HARVARD | BUSINESS | SCHOOL

Design Rules, Volume 2: How Technology Shapes Organizations

Chapter 15 The IBM PC

Carliss Y. Baldwin

Working Paper 19-074

Design Rules, Volume 2: How Technology Shapes Organizations

Chapter 15 The IBM PC

Carliss Y. Baldwin
Harvard Business School

Working Paper 19-074

Design Rules, Volume 2: How Technology Shapes Organizations

Chapter 15 The IBM PC

By Carliss Y. Baldwin

Note to Readers: This is a draft of Chapter 15 of *Design Rules, Volume 2: How Technology Shapes Organizations*. It builds on prior chapters, but I believe it is possible to read this chapter on a stand-alone basis. The chapter may be cited as:

Baldwin, C. Y. (2019) "The IBM PC," HBS Working Paper (January 2019).

I would be most grateful for your comments on any aspect of this chapter! Thank you in advance, Carliss.

Abstract

The IBM PC was the first digital computer platform that was open by as a matter of strategy, not necessity. The purpose of this chapter is to understand the IBM PC as a technical system and set of organization choices in light of the theory of how technology shapes organizations. In Chapter 7, I argued that sponsors of large technical systems (including platform systems) must manage the modular structure of the system and property rights in a way that solves four inter-related problems:

- Provide all essential functional components;
- Solve system-wide technical bottlenecks wherever they emerge;
- Control and protect one or more strategic bottleneck; and
- Prevent others from gaining control of any system-wide strategic bottleneck.

I use this framework to understand how IBM initially succeeded with the PC platform and then lost its position as platform sponsor in the industry it had created.

Introduction

Open product platforms were a new business model for mature firms in the computer industry in the 1980s and 1990s. As Andy Grove described, small firms might enter a new market niche as specialists, but as they became successful and the niche expanded, such firms tended to become vertically integrated. In common with the dominant model of mass production established in the late 19th Century (see Chapters 8-11), the largest firms incorporated many complementary activities and related step processes within their boundaries.

Moore's Law changed the nature of the game. An increase in the exogenous rate of technical change arising from the physics of semiconductors put a premium on modular flexibility over efficiency. (See Chapter 13.) As computer systems became universally modular (following the lead of System/360), the number of thin crossing points multiplied, opening up opportunities for specialist firms making modules. As a result, although vertically integrated firms still dominated the industry, specialized knowledge and capabilities became increasingly dispersed in a growing ecosystem of autonomous firms. Sponsoring an open product platform then became a viable long-term strategy.

The IBM PC was the first computer platform that was open by choice and not because of financial constraints. The purpose of this chapter is to understand the IBM PC as a technical system and set of organization choices in light of the theory of how technology shapes organizations. In Chapter 7, I argued that sponsors of large technical systems (including platform systems) must manage the modular structure of the system and property rights in a way that solves four inter-related problems:

- Provide all essential functional components;
- Solve system-wide technical bottlenecks wherever they emerge;
- Control and protect one or more strategic bottleneck; and
- Prevent others from gaining control of any system-wide strategic bottleneck.

I use this framework to understand how IBM initially succeeded with the PC platform, then lost its position in the marketplace. I first describe how an ecosystem of module suppliers for very small computers emerged in the late 1970s. I then show how IBM leveraged this ecosystem to provide all essential components for the PC system while making very few components itself. Its initial strategy rested on controlling two strategic bottlenecks: (1) standards embedded in the Basic Input Output System (BIOS); and (2) system integration and manufacturing of the computer itself.

IBM's strategy of extreme openness was highly effective in competition with other firms making microcomputers. However, its control over strategic bottlenecks turned out to be weak. The BIOS proved vulnerable to reverse engineering and ceased to be unique. Loss of control of this standard allowed a new group of PC clonemakers to enter the market. When faced with this competition, IBM's capabilities as a system integrator and manufacturer were not sufficient to sustain its logistical platform as a strategic bottleneck. In the early 1990s, the PC division ceased to be profitable. It was sold to Lenovo in 2005.

15.1 The Rise of an Ecosystem

An open product platform requires an ecosystem of suppliers and complementors. In the 1970s, the modular architecture of IBM System/360 provided many opportunities for firms making plug-compatible modules to enter the industry in competition with

IBM.[1] Then, between 1976 and 1980, a new group of firms emerged that made hardware and software for so-called microcomputers. The makers of plug-compatible peripherals and microcomputer hardware and software formed the basis of the ecosystem supporting the IBM PC.

As Gordon Moore predicted in 1965 (see Chapter 12), single-user computers were made feasible by advances in semiconductor chip technology in accordance with Moore's Law. In 1971, Ted Hoff, Stanley Mazor and Federico Faggin of Intel designed the Intel 4004, which became the first commercially available microprocessor. Single-chip microprocessors were then introduced by Motorola, MOS Technology, Zilog, and others.[2] By 1975, microprocessors were available for sale for under a hundred dollars.[3]

In early 1975, a calculator company, MITS, introduced a kit for $397 that could be assembled into a primitive, switch-controlled computer, the Altair 8800.[4] MITS was immediately overwhelmed by orders. Others quickly jumped into this market, offering complementary products and services—software, memory, displays, storage devices, retail distribution, technical support, trade fairs and publications. The race was on. In the greater computer industry, although no one knew it, the "vertical-to-horizontal transition" was underway.

Apple Computer was the winner of the first round of competition in the new microcomputer marketplace. The company's first product, the Apple I, was not a commercial success. Its second, the Apple II, was introduced in April 1977 to great acclaim. By the end of that year, sales were skyrocketing and the company was profitable.[5]

Apple's growth was helped immeasurably by the introduction of a complementary software product, Visicalc. Visicalc, the first spreadsheet program, turned personal computers into an important tool for business analysis. Many users purchased Apple IIs for the sole purpose of running Visicalc. It was the first acknowledged "killer app." [6]

On the strength of its financial performance, Apple went public in December 1980 with a market capitalization of $1.8 billion.[7] It dominated the microcomputer

[1] For data on entry by plug-compatible firms in the wake of System/360, see Baldwin and Clark (2000) pp. 376-381.

[2] Allan (2001) pp. 3/6 – 3/15.

[3] *Ibid.*

[4] *Ibid.* pp. 4/9 – 4/11.

[5] *Ibid.* pp. 5/10.

[6] Downes and Mui (1998).

[7] Deffree (2016).

marketplace until IBM entered in 1981.

In the five years following the introduction of the Altair, the economic environment surrounding small computers changed considerably. In 1976, there were very few companies that were capable of producing hardware or developing software for very small computers. In contrast, by 1980, a large number of firms were making peripheral devices for small computers and a comparable number were writing applications and systems software.

Thus for its new PC system, IBM could draw upon the knowledge and production capacity of this new ecosystem of suppliers and complementors. The ecosystem provided IBM and other PC makers with what Alfred Marshall called "external economies … dependent on the general development of the industry… secured by the concentration of many small businesses of a similar character in particular localities."[8]

The existence of an ecosystem of autonomous firms making complementary products signaled that distributed supermodular complementarity (DSMC) held at this time in the market for very small computers. In the first place, the modular technical architecture of small computer systems meant that their value functions were separable. Consumers could decide how much each optional component was worth to them and purchase it or not accordingly.

Second, the rapid rate of technical change in hardware and software, propelled by Moore's Law, reduced the benefits of integration relative to modular experimentation by independent firms.[9]

Lastly, venture capitalists (VCs), who had emerged as financial intermediaries in the 1960s and 1970s, were eager to fund new startups in exchange for an equity stake in the companies. VC funding covered the startups' initial expenditures, thus aligning value with cost.

In this way, the three conditions for DSMC to hold as a dynamic equilibrium—value separability, benefits of integration less than cost, and value aligned with costs—were satisfied in the small computer ecosystem. Of course, entrepreneurs and venture capitalists at the time did not reason about distributed supermodular complementarity in a

---

[8] Marshall, A. (1890). Richard Langlois was the first to connect the development of the personal computer industry in the 1970s and 1980s to Marshall's theory of external economies. See Langlois (1992) and Langlois and Robertson (1992).

