softens the penalty by populating the processor with gigantic caches and pre-fetching memory before it is needed. As SoCs are founded in a philosophy of cost-reduction, designers cannot afford to dedicate the majority of the cache. Instead, designers bring the processor “close to the memory” by integrating the memory controller with the SoC. (Schaffstein, 2011)

To help illustrate the System on a Chip concept, a block diagram is shown below in Figure 19. This is an architectural representation of what is on a single piece of silicon. This system has a full blend of general purpose and applied blocks. The pure general-purpose logic is the ARM core in the upper left, labeled ARM7TDMI. This system also has a DSP processor, indicated by the Oak DSP Core. Finally, the “hardware coprocs” block refers to hardware co-processors. This and hardware accelerators are additional ways of referring to application specific logic. While the hardware co-processors are only one of twenty-seven blocks, they can potentially take the most design time, as they often have to be custom designed. For much of the rest of the chip, there exist 3rd party solutions. Also note that the majority of the blocks are connected to the ARM bus, which is an free standard protocol ARM provides.

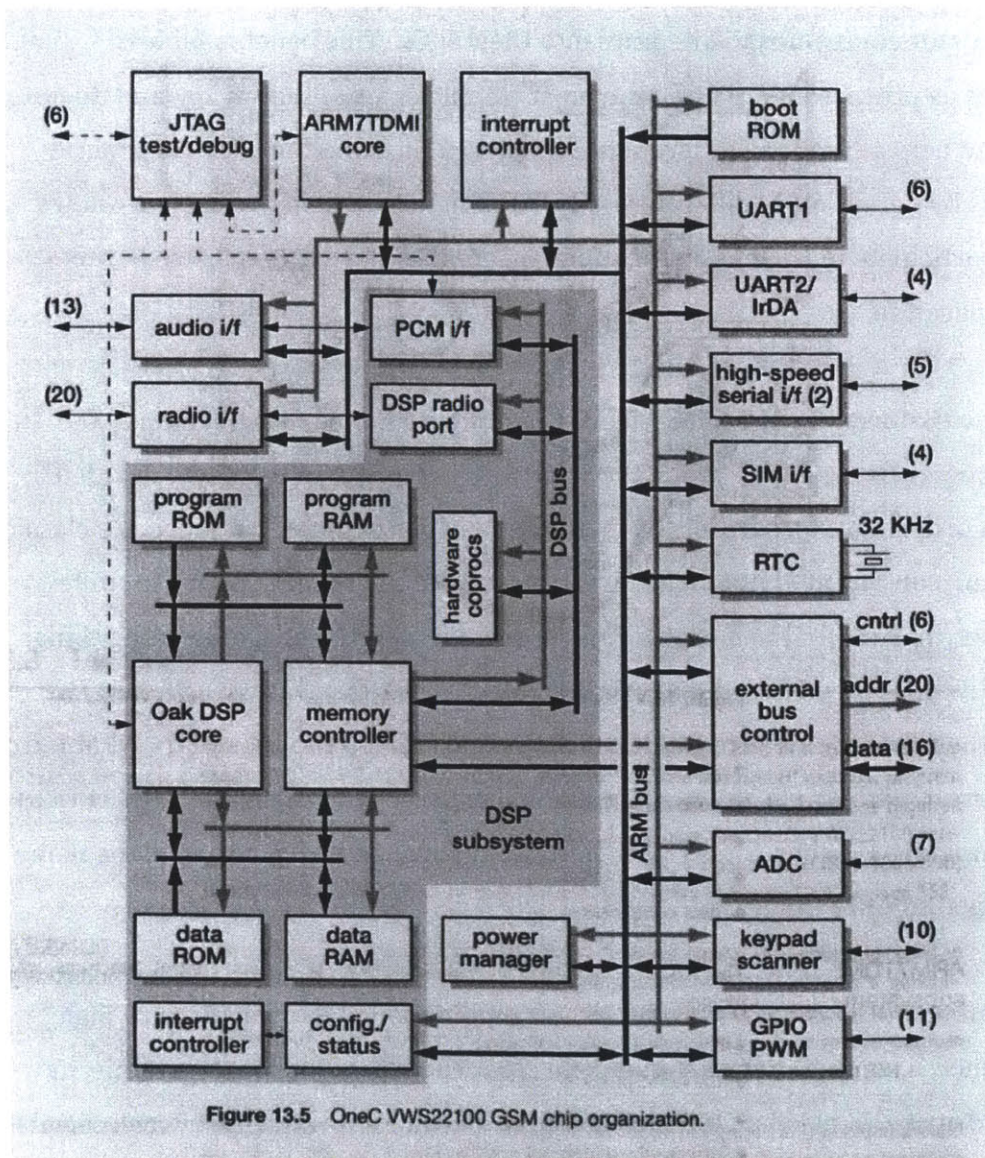


Figure 19 - A sample System on a Chip block diagram¹⁸

The Advanced RISC Machine

For a company designing SoCs, ARM Holdings is a significant partner. ARM Holdings is one of the IP vendors who participate in the “design ecosystem.” Rather than make their own chips, ARM licenses their processor cores to other fabless

¹⁸ Image Source: Furber, Steven. *ARM System-on-Chip Architecture*. 2000.

semiconductor companies to integrate into their SoCs. This benefits fabless companies as processors are often the most complex part of the silicon, and do not do much to help a company differentiate its (designing something that is general-purpose is by definition undifferentiated). ARM in turn benefits because it allows them to participate in various applications where they have no expertise in the target application.

A company that went by the name Acorn Computers Limited first developed the ARM processor between the years 1983 and 1985. The Acorn team had decided to develop a proprietary processor for their next line of machines, but faced the classic engineering constraint of not enough resources or time. To meet their schedule constraints, they elected to design a RISC processor with the goal of keeping things as simple as possible. Thus, the Acorn RISC Machine (ARM) was born. Acorn then changed the name to Advanced RISC Machine, and created the subsidiary ARM Ltd, in partnership with Apple and VLSI Technologies, to sell the use of the ARM core to other companies (Furber, 2000). When the company had its IPO in 1998, the name was changed to ARM Holdings.

ARM found its initial success was due to its combination of low power with high performance, a combination that made it a compelling product for the mobile and embedded space. It also had a relatively small silicon footprint, which was crucial for helping SoC vendors achieve their cost targets. (Furber, 2000) (Levy, 2005) ARM also offers flexibility in its licenses. Most customers simply license the IP core and ARM provides them with a completed net-list, while more ambitious customers can choose to design their own implementations of the ARM Instruction Set.¹⁹ While we cannot say with certainty what the breakdown of ARM's income is, ARM receives payments from customers from a combination of up-front license fees and per-chip royalties.

¹⁹ ARM.com. "Licensing ARM IP." < <http://www.arm.com/products/buying-guide/licensing/index.php> > Observed May 3, 2011.

With its IP licensing model, ARM enters a partnership with its customers, and its ultimate success is dependent on the success of its customers. To this end, ARM has developed a suite of tools and methodologies to help speed the adoption of ARM processing cores and to also help reduce their engineering costs. For example, ARM offers compiler and developer tool-chains to help software developers write code for ARM cores and has created a set of open bus protocols, the language by which different functional blocks inside a SoC can talk to each other. Companies who design their proprietary chips with these buses can be confident that it will be painless to integrate them with an ARM core, and customers who are integrating IP from several vendors knows that they can interoperate as long as they are using ARM's bus standard, the AMBA protocols. ARM has also developed chip design tools, such as functional models, and released them for free, all in a effort to help their customers complete their designs quicker and with less engineering effort. Through its own development efforts and through acquisitions, ARM has set out to become a "one-stop shop" for its customers. (Bray, 1999) (Furber, 2000)(*Portable Design*, 2008)

In the past decade, ARM has been spectacularly successful. This is largely because one of its biggest target markets, smartphones, has exploded in the past few years. Through a combination of engineering and market reasons that will be discussed in the few chapters, ARM is positioned to not only continue its remarkable run, but to grow into markets traditionally dominated by Intel. While the future is unclear, one must ask what is the future of Intel? If the future favors ARM, can Intel adapt? And fundamentally, can a company that has built its success on operational excellence and cutting edge process technology compete with a company that is in a different business altogether, IP licensing?

