Technological Change within Hierarchies:

The Computer Industry

by

Jeffrey L. Funk

Professor

Hitotsubashi University

Institute of Innovation Research

2-1 Naka, Kunitachi, Tokyo 186-8603 Japan

Telephone: 81-42-580-8430

Facsimile: 81-42-580-8410

funk@iir.hit-u.ac.jp

Technological Change within Hierarchies:

The Case of the Computer Industry

Abstract

This paper uses the computer industry to demonstrate a model of technological change that addresses the sources and timing of technological discontinuities and dominant designs. The model emphasizes product design and customer choice hierarchies, design tradeoffs, and technological improvements at lower levels in the product design hierarchy (e.g., improvements in components in an assembled product). Improvements at lower levels in a product design hierarchy drive changes in the design tradeoffs for the product as a whole, which affects the movements up and down the product design and customer choice hierarchies. Movements up the hierarchies may lead to the emergence of a technological discontinuity, which this paper calls a new product class, while movements down the hierarchies may result in the emergence of a dominant design. The use of product design and customer choice hierarchies and the concept of design tradeoffs provide additional insights into how a discontinuity occurs, including the specific changes that occur in the designs, customers, business models, and sales channels during the discontinuity.

1. Introduction

In spite of the recognized importance of technological discontinuities and dominant designs in the existing literature on technological innovation, there are few models that address the sources and timing of them. Anderson and Tushman's (1990) seminal article articulated a cyclical model of technological change where competition between alternative designs, the emergence of a dominant design, and incremental progress follow a technological discontinuity. They and others have shown the difficulties incumbents experience in responding to these discontinuities (Abernathy and Clark, 1985; Tushman and Anderson, 1986; Henderson and Clark, 1990; Utterback, 1994). Still others have extended Anderson and Tushman's (1990) model by showing examples of interactions between component and system innovations/discontinuities (Tushman and Murmann, 1998; Malerba et al, 1999) and how dominant designs can exist at multiple levels in a single product (Utterback, 1994; Tushman and Murmann, 1998; Murmann and Frenken, 2006).

This paper builds on this literature to present a model of technological change that provides greater insights into the sources and timing of technological discontinuities and dominant designs than does the existing literature. The proposed model emphasizes product design and customer choice hierarchies (Clark, 1985), design tradeoffs (Alexander, 1964; Dosi, 1982; Rosenberg, 1963, 1969; Sahal, 1985), and technological improvements at lower levels in the product design hierarchy (e.g., improvements of components in an assembled product). Improvements at lower levels in a product design hierarchy drive changes in the design tradeoffs for the product/system as a whole, which affects the movements up and down the product design and customer choice hierarchies. Movements up the hierarchies may lead to the emergence of a technological discontinuity, which this paper calls a new product class, while movements down the hierarchies may result in the emergence of a dominant design. The use of product design and customer choice hierarchies and the concept of design tradeoffs provide additional insights into how discontinuities occur, including ones that involve an interaction between component and system innovations (Tushman and Murmann, 1998; Malerba et al, 1999), by showing the specific changes that occur in the designs, customers, business models, and sales channels during the discontinuity.

This paper uses data from the computer industry to demonstrate this alternative viewpoint of technological change. The computer industry is an appropriate application for the model due to large amounts of technological change, a large literature on the industry, and the important role that dominant designs play in the industry. The lack of randomness in the choice of industry suggests that we must be careful about generalizing to other industries. Following a description of the proposed model and research methodology, this applies the model to the computer industry.

2. Proposed Model

The proposed model builds on the concepts of hierarchical decision making in complex systems (Simon, 1996; Alexander, 1964) and the use of product and customer choice hierarchies to represent the process by which by which firms translate customer needs into products over time (Clark, 1985). The customer choice hierarchy represents a firm's perception of the ways in which customers make choices in the market and thus how firms define market segments and the problems to be solved in each segment. The product design hierarchy defines the method of problem solving and it includes both alternative designs and sub-problems for both products and processes (Clark, 1985).

The interaction between these hierarchies also includes the determination of a business model (Chesbrough, 2003) and sales and service channels (Abernathy and Clark, 1985).

The introduction of new products and services reflect movements both down and up the hierarchies of product design and customer choice in the industry as depicted in Figure 1. Following a technological discontinuity, design activity shifts from core to periphery at one particular level of the product design hierarchy and it also moves from higher-level to lower-level problem solving (Tushman and Murmann, 1998; Murmann and Frenken, 2006) where these movements down the hierarchies reinforce the design decisions made at higher levels in the hierarchy. The amount of movements down the hierarchies reflects the degree of similarity between different firm’s methods of segmenting customers (customer choice hierarchy) and the different firm’s products in both alternative designs and the definition of sub-problems (product design hierarchy) (Clark, 1985). In terms of sub-problems, the coalescence of customer needs around a few related dimensions and pressures to reduce cost and standardize (Abernathy and Utterback, 1978) may cause firms to redefine the sub-problems in terms of independent modules (Ulrich, 1995; Baldwin and Clark, 2000).

Place Figure 1 about here

The choice of design alternatives and the definition of sub-problems represent a dominant design for the industry, which is consistent with the first half of Suarez and Utterback’s (1995, Figure 1) definition: “a dominant design is a specific path along an industry’s design that establishes dominance among competing paths.” As shown in the upper left hand side of Figure 1, the choice of a specific design alternative defines a single path while the definition of sub-problems into independent modules defines the emergence of independent design paths. Defining a dominant design as a path(s) is consistent with Dosi's (1982) notion of technological trajectories, which define the direction of advance within a technological paradigm (see below), and with other research on dominant designs that emphasizes a stable architecture (Anderson and Tushman, 1990) and the possibility that such a stable architecture can extend to sub-systems and components within a system (Tushman and Murmann, 1998; Murmann and Frenken, 2006).

However, depending on the industry, dominant designs will differ in terms of both the relative importance of alternative designs and sub-problems and the number of levels to which a dominant design proceeds down the design hierarchy (i.e., the degree of commonality between the design paths of different firms). The latter will depend on both the flexibility/robustness of the technology and the extent of common needs among users. The extent of common needs among users sounds similar to the second half of Suarez and Utterback's (1995) definition: "a dominant design will embody the requirements of many classes of users, even though it may not meet the needs of a particular class to quite the same extent as would a customized design."

Returning to movements within the hierarchies, technological improvements at lower levels (e.g., performance trajectories for components in an assembled product) in the product design hierarchy can change the "design tradeoffs" that are implicit at all levels in this hierarchy and thus lead to movements *back up* the hierarchies of both product design and customer choice where many of these improvements may be driven by other industries or even sectors. Both popular journalists (e.g., Gilder, 1990, 1992) and scholars have used similar concepts to explain changes at both the macro- and micro-level. At the macro-level for example, improvements in automobiles in the second half of the 20th century changed the design tradeoffs for cities and thus enabled many countries to redesign them to include suburbs and extended commuting. Similarly, improvements in transportation, communication, and computer systems in the last 10 years have changed the tradeoffs for production systems and one result has been the increased globalization of them (Friedman, 2005).

In terms of the academic literature, the concept of design tradeoffs extends the notion of performance and cost tradeoffs at the customer level, which is widely used in the marketing, decision science, and economics literature (Adner, 2002, Lancaster, 1979; Green and Wind, 1973), to tradeoffs at each level in a product design hierarchy (Alexander, 1964). This concept is similar to Dosi's (1982) characterization of a technology paradigm, which "defines its own concepts of progress based on its specific technological and economic tradeoffs," to Rosenberg's (1963, 1969) concepts of imbalances and technical disequilibria between machines and between the components within them, and to Sahal's (1985) concept of how innovations "overcome the constraints that arise from the process of scaling the technology under consideration."