[9] Except for chipmaking and disk drive production, none of the technologies needed to make small computers was particularly complex. Chipmakers such as Intel and Motorola and disk drive makers such as Tandem and Seagate added microcomputer components to their existing product lines, thus becoming members of the emerging ecosystem. Some established companies, like Tandy, Texas Instruments, Digital Equipment Corporation, Hewlett Packard and IMSAI introduced small computer systems, but the turnover of designs was so rapid that none was able to achieve a dominant position. (Allan, 2001, pp. 4/11 – 4/21.)

systematic way. In their business plans and funding proposals, they did not say, "here is an industry where a cluster of independent companies making supermodular complements can out-compete vertically integrated firms." According to Grove, during the 1980s and early 1990s, most observers expected successful startups to "grow up" to become vertically integrated in the long run.[10]

Nevertheless, entrepreneurs and VCs could perceive the accessibility of the technology; the low costs of entry; the availability of capital to fund new ventures; and the high rewards to founders and firms that succeeded. Potential entrants could also see a large and capable network of suppliers and complementors emerging before their very eyes. Every functional component had many potential suppliers. Furthermore, in the entire sector, there was no strategic bottleneck akin to the Wright Brothers patent (see Chapter 7). The field was wide open.

Give the presence of an ecosystem, it would not be necessary for a new platform sponsor to design and manufacture every functional component inhouse. It would also not be necessary to share the platform system's value with the owner of an existing system-wide strategic bottleneck.

With the backing of IBM's chairman, a pair of IBM managers were ready to seize this opportunity.

15.2 The IBM PC — Origins

In 1980, two IBM managers—William Lowe and Don Estridge—began to see the possibilities inherent in an open product platform for small, single-user computers. They in turn had a direct mandate from the chairman of IBM, Frank Cary, who wanted the company to enter this new, high-growth market as quickly as possible. Because IBM was a late entrant to the market for small computers and because they doubted IBM's capabilities to supply small computer components competitively, Lowe and Estridge elected to make the PC open to both third-party suppliers and complementors.

On August 12, 1981, IBM announced its entry into the small computer marketplace. The IBM Personal Computer, or PC, was by design a versatile, configurable and expandable system. A basic system, using a tape cassette recorder and television set supplied by the user, was set to retail for $1,565. An expanded system, with two floppy disk drives, an IBM display and a printer, sold for $4,425.[11]

These prices were extremely competitive. As a benchmark, at the time of the announcement, IBM offered three other small computer systems, comparable to the new

[10] The first academic papers to investigate supermodular complementarity appeared in the early 1990s. See Milgrom and Roberts (1990, 1995) and Topkis (1998).

[11] IBM Archives, "Personal Computer Announced by IBM."

PC, but each costing $9,000 or more.[12] The Hewlett Packard Apogee (HP-125), introduced the day before the IBM PC, was priced around $10,000.[13]

The IBM PC was a major strategic and organizational departure for IBM. It was developed completely outside the corporate hierarchy by a small team whose director reported directly to IBM's chairman. The product went from authorization to shipment in under 14 months, an incredibly short time for an IBM system.[14]

The IBM PC was both a standards-based and a logistical platform. The managers of the PC business unit published the system's design rules and actively encouraged third parties to design complementary hardware and software that was compatible with the PC's technical standards. The PC managers also outsourced most parts, and distributed PCs through big box retailers such as Sears and ComputerLand. (Initially the IBM salesforce played only a minor role in selling PCs.) The PC thus relied on an ecosystem of external suppliers, distributors, and complementors in a way that no previous IBM product ever had.

Figure 15-1 shows IBM's Personal Computer as it was presented in the first Technical Reference Manual. The design gave users many options to configure their hardware: an extra processor slot, three memory expansion sites, five input/output slots. Not shown in the picture, but mentioned in the original press release were the software programs that were immediately available to run on the PC: EasyWriter™ for word processing; Visicalc™ for spreadsheets; accounting programs from Peachtree Software; and a game from Microsoft.

For users who purchased the floppy disk drive option, there were three "advanced disk operating sytems": PC-DOS™ from Microsoft; CP/M™ from Digital Research and the UCSD p-System™. To encourage external software developers, essentially all of the specifications explaining how the machine worked were published in a Technical Reference Manual (the "purple book") that retailed for around $36.[15]

Making only a few strategically chosen parts of a computer system within a larger ecosystem of suppliers, distributors, and complementors was a new business model for IBM. Initially it proved to be enormously successful and profitable, but within ten years, IBM lost its market leadership position in personal computers. It then watched demand for its larger systems implode because of competition from cheap, PC-compatible clones.

[12] IBM Archives "Before the Beginning: Ancestors of the IBM Personal Computer."

[13] Bricklin (1981).

[14] Chposky and Leonsis (1988); Allan (2002) pp. 9/4 – 9/9.

[15] Williams (1982).

Figure 15-1 IBM Personal Computer: A View of the System

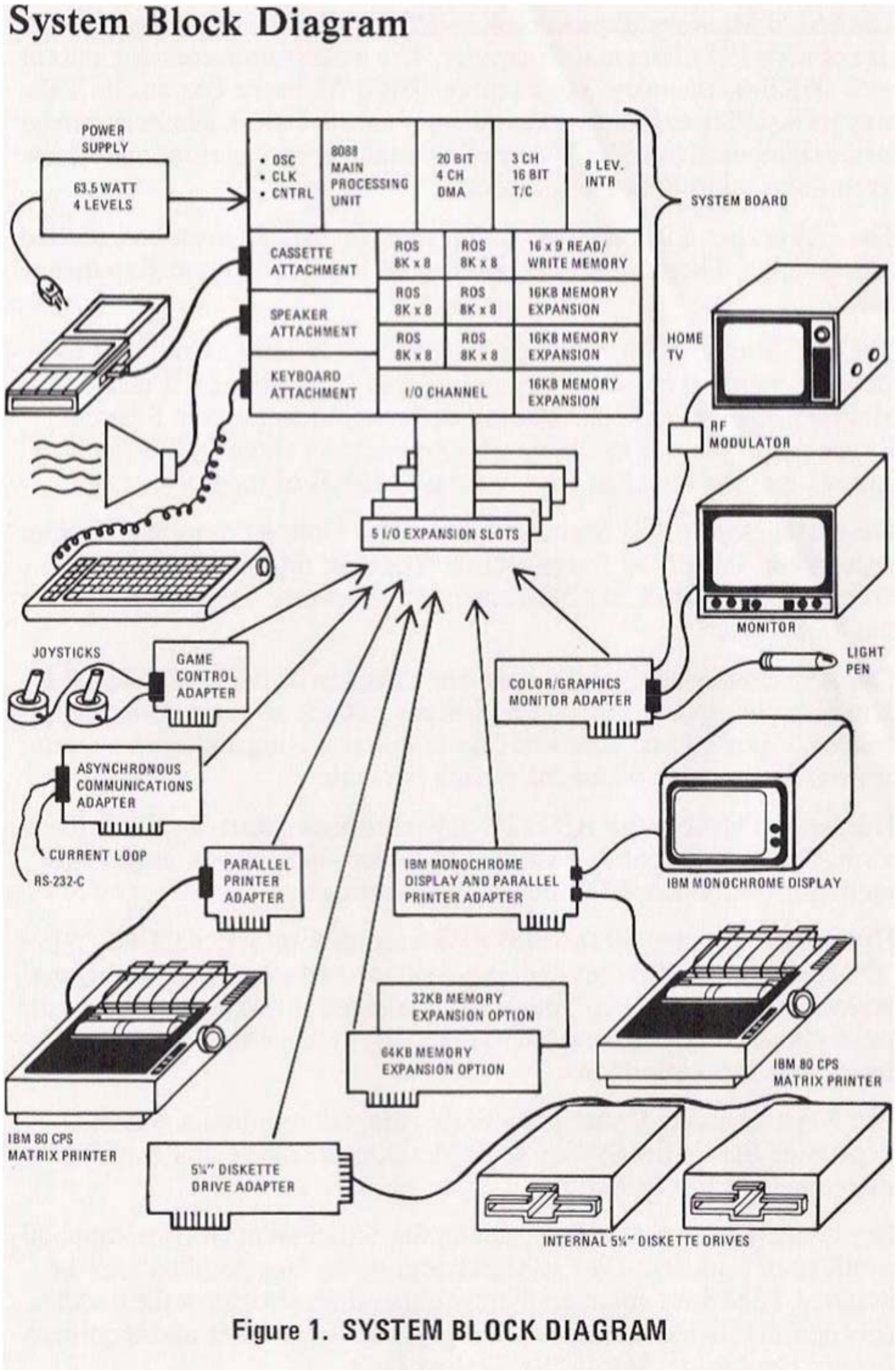


Figure 1. SYSTEM BLOCK DIAGRAM

Source: IBM Personal Computer Technical Reference Manual, First Edition, August 1981. Reprinted by permission.