What is truly tragic for Intel is that they had a moderately successful ARM based design in their XScale product line. XScale chips were descendents of the StrongARM design, and were ARM based solutions targeted for the mobile space. But in 2006, Intel decided that it couldn't be successful in the embedded space, and

sold the XScale technology to Marvell so it could focus much more profitable x86 business. (Carson, 2006) (Schaffstein, 2011)

PART III

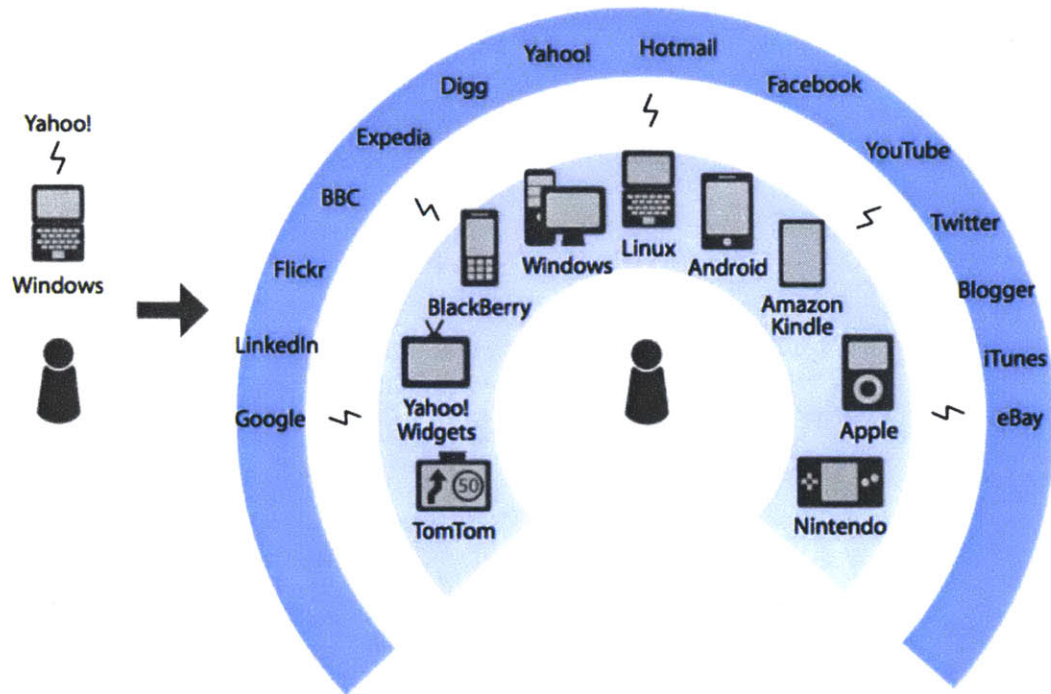
The Future

Chapter 9 – Computing Diversity

In this final third, I will lay out my prediction for the future of the computing industry and how this will affect Intel and ARM. This chapter will discuss the significant changes that are currently underway in the computing industry. The coming “Internet of Things” and significant computing power being embedded in devices, and Cloud Computing represent the two most powerful forces that are shaping what computing will look like in the future. The smartphone and the data center are the poster children of these industry trends and in Chapter 10 I will examine the design constraints of these crucial markets. In particular, Chapters 9 and 10 will emphasize the critical importance of power efficiency, and the trend towards customized System on a Chip designs. In conclusion, Chapter 11 will explore how Intel faces a myriad of interrelated obstacles that will make it impossible for it to continue operating as it has for the past three decades.

A Myriad of Devices

As discussed in earlier chapters, the PC has been remarkably successful. So much that much of computing today is done on devices that can trace their heritage to the PC. The dominance of the PC has resulted in devices, whether they are a desktop, minicomputer or mainframe, having the same underlying PC architecture. Yet recent years have seen the growth of devices, both in consumer devices and high end servers, whose architectures are divergent from the Wintel monoculture that signified the PC. (Gillett, 2010)



56888

Source: Forrester Research, Inc.

Figure 20 Consumer Computing Diversity¹

Smartphones are the best example of this shift, and the huge popularity of the iPhone with consumers has changed the way people use computers. However this shift can be seen across a variety of consumer devices: tablets, e-books, handheld gaming devices, TVs, etc... The common denominator is that these devices are small, have a network connection, and are embedded in the daily life of consumers. Even traditional devices are becoming more connected. HP now offers its consumers a range of ePrint "web-connected" printers, complete with cloud connectivity and app support. While the growth of the PC was characterized by Microsoft's "A PC for Every Desk" mantra, the growth of embedded devices will lead to networked devices everywhere, all powered by cloud-based services.

¹ Gillett, Frank "The Age of Computing Diversity", *Forrester Research*, September 16, 2010, Page 10.



Welcome
Unlock the power of your Web-connected printer!

What is a Web-connected printer?

Many of the newest printers from HP connect to the Internet, enabling these key benefits:

- **ePrint** – Use this HP feature to print from anywhere with a Web connection by sending an email to your printer's email address.
- **Print apps** – Enjoy instant access to printable Web content from your printer's control panel, for select Internet-capable printers.
- **ePrintCenter** – See your printer status, add and remove print apps, and manage ePrint settings and job history from any Web browser, at home or on the go.

In order to use these features, your printer must be connected to a network with Internet access.

[G+](#)
[f](#)
[t](#)
[p](#)
[hp](#)
[Sign In >](#)

Don't have a printer?
[Explore print apps >](#)

Learn more and buy:
[Explore Web-connected printers >](#)
[Learn more about the new DesignJet printers >](#)

Figure 21 - The app enabled printer (From the HP ePrint website)²

While smartphones are leading the charge, this trend will cut a wide swath. We will see network connectivity permeating all aspects of daily life, and all these devices will have to contend with constraints that are far removed from what traditional PCs had to contend with. Mobile devices must satisfy a new set of design constraints, namely size, weight, and battery life. If a smartphone was not large, heavy, and had only 20 minutes of operational battery life, it could hardly be called mobile. In particular, the operational battery life is a particularly vexing constraint. Laptops, in comparison, can have a proportionally much larger battery and much heavier weights are tolerated.

Because of these constraints, these new devices are not the descendents of the PC. They have grown out of the embedded space, and do not lend themselves to a single

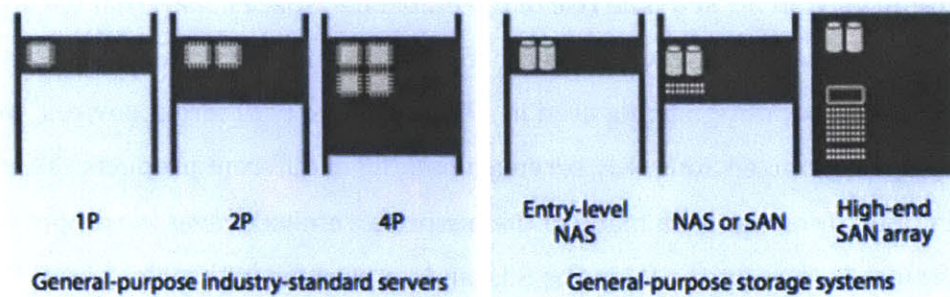
² Image Source: HP ePrint Center. < <http://h30495.www3.hp.com/> > Observed April 13, 2011.

architecture. The constraints of the smartphone are different than the design constraints of your TV set. The rise of the embedded space means that we are entering an age of device diversity, with highly specialized architectures. Instead of a single processor design being used in a huge number of different devices, we will see highly customized SoCs only serving a handful of different products. This development is on par with many of the historical seminal waves in computing: mainframes in the 60s, the PC in the 80s, and networking in the 90s. (*Growth Strategies*, 2005)

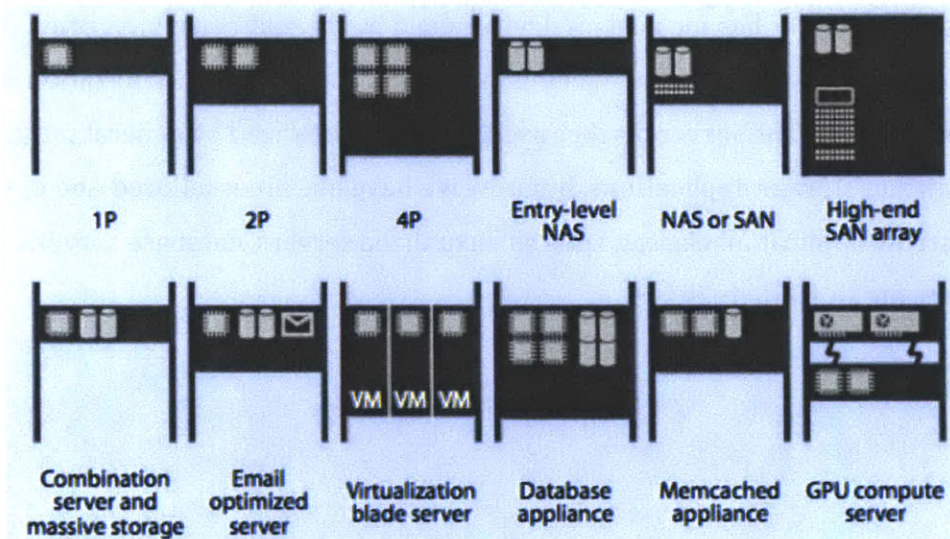
Servers, Cloud Computing and Data Centers

So far this chapter has focused on devices used by the end-consumer, often referred to as client devices. But we are seeing the diversification and specialization of servers as well. The server market used to be characterized by general purpose servers and storage applications, but now we have machines tailored and optimized for narrow application classes, such as virtualized servers, database servers, email servers, etc ... (Gillett, 2010)

Today: Most systems are standard server and storage models



Tomorrow: Diversity of general-purpose and specialized systems to handle diverse workloads



S6888

Source: Forrester Research, Inc.

