HARVARD | BUSINESS | SCHOOL

Design Rules, Volume 2: How Technology Shapes Organizations

Chapter 17 The Wintel Standards-based Platform

Carliss Y. Baldwin

Working Paper 20-055

Design Rules, Volume 2: How Technology Shapes Organizations

Chapter 17 The Wintel Standards-based Platform

Carliss Y. Baldwin
Harvard Business School

Working Paper 20-055

Funding for this research was provided in part by Harvard Business School.

Design Rules, Volume 2: How Technology Shapes Organizations

Chapter 17 The Wintel Standards-based Platform

By Carliss Y. Baldwin

Note to Readers: This is a draft of Chapter 17 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 Wintel Standards-based Platform," HBS Working Paper (November 2019).

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

Abstract

The purpose of this chapter is to use the theory of bottlenecks laid out in previous chapters to better understand the dynamics of an open standards-based platform. I describe how the Wintel platform evolved from 1990 through 2000 under joint sponsorship of Intel and Microsoft. I first describe a series of technical bottlenecks that arose in the early 1990s concerning the "bus architecture" of IBM-compatible PCs. Intel's management of buses demonstrates how, under conditions of distributed supermodular complementarity, a platform sponsor can reconfigure the modular structure of a technical system, property rights within the system, and its own zone of authority to increase system-wide throughput, while protecting its own strategic bottleneck from disintermediation.

I go on to describe how Microsoft used platform envelopment to establish a second strategic bottleneck in productivity software and later to respond to the threat of disintermediation from platform-independent Internet browsers. I end the chapter by discussing the conditions under which shared platform sponsorship can be a long-term dynamic equilbrium.

Introduction

The original IBM PC was both a standards-based and a logistical platform. However, when IBM lost control of the BIOS, a shift occurred in technical and industry architecture. In the new lineup, Intel and Microsoft became sponsors of the standards-based platform by virtue of their control of the system's visible information. However, PC users did not want to assemble their computers as a kit. They preferred to purchase a pre-assembled platform so that they could count on having a complete functioning system "out of the box."

Thus Intel and Microsoft became suppliers to a large number of system integrators, known as OEMs (original equipment manufacturers). IBM and clonemakers like Compaq, Dell and Acer purchased microprocessors from Intel and operating system software from Microsoft, and also purchased many other components—chips, circuit boards, disk drives, displays, keyboards, etc.—from other suppliers. They assembled the components into working computers, and marketed and transported the finished goods to corporations, retailers, and consumers. Consumers could then augment their systems with optional hardware and software according to their needs and preferences.

The purpose of this chapter is to use the theory of bottlenecks laid out in previous chapters to better understand the dynamics of an open standards-based platform. I describe how the Wintel platform evolved from 1990 through 2000 under joint sponsorship of Intel and Microsoft. I first describe a series of technical bottlenecks that arose in the early 1990s concerning the "bus architecture" of IBM-compatible PCs. Intel's management of buses demonstrates how, under conditions of distributed supermodular complementarity, a platform sponsor can reconfigure the modular structure of a technical system, property rights within the system, and its own zone of authority to increase system-wide throughput, while protecting a strategic bottleneck from disintermediation.

I go on to describe how Microsoft used platform envelopment to establish a second strategic bottleneck in productivity software and later to respond to the threat of disintermediation from platform-independent Internet browsers. I end the chapter by discussing the conditions under which shared platform sponsorship can be a long-term dynamic equilbrium.