15.3 The Technical Architecture of the IBM PC

The sponsor of a new technical system must set up a technical architecture and corresponding organization that integrates all essential functional components into a system that works. We can better understand IBM's open platform strategy for the PC using the functional notation developed in Chapters 6 and 7.

At the highest level, the system comprised a platform plus many hardware and software options. Users would decide which options to incorporate into their own systems. Within the constraints imposed by the two basic systems offered (cassette-plus-TV and floppy drive-plus-display) users could also attach and upgrade new hardware and software at will:

Platform □ [*Hardware Options* + *Software Options*] .

IBM supplied some hardware options such as displays and printers, but very few software options. However, in contrast to Apple, which was beginning to wall off its systems from external complementors, IBM did not attempt to prevent others firms from offering optional complements. Instead it invited them in. By definition, an *optional* complement was not *essential*, thus none could be a system-wide technical or strategic bottleneck.

The IBM platform itself was designed to be highly modular. Recall that in a von Neumann computer, the essential platform consists of six functional components: input; storage; memory; control unit; arithmetic unit; and output. By the 1970s, the control and arithmetic units had been merged onto a single chip (the microprocessor), and thus constituted a single module. The other von Neumann components were each separate modules, and different technologies were available to provide each function.

However, the definition of "essential component" had changed since von Neumann's day. The first microcomputer, the Altair, shipped with no system software. The second-generation, Apple II and its competitors in the late 1970s all shipped with an operating system. In effect, this component was becoming an essential part of the platform. IBM addressed this need by providing not one but three operating systems to operate the PC. However, for approximately six months after launch, CP/M and UCSD p-system were not available to run on the new machines.[16] Thus software developers began writing programs only for PC-DOS.[17] Almost immediately, Microsoft's operating system became the *de facto* standard.

Gary Kildall, a pioneer in designing microcomputer operating systems, split his designs into a general purpose operating system (OS) and a basic input-output system

[16] Mace (1982).

[17] Freiberger (1981).

(BIOS), which hid the details of the hardware from the operating system proper. Operating systems that followed Kildall's model thus consisted of two modules, a BIOS and an OS.[18]

A platform sponsor cannot simply supply the essential components of the platform in separate boxes. It must also integrate the components into an artifact that can perform the platform's functions. Just as patterns and stitching are required to make a garment, technical recipes and step processes are needed to bridge the gap between unassembled pieces and a functioning whole. The steps involve essential business processes, including system design, system software development, marketing, manufacturing, and distribution.

Figure 15-2 uses the functional notation of Chapters 6 and 7 to represent the technical architecture of the original PC platform:

Figure 15-2 Functional Components of the Original IBM PC Platform

Input □ *Storage* □ *Memory* □ *Microprocessor* □ *Output* □ *Operating System*

□ *Product Positioning* □ *BIOS Design* □ *Core System Design*

□ *System Integration* □ *Manufacturing*

□ *Advertising* □ *Distribution* □ *Sales*

➔ ***PC Platform***

The first line shows the components of a von Neumann computer, the basic inputs to the platform. The next three lines show the step processes required to design and produce the platform and get it into the hands of consumers.

At the risk of oversimplifying, I have split the marketing function into two parts: product positioning and advertising. Product positioning defines the concept of the product and its relationship to competitive offerings. It is the first stage in the process of product design, defining design goals such as cost, functions, speed of execution, reliability, configurability, and modifiability. Advertising refers to processes for communicating with consumers about the product and its features once it is ready to launch.

Following the convention introduced in Chapter 7, I have indicated potential modules using boxes. The diagram shows what I judge to be the maximal degree of modularity achievable in the early 1980s. In the first line, as indicated, the von Neumann

[18] Cringeley (1992) p. 58.

components and the operating system could all be separate modules.

The product's identity rested, not on these components, but on its competitive positioning and design. The BIOS was a critical part of this design: it was the key interface between hardware and software. Product positioning, BIOS Design and Core System Design involved interdependent, iterative decision-making, thus were part of a single module.

Once the product was designed, the components had to be brought together and assembled into a usable computer. System integration involves planning and synchronizing flows of physical products through a supply chain or production network. In theory, manufacturing could be contracted out, but, in the early 1980s, the interface between product design and manufacturing was not well codified. Nevertheless, the PC managers used the threat of going outside to obtain good terms from manufacturing facilities within IBM.[19]

Advertising, distribution, and sales were separable processes that could be delegated to specialized consultants, ad agencies, and retailers.

To deliver a working platform, the PC's designers had to have a plan for making *all* functional components and performing *all* required steps. However, they did not have to do everything within IBM proper. The PC managers had to decide which components and step processes to make inhouse and which to delegate to other companies. In making these decisions, they had to take account of the location of bottlenecks, technical and strategic, both present and future.

15.4 IBM's Standards-based Platform

William Lowe positioned the IBM PC as machine that could be configured in many different ways according to the user's preferences and budget. User options are the fundamental message conveyed in the system block diagram (Figure 15-1) and were emphasized in the press release introducing the PC. As the figure shows, there were numerous ports and slots in the layout of the machine, which could accommodate optional hardware devices according to the user's preferences.

The standards for designing compatible software and hardware were conveniently packaged and published in two books, which could be purchased for less than $100 at ComputerLand.[20] The IBM PC was thus an open standards-based platform. For the PC managers, the advantage of this strategy was that members of the existing ecosystem,

[19] Chposky and Leonsis (1988) pp. 86-93.

[20] Dan Bricklin (the designer of Visicalc, the first spreadsheet program) quotes Ray Ozzie (the architect of LotusNotes™ and Groove™) as follows: "All you needed was the PC, your external disk...and the two books, the DOS technical reference and that purple PC technical reference, and it was a sense of freedom...as a programmer." Bricklin (2001).

who had experience in designing hardware and software for microcomputers, could provide PC users with more options and better options more quickly than IBM would be able to supply on its own. The availability of many modular options in turn would allow users to configure and upgrade their systems flexibly and economically, *thus increasing demand for the platform*.

This positive feedback loop is characteristic of all platform systems. As shown in Chapter 13, in platform systems, users and options are supermodular complements: more users increase the expected revenue to each option, and more options increase the value of the platform to users.

However, these "indirect network effects" were not well understood in 1981 when the IBM PC was introduced. The first academic papers to describe indirect network effects were Michael Cusumano, Yiorgis Myolandis and Richard Rosenbloom's study of Sony's and JVC's video consoles, and Jeffrey Church and Neil Gandal's seminal papers on network externalities in software.[21] These works followed the launch of the IBM PC by more than a decade. Hal Varian and Carl Shapiro explained network effects and the value of standards to practitioners in *Information Rules: A Strategic Guide to the Network Economy*, published in 1999. Only then did network effects and standards become common parlance among managers.

Before these concepts were sharpened, scholars and managers understood that open product platforms could create more value for users than closed platforms and that feedback loops could accelerate growth. But a crucial question remained: how could the sponsor of an open platform capture enough of the value created to make its investment worthwhile?[22] In the wake of IBM's experience with plug-compatible entrants to System/360, creating a modular system, let alone an open modular system, was akin to playing with fire.[23]

However, as we saw in Chapter 7, the answer to the question "how does a sponsor profit from an open system?" depends on the sponsor's handling of system-wide technical and strategic bottlenecks. If the sponsor can convert a system-wide technical bottleneck into a strategic bottleneck,[24] then, by the threat of exclusion, it can claim a percentage of the value created by the system as a whole. The Wright brothers managed to convert their method of controlling aircraft into a strategic bottleneck via a patent. Taylor and White's patent on the method of producing high-speed steel was overturned,

---

[21] Cusumano et al. (1992); Church and Gandal (1992;1993);

[22] Debates about the virtues of open vs. closed product platforms continue today. See, for example, Parker and Van Alstyne (2018); Parker, Van Alstyne and Jiang (2017).

[23] As a colleague in strategy said to me at the time, "Why would you want to do that?"

[24] Solving a technical bottleneck is not the only route to control of a strategic bottleneck. Some strategic bottlenecks are derived from location (e.g., control of a bridge or a mountain pass); others are based on consumer perceptions and loyalty (e.g., a brand).

ending Bethlehem Steel's hopes of a strategic bottleneck in the machine tool industry. There was no system-wide strategic bottleneck in container shipping, thus most of the value created by that system flowed to users.

Standards defining what needs to be done to achieve interoperability among modules are an essential part of any modular system. If there is only one way to achieve compatibility, then the standards are by definition unique. And if a for-profit firm controls access to the standards, they are the basis of a strategic bottleneck.