Figure 22 - Diversification in Servers³

The reasons behind this specialization are numerous, but one significant factor is the huge spike in demand for data centers. Data centers are the heart of cloud computing, and the growth of cloud-based services has fueled further investments into data centers and related infrastructure. The centralization of tremendous amounts computing power introduces new design constraints, which are helping to

³ Image Source: Gillett, Frank "The Age of Computing Diversity", *Forrester Research*, September 16th 2010, Page 9.

shape the specialization of server architecture. For many data centers, they are limited in the physical footprint of the facility, which has helped fuel the trend towards server convergence. (*eWeek*, 2011) In turn, high density computing has brought forth significant operational constraints, namely powering the data center and cooling the servers, both of which factor into the overall energy consumption. Today in new or retrofitted data centers, efficient energy consumption is a primary design criterion (Cappuccio, 2010).

Macro Level Forces Shaping Micro Level Architecture

Cloud computing combined with mobile devices represents a further evolution in the architecture of networks. This evolution began with the mainframe, where computing power was centralized in the mainframe, and many users used the same machine. The minicomputer and the PC signified a shift towards one or few users per machine, but having a centralized servers and mainframes to share the heavy processing responsibilities. But relatively, the spread of computing power over the network was a much more even distribution. Finally, mobile clients and cloud computing are a shift back towards centralization, but with an important caveat. Virtualization is a technology that allows multiple users to share the same computer but still view the machine as their own private resource. In essence, a machine will be split into N virtual machines, one for each of N users. Data centers are centralizing computing again, but virtualization allows lightweight mobile clients to have more private computing power available to them.

This is a major evolutionary change at the macro level, and it will enforce changes at the micro level. The unique design constraints of mobile devices and data centers will be discussed in greater detail in the next chapter, but they will have significant impact on innovation in microprocessors. In recent years, the biggest innovation in processor cores has been the integration of processor cores with other blocks (Linley, 2010), and much of this integration addresses mobile devices and data

centers. Integration enables smaller form factors, higher system performance, and most importantly more efficient use of power.

After the Wall

In the previous Chapter, I spent a great deal of time talking about the power wall, and how it signals the end of CMOS scaling. If the power wall is an inflection point for the industry, one must ask what is next for semiconductors?

As mentioned in Chapter 6, Intel has responded to the power wall by introducing multi-core chips. Instead of adding complexity to a single core, which would exceed the thermal limits of the chip, they have increased the overall performance by adding parallelism at the processor core level. Again, this breaks the long held Von Neumann paradigm of programming, and is a significant challenge. While software has found effective uses for chips with a handful of cores, the problem does not scale. It is not clear if software will be able to take advantage of the increased processing power that Intel is adding to its chips.

Another response to the power wall is more integration, as mentioned above. By creating special customized functional blocks, work can offloaded the processor, allowing a lower power variant to be used. Integration can also mean mixed signal design, where analog and digital designs are combined on the same die, such as radio antennas or the physical interfaces for high-speed I/Os. This trend of mixed signal design integrated with processing cores has been coined as “More-than-Moore” (Arden, 2010). The innovation focus shifts from the digital logic, which cannot benefit from CMOS scaling anymore, to the analog designs. One important characteristic of analog design is that it is highly sensitive to the operating environment and often requires customization for the application. This only serves to further emphasize the diverse chipset ecosystem first mentioned earlier in the chapter.

In summary, the technology trends today are emphasizing a diverse product ecosystem and power efficient designs. Intel owes its success to the PC era, which was characterized by the dominance of a single architecture. What does it mean for Intel if we are now entering the post-PC era?

Chapter 10 – Where the Battle is Being Fought

In this chapter, we will investigate two market segments, smartphones and data centers, that are both changing rapidly, and will likely have an outsized influence on the continuing evolution of computers. Both of these markets are key to the rise of cloud computing, with mobile internet devices bring network connectivity to all aspects of daily life and data centers making tremendous computing power available on the other end of these network connections.

Smartphones

While not the first smartphone, the iPhone was a seminal device. It was announced in January of 2007¹, and signified that smartphones had potential beyond the business market. Since the launch of the iPhone, smartphones have grown explosively, especially with the introduction of Android based devices, and their growth has not shown any signs of slowing down. Combined sales of smartphones totaled 67 Million in 2010 and a recent survey indicated that consumers were more likely to buy a smartphone than any other electronic device in 2011, including PCs and laptops. For 2011, sales projections for smartphones top 95 Million. (Gartner, 2011)

One significant difference between mobile devices like smartphones and the traditional PC is that they must run off a battery. The power consumption then takes on disproportional significance because the faster you draw down your battery, the less useful a mobile device becomes. Laptops have the same constraint, but because of where they used (more often than not, they are used at a desk where a power source is readily available and not on a lap) and the relative size of their battery it is not as dominating a concern as in phones. Figure 23 shows a comparison of power dissipation between PCs, laptops and cell phones. Cell phones are nearly two orders of magnitude below laptops in power consumption. To be

¹ Wikipedia < <http://en.wikipedia.org/wiki/Iphone> > Observed April 22, 2011.

fair, smartphones require a good deal more power than their predecessors, but the gap is still significant.

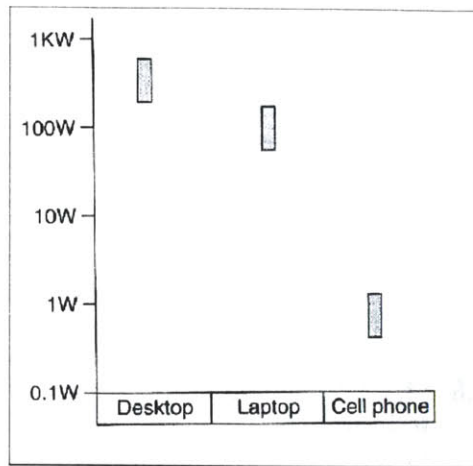


Figure 23 - Relative Device Power Ranges²

Portability is a defining feature of mobile devices, and this is determined by keeping overall device size and weight low. As the battery is both the largest component and the heaviest by unit-volume, there is significant competitive pressure to keep the battery small. This in turn makes system power efficiency an even greater imperative. To complicate matters, batteries are subject to the square cube law. Dimensional scaling can have a much greater impact on battery capacity, which is determined by volume, than on something like circuit size, which is a function of area. To help illustrate this math, imagine a battery in a cube shape. If you reduce the battery size in each dimension by 10%, you actually loose 27% of the overall battery capacity. This is the square cube law in action.

Thus we see enormous pressure in smartphones towards integrated SoC based solutions, as this can have a positive effect on battery life, size, and weight. The size benefits of a SoC are obvious, as an integrated processor, memory controller and modem on one die will take up less room than separate dies, each with their own packaging, and the requisite PCB circuitry required to stitch it all together. Recall

² Image Source: Kumar, Rakesh. *Fabless Semiconductor Implementation*. 2008. Page 128.

from Chapter 8 that with ability to blend general-purpose blocks like processor cores and application specific logic, SoC designers can often come up with the most cost effective circuit to achieve the performance goals of a system. Chapter 8 considered cost primarily as a function of circuit size, but circuit size is also highly correlated with power consumption. If you are implementing a system with a smaller circuit, you are most likely designing a system with lower power needs.

One needs look no further than the HTC Aria for example of how the combined constraints of power, size, and weight are driving smartphone designers to integrate more and more of their design into System on a Chip solutions. While they were both released in the summer of 2010, the Aria is 10% shorter and nearly 20% lighter than the iPhone 4³. Another example is Apple and their purchase of the microprocessor design companies PA Semi⁴ and Intrinsity⁵. Previously Apple had relied on outside suppliers for their silicon, but as smartphones and tablets became central to their corporate strategy they decided to bring design capabilities internal.