17.1 Intel and the Problem of Buses

Toward the end of the 1980s, the bus and chipset design of the PC became a technical bottleneck standing in the way of higher system-level performance. This section describes the initial "problem of buses" from Intel's perspective. It illustrates the following theoretical principles:

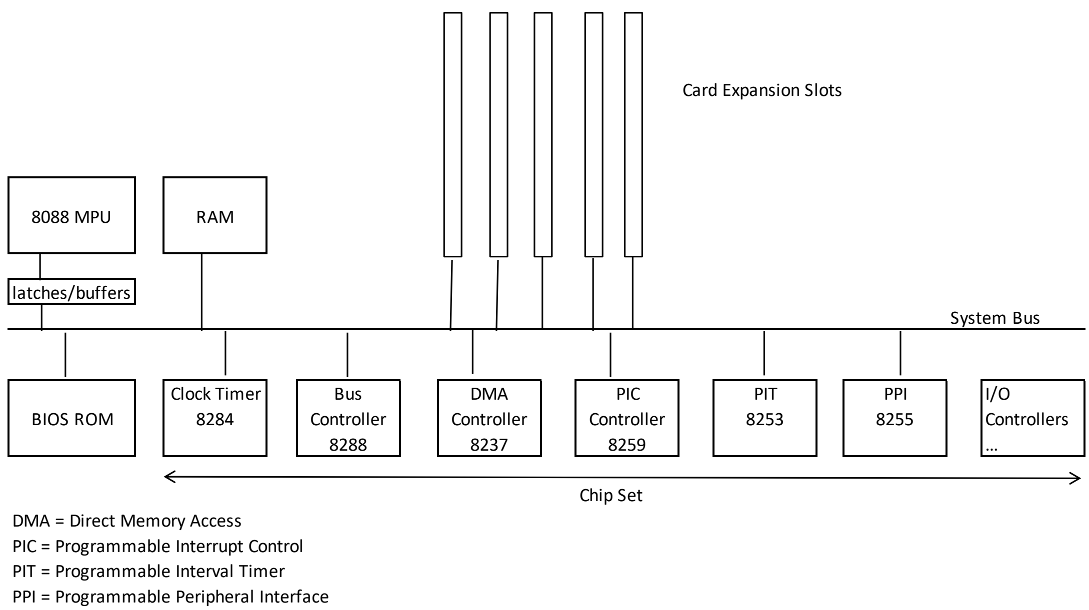
- Standards are not fixed. Additional standards are needed to support more complex technical architectures.
- Backward compatibility is an important factor that often drives the adoption of new standards.
- Industry groups (consortia) can be organized to promulgate new standards.
- Flow bottlenecks in the "core" of a platform may slow down instruction processing for the system as a whole.
- Standards can be *modularized* to allow flexible evolution of the system as a whole.

Standards are not fixed. Additional standards are needed to support more complex technical architectures.

As mentioned in previous chapters, bus specifications are part of a computer's visible information. As such, they can be the basis of a strategic bottleneck, if they have the status of protected intellectual property. Bus standards specify hardware designs, thus can be protected via patents. However, to warrant a patent, a design must pass the test of being "novel" and "non-obvious." The IBM PC's bus design was extremely simple and was not patented.[1] Thus once IBM's grip on the PC BIOS was broken, clone-makers could simply install a PC-compatible bus, with no worries about infringing on IBM's intellectual property.

Figure 17-1 is a simplified diagram of the motherboard of the first three IBM PC product lines, the PC, the PC-AT and the PC-XT. The boxes represent chips or cards and the lines represent buses with associated wires, ports and cables. The entire system was connected via a single system bus. Numerous single-function controller chips, collectively known as the "chipset," supported the transfer of data to and from the microprocessor (MPU). Notably, many of the chips in the original PC chipset were manufactured by Intel, but these were relatively simple chips that could be sourced from other vendors as well.

Figure 17-1 Schematic Diagram of IBM PC Motherboard



Source: Constructed by the author based on the IBM PC Technical Reference Manual, Mueller (2005), and Benschop (2011).

This configuration became known as the Industry Standard Architecture (ISA). As processors ran at higher speeds in the late 1980s, clone makers created two buses, an internal system bus with high bandwidth (32 bits) and an ISA bus with lower bandwidth (16 bit). The two buses were connected by a special chip called a "bridge."