The extent of the movements back up the product design and customer choice hierarchies define the degree of the technological discontinuity. For example, although some research has defined the introduction of transistors, integrated circuits (ICs), and semiconductor memory in mini-computers as technological discontinuities (Tushman and Anderson, 1986; Anderson and Tushman, 1990), these discontinuities clearly involve smaller movements back up the hierarchies than the introduction of mainframe, mini-, and personal computers, which this paper addresses. In terms of the largest movements back up the hierarchies, new product classes that are primarily due to movements back up the customer choice hierarchy are often called niche innovations (Abernathy and Clark, 1985) or disruptive technologies (Christensen, 1997). Ones that are primarily due to movements back up the product design hierarchy are often called revolutionary (Abernathy and Clark, 1985) or architectural (Henderson and Clark, 1990) innovations.

By showing how these discrete innovations fit within the proposed model, future research with the proposed model can refer to the research on these discrete innovations when analyzing how firms have moved back up the product design and customer choice hierarchies in response to changes in the design tradeoffs. Future research with the proposed model should consider the roles of organizational structure (Henderson and Clark, 1990), capabilities (Tushman and Anderson, 1986; Afuah and Bahram, 1995), complementary assets (Teece, 1986), and managerial cognitive representations (Kiesler and Sproull, 1982; Tripsas and Gavetti, 2000).

There are a number of mathematical models that can be applied to the movements up and down the hierarchies. The concepts of value trajectories and indifference curves can be used to model competition between different product classes (Adner, 2002). New product classes must also overcome the network effects of the existing product class and create a critical mass of users (Rohlfs, 2001). For example, customers often perceive a tradeoff between the performance of a new product class and its level of compatibility with an existing product class. Without compatibility with the existing product class, the new product class must have a large performance advantage over the existing product class in order for users to forgo the network effects, including both indirect (complementary) and direct ones, of the existing product class (Shapiro and Varian, 1999).

If we focus on the compatibility aspects of modular designs, Shapiro and Varian's tradeoff between performance and compatibility can also be applied to movements down the hierarchy that deal with defining sub-problems in a modular way, which has received a great deal of emphasis in the literature (Langlois and Robertson, 1992; Ulrich, 1995; Sanchez and Mahoney, 1996; Baldwin and Clark, 2000). Following the introduction of a new product class, there is decreasing marginal utility from increases in product performance (Anderson and Tushman, 1990) via an integral design and increasing marginal utility from network effects via a modular design. Although the utility from network effects are typically modeled in terms of the number of users (Shapiro and Varian, 1999), we can plot the marginal utility as a function of time since the number of users increases over time in a growing market, where this marginal utility eventually declines as shown in Figure 2 (Rohlfs, 2001). Figure 2 summarizes this tradeoff over time and helps us better understand how, when and why the emergence of a dominant design lags the emergence of a technological discontinuity/new product class (Anderson and Tushman, 1990).

Place Figure 2 about here

3. Research Methodology

The author analyzed the primary and secondary literature on the computing industry including academic papers and books from the management, economic, and historical fields, practitioner-oriented accounts, and encyclopedic histories. Through analysis of this literature, the author identified the: 1.) changes in product class through major movements back up the product design or customer choice hierarchies (and changes in business models and sales channels); 2.) technological improvements at lower levels in the hierarchy (i.e., improvements in components) that have changed the design tradeoffs thus leading to movements back up the hierarchies; 3.) movements down the hierarchies in terms of the choice of alternative designs and definitions of sub-problems in each product class; and 4.) the dominant design for each product class and the technological improvements at lower levels in the hierarchy that have impacted on the timing of their emergence.

4. Results: Brief History of the Computer Industry

Table 1 summarizes the major product classes in the computer industry and the changes in product design and customer choice hierarchies, sales channels, and business models (as compared to the previous product class) when the technological discontinuity occurred. Each product class involved movements back up the product design hierarchy where the mini-, personal, and portable computers initially represented scaled down versions of the previous product class. These new product classes used slower processors, smaller memory, shorter word lengths, and smaller instruction sets, and thus had significantly lower performance than the previous product class (Smith, 1989). There were also movements back up the customer choice hierarchy (including early users and initial applications) and changes in the sales channels and business models for the mini-, personal, and some forms of portable computers.

The technological improvements at lower levels in the computer hierarchy that have driven changes in the design tradeoffs and caused movements back up the product design and customer choice hierarchies and thus the emergence of new product classes include improved vacuum tubes, semiconductors (See Figures 3 and 4), and magnetic recording media (See Figure 5). These improvements changed the design tradeoffs for computers and thus enabled the emergence of new product classes including the ones that represented scaled down versions of the previous product class. One implication of these changes in design tradeoffs for computers has been a change in the tradeoff between price and performance for users of them. As shown in Figure 6, mainframe, mini-, and personal computer users had different tradeoffs for price and performance partly because they were using the computers for different applications (See Table 1), which are described in more detail in the subsequent sections.

The technological improvements at lower levels in the hierarchy have also impacted on the timing of dominant designs. Table 2 summarizes the dominant designs, their year of release, and the tradeoff between the decreasing marginal utility from increases in product performance via an integral design and the increasing marginal utility from network effects via a modular design (See Figure 2). The subsequent sections describe the role of these tradeoffs in the emergence of a dominant design for each product class and how these dominant designs have reinforced multiple movements down the product design and customer choice hierarchies.

Place Tables 1 and 2 and Figures 3-6 about here

4.1 Mainframe computers

Improvements in vacuum tubes and magnetic recording media enabled the creation of the first digital computers at British and U.S. universities in the mid-1940s and these improvements and the concept of stored program control (from von Neumann) enabled the creation of the industry in the early 1950s (Flamm, 1988; Ceruzzi, 1998). The improvements in vacuum tubes and magnetic tape were driven by the diffusion of radios (Inglis, 1991) and music players. Although the earliest computers used a combination of punched cards and paper tape, these were gradually replaced by magnetic tape in the early 1950s (Daniel et al, 1999).

The initial customers for these mainframe computers in the early 1950s were well-established users of analog computers, i.e., punched-card equipment (Pugh and Aspray, 1996; van den Ende and Dolfsma, 2005), which suggests that the hierarchy of customer choice for computers was initially similar to the one for punched-card equipment. These customers include the U.S. government and railroad and insurance companies. The required compatibility between these punched cards and computers is one reason for IBM's early domination of the computer business in the US and in the world. IBM controlled 90% of the U.S. punch card market and similar levels in the rest of the world in the 1950s if licensing agreements are included (Pugh and Aspray, 1996; Shurkin, 1984; Flamm, 1988).

The decision to use stored program control and magnetic core memory can be interpreted as the first moves down the product design hierarchy and the emergence of a dominant design. Stored program control required various levels of memory where each level of memory had different tradeoffs with respect to access time, cost, and capacity. These different tradeoffs caused different technologies to be initially used in the backup memory (magnetic tape), registers (primarily vacuum tubes), and cache memory where magnetic cores were finally chosen as the method of cache memory in the early 1950s over cathode ray tubes, mercury relay lines, and magnetic drums. The choice of and improvements in magnetic cores changed the performance bottleneck to logic circuits where the introduction of and improvements in transistors again changed the performance bottleneck to software (Push and Aspray, 1996; Flamm, 1988).