Furthermore, the SoC approach gives software dynamic capabilities to manage power. If a chip is also equipped with a domain specific units like a Digital Signal Processor or an application specific block like a video decoder, compute intensive tasks can be offloaded from the core processor. This allows software to put the CPU in low power sleep modes, not to mention that it may allow the system to get away with an overall smaller CPU (Gwennap, 2010). ARM has enthusiastically embraced this paradigm. In it's processor roadmap, ARM describes a vision of "Big/Little" multi-core processing. For example, the "Kingfisher" being developed by ARM is described as a "companion" processor and will be paired with a larger processor

³ Mobiledia.com. "HTC Aria Review." July 2010 < <http://www.mobiledia.com/reviews/htc/aria/page1.html> > Observed April 23, 2011.

⁴ Forbes.com < http://www.forbes.com/2008/04/23/apple-buys-pasemi-tech-ebiz-cz_eb_0422apple.html > Observed May 8, 2011.

⁵ Arstechnica.com < <http://arstechnica.com/apple/news/2010/04/apple-purchase-of-intrinsity-confirmed.ars> > Observed May 8, 2011.

core, such as a Cortex-A9 or its follow-on. During system operation, simple tasks such as playing a MP3 file can be switched dynamically over to a smaller, less power hungry processing core. (Moynihan, 2011)

Smartphones are proving significant not just because of their impressive sales. They signify the growing sophistication of the embedded space, and its potential for diffusion to applications traditionally dominated by PCs. One need look further than the numerous tablet devices released in the last two years for an example. Tablets based on PCs have struggled to gain traction for years, but with the release of the iPad in 2010, the segment has exploded. In 2010, Apple sold 10 Million iPads, outpacing the sales for its original iPhone⁶. Close on the heels of Apple's success, we have seen the release of tablets from Motorola, Samsung, and RIMM. What is fascinating about these tablets is that they are all based on smartphone architectures. Although the hardware is tightly integrated and customized, we have seen the emergence of a new computing platform.

This platform will likely expand up into PCs as well as out into other embedded devices. Gartner research has highlighted an emerging trend of hybrid devices that use smartphones and tablets as their computing engine. For example, many devices in the home, such as home stereos or TV set top boxes, can now be controlled via a Wi-Fi connected smartphone or tablet. This trend will only accelerate, and we will see tighter integration between mobile devices and long established products like cars and refrigerators (Gartner, 2011).

⁶ Dailywireless.org. December 29, 2010 < <http://www.dailywireless.org/2010/12/29/2010-ipad-sales-10m/> > Observed April 23, 2011.



Figure 24 - Examples of Hybrid Devices⁷

Data Centers and Servers

As smartphones came onto the scene, ARM had a distinct advantage. It had been the dominant processor vendor in cell phones and other embedded devices, and would likely be considered the incumbent as smartphones were first designed. With servers, it is a different story. This is a market dominated by Intel and has aligned well with their focus on processor performance. So why is there now so much attention being paid to ARM's attempts to enter the server market?

In short, the way that data centers are being designed and managed is changing. Newly constructed centers or centers that are being redesigned have a new emphasis on increasing computing density and lower power consumption. Much of this stems from trying to keep costs down, both in initial expenditures and operationally. The greatest operational costs in a data center are supplying power and cooling the center, which will only be exacerbated by denser data centers. (Cappuccio, 2010) (Cappuccio, 2011) A data center built with power efficient

⁷ Image Source: Gartner Research "Tablets and Smartphones Give Rise to New Hybrid Devices." April 12, 2011.

processors has the dual benefits of requiring less power and generating less waste heat that needs to be cooled. There are also pressures stemming from “Green” pressures. Policy makers are looking for opportunities to reduce energy consumption nation wide, and in 2006 servers and data centers consumed 61 Billion kilowatt-hours, nearly 1.5% of total US electricity consumption. (Brown, 2010)

Thus it is no wonder that we have seen a spate of systems and chips aimed at the server market all using ARM processors. Calxeda, a Texas startup formerly known as Smooth-Stone, is building servers with ARM at its core⁸. Not to be left out, SoC giant Marvell is also developing ARM-based silicon for use in servers⁹. NVidia has garnered a great deal of attention with its announcement of a chip that will combine its GPU cores with ARM cores for use in both PCs and high performance servers. The prospect of ARM based servers has also benefited from support in an unlikely place. Recognizing the potential shift underway, Microsoft has announced that the next version of Windows will run on both x86 and ARM ISAs (Wall Street Journal, 2011).

Energy efficiency is clearly growing in importance, but what about the functional requirements of the server market? Many doubt that low powered processors from mobile devices have the required features to make their way into servers (Shilov, 2011). For example, while 32 bit addressing is sufficient for mobile devices, it is not sufficient for high-end servers where most processors support 64 bit addressing. Currently, neither Intel’s low power processor line ATOM nor ARM support 64 bit addressing. Virtualization is another critical technology, especially for cloud-based services where users need to be isolated from other users. Yet, many of these

⁸ Gigaom.com “Smooth-Stone Bets ARM Will Invade the Data Center”, April 9, 2010 < <http://gigaom.com/2010/04/09/smooth-stone-bets-arm-will-invade-the-data-center/> > Observed April 23, 2011.

⁹ EE Times. “Marvell plans 40-nm ARM server processors.” < <http://www.eetimes.com/electronics-news/4199239/Marvell-ARM-Servers> > Observed April 23, 2011.

critical features will come to mobile devices, even without the pressure from the server market. ARM has clearly laid out that 64 bit processors are in their product roadmap (Schaffstein, 2011). Virtualization is making its way into mobile devices, largely stemming from the demands of the banking industry, who desire greater security as more financial transactions occur on mobile devices (Lee, 2011).

However, the server market still highly prizes raw CPU throughput and performance. While data centers designed around a large numbers of ARM cores versus a smaller number of high performance x86 Xeon processors might be more power efficient, some applications will still require “Brawny” processor cores (Hölzle, 2010). So it is still not clear if ARM will successfully outflank Intel in the server space. ARM certainly provides a compelling power story, but it also must compete with Intel on CPU throughput, an arena Intel is well accustomed to. The answer will only be revealed with time, but Intel must take this threat to its most profitable segment seriously.

Chapter 11 - ARM and Intel In the New Microprocessor Ecosystem

In this chapter the multifaceted challenges Intel will be discussed at length, and the arguments herein will draw on much of what has been discussed in previous chapters. We will begin with an analysis of the obstacles currently faced by Intel in their core technologies as they grapple with shifting from processor performance to processor energy efficiency and the rising economics of cutting edge processor fabrication. We will then discuss the technical ecosystem at large, and why Intel struggles with highly integrated products and product design. Finally, we will investigate why Intel has fundamentally the wrong business model to compete in a world with diverse, integrated system architectures.

The Challenges In Microprocessors

The Architectural Complexity Problem

As discussed in Chapter 8, Intel's pursuit of greater processor performance has left them with a legacy of very large and complex implementations of x86 ISA. Much of the innovations introduced in the 1990s, such as out-of-order execution or wide and long super-scalar pipelines, came at the cost of greater architectural complexity with minimal performance gain. The combination of switching to multi-core designs and the rise of mobile applications caused Intel to seriously tackle the problem of power consumption.

Thus in 2004, Intel kicked off a project, codenamed Bonnell, to design a low power processor core that could be used as the key building block for multi-core chips. The project was given to a design team based in Austin, Texas (which is home to Mount Bonnell), whose previous project was ironically a power hungry variant of the Pentium 4 processor family. While their previous project was cancelled, Bonnell went on to become what is now known as the ATOM processor (Shimpi, 2008).

When the ATOM processor was launched, it came in a variety of packages with a TDP¹ range of .65 to 13 Watts². For comparison, the Intel Core 2 Duo family runs from 5 Watts to 150 Watts.³ So what is different about the ATOM processor? There are a myriad of design changes, but at its essence much of ATOM's design was taking the existing x86 designs and scaling it back. ATOM has a 2 issue super-scalar pipeline, which means two instructions can be processed at once, while most x86 processors at that time were 3 or 4 issue. The ATOM processor has almost no ability to execute instructions out of order, except for the most narrow of cases. You may recall from Chapter 6 that x86 was able to mimic the pipeline of RISC designs by breaking up instructions into micro-ops. In contrast, the Bonnell design team stripped out most of these micro-ops and treated many x86 instructions as single operations. (Shimpi, 2008)

What is so fascinating about these design decisions is that they are a deliberate rolling back of many of the architectural innovations introduced to improve processor performance. They were simply too expensive to keep in the processor core. To be fair, the ATOM line does have many design features that are not a direct contradiction of the innovations of the past, such as larger cache cells for lower power and a binary clock distribution scheme. The ATOM processor spends only 10% of its power budget on clock distribution, while large contemporary CPU cores can spend up to a third of their power. (Shimpi, 2008)

ATOM is certainly an achievement for the Bonnell team. They created an x86 with dramatically lower power consumption than other Intel x86 products. But is it enough? Despite the major design shifts, ATOM does not beat ARM based processors in either active or idle power consumption (Shimpi, 2008). Intel would

¹ TDP stands for Thermal Design Point or Thermal Design Power. It is a measure of the maximum power a processor core is expected to draw running real applications. Source: Wikipedia

² Intel.com Observed April 19, 2011.