[1] The simplicity of the PC bus architecture is indicated by the fact that its entire description, including all commands took up three pages in the Technical Reference Manual. In contrast, the description and command listing of the ROM BIOS required over 100 pages.

Backward compatibility is an important factor that often drives the adoption of new standards.

Industry groups (consortia) can be organized to promulgate new standards.

IBM sought to re-establish control of the PC's visible information by developing and patenting a new Micro Channel™ bus architecture. It was willing to license the Micro Channel to clone manufacturers for a fee of 1% to 5% of revenue.[2] Unfortunately, the Micro Channel bus was not backward compatible with existing expansion cards and would require users to scrap large amounts of legacy hardware and software.[3]

Responding to the MicroChannel's lack of backward compatibility and IBM's high fees, a consortium of clone-makers, led by Compaq, developed a patented Extended Industry Standard Architecture (EISA), which was backward compatible with the previous Industry Standard Architecture (ISA).

Flow bottlenecks in a platform may slow down instruction processing for the system as a whole.

The PC's bus architecture posed problems for Intel on two levels. First, an inadequate bus system "starved" the processor by slowing down data transfers between the processor and other parts of the system. Spurred by competition from other chipmakers (see Chapter 11), Intel was on track to introduce a series of ever-faster microprocessors as quickly as Moore's Law would allow. But without high-bandwidth buses, the microprocessor would be idle most of the time. This would drag down the performance of the user's system with a commensurate negative effect on demand.

Fortunately, Intel had two product divisions that dealt directly with the flow of instructions beyond the microprocessor. First, it had a systems division that offered fully functional "white box" PCs (PCs minus a screen and a keyboard) as well as PC motherboards to companies such as AT&T, Prime Computer and Unisys. The systems business generated $400 million in revenue in 1988, but was resented by Intel's major customers.[4] However, it meant that Intel had the inhouse capabilities needed to design and manufacture small computers.

Second, in the late 1980s, Intel created a chipset division to explore new ways of packaging the controller chips in the chipset (see Figure 17-2). This group had designed a chipset for the IBM Micro Channel bus architecture in 1988 and a chipset for the EISA architecture in 1989. The EISA chipset was something of a breakthrough because it reduced the number of chips needed to control data flow on the motherboard from 98 to

[2] Lewis (1988).

[3] Ferguson and Morris (1993) p. 58.

[4] Casadesus-Masanell et al. (2010) p. 8.

two.[5] Nevertheless, the chipset division struggled in the market as it competed with the product design groups of the major OEMs and vendors such as Chips and Technologies.

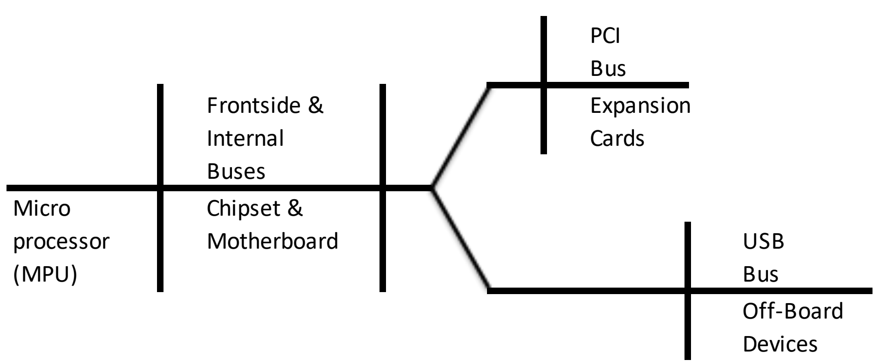
Thus as Intel engineers began to analyze the PC's bus architecture in 1991, they could draw on significant inhouse expertise regarding the two contending state-of-the-art bus architectures as well as the design and manufacture of chipsets, motherboards and even whole systems. In effect, although the microprocessor, chipset, and systems divisions were independent profit centers, Intel as a company had the ability to perform all the step processes needed to go from a processor to a whole system.