Improving software, which represents the next move down the product design hierarchy, was much more difficult than improving hardware. Although users developed most of the software for the first computers (Campbell-Kelly, 2003), development quickly moved to cooperative projects between users and manufacturers (e.g., the Sabre project with IBM and American Airlines) and to a greater extent the full development of software by manufacturers. Manufacturers developed programming languages such as Fortran and Cobol (Campbell-Kelly, 2003) and general-purpose software such as inventory, accounting, logistics, financial, actuarial, and payroll, which they offered for free to their customers as part of the customers leasing the hardware (Cerruzi, 1998). The emergence of this general software is consistent with a dominant design that embodies the requirements of many users (Suarez and Utterback, 1995) and also reflects the emergence of well-defined segments in the customer choice hierarchy. To protect their hardware, the computer manufacturers prohibited hardware modifications including the use of custom-built input/output devices and did this by not providing detailed specifications for the computers (Rifkin and Harrar, 1987; Mowery, 1996).

IBM's introduction of a compatible line of computers called the IBM System/360 in 1964 can be seen as the next move down the design hierarchy and the first one that involved defining sub-problems in terms of relatively independent modules including peripherals, software, and hardware. The decision to define sub-problems in terms of these independent modules can also be represented in terms of a tradeoff between decreasing marginal utility from increases in product performance via an integral design and increasing marginal utility from network effects via modularity (See Table 2 and Figure 2). For example, as shown in Table 3, between 1953 and 1964 increases in memory capacity (66 times) and speed (43 times) for hardware enabled the use of more complex software (100 times the number of lines of code) but the marginal utility of this greater software complexity/performance was falling. At the same time, the increasing installed base of computers, applications for these computers, and the relatively small increases in programmer productivity (2-3 times) increased the need for the reuse of software and thus a modular design.

Place Tables 3 about here

IBM designed its subsequent machines to be backward compatible with the System/360, other firms released IBM-compatible products such as magnetic memory (Pugh, 1995), and firms began offering packaged software following IBM's decision to un-bundle hardware and software in 1969 (Mowery, 1996: Campbell-Kelly, 2003). One reason that firms did these things is because subsequent improvements in the performance of components such as transistors, integrated circuits, and magnetic memory reinforced the IBM System/360 as a dominant design. Although some of these improvements in component performance led to increases in word size and instruction set, these improvements were compatible with the System/360 and were aimed at the same applications via the same business model. Thus the increased word size and instruction sets had little effect on the customer choice hierarchy and largely represented lateral movements in the product design hierarchy.

4.2 Mini-computers

Reductions in feature size and thus improved performance in ICs, which were primarily driven by military applications, changed the design tradeoffs and led to movements back up the customer choice and product design hierarchies and the emergence of a new class of computers called mini-computers in the mid-1960s. Although DEC had sold logic modules from the late 1950s, DEC's PDP-8, which was released in 1965, is usually considered the first mini-computer (Rifkin and Harrar, 1988; Baldwin and Clark, 2000). In terms of movements back up the product design hierarchy, the mini-computer was a scaled down version of the mainframe that used a shorter word length (12-bits) and instruction set and made the mini-computer much cheaper to produce than the smallest IBM System/360 machine (Ceruzzi, 1998). For example, if a user chose to rent this machine, it could do so for about $525 a month or 6% the cost of IBM's smallest System/360, the Model 30 (Mowery, 1996).

Movements back up the customer choice hierarchy are represented by the differences between the customers and applications for the mini- and mainframe computers and the different tradeoffs between price and performance that mini- and mainframe computer users make (See Figure 6). Scientific and engineering departments used mini-computers for product design and process control where they developed their own software and made modifications to the input-output devices. The user's development of software and modifications to the hardware were made possible by DEC's business model that included extensive product documentation and the sale as opposed to only leasing of the mini-computers. The need to move back up the customer choice hierarchy is usually cited as a major reason why only one incumbent, IBM, offered a mini-computer and this was done ten years after DEC introduced the PDP-8 (Christensen, 1997). On the other hand, a number of new entrants such as Data General, Prime Computer, HP, and Wang did focus on the new customers and applications served by the mini-computer (Flamm, 1988; Rifkin and Harrar, 1988).

The first movements down the product design hierarchy are represented by mini-computers that used IC-based processors, MOS memory, and magnetic disks (Jackson, 1997). The improved processors and memory enabled DEC to expand the word length and instruction set, which were needed to handle more complex software and later to provide compatibility with 8-bit computers (Rifkin and Harrar, 1988). IBM's System/360 had defined word lengths with multiplies of 8-bits (now considered one byte) as an industry standard and DEC made the word length of its mini-computers compatible with this standard through the introduction of its PDP-11 line of 16-bit machines in 1970 (Rifkin and Harrar, 1988).

The timing of DEC's introduction of the PDP-11 is consistent with the tradeoff between decreasing marginal utility from increases in product performance via an integral design and increasing marginal utility from network effects via a modular design (See Table 2), as is represented by Figure 2. Although DEC initially focused on increasing the complexity of software that could be run with its machines by doubling and tripling the 12-bit word length of its PDP-8 machine, its introduction of a 16-bit machine reflected the increasing importance of compatibility between mini-computers and between them and mainframe computers (Rifkin and Harrar, 1988).

As with the mainframe computers, further improvements in integrated circuits, magnetic memory, and software development tools led to increases in word length, improvements in the performance of mini-computers, and an expansion of their market where time-sharing allowed even smaller organizations to gain access to computers (Rifkin and Harrar, 1988). DEC's introduction of the VAX line of mini-computers in the mid-1970s reinforced the PDP-11 as a dominant design and facilitated the emergence of packaged software such as word processing, database, and CAD (computer-aided design) systems (Mowery, 1996; Campbell-Kelley, 2003). The emergence of this packaged software is consistent with a dominant design that embodies the requirements of many users (Suarez and Utterback, 1995) and also reflects the emergence of well-defined segments in the customer choice hierarchy.

4.3 Personal Computers

Improvements in ICs, semiconductor memory, magnetic recording density, and cathode ray tubes changed the design tradeoffs for computers and led to movements back up the customer choice and product design hierarchies and the emergence of a new class of computers called personal computers (PCs) in the mid-1970s. Reductions in feature size and thus increases in the number of transistors that could be placed on a chip enabled Intel to place a central processing unit on a single chip called a microprocessor, which was initially used in calculators and aviation and scientific instruments (Jackson, 1997). Improvements in magnetic recording density made it possible for hard disk manufacturers to introduce smaller disk drives (5.25 inch and later 3.5 inch), which matched the needs of PCs better than mini- and mainframe computers (Christensen, 1997).

In terms of movements back up the product design hierarchy, the PC was a scaled-down version of the mainframe that used a smaller CPU, semiconductor memory, instruction set, and word length (8-bits). This made the PC much cheaper to produce than the smallest mini-computer and thus provided users with a different tradeoff between price and performance (See Figure 6). In terms of movements back up the customer choice hierarchy, individuals, including hackers and small firms were the initial customers where hackers were happy to write their own software. The first packaged software applications were game and education software and business software such as spreadsheet and word processing were not introduced until 1979 and did not become the main PC applications until the early 1980s (Campbell-Kelly, 2003). The new business model was the sale of PCs and pre-packaged software through the mail and in retail outlets such as Computerland.