As a computer system, the IBM PC had three sets of system-wide standards that designers of compatible hardware and software had to obey. First, the microprocessor instruction set mapped machine commands to specific circuits paths in the microprocessor. Second, the hardware-software interface specified by the BIOS and buses defined interactions between hardware and software. Lastly, users and software developers were increasingly dependent on operating system instructions (these later came to be called "application programmer interfaces" or APIs).

The PC managers in Boca Raton were fully aware of the status of the microprocessor instruction set and the BIOS as possible strategic bottlenecks in the open system they were creating. In setting up their contract with Intel, the supplier of the microprocessor, they required Intel to share its product and process designs with twelve other suppliers. They also chose for IBM to directly own the BIOS instructions and protect them via copyright.

The PC managers were less aware of the strategic potential of the operating system. They contracted with Microsoft for rights to use its disk operating system (DOS) in perpetuity.[25] But they did not anticipate the challenge of obtaining rights to a graphical user interface (Windows) built on top of DOS. (Intel's and Microsoft's dealings with IBM in the late 1980s and early 1990s are discussed below and in the next chapter.)

15.5 IBM's Logistical Platform

Standards were not IBM's only platform in the PC architecture. The managers in Boca Raton also created a logistical platform to manage suppliers, system integration, and the manufacturing of the computer itself.

For the managers charged with bringing the IBM PC project to fruition, the existence of a vibrant and growing microcomputer ecosystem meant that they could use pre-existing components and capacity to make the first PCs. In July of 1980, William Lowe promised IBM's Management Committee that the new machine would be ready to launch in one year. This deadline could not be met without using components that already existed. IBM, having ignored the growth of the microcomputer market, made virtually

[25] Microsoft' operating system was variously known as PC-DOS and MS-DOS. They were essentially the same codebase.

none of these components inhouse.

Another reason to favor external suppliers was to avoid corporate infighting. By the late 1970s, IBM had evolved into a set of powerful fiefdoms.[26] Many divisional managers saw the PC as a major threat to their existing products and profits. For example, the Office Products Division worried that the PC might replace their lucrative line of Selectric typewriters. (It did.) Managers in the mainframe divisions saw that the price-performance of microprocessors was so much better than mainframes that IBM's entire revenue stream was at risk. Thus for many of IBM's senior divisional managers, personal computers were not an opportunity to be welcomed but a threat to be avoided. In their view, IBM should not be accelerating the growth of the PC market, but postponing it for as long as possible.[27]

For both of these reasons, the managers of the PC project strongly favored external over internal suppliers. They did select some IBM plants to manufacture different parts of the platform, but, in a departure from IBM's longstanding practice, the internal vendors had to bid competitively against outside firms and sign fixed-price agreements.[28]

IBM thus depended on what came to be called a "modular production network" to manufacture the first PCs.[29] Any manufacturing system requires system integration to bring necessary inputs and activities together in a synchronized fashion. System integration of a modular production network means, first, contracting with suppliers; then scheduling deliveries; combining the parts according to a specific manufacturing design; and transporting finisihed goods to distributors and end users. System integration requires simultaneous management of step processes involving production and transportation and options such as sourcing, pricing, configuration, scheduling, and promotions.

From inception, the IBM PC was on the tightest of schedules. To get the job done (thus fulfilling their promise to Frank Cary and the Management Committee), the PC managers elected to manage system integration themselves and delegate manufacturing of the computer to a single IBM assembly plant.[30] According to IBM's official history, "the manufacturing strategy was to simplify everything, devise a sound plan and not deviate." [31]

[26] Chposky and Leonsis (1988); Ferguson and Morris (1993).

[27] Chposky and Leonsis (1988).

[28] *Ibid.* pp. 86-93. These internal contracts were not enforceable in a court of law, but were respected within IBM.

[29] Sturgeon (2002).

[30] Other plants made printers and displays.

[31] IBM Archives, "The Birth of the IBM PC."

System integrators determine which components are included in the system thus have bargaining power vis a vis suppliers. A single manufacturing facility can achieve scheduling efficiencies and maximum economies of scale. At the time of the PC's creation, leverage over suppliers and economies of scale in manufacturing were believed to be a strategic bottleneck on a par with control over the BIOS standard. In the end, however, this belief proved ill-founded.[32]

15.6 Protecting an Open Product Platform through Control of Strategic Bottlenecks

When a platform sponsor embraces openness, it must have a strategy for capturing value while directly controlling only part of the overall system. In general, a firm can realize super-normal profits at the platform level if (and only if) it supplies a *unique and essential* component in the overall platform system. In Chapter 7, I equated control of a unique and essential component to possession of a *strategic bottleneck*.

The most important source of strategic bottlenecks in computer systems is the visible information that users and designers need to know to operate the machine and design compatible components. As indicated above, in the original IBM PC, the components containing important visible information were (1) the microprocessor, which converted code to circuit paths; (2) the BIOS and buses, which allowed software to give instructions to specific hardware devices; and (3) the operating system, which provided user-friendly ways of interacting with hardware and other software.

To increase the speed of execution, many early programs, such as Lotus 1-2-3, were written in Intel's 8088 assembly language, which mapped directly to the microprocessor instruction set.[33] Professional programmers could thus bypass the operating system and even the BIOS if they chose to do so. But many found it convenient to incorporate BIOS and DOS instructions into their code, thus avoiding tedious recoding of common functions. At the same time, millions of ordinary users used DOS commands as their principal way of interacting with their machines.

In this fashion, the specific instructions used to command the microprocessor, the BIOS and the operating system became embedded in both hardware and software. Programs and files written with these instructions would only operate correctly on machines that recognized the instructions. In the language of Chapter 5, strong one-way complementarity developed between the instruction sets and essentially all hardware and software for the IBM PC. Because of this complementarity, purchasers of new platforms would pay more for machines that used these instruction sets than those that did not.

The microprocessor instruction set was owned by Intel. At the time, chip designs

---

[32] Ferguson and Morris (1993).

[33] Cringeley (1992) pp. 154-157.

were not protected intellectual property, although production methods were considered trade secrets. Intel supplied the chips to IBM on a non-exclusive basis—they were not prohibited from selling the chip to other parties. At IBM's behest, they also entered into second-sourcing agreements with twelve other semiconductor companies.[34] Thus at the outset, the instruction set and chip design were not exclusively controlled by Intel. However in 1984, the Semiconductor Protection Act gave chip designers exclusive rights to their specific circuit layouts for ten years.[35] This fact influenced Intel's later actions.

The BIOS was owned by IBM and protected by copyright. The buses were hardware, thus not subject to copyright. They were also not sufficiently original to be patentable. Hence the buses were not protected. We will revisit the issue of buses in the next chapter.

DOS was owned by Microsoft, licensed to IBM (again not exclusively) and protected by copyright, complexity, and secrecy of the source code. As a general rule, Microsoft did not allow outsiders, including IBM, to view its source code. It is not even clear who within Microsoft was allowed to see the entire codebase. (Dividing up the right-to-know is a classic way of protecting valuable intellectual property.[36])

Notably IBM owned outright only one of the three critical instruction sets, although its customers and complementors depended on all three. Thus IBM was theoretically vulnerable to holdup by its two main suppliers. If Intel or Microsoft could deny IBM access to the instructions they owned, they could bargain for a share of the surplus value created by the PC ecosystem as a whole.

IBM initially protected itself from holdup through its power as a buyer to control what went into a PC. Using this power, it demanded extensive second-source arrangements from Intel as well as the right to manufacture the 8088 and successor chips. From Microsoft it obtained the right to license DOS and any successor programs but did not ask for the right to view or modify Microsoft's source code.[37] It did not require exclusive purchasing arrangements with its two key suppliers—such actions might have raised serious antitrust concerns.

Finally, IBM protected itself from holdup by controlling the BIOS. Another computer system—a potential competitor—could use an Intel processor and Microsoft DOS, but it would not be fully compatible with IBM machines unless it also used BIOS

---

[34] Yoffie et al. (1989); Froot (1993).

[35] See Graham (1999) pp. 12-13 for a summary of the Act. Brooktree Corporation v. Advanced Micro Devices, Inc., 977 F.2d 1555 (Fed. Cir. 1993) provides case law.

[36] Henkel, Baldwin and Shih (2013); Baldwin and Henkel (2015).