In 1991, in recognition of the fact that Intel would prosper only if the entire market for PCs grew rapidly, the company established the Intel Architecture Lab (IAL) with a mission to "grow the overall market … by getting new applications and finding new users for the PC."[6] The IAL had a mandate from top management to recognize complementarities between the platform and complementors and to work with members of the ecosystem as needed.

Standards can be modularized to allow flexible evolution of the system as a whole.

Intel first approached the looming technical bottleneck in the bus architecture by modularizing buses based on their distance from the microprocessor, as shown in Figure 17-3. In the figure chips and other hardware devices are shown below the horizontal lines, while buses are shown above the lines.

Figure 17-2 PC System Arranged by Distance from Microprocessor (MPU)



The motherboard contained the microprocessor, the chipset, and RAM memory chips. Here Intel designed a new proprietary chipset and set of buses that were highly integrated with its processor designs. At the next remove were various expansion slots on the motherboard where cards could be inserted that handled functions such as graphics, networking, sound and internal storage. For this set of functions, Intel created the "PCI

[5] Intel Timeline 1980-1989.

[6] Gerald Holzhammer (1997) Co-Director of IAL, as quoted by Gawer and Cusumano (2002), pp. 23-24.

bus." At the furthest remove were a staggering variety of off-board peripheral devices such as keyboards, mice, external disk drives, printers, cameras, etc. Here Intel collaborated with other major players in the industry to design a "universal serial bus" (USB) that made it easy for users to connect, chain together, and mix and match hardware devices with their PCs.

Modularization of standards combined with management of property rights can partition the system and hide information in ways that make continued evolution possible. The Intel partition was one such modularization. However, as new graphics, video, storage, and networking functions were added to the initial PC architecture, specialized buses such as VESA, IDE, SATA, and Ethernet were proposed and promoted by various industry groups. Multiple, competing standards in turn gave rise to attempts by OEMs to simplify designs by reducing the number of buses and corresponding ports in their machines. The latest Apple Macbooks, for example, have only a single external bus (USB-C) and a uniform set of ports.

The following sections describe how during the early 1990s, Intel addressed the "problem of buses" by (1) changing the modular structure of the PC platform; (2) reallocating property rights, reserving some to itself and allocating others to standard-setting organizations; (3) consolidating its zone of authority to protect the microprocessor as its strategic bottleneck within the PC platform system.

17.2 The Core of the Platform

This section describes how Intel redesigned the chipset and internal buses on the PC motherboard. It illustrates the following theoretical principles:

- Integration to increase processing speed in the core of the platform;
- Integration to deter reverse engineering;
- (Partial) platform envelopment;
- Accommodation of diverse preferences via a flexible production platform.

Each of these is discussed in subsections below.

Integration to increase processing speed in the core of the platform

The basic dynamic of Moore's Law requires placing more elements on each chip as the size of individual gates and circuits declines (see Chapter 11). Thus all companies making chips were bound to reduce modularity to pack more functions on a chip and to make the chips run faster.

The trend towards reducing the number of chips in a chipset was evident from the mid-1980s. In 1988, Intel's chipset division decreased the number of chips in an EISA-compliant chipset from 98 to 2.[7] In 1989, Chips and Technologies, the market leader,

[7] Intel Timeline 1980-1989; "First EISA Chips Delivered," (1989); Gawer and Cusumano (2002) p. 37.

introduced a single chip controller for IBM-AT computers.[8] Intel engineers continued to advocate a two-chip design.

Figure 17-2 is a diagram of Intel-style motherboard from the mid-1990s.[9] The architecture has fewer chips than the original PC (see Figure 17-2), but there are now several special-purpose buses each associated with a different standard. Numerous controller chips have been consolidated into two larger chips, the Northbridge and the Southbridge.