³ Wikipedia: Intel Core 2 < http://en.wikipedia.org/wiki/Intel_Core_2 >, Observed April 19, 2011.

claim that the power gap will narrow with further generations of the ATOM processor. But the Christen disruption framework discussed in Chapter 1 argues that companies will struggle to create products that are competitive in a new performance dimension, which in this case is power. And there are tell tale signs of trouble here.

For example, the ATOM processor has a pipeline of 16 stages, while the Core 2 Duo has 14 pipeline stages. Longer pipelines are associated with higher complexity, higher performance, and greater power consumption. While some of the stages are the result of design decisions made to lower power, several other stages are required to support simultaneous multi-threading and the clock frequencies of ATOM, or in other words the pipeline reached this length in an effort to achieve performance targets. (Shimpi, 2008) A more human tell-tale sign of Intel's trouble may be the departure of Anand Chandrasekher, the VP of the Ultra-Mobility Group and closely associated with the ATOM line, from Intel this past March.⁴ This has led many to question Intel's dedication to the mobile space.

Company Culture's Influence on Technical Design Decisions

At this point, the discussion will turn to how organizational history and culture can influence how design teams approach a problem and influence their decisions. Much of this is rooted in Chapter 5, and its discussion of the role of culture. The salient example is Polaroid, who failed to bring a compelling digital camera to market despite having an enormous technical lead.

When an engineering team approaches a design parameter, such as the power consumption of the core, there are several ways to frame the problem. On one end

⁴ Intel Newsroom. < http://newsroom.intel.com/community/intel_newsroom/blog/2011/03/21/chip-shot-anand-chandrasekher-to-leave-intel-mike-bell-dave-whalen-to-lead-ultra-mobility-group > Observed April 20, 2011.

of the spectrum, you can view it as a constraint that needs to be worked around or a condition that must met in order to focus on your other goals. Further along spectrum, a team can view a constraint as a trade-off point. For Intel, shifting from a constraint-based view to a trade-off approach would be the difference between stating a TDP target that an overall core must come in below, versus setting an instruction-per-watt target for the core. As power has become the primary concern in their target market, Intel has made an important shift. The Bonnell team adopted a rule of thumb where a performance improvement of 1% could only implemented if it increased power consumption by 1% or less (Shimpi, 2008).

But even though they are now focused on the power-performance trade off, Intel still carries with them the legacy of their previous mental models. The 1% for 1% rule of thumb still treats power consumption as a tax to be paid for increased performance. Contrast this with the ARM design team, who tackled the power constraint from the other end of the spectrum. Rather than considering power consumption to be a challenge to be overcome, they made it an explicit design goal to be a low power processor core early in the history of the ARM processor (Levy, 2005). Over time they have come to the same trade-off view as Intel, but they have a cultural legacy of low power design and focus. As a result, the ATOM product line has had to focus on bringing power consumption down in order to meet the design constraints of mobile devices, while ARM can focus on narrowing the performance gap between their product and Intel's. If we were to view this again through the lens of a Christensen disruption, Intel's existing customers and existing value network force them to put overall performance first, while ARM's own value network forces them to never take their eye of designing low power products.

The Commoditization of Wafer Fabrication

In Chapter 9 we discussed the "power wall" at length and what it means for CMOS-scaling. CMOS-scaling is the technique of shrinking a circuit along all dimensions by the same relative factor. The dimensions of a transistor designed for a 180

nanometer process are proportionally the same as a transistor designed for a 130 nanometer process. Intel has relied on CMOS-scaling as a powerful tool to dramatically increase processor performance while keeping product costs relatively constant. However as was discussed earlier CMOS-scaling has run out of steam and as Intel transitions to smaller and smaller geometries, the final product advantages are shrinking. The compelling argument for CMOS-Scaling was that it allowed the fabrication process to advance and mature without incurring a high cost on the design of a circuit. There are theoretical options to shrink circuits beyond the power wall, such as FinFETs or Silicon-On-Insulators designs, but these are fundamental shifts in the transistor design. Any transition to new transistor and gate design will now extract a significant design cost (Nowak, 2002) (Haensch, 2006).

What all this means is that it will require exponentially more effort and more investment for Intel to maintain a process advantage over other wafer fabricators, and even if it does so, the payoffs may not be worthwhile. Pure-play foundries like TSMC and UMC have dramatically improved their capabilities and shrunk the process advantage that Intel has struggled to maintain. In their currently operating facilities, Intel's leading edge process is 32 nanometers⁵, and has announced the construction of a 14 nm fab⁶. In contrast, TSMC currently offers a 40-nanometer process and is developing a 20-nanometer process⁷.

⁵ Intel Newsroom. <
http://newsroom.intel.com/community/intel_newsroom/free_press/blog/2010/10/21/moores-law-around-the-world-in-bricks-and-mortar/> Observed April 20, 2011.

⁶ Intel Newsroom, "Intel to Invest More than \$5 Billion to Build New Factory in Arizona", <
http://newsroom.intel.com/community/intel_newsroom/blog/2011/02/18/intel-to-invest-more-than-5-billion-to-build-new-factory-in-arizona> Observed April 20th, 2011

⁷ TSMC.com, Observed April 20, 2011.

While Intel has always maintained a technological lead with smaller performance geometries, independent foundries such as TSMC are matching Intel in the transition from 300mm wafers to 450mm wafers. A larger wafer allows more die to be cut from the same wafer and can ultimately lead to cost reductions ranging from an initial 10% up to a potential 50% cost advantage over 300 mm wafers, the current industry standard. (LaPedus, 2011) (Kumar, 2008) This is a telling difference, as a larger wafer does not offer any performance advantage like a smaller geometry. It is clear independent foundries are focused on commoditizing semiconductor fabrication, and are content to cede the performance bleeding edge to Intel. The escalating difficulty of maintaining a process advantage while simultaneously facing rapid commoditization of previous generations puts Intel in a difficult spot.

The Challenge In Modular Design

The NRE Economy

Recall that in Chapter 9, I outlined the move to diverse integrated product architectures, which in turn requires more specialized components like System on Chip designs instead of generic microprocessors. But doesn't customization incur a cost? This is one of the most fundamental lessons in economics. Henry Ford was able to achieve mass-market success with the Model T by standardizing the product and exploiting the inherent cost advantages in mass production. If the future is one of custom designs and smaller niche volumes, wouldn't this open a window for Intel to compete on price? Why don't customized designs incur higher costs?

The first part of this answer is that the semiconductor industry long ago decoupled fabrication. Intel does optimize its plants for long runs of the same product, but TSMC and other independent foundries specialize in doing short runs of different products for different customers (Gwenapp, 2010). By grouping customers by their selected process technology, foundries can make a different product simply by

swapping the design masks. The cost differential in manufacturing from mass production is largely negligible. Furthermore, a customized component may result in a significantly lower overall system cost. For example, if a system uses a lower power custom SoC over a generic processor, the system cooling requirements could be lower, allowing a manufacturer to save on total system cost.

But customization does incur higher design costs. In industry terms, any engineering costs incurred during the design of a chip, from digital logic design, to test design, to layout, is referred to as non-reoccurring engineering, or NRE. The drivers and consequences of NRE are complex. The more complex a design, the higher the NRE will be. And the larger the volumes of a chip sold, the less the NRE matters in the final cost structure of the product. When Intel designs a cutting edge processor, the NRE is amortized over hundreds of millions of units. But when you design a SoC customized for a niche application, the NRE is a significant factor. Further more, as process geometries shrink, NRE costs increase as a rule, regardless of the step in the design process. Figure 25, plots these various costs, such as the design, verification, and software costs.