The first movements down the hierarchies can be seen in the release of PCs in 1977 (Commodore PET, Tandy TRS-80 Model I, Apple II) that contained monitors (i.e., cathode ray tubes), keyboards, floppy disks, and later hard disks. These features were not available on the PCs released in 1975 and 1976 and the existence of floppy disk drives supported a market for packaged software (Campbell-Kelly, 2003) where large amounts of third party software were developed for the Apple computer and computers that ran the CP/M operating system (Gupta and Toong, 1985; Steffens, 1994). Like the mini-computer, the emergence of this packaged software is consistent with a dominant design that embodies the requirements of many users (Suarez and Utterback, 1995) and also reflects the emergence of well-defined segments in the customer choice hierarchy.

IBM's introduction of its PC in 1981 further expanded the market for packaged software and a design that embodied the requirements of many users. Many researchers argue that the greater openness and modularity of the IBM PC and the stronger brand name of IBM are major reasons for the establishment of the IBM PC as the dominant design partly since they contributed to the emergence of a large amount of third-party software and hardware (Teece, 1986; Langlois, 1993; Grindley, 1995; Rohlfs, 2001). Unintentionally, the ability to reverse engineer IBM's BIOS chip and the open modular architecture enabled other firms to manufacture "clones," which made the IBM PC an even more open and modular product than IBM had intended, where the critical components were provided by Microsoft and Intel. One reason that IBM elected to offer an open and modular architecture was because it was entering the market late and thus did not have time to internally develop the operating system and microprocessor. The reason IBM entered late was because it initially believed that few people wanted a personal computer (i.e., slow to go back up the customer choice hierarchy), an argument that DEC's President Kenneth Olsen also made (Grindley, 1995; Carrol, 1983; Rifkin and Harrar, 1988).

However, the emergence of the IBM PC as a dominant design is also consistent with the tradeoff between the decreasing marginal utility from increases in product performance via an integral design and the increasing marginal utility from network effects via a modular design (See Figure 2). The IBM PC provided a superior tradeoff between performance and compatibility through the use of a hybrid 8- and 16-bit microprocessor. The 8-bit capability enabled the IBM PC to run existing packaged software that had been developed for the CP/M operating systems (network effects) while the 16-bit capability enabled the IBM to run new and better software (i.e., increased performance). The success of WordPerfect and Lotus provide evidence of this increased performance. They introduced word processing and spreadsheet software that were customized for the 16-bit microprocessor and thus ran faster than WordStar and Visicalc did on the IBM PC since both Visicalc and WordStar had been optimized for the 8-bit chips used in Apple's computers and in those computers that ran the CP/M operating system (Steffens, 1994).

Previously released computers, even if they had offered the degree of openness available with the IBM PC, probably did not provide sufficient performance to become a dominant design while subsequently released computers could not overcome the network effects of the IBM PC/Wintel. Obviously the Altair and other early computers did not offer sufficient performance to become a dominant design while the success of WordPerfect and Lotus suggests that the IBM PC did offer significant performance improvements to users over previous computers like those from Apple. On the other hand, the fact that firms did not release a PC with a 32-bit micro-processor for more than 10 years suggest that these firms believed that performance improvements could not overcome the network effects achieved by the IBM PC and many now argue it is very difficult for new products to displace the Wintel standard given Wintel's strong network effects (Bresnahan, 2004).

Nevertheless, the tradeoff between product performance via an integral design and network effects via a modular design is only clear in hindsight. One reason that many firms, such as Xerox, DEC, HP, Zenith, and Wang were slow to introduce 16-bit machines in the early 1980s was because they did not want to abandon CP/M, which they considered to be the industry standard and only ran on 8-bit machines (Steffens, 1994). Even Apple sold expansion boards so that software developed for the CP/M operating system could be run on the Apple computer (Steffens, 1994; Mowery, 1996; Ferguson and Morris, 1993; Campbell-Kelly, 2003) and articles published as late as 1985 gave the CP/M operating system the same status as the IBM PC and Apple computers (Gupta and Toong, 1985).

Following the establishment of the IBM PC as the dominant design for PCs, improvements in microprocessors, semiconductor and hard disk memory, display monitors, modems, and printers continued to drive improvements in the performance of PCs throughout the 1980s and the 1990s. In turn these improvements led to a rapid diffusion of PCs in companies, homes, and universities where there was positive feedback between the diffusion of PCs and the Internet. These improvements have not required large movements back up the product design hierarchy for PCs[1] while the diffusion of the Internet has led to changes in the sales channel and business models. New entrants like Dell moved faster to sell computers over the Internet and introduce the appropriate business model than other firms such as Compaq that were restrained by their existing sales channels (Kenney and Curry, 2001).

4.4 Portable Computers

Improvements in microprocessors, semiconductor and hard disk memory, and displays continue to change the design tradeoffs and have led to repeated movements back up the customer choice and product design hierarchies and the emergence of new product classes for computers. Although Osborne Computers introduced the first portable (better known as a luggable) computer in 1981, since Osborne was not able to find new applications and users and its computer was incompatible with desktop machines due to its slow processing speed, the market for portable computers did not begin to grow until the late 1980s when faster processors provided compatibility with desktop machines and LCD screens, better batteries, and smaller components like hard-disk drives made laptop machines possible (Steffens, 1994).

Similar improvements in component performance led to another movement back up the customer choice and product design hierarchies and the introduction of a new type of portable computer called PDAs (Personal Digital Assistant) in the early 1990s. Some

[1] Exceptions include: 1) the redesign of the PC architecture, which was led by Intel, in order to prevent the PC bus from being a bottleneck to performance (Gawer and Cusumano, 2002); and 2) Microsoft's introduction of a graphical user interface and the integration of Word, Powerpoint, Excel, and later Explorer (Campbell-Kelly, 2003).

firms such as Apple and Microsoft focused on the existing hierarchy of customer choice of PC users but the complexity of these functions caused the products to be expensive and very slow. For example, it took six minutes to input a single address and recognition of hand written words did not exceed 80% in Apple‘s Newton, which was released in 1993. Microsoft’s PDA also had similar problems in spite of the fact that it released its PDA in 1996 or one year after Palm released its successful Palm Pilot (Butter and Pogue, 2002).

On the other hand, some firms did not find the right combination of functions and price even when they went back up the customer choice and design hierarchies to look for them. For example, some calculator manufacturers initially used the customer choice hierarchy from calculators to choose product features and prices for PDAs. For example, as a partner to Palm, Casio recommended in 1992 that Palm produce a PDA with 100 hours of battery life and a large number of features including most facsimile and printer drivers. The requirement for long battery life required Palm to use a very slow microprocessor (7 MHz), which was not fast enough to quickly and effectively handle many features including the facsimile and printer drivers (Butter and Pogue, 2002).

Nevertheless, Palm was the first firm to effectively move back up the product design and customer choice hierarchies and release a product in 1996 that contained the specific functions and price demanded by the market. Palm reduced the number of functions to a calendar, address book and memo pad and reduced the battery requirements to a few hours; the latter enabled the use of a faster microprocessor that could effectively perform the reduced number of functions. The Palm Pilot used inexpensive batteries, an inexpensive touch-tone display, and a synchronization cradle, and it eliminated the keyboard and other features such as backlighting and expansion cards in order to sell the product for less than $300. The success of the Palm Pilot and the network effects associated with this success, which included an increasing amount of third party software, caused the Palm Pilot to become the dominant design for PDAs that handled schedules and address books (Butter and Pogue, 2002).

However, improvements in chips, magnetic recording density, and wireless and later wireline systems have changed the design tradeoffs many times and required multiple movements back up the product design and customer choice hierarchies, which other firms have done better than Palm has. For example, improvements in chips and wireless systems enabled Research in Motion (RIM) to include a phone receiver and mail client in a PDA thus enabling RIM to offer its now famous Blackberry device. Apple computer included a small magnetic hard disk and microprocessor in its i-Pod that enable users to play songs they have downloaded on their PCs. NTT DoCoMo modified Internet standards including the markup language, mail client, and music and image formats for the small processing power, keyboard, and screen of the mobile phone in its successful i-mode service.