In the top half of the figure, transfers to and from the CPU and the graphics chip and memory had to take place very quickly. In the bottom half, transfers within the motherboard or between the motherboard and external devices such as disk drives, the keyboard, a mouse, etc., took place more slowly. At the very bottom a chip containing legacy BIOS commands was connected to the Southbridge via a "low pin count" (LPC) bus. (As the design of PCs evolved, the slow BIOS was less involved in driving specific hardware devices, but, for many years, it was still necessary to maintain backward compatibility.[10])

Within this new technical architecture, Intel exercised property rights to bring the central core of the computer within its zone of authority. The Northbridge and Southbridge chipsets were proprietary, protected by numerous patents. The same was true of the designs of the frontside bus and the internal bus. Unpatented parts of the design were treated as trade secrets protected by strict non-disclosure agreements with OEMs.[11]

Integration of the processor and chipset was not Intel's only option. Intel could have licensed the chipset and bus designs widely, allowing its licensees to compete to supply these components at the lowest possible cost. In 1987, Sun Microsystems used a wide licensing strategy to promote its SPARC processor architecture. ARM Holdings has succeeded with a business model based entirely on licensing "IP cores," which are reusable and recombinable parts of a chip design.

---

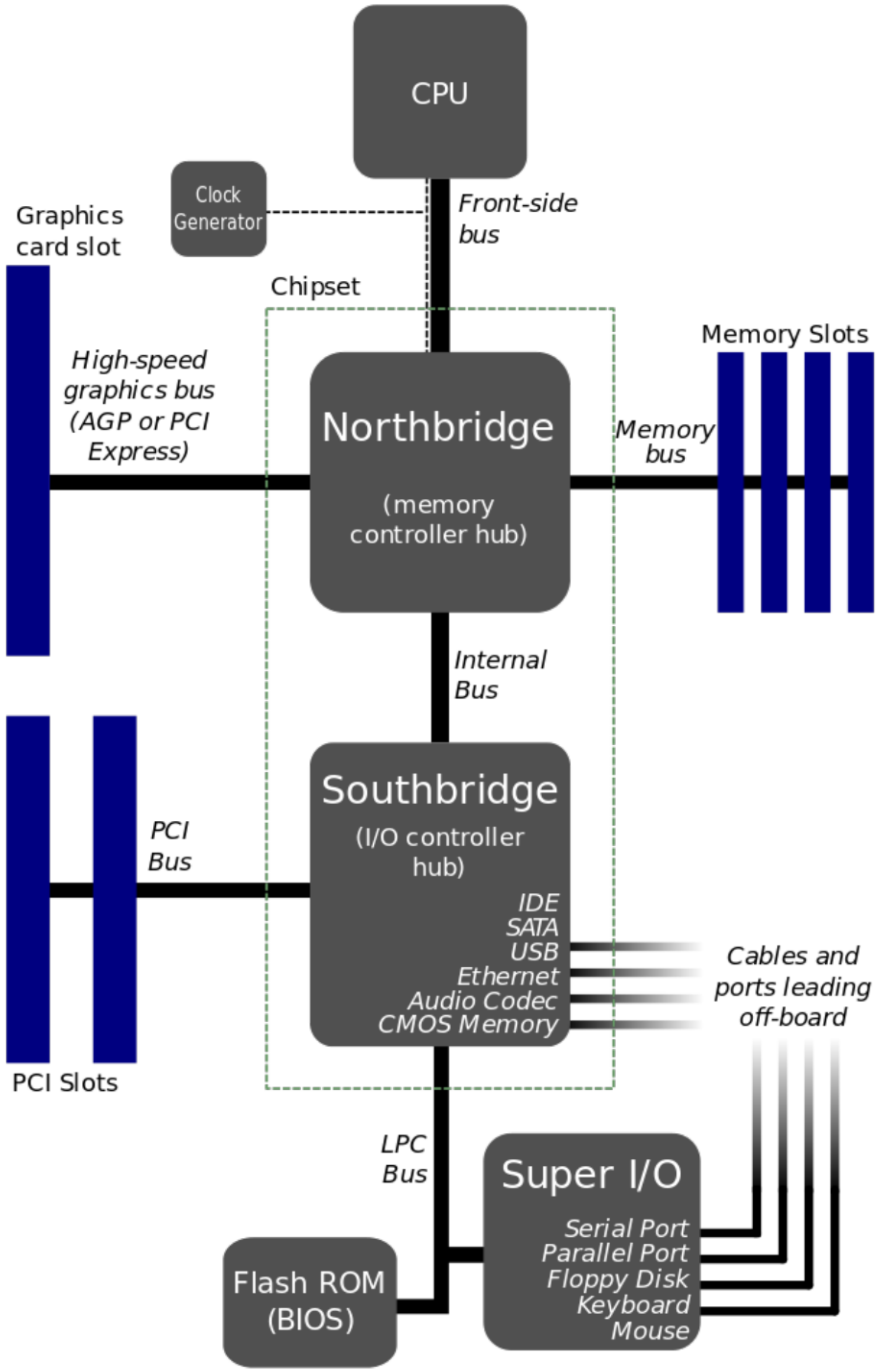
[8] Copeland (1989).