[37] Ferguson and Morris (1993) pp. 70-71.

commands to connect to hardware devices and load the operating system.[38]

Figure 15-3 shows the IBM PC's initial technical architecture including the bottlenecks:

Figure 15-3 Functional Components of the Original IBM PC Platform Indicating Technical and Strategic Bottlenecks

Input □ *Storage* □ *Memory* □ *Microprocessor** □ *Output* □ *Operating System**

□ *Product Positioning* □ *BIOS Design**IBM □ *Core System Design*

□ *System Integration* □ *Manufacturing*

□ *Advertising* □ *Distribution* □ *Sales*

➔ ***PC Platform***

When the PC was ready to ship, no essential component was lacking. There were no remaining system-wide technical bottlenecks. (In contrast, many technical bottlenecks remained in the optional components. Most importantly, two of the three operating systems, CP/M and UCSD's p-system, were not yet available.)

The microprocessor and operating system were unique. Following the conventions laid out in Chapter 7, I have placed asterisks on those components. However, IBM's contracts with Intel and Microsoft meant that these companies could not prevent IBM from using their designs. Intel was obligated to share its chip layout and manufacturing methods with a number of other suppliers, and IBM had the right to use DOS on any of its machines. Although these components both essential and unique, their owners could not prevent IBM from using them. They were not, *at this time*, strategic bottlenecks.

In contrast, the BIOS was unique, and controlled by IBM. Thus it qualified as a strategic bottleneck—the only strategic bottleneck in the entire technical architecture. (Some might claim that IBM had unique capabililties in system integration and manufacturing. However, as evidenced by the rapid entry of numerous "almost compatible" competitors, many companies had the ability to manufacture comparable machines. The trick was not to make the machine but to make a *truly compatible* machine.[39]

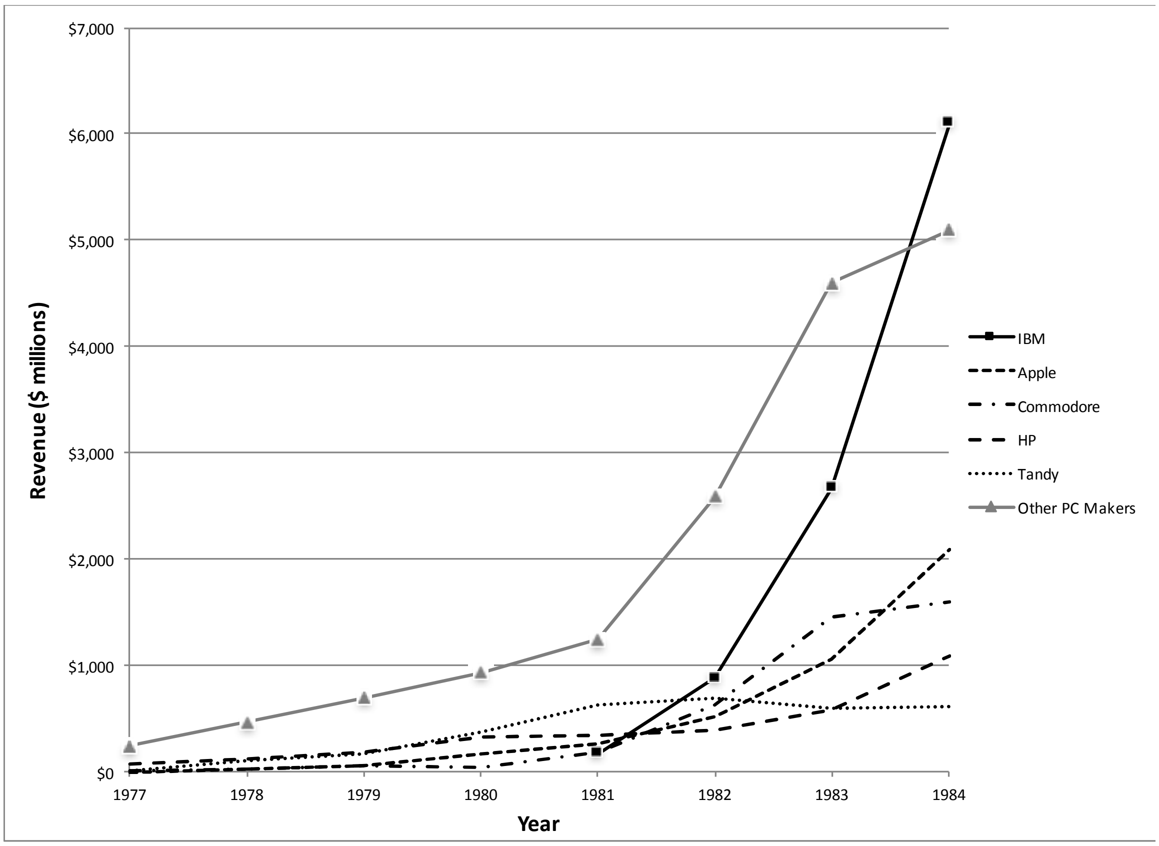
[38] Montague et al. (1983).

[39] Montague et al. (1983); Ward (1983).

15.7 The BIOS Protects IBM Position

The initial PC introduction was blessed with extraordinary good luck. The positioning of the machine as a system with many options was perfectly in accord with what users wanted. Sales took off. Suppliers worked overtime and complementors appeared in droves. Figure 15-4 shows revenues for all manufacturers of small computer systems from 1977 through 1984. Sales of almost all personal computers accelerated in 1981, but no company grew as fast as IBM.

Figure 15-4 Personal Computer Revenues 1977-1984



Source: IDC, WW Quarterly Personal Computer Device Tracker, 2016Q4 Historical Release, Publication Date: February 10, 2017.

A number of companies, including Tandy, Texas Instruments, Olivetti, Wang and Zenith responded to the IBM PC's success by introducing machines that purported to be compatible with the IBM machines. Microsoft and Intel, as was their right, sold DOS and microprocessors to these "almost compatible" competitors. But there are degrees of compatibility and none of the rivals worked seamlessly with IBM's machines.[40]

Advanced applications, including spreadsheets, word processors, and any program involving graphics needed to write instructions to the hardware directly in order to achieve acceptable execution times. This in turn meant using both Intel's 8088 assembly language *and* the IBM BIOS instructions. For example, the spreadsheet program Lotus 1-2-3 was written specifically for the IBM PC using both instruction sets. Because its code was "native," 1-2-3 was much faster than any other spreadsheet program on the market. In contrast, Visicalc, the original spreadsheet program, was originally written for the Apple II and then "ported" to the PC. Its performance was very slow as a result. 1-2-3 quickly became the market leader in the all-important spreadsheet category.[41]

IBM's position as the owner of the visible information contained in the BIOS was fortified by an important legal decision in 1983 in the case Apple v. Franklin. In this case, the appeals court ruled that a computer program expressed as object code, that is, in machine-readable format, is subject to copyright protection.[42] After the decision, IBM began enforcing its copyright on the PC BIOS. Following threats of legal action, a number of PC clonemakers were forced to cease production.[43]

As long as IBM controlled the BIOS, IBM and its suppliers were in symmetric positions. IBM needed the Intel instruction set and the Microsoft operating system to sell its platforms, but Intel and Microsoft needed the BIOS to to be "IBM-compatible." However, in setting up this symbiotic relationship with Intel and Microsoft, IBM's managers did not anticipate the effects of a new technical and legal strategem known as "clean room reverse engineering."

15.8 Reverse Engineering and IBM's Loss of Advantage

The legally acceptable path to a fully compatible BIOS lay in an arduous process known as "clean room reverse engineering." The method requires two teams of software developers. One team must be made up solely of individuals who can prove they have never seen the target codebase. The other team studies the code both as written and in operation and writes a complete description of what happens in response to each command. The first team reads this description, then writes a program that behaves in *exactly* the same way. The result is a new program that provides identical functionality, but by definition is not a copy.

---

[40] Ward (1983).

[41] Cringeley (1992) pp. 150-158.

[42] Apple Computer, Inc. v. Franklin Computer Corporation, U.S. Court of Appeals Third Circuit; August 30, 1983; 714 F.2d 1240.

[43] Caruso (1984); Sanger, David E. (1984).

The cost of clean room reverse engineering increases exponentially with the size and complexity of the software system. The PC BIOS was susceptible to this technique because, following Kildall's design, it was small and separate from the rest of the operating system. A less modular system with a larger BIOS might have been immune to the threat. Indeed, Scott Mueller, an expert on PC architecture, has noted:

> [The Macintosh] BIOS is very large and complex and is essentially part of the OS, unlike the much simpler and more easily duplicated BIOS found on PCs. The greater complexity and integration has allowed both the Mac BIOS and OS to escape any clean-room duplication efforts.[44]

Compaq Computer was the first company to successfully reproduce the IBM BIOS via clean room procedures. It reportedly spent $ 1 million and the better part of a year on the project.[45] As a result of this investment, Compaq computers achieved the highest degree of compatibility with the IBM PC. According to Ronnie Ward writing in *BYTE* magazine in 1983:

> … these computers [can] run the top-selling software for the IBM PC … use add-on boards designed for the PC and read and write IBM PC disks. They [also] provide the same user interface … .[46]

Compaq's first product was an immediate success, selling 53,000 units and earning revenues of $169 million in its first year (1983). That same year, IBM had revenues of $2.7 billion on 670,000 units while Apple had revenues of $1.1 billion on 640,000 units.[47]

Compaq used its reverse-engineered BIOS for its own production. However, in 1984, Phoenix Technologies came out with an IBM-compatible BIOS chip design that was also obtained via clean room methods.[48] Phoenix began selling its reverse-engineered BIOS in May 1984, and thus the PC clone industry was born.