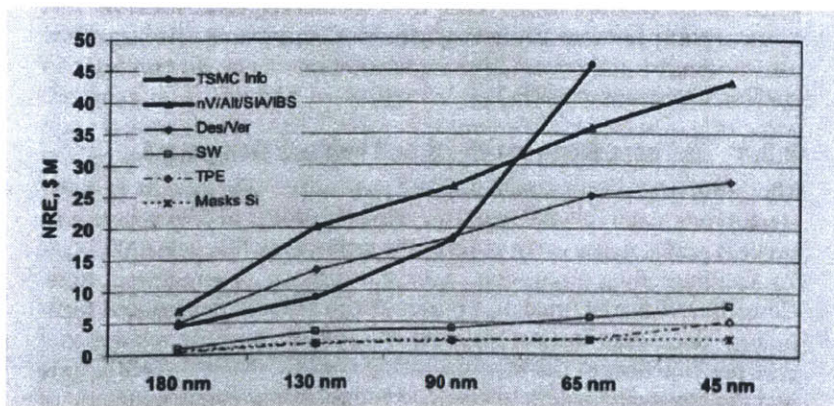


Figure 25 - The costs associated with all aspects of design are increasing⁸

The semiconductor industry has responded to this through the modularization of designs, and tools focused on reducing NRE. The most prominent example of this are the 3rd party IP vendors. Instead of designing all parts of the SoC from scratch, a

⁸ Source: Kumar, Rakesh. *Fabless Semiconductor Implementation*. 2008. Pg 235.

robust IP market has evolved where functional blocks can be licensed and integrated into the design. By selling a design across to multiple companies, IP vendors can often offer their design at a lower cost than the NRE a company would incur designing the IP themselves. Furthermore, new design always incurs a risk that it will not work as expected. As their product is used in a variety of products, 3rd party IP vendors often present a less risky choice for SoC designers. ARM is the most visible and most successful of the 3rd party IP vendors. The modularization of semiconductor design is ARM's raison d'être.

When a company can license from a market of proven IP designed common applications, they can also focus their engineering efforts on differentiating design. Recall the "Good NRE" and "Bad NRE" discussion from Chapter 1. A modular design ecosystem allows design teams to more effectively deploy their engineering resources and achieve a better balance of Good NRE and Bad NRE. This is evocative of the "small footprint" strategy described in Chapter 4. (Baldwin and Clark, 2006)

The fabrication ecosystem has also evolved to reduce NRE. Companies like UMC and TSMC charge companies an initial fee for each new design, known as the mask fee, which can significantly drive costs up if multiple revisions are needed to perfect a design. To help mitigate the risk of higher NREs, fabless semiconductor companies often offer a "shuttles." A shuttle lets multiple customers share the same wafer and costs significantly less than a full mask set fee. A customer cannot go to production with a shuttle chip, but it does allow them to build a low cost prototype and provide early samples to software developers in turn. Shuttles help reduce overall NRE costs and improve the time to market.

System Optimization

As we enter the era of computing diversity, each device will have its own unique combination of performance capabilities and features. In addition, the semiconductors designers create to power these devices must choose between the iron triangle of tradeoffs of power, performance, and cost. The upshot of highly

integrated customized components is that it makes it a much easier proposition to optimize the overall system. The modular design ecosystem allows device manufacturers to choose a combination of best in class IP blocks. While Intel may release a version of ATOM coupled with a H.264 decoder, a custom SoC can combine a standard ARM processor with any decoder a customer chooses. Intel's offerings bring to mind the saying "you can choose any color, as long as its black." (Demerjian, 2011)

Again the foundries contributions to the ecosystem are helping designers optimize the total system. All the major independent foundries offer different sets of library gates that help tailor the circuit for different applications. For example, after it passed the 90 nanometer process node TSMC started offering Low Power (LP) and General Purpose (GP) libraries. The GP was fast but power intensive, and the LP was slower. SoC designers would choose a library that best matched their target application. In addition TSMC now supports mixed library designs, where some portions of the chip can be built with different sets of library gates, allowing for greater customization and system optimization. In short, independent foundries and the 3rd party IP market have greatly reduced the required NRE to design a new customized chip. And with the ability to customize each designer can choose the combination of speed, power, and cost that best satisfies their design constraints (Schaffstein, 2011).

Developer Tools and Environments

Intel boasts a great deal of industry tool support, but most of that is relegated to software development tools, such as compilers, and hardware drivers. On the other hand, the fabless semiconductor industry boasts a wide selection of semiconductor design tools, addressing all aspects of the design process. These tools run the gamut from logic design simulators, to circuit power optimization. Thus when building a SoC, designers can avail themselves of a wide selection of software tools to help optimize their design, reduce NRE, and improve time to market. (Schaffstein, 2011)

Furthermore, the fabless semiconductor enjoys excellent tool-chains to support software development. Much this comes from the market position of ARM, with much if its industry support and adoption stemming from the fact that it is the dominant processor core vendor for fabless semiconductor companies. But ARM has also deliberately cultivated excellent software support and enabled developers to quickly write software for ARM processors. (Lee, 2011) (Levy, 2005) ("CEO Interview: Warren East ARM", 2008)

The support of software developers is a critical point. As Intel tries to move into the mobile space, it is trying to displace a processor ISA that enjoys wide adoption. Recall from Chapter 6 that the primary reason that Intel was able to fend off the challenge from higher performing RISC processors in the PC was because of the tremendous amount of software that was already compatibility with the x86 ISA. Has ARM reached the software compatibility tipping point? It is telling that Microsoft, the other giant of the PC industry and perhaps the software company most closely tied to the x86 architecture, demonstrated the next version of Windows will run on an ARM processor⁹. Furthermore, Microsoft also acquired an ARM architectural license, signaling their intention to begin designing their own SoCs.¹⁰

The Challenge With the Business Model

Vertical Versus Horizontal

In many ways, the challenges that Intel currently face mirror the Intel's rise to dominance with the advent of the PC. Before the PC companies that designed and manufactured all the components in the computer, such as IBM or DEC, dominated the industry. The open modular architecture of the PC allowed companies like Intel

⁹ ZDNet.com. January 5, 2011 < <http://www.zdnet.com/blog/microsoft/ces-microsoft-shows-off-windows-8-on-arm/8339> > Observed April 20, 2011.

¹⁰ Clarke, Peter. "Microsoft-ARM deal is a consumer, computing game-changer." *EE Times*. July 23, 2010 < <http://www.eetimes.com/electronics-news/4204864/Microsoft-ARM-deal-game-changer> > Observed April 20, 2011.

and Microsoft to carve out large chunks of the value chain for themselves. We are seeing the same pattern play out again with the highly segmented semiconductor industry and SoC design. Is a conflict of vertically integrated companies versus a horizontally segmented ecosystem.

Just like Intel didn't have to excel at hard drive design, or write operating systems, the 3rd party IP vendors allow SoC designers to focus on the portion of their design that differentiates themselves from their competition. And in return 3rd party vendors like ARM benefit regardless of a products' ultimate design. Through its licensing model ARM can participate in a variety of target applications where it has almost no domain knowledge. Intel, in contrast, must limit itself to targeting a handful of applications that will yield the greatest volumes. In short, Intel has to make a few well-targeted bets, while ARM can flood the market with different SoC designs and nearly no costs to themselves.

For companies further up the value chain, such as mobile handset manufacturers or mainframe suppliers, there is a powerful incentive to pursue a "second-source" strategy for components in their supply chain. Second sourcing means exactly what it says. If you have more than one source for a component, you have more leverage in price negotiations. If a company designs an x86 ATOM based SoC into their design, they likely are dependent on Intel exclusively, with the outside possibility of switching to an offereing from AMD. However, given ARM's widespread adoption by a variety of companies, there is no shortage of ARM suppliers to choose from. And while SoCs are rarely "socket compatible" with each other due to customization, a device manufacturer can swap ARM SoC suppliers in each new product generation without breaking software compatibility.

The Losing Economics of Fab Development

With the costs of building cutting edge rising exponentially, many companies are asking if they can afford to keep their own fabs. In addition, with the power-wall standing firm, radical new transistor designs and methodologies needed, such as the

aforementioned FinFET or Silicon-On-Insulator paradigm shifts. This will create further costs as the industry shifts from its traditional CMOS designs.

This is one of the driving reasons behind the emergence of the “fab-lite” company. Many previously integrated semiconductor companies are now shedding themselves of their fabs and are now using partnerships with either other vendors or independent foundries to develop new fabrication processes. A famous example is the other x86 company AMD, who spun off their foundries into a new independent company in 2008¹¹.

Intel has employed a “Tick-Tock” product roadmap to introduce new process generations. New process geometries are broken in using a “Tick” product, which is an established product that is ported to the new process. In this way, they hedge the risk inherent in a new process by porting an established design. In the “Tock” phase, they create a new design for the now semi-mature process. This design in turn may be used as the following generation’s “Tick” project (Kumar, 2008). It is a clever way to align product roadmaps with process roadmaps, but when a new process also incurs radical new design paradigms at the gate level, Intel may have to acclimate to a much higher level of risk with their “Tick” projects.