It is likely that improvements at lower levels in the product design hierarchy, their impact on design tradeoffs, and how firms respond to these changes in design tradeoffs will continue to shape competition between these portable devices. For example, improvements in ICs, memories, and also wireless networks have enabled mobile phones, i.e., smart ones, to be used as music players and further improvements may cause mobile phones to replace i-Pod-like devices in this application. On the other hand, improvements in processing power and magnetic disks have enabled Apple Computer to expand the functions of its i-Pod to video viewing. Although improvements in flash memory and other chips may cause mobile phones to also be used in this application, competition between phones and i-Pod-like devices will probably also depend on the comparative size of the screens, the amount of memory demanded by users, and the relative improvements in flash and hard disk memory.

In terms of business applications, NTT DoCoMo modified Internet mail for the small size of the mobile phone thus enabling all of its subscribers to receive short Internet mail messages on their phones, something that is only a standard feature on phones in Japan. Improvements in mobile phone microprocessors continue to expand the Internet mail capabilities of standard Japanese phones thus enabling users to obtain a similar level of service from regular mobile phones as from RIM's Blackberry device. The relative improvements in processing speed, memory, and screen size, their impact on design tradeoffs, and how firms respond to these changes in design tradeoffs will impact on this competition between Blackberrys and phones in applications such as mail, browsing, and access to corporate databases.

5. Discussion

The purpose of this paper was to introduce a model of technological change that addresses the sources and timing of technological discontinuities and dominant designs better than the existing literature. The use of a single industry suggests that we must be careful about generalizing to other industries. With this caveat in mind, this paper has made several contributions to our understanding of both technological discontinuities and dominant designs.

With respect to technological discontinuities, which this paper calls new product classes, the use of product design and customer choice hierarchies and the concept of design tradeoffs provide insights that are not found in the existing literature.

Technological improvements at lower levels in the product design hierarchy change the design tradeoffs and thus require firms to rethink the product design and customers where many of these improvements are driven by other industries. For example, the improvements in vacuum tubes and magnetic tape that changed the design tradeoffs for computers (analog ones) were driven by the radio and recorded music industries. The improvements in ICs that changed the design tradeoffs for computers and led to the emergence of mini-computers were driven by military applications while the improvements in microprocessors that also changed the design tradeoffs for computers and led to the emergence of PCs were driven by calculators and aviation and scientific instruments.

The exact timing of the discontinuities have depended on how firms use these improvements to rethink their products, customers, business models, and sales channels, which will partly depend on the design tradeoffs that are inherent in the product design hierarchy. Movements back up the customer choice hierarchy reflect changes in the users and applications and any movements back up this hierarchy may reduce the improvements in performance and cost that are needed for growth in the new product class to occur. For example, the existence of scientists, engineers, and hackers that required lower performance than did users of previous product classes made it possible for the mini- and personal computers to diffuse before their performance had reached the level of the previous product class. These new users also reflected the changes in design tradeoffs that improvements at lower levels in the hierarchy brought about. Differential rates of improvements in components not only changed the tradeoffs between vacuum tubes, transistors, and ICs; improvements in these components, particularly ICs also changed the tradeoffs between price and performance for users

(See Figure 6) and thus enabled the change in users.

These results go beyond those of previous research that have shown examples of interactions between component and system innovations/discontinuities (Tushman and Murmann, 1998; Malerba, et al, 1999). For example, Malerba et al (1999) describe how innovations such as the microprocessor enabled the development of the PC. The proposed model represents this phenomenon at a much deeper level by describing the interaction between improvements in components, design tradeoffs, and movements back up product design and customer choice hierarchies where there were *independent* movements back up them for both computers and to semiconductors. Although the semiconductor industry is not the main subject of the paper, the initial use of microprocessors in calculators and aviation and scientific instruments represents the movements back up the customer choice hierarchy for semiconductor manufacturers.

With respect to dominant designs, this paper extends Suarez and Utterback's (1995) concept of a dominant design as a design path and shows the relationship between movements down the product design and customer choice hierarchies and the emergence of a dominant design. For example, the release of the IBM System 360 followed and reinforced a number of other movements down the product design hierarchy including the use of stored program control, magnetic core memory, transistors and general purpose software each of which can be seen as part of the dominant design for mainframe computers. The multiple movements down the hierarchy that preceded the release of the PDP-11 included mini-computers with MOS memory and magnetic disks. For PCs, the release of the IBM PC followed and reinforced multiple movements down the product design hierarchy including the release of computers with monitors, keyboards, and floppy and hard disks.

The emergence of this dominant design path is also consistent with a dominant design that embodies the requirements of many users (Suarez and Utterback, 1995) where *the definitions* of these users represent movements down the customer choice hierarchies for mainframe, mini-, and personal computers. For example, the widespread use of general purpose software in mainframes such as inventory, accounting, logistics, financial, actuarial, and payroll suggests that this software embodied the requirements of many users and reflected the emergence of well-defined segments in the customer choice hierarchy. Similar arguments can be made for the most widely used software in mini- (word processing, database, and CAD) and personal (game, education, and spreadsheet software) computers.

A second contribution to the area of dominant designs concerns the timing of designs that define sub-problems in a modular way. Although it is possible to define a number of design decisions in mainframe, mini-, and personal computers as modular design decisions, this paper has focused on those products (and the design decisions they embody) that are emphasized as dominant designs in the literature such as the IBM System/360, PDP-11, and the IBM PC. All of these dominant designs involved defining hardware and software design problems in a modular way and they showed the tradeoff between decreasing marginal utility from increases in product performance and increasing marginal utility from network effects. The improvements at lower levels in the product design hierarchy drove improvements in overall product performance and thus delayed the emergence of dominant designs.

The characterization of this tradeoff between decreasing marginal utility from increases in product performance and increasing marginal utility from network effects helps us better understand why a dominant design would lag the emergence of a technological discontinuity/new product class (Anderson and Tushman, 1990) and suggests that when there are very large network effects, the dominant design might not lag a technological discontinuity. The characterization of this tradeoff also helps firms better understand the timing of dominant designs and how they should manage the tradeoff between compatibility (with existing products) and performance.

Finally, these results also help us understand how to analyze the future of the computer industry in terms of both dominant designs and technological discontinuities (i.e., new product classes). The discussion in the last four paragraphs can help firms better understand how a dominant design may emerge in for example mobile Internet phones. As for technological discontinuities, a new product class must overcome the network effects that exist with the existing product class. With PCs, it appears that the chances of this occurring are small thus lending support to governmental action (Bresnahan, 2004). On the other hand, it appears that improvements in chips, magnetic recording density, and wireless and wireline systems continue to change the design tradeoffs for portable computers. Analyzing how these improvements change the design tradeoffs and how these changes in design tradeoffs impact on movements up and down the hierarchies can help us better understand how today's successful products (e.g., i-Pod and Blackberry-type devices) may be surpassed in the near future with new products that represent movements back up the product design and customer choice hierarchies.

6. References

Abernathy, W., Clark, K. 1985. Innovation: Mapping the winds of creative destruction, Research Policy 14, 3-22.

Abernathy, W. and Utterback, J. 1978. Patterns of innovation in technology, Technology Review 80, 40-47.