With the entrance of many low-cost, fully compatible clones, IBM's revenue leveled off and its market share began to fall. Figure 15-5 shows estimated market shares for IBM, Apple and clone-makers from 1981 through 1994. IBM's market share peaked at 38% in 1985 and then fell dramatically. By 1994, its market share had fallen to 13%

---

[44] Mueller, 2003, p. 28.

[45] Cringeley (1992) p. 171.

[46] Ward (1983).

[47] IDC, WW Quarterly Personal Computer Device Tracker, 2016Q4 Historical Release, Publication Date: February 10, 2017.

[48] Langdell (1984).

just below Apple's at 14%.

Figure 15-5 Market Share of IBM PC vs. Apple and Clones 1981-1994

Source: Author's calculations based on IDC, WW Quarterly Personal Computer Device Tracker, 2016Q4 Historical Release, Publication Date: February 10, 2017.

In contrast, from 1984 to 1994, the clone market grew dramatically. Compaq was joined by other large sellers, Dell, Gateway and AST. Then, in 1987, Hewlett Packard switched from using their own proprietary microprocessors to making IBM-compatible machines. Following the introduction of the Phoenix BIOS in May 1984, hundreds of smaller firms also began to offer IBM-compatible PCs. Usually they purchased "white boxes," that is, unbranded machines, from Taiwanese manufacturers like Asus and sold them through different distribution channels at very low prices.

As the figure shows, the white box companies were initially IBM's fiercest competitors. However, by the early 1990s, the five major clonemakers had a slightly larger market share than all other clones (36% vs. 33%). Still, the market for IBM-compatible PCs remained highly fragmented. In 1994, Compaq was the market leader with an 18% share; IBM second at 13%; Dell had 6%; Gateway, AST and HP had 4% each.

When its copyrighted BIOS ceased to be unique, IBM could no longer control the rate of improvement and turnover in PC designs. Evidence of IBM's loss of control appeared as early as 1986. In that year, Intel introduced its 32-bit 80386 microprocessor, priced at $150 compared to $40 for the 80286. In a departure from previous practice, Intel insisted on becoming the sole source for the new processor generation.[49]

Preferring the lower cost chip and perhaps fearing cannibalization of its minicomputer products, IBM elected to stay with the older generation of chips. Seizing the opportunity, Compaq adopted the new microprocessor and promoted it heavily. Compaq's machines were an instant market success.[50] (In that year, profits on microprocessors in part offset Intel's heavy losses on memory chips.)

At the same time, in a complex and stealthy series of moves, Microsoft was in the process of severing its relationship with IBM. IBM's managers were backing a ground-up rewrite of the PC operating system, a codebase known as OS/2. OS/2 was designed to have several advanced features such as multi-tasking, but it was initially tailored the the Intel 16-bit 80286 chip, which (in 1986) was already being superseded by the more powerful 32-bit 80386 chip. (The initial OS/2 codebase was reportedly written in 80286 assembler language.[51] If true, this was a technical error of major proportions. At the time, Microsoft, DEC and other companies making operating systems were beginning to write code in the high-level, portable language, C.[52] As Moore's Law drove the cost of computation down, assembler code, which had been the key to fast execution in early PCs, was being replaced by high-level languages and optimizing compilers.[53])

In contrast to the ground up approach in OS/2, Gates and Microsoft developers conceived of Windows™ as a graphical user interface (GUI) running on top of DOS. Thus, from the beginning, OS/2 and Windows were incompatible. Each was a candidate to become the standard for the next generation of personal computers, with all the advantages that position entailed. After several years of tense relations, in 1991, IBM and Microsoft ceased the pretense of joint development and split up, with IBM taking OS/2 and Microsoft taking Windows.

In the subsequent battle to become the next-generation operating system for PCs, Windows won decisively.[54] IBM thus lost its chance to secure this strategic bottleneck. In the next chapter, we shall see that IBM also tried but failed to turn the next generation of

---

[49] Yoffie et al. (1989); Froot, K.A., (1992); Casadesus-Masanell, R., D. Yoffie, and S. Mattu (2010).

[50] Chposky and Leonsis (1988) p. 211; Ferguson and Morris (1992) pp. 55-59.

[51] Ferguson and Morris (1993) p. 76.

[52] Zachary (1994).

[53] Hennessy and Patterson (1990).

[54] Carroll (1993).

buses into a strategic bottleneck.)

Lacking control of a strategic bottleneck, IBM was forced to compete with the clones on the basis of price and efficiency. It was no contest. In the words of Charles Ferguson and Charles Morris, once the company lost control of the standards:

> IBM was just another clonemaker, but the one with the most pretensions, the biggest overhead, the highest prices, and a rapidly falling market share.[55]

During the recession of 1990-1991, cheap, networked IBM-compatible PCs began to place intense pressure on IBM's traditional mainframe pricing models. By 1991, all of IBM was losing money. Before writeoffs, IBM's computer business made only $942 million, down from $11 billion the year before.[56]

We do not know what the profits from the PC business were at this time: IBM did not break them out in its financial statements. However, we do know that during the 1990s, the IBM's share of the PC market dropped from 23% to 7%.[57] It could not hold its own in head-to-head competition with the clones. The PC division was sold to the Chinese firm, Lenovo, in 2005. In the three-and-a-half years preceding the sale, the division lost almost $ 1 billion.[58]

With IBM no longer a key player, advantage shifted to the firms controlling the other essential standards—Intel and Microsoft. IBM-compatible clones needed to provide users and developers with access to the microprocessor instructions and operating system APIs. Thus, through most of the 1980s and 1990s, virtually every manufacturer of an IBM-compatible personal computer was a customer of both Intel and Microsoft.

Figure 15-6 shows the technical architecture and strategic bottlenecks of the PC platform after (1) Phoenix reverse engineered the PC BIOS (1984); (2) Intel sole sourced the 80386 chip (1986); and (3) Microsoft introduced Windows 3.0, its first successful graphical operating system (1990).

---

[55] Ferguson and Morris (1992) p. 59.

[56] Moffat (1992).

[57] Author's calculations based on IDC, WW Quarterly Personal Computer Device Tracker, 2016Q4 Historical Release, Publication Date: February 10, 2017.

[58] Bloomberg News (2004).

Figure 15-6 Functional Components of the IBM PC Platform Indicating Strategic Bottlenecks after 1990

Input □ *Storage* □ *Memory* □ *Microprocessor*[*Intel] □ *Output* □ *OperatingSystem*[*MSFT]

□ *Product Positioning* □ *BIOS Design* □ *Core System Design*

□ *System Integration* □ *Manufacturing*

□ *Advertising* □ *Distribution* □ *Sales*

➔ ***PC Platform***

After the IBM BIOS was reverse engineered, the BIOS was no longer unique, thus it could not serve as the basis for a strategic bottleneck. In contrast, after Intel went sole source on its chip designs, the 80386 and subsequent Intel microprocessors *were* unique, hence became a strategic bottleneck controlled by Intel. Similarly, once the success of Windows was assured, Microsoft controlled a strategic bottleneck in next-generation operating system.