[9] There are quite a few variants on this basic architecture, mainly involving the placement of buses. This diagram captures a common configuration.

[10] Mueller (2005).

[11] Gawer and Cusumano (2002) pp. 34-37.

Figure 17-2 Schematic Diagram of Intel Motherboard from mid-1990s



Source: Moxfyre at English Wikipedia (2009) "Motherboard diagram.svg" https://commons.wikimedia.org/wiki/File:Motherboard_diagram.svg (viewed 3/1/16); Reproduced with attribution under the Creative Commons license.

Integration to Deter Reverse Engineering

However, Intel's approach to the chipset and internal bus served a second purpose: it made it more difficult to reverse engineer Intel microprocessors.

The new bus architecture allowed Intel to create "one-way interface standards" between external devices and the central processor.[12] For makers of peripheral devices, Intel (or a standards-setting organization of which Intel was a leading member) would publish a standard, indicating what the device should do to work correctly with the processor. But instead of connecting directly with the processor through a single bus controller, as in the ISA architecture, the external devices connected with some part of the Northbridge or Southbridge. The Northbridge and Southbridge in turn used proprietary internal buses and secret protocols to send and receive information to and from the processor.[13]

The upshot of these architectural changes was to close off a portion of the system from third parties. The CPU, Northbridge and Southbridge chips were still nominally modules that could be purchased separately. But the means of hooking the modules together—the design rules governing their interoperability were unknown. A competitor by definition could not observe how the processor responded to instructions from the Northbridge or Southbridge. He or she could only observe how the three-chip system responded to instructions coming from outside. The competitor would then have to create three interacting components not one. The cost of reverse engineering increases with the complexity of the system being simulated, thus integrating the chipset with the processor increased the costs and delayed the success of rival chipmakers.

Intel's microprocessors were under constant threat of reverse engineering by Advanced Micro Devices (AMD). In conjunction with its Athlon series of processors, in 1999, AMD introduced its own chipsets. Unlike previous AMD processors, the Athlon and subsequent families were software compatible, but not hardware compatible with Intel designs.

Today AMD is a fabless semiconductor firm, which outsources chip fabrication to TSMC and other foundries.[14] (See Chapter 12.) Intel and AMD designs require different motherboards, but both companies design chips that implement versions of the 80x86 instruction set. Most software programs will run on both types of machine.

[12] Mackie-Mason. and Netz (2007).

[13] Initially the Northbridge and Southbridge chips were connected via a PCI bus, which was a published standard. In the late 1990s, Intel introduced a dedicated hub interface that had twice the throughput of PCI. The new interface also reduced congestion in the PCI bus itself and increased throughput for devices connected directly to the Southbridge (Mueller, 2005, p. 245).