Adner, R. 2002. When are technologies disruptive? A demand-based view of the emergence of competition, Strategic Management Journal 23 (8), 667 – 688.

Afuah, A., Bahram, N., 1995. The hypercube of innovation, Research Policy, 24: 51-76.

Alexander, C. 1964. Notes on the Synthesis of Form, Cambridge, MA: Harvard University Press.

Anderson, P. and Tushman, M. 1990. Technological discontinuities and dominant designs: A cyclical model of technological change, Administrative Science Quarterly 35, 604-633.

Baldwin, C. and Clark, K. 2000. Design Rules, Volume 1: The Power of Modularity: Cambridge, MA: MIT Press.

Bresnahan, T. 2004. Creative destruction in the PC industry, Presented to the Conference on IT Innovation, Tokyo, December (http://www.stanford.edu/~tbres).

Butter, A. and Pogue, D. 2002. Piloting Palm: The inside story of Palm, Handspring, and the birth of the billion-dollar handheld industry, NY: John Wiley & Sons.

Campbell-Kelly, M. 2003. From airline reservations to sonic the hedgehog: A history of the software Industry, Boston: MIT Press.

Carrol, P. 1983. Big blues: the unmaking of IBM, NY: Crown Books.

Ceruzzi, P. 1998. A history of modern computing, Cambridge, MA: MIT Press

Chesbrough, H. 2003. Open innovation: The new imperative for creating and profiting from technology, Boston: Harvard Business School Press.

Christensen, C. 1997.The innovator's dilemma, Boston: Harvard Business School Press.

Clark, K. 1985. The interaction of design hierarchies and market concepts in technological evolution, Research Policy 14, 235-251.

Daniel, E., Mee, D., and Clark, M. 1999. Magnetic recording: The first 100 years, NY: IEEE Press.

Dosi, G. 1982. A suggested interpretation of the determinants and directions of technical change, Research Policy 11 (3), 147-162.

Ferguson, C. and Morris, C. 1993. Computer Wars: The Post-IBM World, NY: Times Books.

Flamm, K. 1988. Creating the computer: government, industry, and high technology, Washington DC: Brookings Institution.

Friedman, T. 2005. The World Is Flat: A Brief History of the Twenty-first Century, NY: Farrar, Straus and Giroux.

Gilder, G. 1990. Microcosm: The Quantum Revolution in Economics and Technology, NY: Free Press.

Gilder, G. 2002. Telecosm: The World After Bandwidth Abundance, NY: Touchstone.

Green, P. and Wind, Y. 1973. Multi-attribute Decisions in Marketing: A Measurement Approach, Hinsdale, IL: Dryden Press.

Grindley, P. 1995. Standards strategy and policy: Cases and stories. Oxford: Oxford University Press.

Gupta, A. and Toong, H. 1985. The first decade of personal computers, in A. Gupta, H. Toong (Eds), Insights into the personal computer, NY: IEEE Press.

Henderson, R. and Clark, K. 1990. Architectural innovation: The reconfiguration of existing product technologies and the failure of established Firms, Administrative Science Quarterly 35, 9-30.

Inglis, A. 1991. Behind the Tube: A History of Broadcasting Technology and Business, Boston: Focal Press.

Jackson, T. 1997. Inside Intel: Andy Grove and the Rise of the World's Most Powerful Chip Company, NY: Dutton.

Kenney, M. and Curry, J. 2001. We Want to be Like Mike: The Internet and the PC Value Chain, Tracking a Transformation: E-commerce and the Terms of Competition in Industries, The BRIE-IGCC Economy Project, Washington D.C.: Brookings.

Kiesler S and Sproull, L. 1982. Managerial response to changing environments: Perspective on problem sensing from social cognition, Administrative Science Quarterly 27, 548 – 570.

Lancaster, K. 1979. Variety, Equity, and Efficiency, NY: Columbia University Press.

Langlois, R. 1993. External Economics and Economic Progress: the case of the Microcomputer Industry, Business History 66, 1- 50.

Langlois, R. and Robertson, P. 1992. Networks and innovation in a modular system: lessons from the microcomputer and stereo component industries, Research Policy 21, 297-313

Malerba, F., Nelson, R., Orsenigo, L., Winter, S. 1999. "History-Friendly Models of Industry Evolution: The Computer Industry," Industrial and Corporate Change 8, 3-40.

Moore, G. 2004. "No Exponential is Forever……..," ftp://download.intel.com/research/silicon/Gordon_Moore_ISSCC_021003.pdf

Mowery, D. 1996, The International Computer Software Industry, NY: Oxford

University Press.

Murmann, J. and Frenken, K. 2006. “Toward a Systematic Framework for Research on Dominant Designs, Technological Innovations, and Industrial Change,” in press, Research Policy.

O’Neil, P. 2003. “CMOS Technology and Business Trends: Can the semiconductor industry afford to continue advancing,”
http://ewh.ieee.org/r5/denver/sscs/Presentations/2002.12.ONeill.pdf

Pugh, E. 1995. Building IBM: Shaping an industry and its technology, MIT Press.

Pugh, E. and Aspray, W. (1996). A history of the information machine, IEEE Annals of the History of Computing 18 (2), 70-76.

Rifkin, G. and Harrar, G. 1983. The ultimate entrepreneur: The story of Ken Olsen and Digital Equipment Corporation, NY: Contemporary Books.

Rohlfs, J. 2001. Bandwagon Effects in High-Technology Industries, Cambridge, MA: MIT Press.

Rosenberg, N. 1963. Technological Change in the Machine Tool Industry, 1840-1910, The Journal of Economic History 23 (4), 414-443.

Rosenberg, N. 1969. The Direction of Technological Change: Inducement Mechanisms and Focusing Devices, Economic Development and Cultural Change 18 (1), 1-24.

Sahal, D. 1985. Technological guideposts and innovation avenues, Research Policy 14: 61-82.

Sanchez, R. and Mahoney, J. 1996. Modularity, flexibility, and knowledge management in product and organization design, Strategic Management Journal 17, 63-76.

Shapiro, C. and H. Varian 1999, Information Rules, Harvard Business School Press.

Shurkin, J. 1984. Engines of the Mind: A History of the Computer, NY: Norton &

Company.

Smith, R. 1989. A Historical Overview of Computer Architecture, IEEE Annals of the History of Computing 10 (4), 277-303.

Simon, H. 1996. The Sciences of the Artificial, Cambridge: MIT Press, 1996.

Steffens, J. 1994. Newgames: Creating competition in the PC revolution, NY: Pergamon Press.

Stevens, L. 1999. Data storage on hard magnetic disks, in E. Daniel, C. Mee, M. Clark (Eds), Magnetic recording: The first 100 Years, NY: IEEE Press.

Suarez, F. and Utterback, J. 1995. Dominant Designs and the Survival of Firms, Strategic Management Journal 16, 415-430.

Teece, D. 1986, Profiting from technological innovation: Implications for integration, collaboration, licensing, and public policy, Research Policy 15, 285 - 305.

Tripsas, M. and Gavetti, G. 2000. Capabilities, cognition, ad inertia: evidence from digital imaging, Strategic Management Journal 21, 1147 – 1161.

Tushman, M. and Anderson, P. 1986. Technological discontinuities and organizational environment, Administrative Science Quarterly 31, 439-456.

Tushman, M. and Murmann, J. 1998. Dominant Designs, Technology Cycles, and Organizational Outcomes, Research in Organizational Behavior 20, 231-266.

Ulrich, K. 1995. The role of product architecture in the manufacturing firm, Research Policy 24, 419-440.