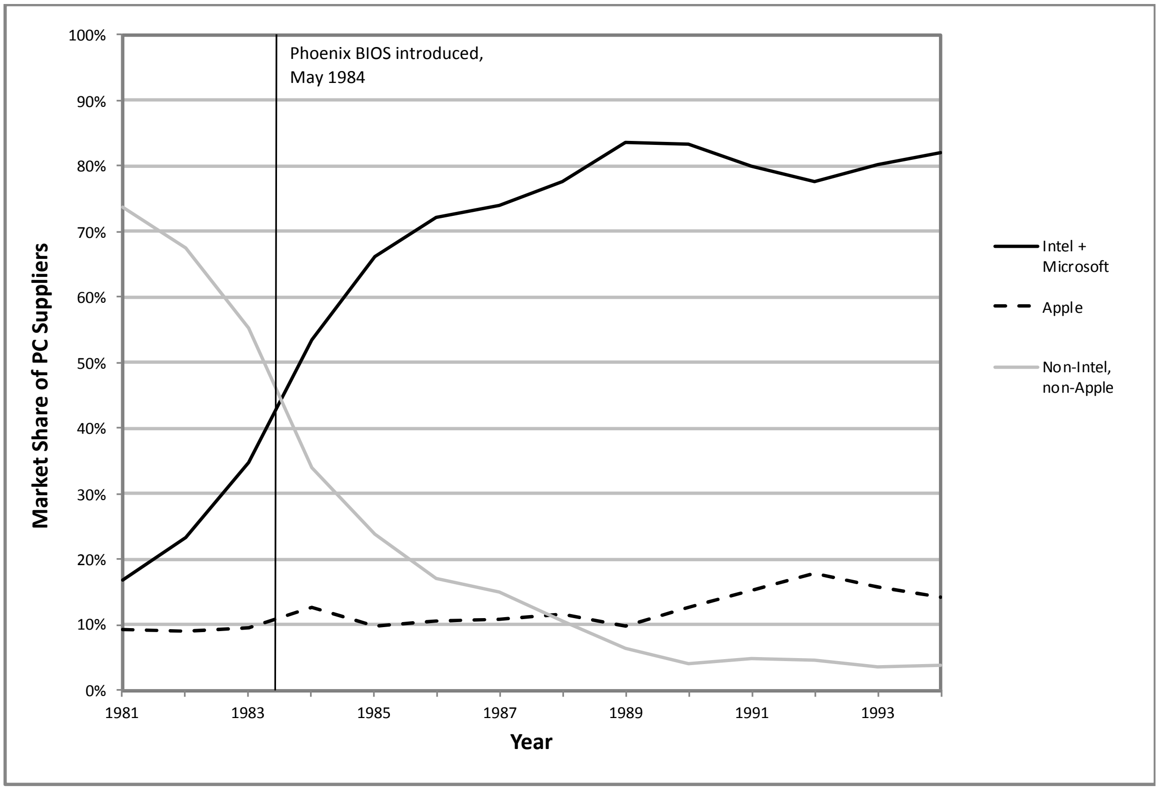
The two companies thus became joint sponsors of the so-called "Wintel" platform. We will look at how the companies managed technical and strategic bottlenecks and their overlapping ecosystems in the next two chapters. Before concluding this chapter, however, it is useful to look at how the Intel-Microsoft standards-based platform fared against the competition.

15.9 The PC Ecosystem vs. the Rest

Figure 15-7 shows the market share of the Intel-Microsoft ecosystem vs. that of Apple and other non-Intel computers from 1981 through 1994. Following the introduction of the IBM PC in 1981, the Intel-Microsoft share rose as IBM's share grew and other PC makers sought to sell "almost compatible" machines. With the introduction of the Phoenix BIOS in 1984, "almost compatible" became "actually compatible." Over the next five years, Intel-Microsoft's share of the total market rose to 80%. This growth was entirely due to clones as IBM's market share declined during this time period (see Figure 15-5).

At the same time, Apple's market share remained nearly flat while the share of other non-Intel computers dropped precipitously. While non-Intel, non-Apple computers accounted for over 70% of the market in 1981, by 1994, their share was under 5%.

Figure 15-7 Market Share of Intel and Microsoft vs. Apple and Others 1981-1994



Source: Author's calculations based on IDC, WW Quarterly Personal Computer Device Tracker, 2016Q4 Historical Release, Publication Date: February 10, 2017.

The figure contains two lessons. First is the importance of compatibility and the power of a dominant standard. Fully 80% of the market elected to be compatible with the IBM PCs. Only the Apple Macintosh, which found a niche in graphically intensive desktop publishing, was able to sustain an incompatible standard. (In the early 1990s, Apple introduced Apple File Exchange and PC Exchange, which allowed Macintosh computers to read PC-formatted disks and files. These programs significantly decreased the cost of incompatibility for Macintosh users.)[59]

The second lesson concerns the advantages of an open product platform vs. a closed platform. In the mid-1990s, the Apple Macintosh product platform was more closed than the PC platform. Unlike IBM, Apple did not encourage third-parties to develop peripheral devices. Much of the software that ran on Macintoshes was also created by Apple. Finally, except for a brief period in the 1990s, Apple did not license its operating system to clonemakers.

[59] Johnson, H. (undated).

The PC, in contrast, was open to third-party hardware and software from the very beginning. Hence the chart can be read as evidence of the constrasting evolution of a highly modular, open product platform in competition with a more integrated, closed product platform. In this encounter, the open platform won decisively. However, in the general debate on the virtues of open vs. closed systems, this is only one data point.

It is sometimes claimed that fierce price competition among clonemakers drove down the price of PCs relative to Macintosh computers. This claim is not supported by the data on average selling prices (ASPs). Between 1984 and 1994, ASPs for Intel-Microsoft and Apple computers remained close together. A $300 gap in ASPs appeared in 1996 and increased to $500 in 2000.[60] But, as the figure shows, by the early 1990s, the dominance of the PC platform was already established.[61]

15.10 Conclusion—How Technology Shapes Organizations

It is safe to say that, in the early 1980s, when IBM rode the wave of the PC and then lost its place, very little was known about how to manage or profit from an open product platform. This was uncharted territory. According to Grove, a mature, successful company:

> would have its own semiconductor chip implementation, build its own computer around these chips according to its own design and in its own factories, develop its own operating system software … and market its own applications software…. This combination … would then be sold as a package by the company's own salespeople. This is what we mean by a vertical alignment.[62]

The main advantage of vertical alignment was seamless technical integration with all parts efficiently working together. The disadvantages arose from the fact that there were no options outside the vertical stack. This is precisely the tradeoff modeled in Chapter 13. As the model shows, as the external rate of technical change increases, the benefits of modular flexibility come to outweigh the benefits of higher efficiency obtained through integrated designs and production processes.

By 1980, virtually all new computers were designed as modular systems. This was a reasonable strategic response to the high rate of technical change imposed on computer companies by the dynamics of Moore's Law. However, there was implicit tension between the traditional business model of vertical integration and the new technical architecture characterized by easily swappable modules and low transaction

[60] Author's calculations based on IDC, WW Quarterly Personal Computer Device Tracker, 2016Q4 Historical Release, Publication Date: February 10, 2017.

[61] The larger PC platform offered more low-cost options than the Macintosh. This may have led to a perception among consumers that Macintosh computers cost more.

[62] Grove (1996) pp. 40-41.

costs. It was only a matter of time before firms in the computer industry began to experiment with *open* platform strategies. IBM's managers in Boca Raton ran one of the first of these experiments.

The IBM PC succeeded against Apple and other early entrants in establishing an open product platform and ecosystem as the dominant technical and organizational architecture for personal computers. However, the company failed to maintain control of a strategic bottleneck, thus, in the long run, IBM was unable to capture significant value from the platform system it created.

By revising the terms of their contracts with IBM, Intel and Microsoft were able to gain control of the unique and essential standards embedded in the processor instruction set and the operating system APIs. After the fall of IBM, they became joint sponsors of the Wintel platform.

In the next three chapters, I will discuss the experiences of Intel, Microsoft, and Dell Computer in the 1990s and early 2000s. I have selected these firms because all three were successful in capturing value as platform sponsors. Intel and Microsoft were sponsors of standards-based platforms, while Dell was the sponsor of a logistical platform. Standards-based and logistical platforms are in fact different forms of organization. Each type of platform poses different opportunities and challenges for the platform sponsor, thus it is useful to study each in isolation. That is my goal in the following chapters.

References

Adner, R. (2017) “Ecosystem as structure: an actionable construct for strategy,” *Journal of Management*, 43(1), pp.39-58.

Adner, R. and Kapoor, R. (2010) “Value Creation in Investment Ecosystems: How the Structure of Technological Interdependence Affects Firm Performance in New Technology Generations,” *Strategic Management Journal* 31:306-333.

Allan, R.A. (2001) *A History of the Personal Computer: The People and the Technology*; Allan Publishing, London, Ontario, Canada.

Apple Computer, Inc. v. Franklin Computer Corporation, U.S. Court of Appeals Third Circuit; August 30, 1983; 714 F.2d 1240. http://digital-law-online.info/cases/219PQ113.htm (accessed January 17, 2018).

Baldwin, C.Y. and Henkel, J., 2015. Modularity and intellectual property protection. *Strategic Management Journal*, *36*(11), pp.1637-1655.