Cannibalization and Competition

Competitively, Intel faces a myriad of challengers, including itself. As low power processors are adopted for use into servers, Intel faces the unappealing reality that their low margin ATOM processors could be stealing business from their very profitable Xeon line (Demerjian, 2011)

Furthermore, the competitors outside of Intel are fragmented. Intel is competing with ARM on the ISA, TSMC and the independent foundries on process innovation, and a plethora of SoC companies (Marvell, Qualcomm, NVidia, Samsung, Apple,

¹¹ The Inquirer.net < <http://www.theinquirer.net/inquirer/news/1019627/amd-foundry-spinoff-details> > Observed April 20, 2011.

MediaTek, etc . . .) for design wins. These competitors also feed off each other. More firms choosing to outsource fabrication creates opportunities for ARM, and in turn the market dominance of ARM will drive more companies to try their hands at SoC design. Instead of competing with other vertically integrated companies such as the AMD of old who had their own strengths and flaws, Intel cannot afford to miss a step any segment. In contrast, by not asserting itself too strongly and providing best in class power performance, ARM has quietly made itself ubiquitous. (Turley, 2010)

Conclusion

The Intel Age of Computing is over. While it is a transition that will happen slowly, the x86 instruction set is no longer the dominant architecture of computing. It has been done in by a combination of architectural innovations, technological realities, and innovative business models. Intel as we know it is doomed.

Power has become a dominant performance indicator at precisely the moment Intel is most vulnerable. If CMOS-scaling had not run headfirst into the power-wall, Intel would likely be able to use its process advantage to drive power consumption down without making any changes to the architecture. As consumer attention shifts to power-efficiency, processor performance is no longer valued as highly. Incredibly complex and large circuits are required to continue to push the performance envelope. Intel must charge significant premiums to cover both the per-unit cost and the staggering capital expenditures to build a cutting-edge fabrication plant.

Cost and energy efficiency pressures and the rise of a diverse class of embedded systems will continue to push along the transition to System on a Chip solutions. System on a Chip designs customized for their applications can only exist in an ecosystem that embraces the third party IP and independent foundries. Intel's business model is incompatible. It is predicated on co-specialization of design and manufacturing extracting huge amounts of value from the entire chain, from initial design efforts to the finished components.

As a result, Intel is all but locked out of a vibrant new computing segment and one that promises to have outsized impact, perhaps as much as the first IBM 5150. As the struggle to come up with a compelling story for smartphones, their most lucrative market is at risk as ARM makes a challenge to the dominance of the Xeon product line in high-end servers. As embedded devices continue to grow upwards, much like the PC did thirty years ago, *all* of Intel's business becomes at risk. There

are signals from the market that this shift is already happening. Some inside sources are now claiming that Apple is going to phase out Intel processors from its laptop line in favor of ARM¹.

Intel's problem is exacerbated by the fact that the investments required are so lopsided. Each successive generation of fabrication plants is exponentially more expensive. Intel announced that its newest fab will cost \$5 Billion², and that does not include the massive R&D investments required to develop the process. Just recently, Intel announced that their next fab will also build circuits using the fin-FET 3-D transistor design³. This is a massive bet, and represents a significant transition for Intel. However, it was a necessary bet to make if Intel wanted to continue focusing on circuit speed. This announcement is not surprising, as Intel has prided themselves on their advanced processes. (Recall the marketing campaign centered around dancing wafer process engineers in their clean suits). But if fin-FET designs prove to be a painful transition, what then? When the economics catch up with them, will they be able to divorce themselves from their fabrication plants?

So what is Intel to do? First it must recognize that the dominant design has already crystallized for smartphones, tablets, and by extension the embedded space. While it might seem counter-intuitive to have a dominant design in an industry that will be characterized by customized device, the dominant design takes the form of a System on a Chip with one or more ARM cores, an integrated in memory controller, and a combination of 3rd party IP blocks and application specific logic. If Intel wants to participate in this market, it must embrace the dominant design.

¹ Demerjian, Charlie. "Apple dumps Intel from laptop lines" *SemiAccurate.com*, May 5, 2011 < <http://semiaccurate.com/2011/05/05/apple-dumps-intel-from-laptop-lines/> > Observed May 7, 2011.

² Bloomberg News. February 18, 2011 < <http://www.bloomberg.com/news/2011-02-18/intel-plans-to-build-5-billion-chip-plant-in-arizona-hire-4-000-workers.html> > Observed April 23, 2011.

³ Markoff, John "Intel Increases Transistor Speed by Building Upward." *New York Times*, May 4, 2011 < <http://www.nytimes.com/2011/05/05/science/05chip.html> > Observed May 6, 2011.

Intel must not only participate in licensing their IP to other silicon vendors, but they must develop a library of IP to license *outside* of processor cores. And any SoC they design must primarily feature an ARM processor core if they want it to gain any traction. Intel clearly made a mistake when they gave up on their own ARM implementation and sold XScale to Marvell. If they want to have a chance in the vital embedded space, they must embrace modular design, admit their losses and become an ARM licensee, and be willing to outsource the semiconductor fabrication. In short, they must turn their business upside down. However, they continue to make money from PCs, laptops and servers for the time being. The Christensen disruption framework would say that this is nearly an impossible feat to pull off, as the value network Intel has built up over 30 years would pull it in exactly the opposite direction.

Intel must embrace the commoditization of wafer fabrication. There are some signs that this is already happening, as Intel has agreed to fabricate FPGAs for two startups, Achronix and Tabula⁴. The volumes of these deals are drops in the bucket, but they are a dramatic departure for Intel. As volumes in their old fabrication plants wind down, Intel can fill their capacity by bringing on more fabless semiconductor companies as customers. However, the FPGA customers may be as much about trying to find a workable platform for the ATOM processor than a try venture into the pure-play foundry market. Another rumor that has only surfaced in the last week is that Intel is trying to woo Apple away from Samsung⁵. The irony of this rumor is that if Intel does succeed, they would be building ARM processors. If Intel does commit itself to the foundry market, it will have to maneuver carefully to avoid conflicts of interests. Intel is such a large company that one would have to

⁴ SemiAccurate.com. "Intel picks up a second foundry customer, Tabula." April 18, 2011 < <http://semiaccurate.com/2011/04/18/intel-picks-up-a-second-foundry-customer-tabula/> > Observed April 23, 2011.

⁵ Barak, Sylvie. "Could Intel Churn out ARM chips for Apple?", *RCR Wireless*. May 4, 2011 < <http://www.rcrwireless.com/article/20110504/CHIPS/110509966/0> > Observed May 6, 2011.

imagine that any business that a foundry division could win would be in turn competing with some other Intel offering, not to mention that Intel would face a tremendous headache managing their plant capacity between supplying their own products and meeting their obligations to their foundry customers.

AMD may very well be a picture of Intel in 10 years. In a move that shook up the industry, AMD had to give up on owning its own fabs, as they became too expensive to maintain. Two years ago, AMD spun all of its plants into an independent company, GlobalFoundries⁶. Today, AMD is the subject of a rash of rumors, and there is much speculation that it may become an ARM licensee. As a company that was focused on capturing a market defined by Intel, this is a significant development. But many believe that this moment is a paradigm shift, and it would make sense for AMD to abandon x86. AMD famously became the second source for x86 at the advent of the PC because IBM would not give Intel their business without an access to a second source, but today if AMD feels the “x86 architecture is not worth second-sourcing what does that say about the value of the first-source chips?” (Clarke, 2011)

Beyond the questions of technology and design strategies, if Intel is to survive, it must change its culture. The examples of IBM and Polaroid have shown how critical culture is. Lou Gerstner famously said, “I came to see, in my time at IBM, that culture isn’t just one aspect of the game – it is the game.” For the past three decades, it has been the center of gravity for computing hardware. No other company can match their operational acumen and capabilities. This is a bitter pill for anyone to swallow. The Intel Age is over. Time will tell if it is the end of Intel as well.

⁶ EE Times. April 8, 2010 < <http://www.eetimes.com/electronics-news/4088550/AMD-recognizes-325M-from-GlobalFoundries-spinoff> > Observed May 1st, 2011.