Utterback, J. 1994. Mastering the dynamics of innovation, Boston, MA: Harvard Business School Press.

van den Ende, J. and Dolfsma, W. 2005. Technology-push, demand-pull and the shaping of technological paradigms, Journal of Evolutionary Economics 15 (1), 83 - 99.

Table 1. Major Product Classes in the Computer Industry and Relevant Changes during the Technological Discontinuity

Product Class	Movements back up the Hierarchies and Other Changes				
	Product Design	Customer Choice		Sales Channels	Business Model
		Early Users	Applications		
Main-Frame	Add vacuum tubes to punch card equipment	No changes (Existing punch-card users and their business systems)		No changes (Existing sales force)	No changes (Lease computers and software)
Mini-Computer	Scaled-down version of mainframes	Scientific & engineering companies	Engineering analysis and process control	Corporate mail orders, later sales force	Sell not lease. Extensive documentation.
Personal Computer (PC)	Scaled-down version of mini-computers	Individuals (home, university, small business)	Games spreadsheet, word processing	Individual mail order and later retail, Internet	Modular and open systems, sale of packaged software
Portable	Scaled-down version of PCs	Different for laptops, PDAs and "smart" phones		Retail and Internet	Different for laptops, PDAs and "smart" phones

PDAs: Personal digital assistants; Sources: (Rifkin and Harrar, 1983; Flamm, 1988; Langlois, 1993; Ceruzzi, 1998; Campbell-Kelly, 2003; Butter and Pogue, 2002)

Table 2. Dominant Designs for the Major Product Classes in the Computer Industry

Product Class	Dominant design	Year Released	Decreasing Marginal Utility from Increases in Product Performance	Increasing Marginal Utility from Network Effects (Reuse of Software)
Main-Frame	IBM 360	1964	Improved hardware and thus more complex software	Rising installed base and need for software reuse
Mini-Computer	DEC's 16-bit PDP-11	1970	Above manifested in form of longer word length	Above manifested in increased need for compatibility
Personal Computer (PC)	IBM PC (later called Wintel)	1981	Above manifested in more complex software	Above manifested in increased need for compatibility

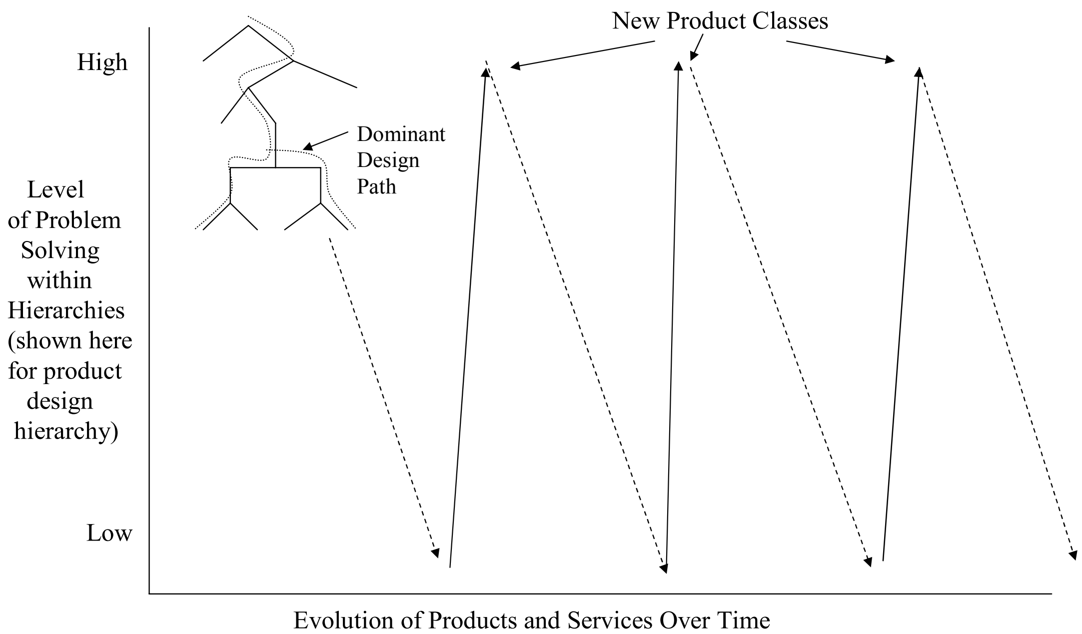
Sources: (Rifkin and Harrar, 1983; Flamm, 1988; Langlois, 1993; Steffens, 1994; Ceruzzi, 1998; Campbell-Kelly, 2003)

Table 3. Changes in Computers Between 1953 and 1964

Item	Changes
Memory capacity	Increased by 66 times
Speed	Increased by 43 times
Price	Dropped by 40 times to 2.5% the 1953 levels
Lines of code	Increased by 100 times
Programming productivity	Increased by 2-3 times

Source: summarized from data in Figure 4.1, Table 4.2, and text of Campbell-Kelly, 2003

Figure 1. Evolution of Level of Problem Solving in Hierarchies as a Function of Time



Note: Dotted lines represent movements down the hierarchies and solid lines represent movements back up the hierarchies

Figure 2. Tradeoff Between Product Performance via an Integral Design and Network Effects via a Modular Design