Baldwin, Carliss Y. and Kim B. Clark (2000). *Design Rules, Volume 1, The Power of Modularity,* Cambridge, MA: MIT Press.

Bloomberg News (2004) "I.B.M. Division Headed to China Has Made No Profit in 3 1/2 Years," https://www.nytimes.com/2004/12/31/technology/ibm-division-headed-to-china-has-made-no-profit-in-3-12-years.html (viewed 1/18/19).

Bricklin, D. (1981) IBM PC Announcement 1981, Transcript of Software Arts staff meeting, August 12, 1981, http://www.bricklin.com/ibmpcannouncement1981.htm (viewed 3/3/17).

Bricklin, D. (2001) "The IBM PC BIOS listing," http://danbricklin.com/log/2001_08_21.htm (viewed 3/3/17).

Brooktree Corporation v. Advanced Micro Devices, Inc., 977 F.2d 1555 (Fed. Cir. 1993)

Carroll, P. (1993) *Big Blues: The Unmaking of IBM*, Crown Publishers, New York.

Caruso, Denise (1984), "IBM Wins Disputes Over PC Copyrights", InfoWorld, p. 15; https://books.google.com/books?id=gy4EAAAAMBAJ; (viewed 1/17/18).

Casadesus-Masanell, R., Yoffie, D. and Mattu S. (2010) “Intel Corporation: 1968-2003,” Harvard Business School Publishing.

Chposky, J. and T. Leonsis (1988) *Blue Magic: The People, Power and Politics Behind the IBM Personal Computer*; Facts on File Publications, New York, NY.

Church, J., and Gandal, N. (1992). Network effects, software provision, and standardization. *The Journal of Industrial Economics*, 85-103.

Church, J., and Gandal, N. (1993). Complementary network externalities and technological adoption. *International Journal of Industrial Organization, 11*(2), 239-260.

Cringely, Robert X. (1992) *Accidental Empires*, Reading, MA: Addison-Wesley.

Cusumano, M.A., Mylonadis, Y. and Rosenbloom, R.S., (1992) “Strategic maneuvering and mass-market dynamics: The triumph of VHS over Beta.” *Business history review*, *66*(1), pp.51-94.

Deffree, Suzanne (2016) “Apple IPO makes instant millionaires, December 12, 1980” http://www.edn.com/electronics-blogs/edn-moments/4403276/Apple-IPO-makes-instant-millionaires--December-12--1980 (viewed 3/3/17).

Downes, L. and Mui, C. (1998) *Unleashing the killer app: digital strategies for market*. Harvard Business School Press, Cambridge, MA.

Ferguson, Charles H. and Charles R. Morris (1993) *Computer Wars: How the West Can Win in a Post-IBM World*, New York, NY: Times Books.

Freiberger, P. (1981) “Some Confusion at the Heart of IBM Microcomputer,” *InfoWorld*, October 5, 1981. https://books.google.com/books?id=JT0EAAAAMBAJ&pg=RA1-PA50#v=onepage&q&f=false (viewed 1/12/19).

Froot, K.A., (1992) “Intel Corporation, 1992,” Harvard Business School Publishing.

Graham, L.D. (1999) *Legal Battles that Shaped the Computer Industry*, Quorum Books, Westport, CT.

Grove, Andrew S. (1996). *Only the Paranoid Survive*, New York: Doubleday.

Henkel, J., Baldwin, C.Y. and Shih, W., (2013) “IP modularity: Profiting from innovation by aligning product architecture with intellectual property.” *California management review*, *55*(4):65-82.

Hennessy, John L. and David A. Patterson (1990) *Computer Architecture: A Quantitative Approach*, San Mateo, CA: Morgan Kaufmann.

Iansiti, Marco and Roy Levien (2004). *The Keystone Advantage: What the New Dynamics of Business Ecosystems Mean for Strategy, Innovation, and Sustainability*, Boston: Harvard Business School Press.

IBM Archives (undated) “Before the Beginning: Ancestors of the IBM Personal Computer,” http://www-03.ibm.com/ibm/history/exhibits/pc/pc_1.html (viewed 1/17/19).

IBM Archives (undated) “Personal Computer Announced by IBM,” http://www-03.ibm.com/ibm/history/documents/pdf/pcpress.pdf (viewed 1/17/19).

IBM Archives (undated) “The Birth of the IBM PC,” https://www.ibm.com/ibm/history/exhibits/pc25/pc25_birth.html (viewed 1/6/19).

IBM Personal Computer Technical Reference Manual, First Edition, August 1981.

Johnson, H. (undated) “Apple Mac to IBM-PC file exchange and diskette exchange,” http://www.retrotechnology.com/herbs_stuff/mactopc.txt (viewed 1/13/19).

Langdell, James (1984) "Phoenix Says Its BIOS May Foil IBM's Lawsuits". *PC Magazine*, 3(13):56 https://books.google.com/books?id=Bwng8NJ5fesC (viewed 1/17/18).

Langlois, R.N. (1992) "External Economies and Economic Progress: The Case of the Microcomputer Industry," *Business History Review* 66(1):1-50.

Langlois, Richard N. and Paul L. Robertson (1992). "Networks and Innovation in a Modular System: Lessons from the Microcomputer and Stereo Component Industries," *Research Policy,* 21(4): 297-313.

Mace, S. (1982) "IBM releases CP/M-86 for the Personal Computer after delay," InfoWorld, April 26, 1982. https://books.google.com/books?id=azAEAAAAMBAJ&pg=PA8#v=onepage&q&f=false (viewed 1/12/19).

Marshall, A. (1890) Principles of Economics, Book IV, Chapter 9, Paragraph 25, http://www.econlib.org/library/Marshall/marP23.html#IV.IX.25 (viewed 3/5/17).

Milgrom, P., & Roberts, J. (1995). Complementarities and fit strategy, structure, and organizational change in manufacturing. *Journal of accounting and economics*, 19 (2): 179-208.

Milgrom, Paul and John Roberts (1990) "The Economics of Manufacturing: Technology, Strategy and Organization," *American Economic Review* 80 (3): 511-28.

Moffat, S. (1992) "IBM Reports First Loss--$2.8 Billion : Business: The 1991 decline is blamed on the recession and costs of staff reductions. The firm's stock has fallen, but analysts believe it will weather the storm." *Los Angeles Times,* http://articles.latimes.com/1992-01-18/news/mn-65_1_staff-reductions (viewed 1/18/19).

Montague, C., Howse, D., Mikkelsen, B., Rein, D., and Mathews, D. (1983) "Technical Aspects of IBM PC Compatibility," *BYTE Magazine* 8(11):248–253; https://archive.org/details/byte-magazine-1983-11/page/n247 (viewed 1/18/19).

Moore, James F. (1996). *The Death of Competition: Leadership & Strategy in the Age of Business Ecosystems*. New York: HarperBusiness

Mueller, Scott (2003), *Upgrading and Repairing PCs*, 15th ed., Indianapolis: Que Publishing.

Parker, G. and Van Alstyne, M. (2018) "Innovation, Openness, and Platform Control," *Management Science*, 64(7):3015-3032.

Parker, G., Van Alstyne, M. and Jiang, X., (2017). Platform Ecosystems: How Developers Invert the Firm. *MIS Quarterly*, *41*(1):255-266

Sanger, David E. (1984) "Eagle's Battle for Survival," New York Times; http://www.nytimes.com/1984/06/09/business/eagle-s-battle-for-survival.html (viewed 1/17/18).

Shapiro, Carl and Hal R. Varian (1999). *Information Rules: A Strategic Guide to the Network Economy*, Boston, MA: Harvard Business School Press.

Sturgeon, Timothy J. (2002). “Modular Production Networks: A New American Model of Industrial Organization,” *Industrial and Corporate Change*, 11(3): 451-496.

Topkis, Donald M. (1998) *Supermodularity and Complementarity*, Princeton University Press, Princeton, NJ.

Ward, Ronnie (1983). "Levels of PC Compatibility" *BYTE Magazine* 8(11):248–249; https://archive.org/details/byte-magazine-1983-11/page/n249 (viewed 1/17/18).

Williams, G. (1982) “A Closer Look at the IBM Personal Computer,” *BYTE Magazine* 7(1):36-68. https://archive.org/details/byte-magazine-1982-01/page/n37 (viewed 1/18/19).

Yoffie, D. B., Laurie, R. and Huston, B. (1989) "Intel Corporation 1988," Harvard Business School Publishing, Boston, MA.

Zachary, G. Pascal (1994) *Showstopper! The breakneck race to create Windows NT and the next generation at Microsoft*, Boston, MA: Little, Brown.