Bibliography

1. Abernathy, William J. and James M. Utterback, "Patterns of Industrial Innovation." *Technology Review*. Vol. 80 No. 7 1978
2. Ante, Spencer E. "BIG BLUE'S BIG BET ON FREE SOFTWARE." *BusinessWeek* 3761 (2001)
3. Arden, Wolfgang et Al. "More-Than-Moore: White Paper." *International Technical Roadmap for Semiconductors*, 2010
4. Baldwin, Carliss and Kim Clark. "Architectural Innovation and Dynamic Competition: The Smaller 'Footprint' Strategy". *Working Paper*. 2006
5. Baldwin, Carliss and Kim Clark. *Design Rules: Volume 1. The Power of Modularity*. 2000
6. Bray, Neil. "Designing for the IP Supermarket." *Fall VIUF Workshop*. October 1999
7. Brown, David and Charles Reams. "Toward Energy Efficient Computing." *ACM Queue*, 2010. < <http://queue.acm.org> >
8. Burrows, Peter. "What 'Beyond the PC' Means for PC Makers." *Business Week Online*. March 8th, 1999
9. Carson, Phil. "Behind the Intel sale: possibly hampered in wireless by PC successes. (cover story)." *RCR Wireless News* 03 July 2006: 1+.
10. Cappuccio, David "What to Consider When Designing Next-Generation Data Centers." *Gartner Research*. September 10th, 2010
11. Cappuccio, David "Shrinking Data Centers: Your Next Data Center Will Be Smaller Than You Think." *Gartner Research*. March 4th, 2011
12. "CEO Interview: Warren East ARM", *Portable Design*, November 2008
13. Christensen, Clayton. *The Innovators Dilemma*. 1997
14. Christensen, Clayton M., and Michael Overdorf. "Meeting the Challenge of Disruptive Change. (cover story)." *Harvard Business Review* 78.2 (2000)
15. Clark, Don. "Intel's New Chip Aimed at Big Servers." *Wall Street Journal* 31 Mar. 2010
16. Clarke, P.. "Power imperative favors ARM's client-to-server play." *Electronic Engineering Times* 27 Sep. 2010
17. "Data Center Designs a Major Factor in New IT Gartner." *eWeek*. March 16th, 2011
18. Demerjian, Charles. "Atom is dead, Slowly Strangled by Intel." *SemiAccurate.com*, Feb 15th 2011 < <http://semiaccurate.com/2011/02/15/atom-dead-strangled-slowly-intel/> > Observed April 20th, 2011
19. Demerjian, Charles. "Intel Desperately Tries to Deflect ARM's March Into Servers." *SemiAccurate.com*, March 15th 2011 < <http://semiaccurate.com/2011/03/15/intel-desperately-tries-to-deflect-arms-march-into-servers/> > Observed April 20th, 2011

20. Faletra, Robert. "Tablets Are The Future." CRN 1 Jan. 2011: ABI/INFORM Global, ProQuest. Web. 29 Mar. 2011.
21. "The Fifth Wave of Computing." *Growth Strategies*. July 2005
22. Furber, Steven. *ARM System-on-Chip Architecture*. 2000
23. Galli, Peter. "Why Unix can't win." *eWeek* 22.1 (2005)
24. Garr, Doug. *IBM Redux*. 1999
25. Gartner Research "Findings: Smartphones Top U.S. Consumers' Intended Purchases for 2011." February 14th, 2011
26. Gartner Research "Tablets and Smartphones Give Rise to New Hybrid Devices." April 12th, 2011
27. "Getting personal." *Economist* 380.8488 (2006): 57-58. Business Source Complete. EBSCO. Web. 1 Apr. 2011.
28. Gillett, Frank "The Age of Computing Diversity." *Forrester Research*. September 16th 2010
29. Grinnell, J., and C. Muise. "Dell Computers: Competing Toward Decline?" *Journal of Business Case Studies* 6.3 (2010)
30. Grove, Andrew. *Only the Paranoid Survive* 1996
31. Gwennap, Linley "Editorial: What is a Microprocessor?" *The Microprocessor Report*. September 2010
32. Gwennap, Linley "Editorial: Intel Eyes Foundry Market" *The Microprocessor Report*. November 2010
33. Haensch, W E J Nowak and R H Dennard, P M Solomon, and et al. "Silicon CMOS devices beyond scaling." *IBM Journal of Research and Development* 50.4/5 (2006)
34. Hannan, Michael and John Freeman. *Organizational Ecology*. 1989
35. Harreld, J. Bruce et al. "Dynamic Capabilities at IBM: Driving Strategy Into Action." *California Management Review*. Vol. 49 No. 4 2007
36. Haynes, Peter. "The third age." *Economist* 332.7881 (1994)
37. Hölzle, Urs. "Brawny cores still beat wimpy cores, most of the time." *IEEE Micro*, 2010
38. Iansiti, Marco and Roy Levien. "Strategy as Ecology." *Harvard Business Review* March 2004
39. Jackson, Tim. *Inside Intel: Andy Grove and the Rise of the World's Most Powerful Chip Company*. 1997
40. Kumar, Rakesh. *Fabless Semiconductor Implementation*. 2008
41. LaPedus, Mark. "TSMC tips 450-mm fab." *EE Times*. January 31st, 2011. < <http://www.eetimes.com/electronics-news/4212716/TSMC-tips-450-mm-fab> > Observed April 20th, 2011
42. Lee, Ruby. Personal Interview. March 28th, 2011
43. Levy, Markus "The History of the ARM Architecture". *ARM IQ Magazine*. Vol. 4 No 1. 2005
44. Mann, Daniel. "Why an embedded X86 CISC beat RISC." *Electronic Engineering Times*. 1997
45. Manners, David. "Foundries get upper hand." *Electronics Weekly* 2039 (2002)
46. Meredith, Robyn. "It Takes A Crisis." *Forbes*. Vol. 182 Issue 2. Pg 96 – 102

47. Moore, Gordon. "Cramming More Components Onto Integrated Circuits." *Electronics*. Vol. 38 No. 8 1965
48. Moore, James F. "Predators and Prey: A New Ecology of Competition." *Harvard Business Review*. May 1993
49. Moynihan, Finbarr. Personal Interview. April 21st, 2011
50. Nash, Kim S.. "Law Firm Migrates from Wang." *Computerworld* 16 Nov. 1992: ABI/INFORM Global, ProQuest. Web. 4 Apr. 2011
51. Nowak, E J. "Maintaining the benefits of CMOS scaling when scaling bogs down." *IBM Journal of Research and Development* 46.2/3 (2002)
52. Olukotun, Kunle and Lance Hammond. "The Future of Microprocessors." *ACM Queue*, September 2005. < <http://queue.acm.org> >
53. Patterson, David. "The Trouble With Multi-Core." *IEEE Spectrum*. Vol. 47 Issue 7. 2010
54. "Record Earnings Won't Tell Full Intel Tale." *Wall Street Journal* (Online) 12 Jan. 2011
55. Reed, Sandy. "Not so long ago, you had to pick IBM or Compaq." *InfoWorld* 22 Jan. 1996: ABI/INFORM Global, ProQuest. Web. 29 Mar. 2011.
56. Reimer, Jeremy. "Total Share: 30 years of personal computer market share figures." *ArsTechnica.com*. December 2005 < <http://arstechnica.com/old/content/2005/12/total-share.ars/> > Observed on April 4, 2011
57. Scannell, Ed. "Big Blue's Bet Pays Off: IBM's open-source gamble creates opportunities for partners." *VARBusiness*. December 5, 2005
58. Schaffstein, Michael. Personal Interview. April 6th, 2011
59. Shilov, Anton. "Intel, Google Doubt ARM and Atom Have Chances in Servers." *Xbitlabs.com*, March 1st, 2011
60. Shimpi, Anand Lal. "Intel's Atom Architecture: The Journey Begins." *Anandtech.com*. April 2nd, 2008. < <http://www.anandtech.com/show/2493/1> > Observed April 19th, 2011.
61. Sperling, Ed. "Make Way For The Consumer Era." *Electronic News* (10616624) 49.6 (2003)
62. Tripsas, Mary and Giovanni Gavetti. "Capabilities, Cognition, and Inertia: Evidence From Digital Imaging." *Strategic Management Journal* 21: 1147-1161. 2000
63. Turley, Jim. "Editorial: The Race to the Bottom" *The Microprocessor Report*. July 2010
64. Tynan, Dan. "1978-1985: The Dawn of the PC." *InfoWorld* 15 Dec. 2003: ABI/INFORM Global, ProQuest. Web. 29 Mar. 2011.
65. Utterback, James M. *Mastering the Dynamics of Innovation*. 1994
66. Vijayan, Jaikumar. "PC servers bulk up." *Computerworld* 27 Feb. 1995
67. "Warren East ARM." *Portable Design* 14.10 (2008)
68. Wilcox, Joe. "The Compaq portable." *Computer Reseller News* 15 Nov. 1998: ABI/INFORM Global, ProQuest. Web. 29 Mar. 2011.
69. Wladawsky-Berger, Irving. "Reflections on Surviving Disruptive Innovations." *blog.irvingwb.com*. December 7th, 2009 <

<http://blog.irvingwb.com/blog/2009/12/reflections-on-surviving-disruptive-innovations.html>>

70. Wladawsky-Berger, Irving. "Looking Out for Asteroids". blog.irvingwb.com. July 21st, 2005 <

http://blog.irvingwb.com/blog/2005/07/looking_out_for.html >