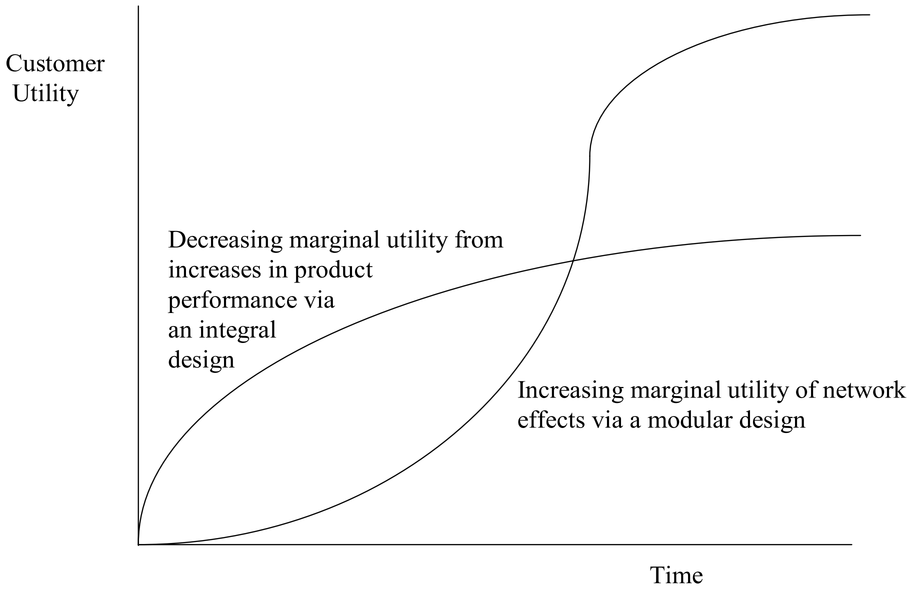
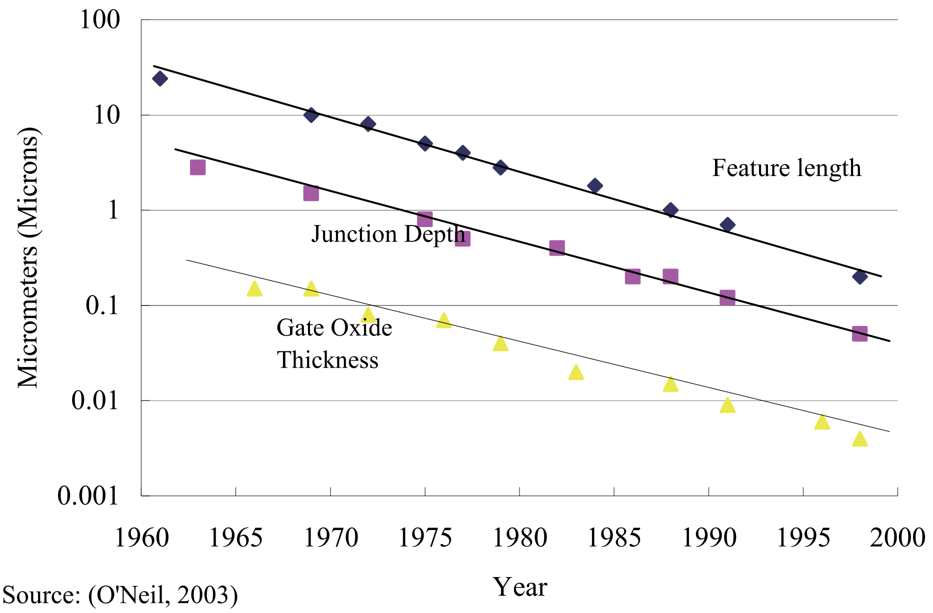
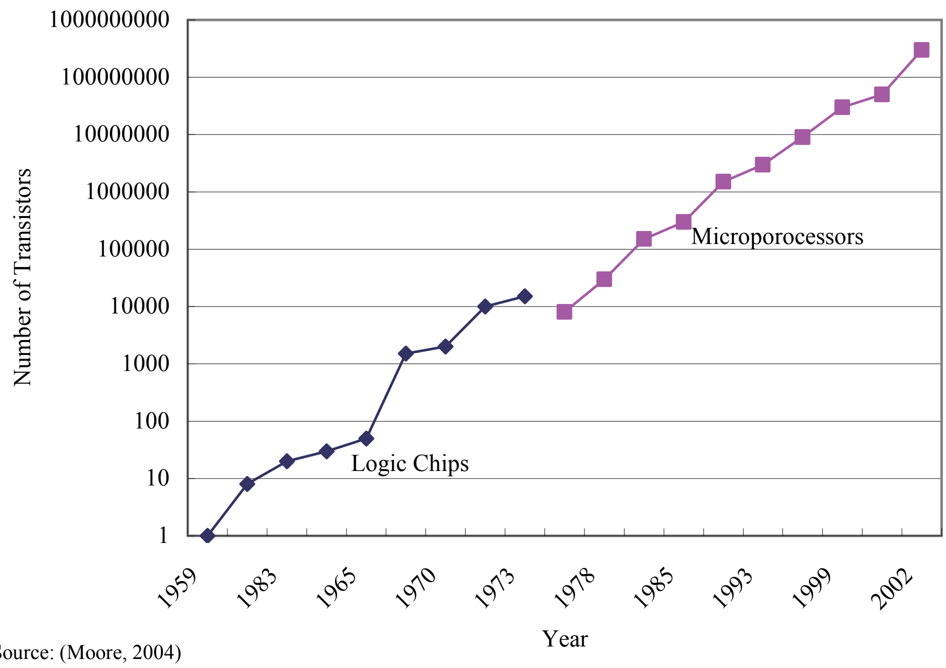


Figure 3. Declining Feature Size



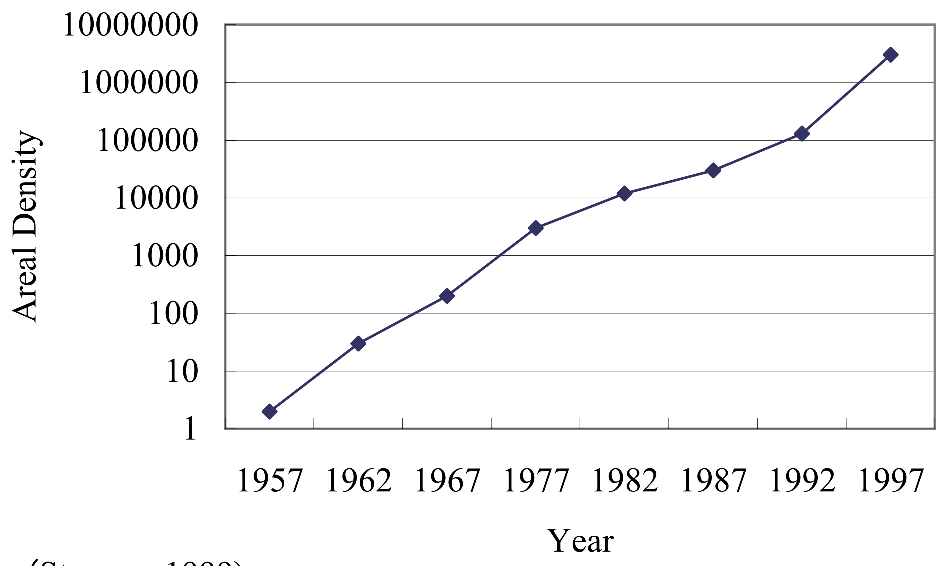
Source: (O'Neil, 2003)

Figure 4. Number of Transistors Per Chip



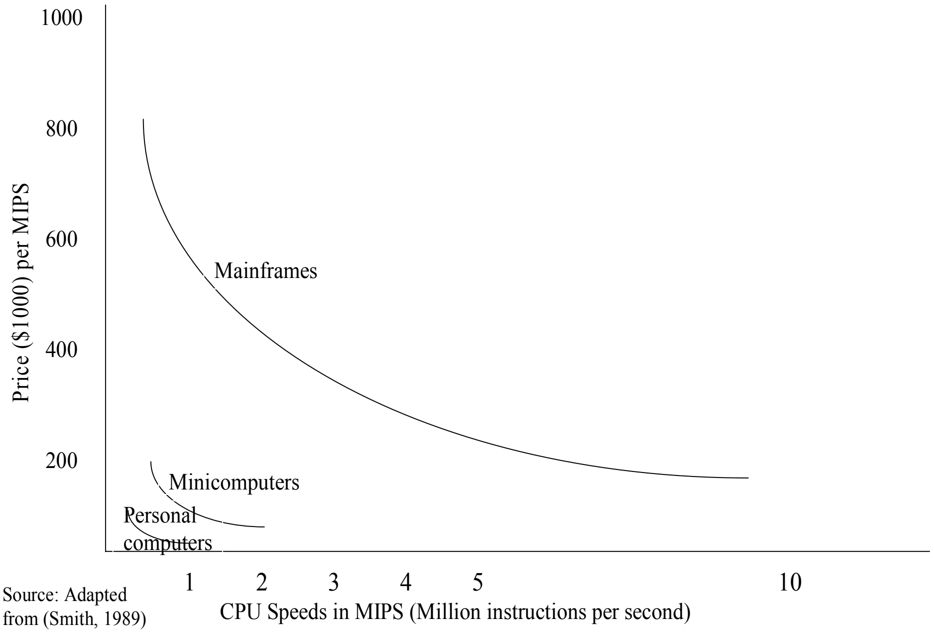
Source: (Moore, 2004)

Figure 5. Growth in Hard Disk Recording Density



(Stevens, 1999)

Figure 6. Relationship Between Prices and Performance (1981 data) for Different Product Classes of Computers that Reflect the User's Different Tradeoffs Between Price and Performance for Different Product classes of computers (1981 data)



Source: Adapted from (Smith, 1989)