[14] Manners, D. (2010); Chiapetta, M. (2018).

(Partial) Platform Envelopment

Bus standards are visible information needed to design transfers of instructions to and from the central processor and the chipset. Designers of the next generation of the chipset could potentially implement a new bus architecture that would control all traffic between the microprocessor and the rest of the system. If the chipset-cum-bus-architecture succeeded in the marketplace, that would create a third system-wide strategic bottleneck. IBM had tried and failed at such a move with the Micro Channel. A consortium of OEMs had also tried and failed with the EISA bus architecture. Chips and Technologies was positioned to make the next attempt.

By integrating the design of the microprocessor with the internal buses and the Northbridge and Southbridge chipsets, Intel was in effect "enveloping" the chipset platform into its microprocessor platform.[15] Chips and Technologies, in contrast, did not control a microprocessor, thus Intel's move effectively disintermediated its chipset product line. In contrast, AMD did have its own microprocessors, and thus could create a complete three-chip solution. Envelopment created challenges for AMD; but it was fatal for Chips and Technologies.

Although Intel was well-entrenched in the microprocessor market, it was a comparative newcomer to the chipset and motherboard marketplaces. For this reason, it did not implement the second part of the envelopment strategy by withdrawing its stand-alone product, thus forcing customers to purchase its chipsets and motherboards.

Flexible Production Platform

Instead, Intel made a bet through its existing product divisions on a flexible production platform. In 1993, the company began the mass producing chipsets in its chipset division in Folsom, California. The Intel chipset made use of the new PCI 2.0 external bus architecture discussed in the next section. The systems division then used Intel processors, chipsets, and the new bus architecture to make motherboards for the "white box" channel. However, the CPUs, chipsets and motherboards remained separate products, sold by different product divisions.

As with GM in the 1920s and 1930s, Intel aimed to offer a product suited to "every purse and purpose" (see Chapter 10). By the mid-1990s, a company selling IBM-compatible PCs could get a specification from Intel and design their own chipset; purchase a chipset from Intel and design their own motherboard; or buy the whole motherboard from Intel. Companies like Compaq, HP and IBM generally chose to design their own motherboards, while companies like Dell and Gateway took the so-called "white box" route and purchased their motherboards from Intel and other vendors.[16] Intel continued to supply customers who wished to design their own chipsets and

---

[15] Eisenmann, Parker and Van Alstyne (2011).

[16] Mueller (2005).

motherboards with confidential technical information and prototype chips under strict non-disclosure agreements.[17]

By the late 1990s, Intel dominated the markets for chipsets and motherboards. According to MarketResearch.com, in 1998, Intel had revenues of $1.2 billion from chipsets for a staggering 81% market share and revenues of $2.4 billion from motherboards with 35% market share.[18] The former market leader, Chips and Technologies, dropped out of the market and was acquired by Intel in 1998.[19]

Of course, the lion's share of Intel's $26 billion in revenue in 1998 arose from the fact that virtually every PC-compatible computer sold anywhere in the world contained an Intel microprocessor. (See Figure 15-7.) Revenues from chipsets and motherboards were icing on the cake.

The history of the Intel chipset and internal buses demonstrates that the same technical move can serve several different purposes. Intel's goals were threefold: to speed up the system by addressing potential bottlenecks in the flow of instructions to and from the processor; (2) to deter (or at least slow down) reverse engineering of its microprocessors by AMD; and (3) to ensure that rivals, such as IBM and Chips & Technologies did not use buses to establish a third strategic bottleneck in the PC technical system. Intel achieved these goals by integrating the designs of the processor, internal buses and chipset and hiding their internal structure from third parties. In effect these three components became a single module within the computer system as a whole.

17.3 Expansion Cards and the PCI Bus

This section describes Intel's handling of the Peripheral Component Interconnect Bus, which linked the processor and chipset with special-purpose expansion cards mounted in slots on the motherboard. It illustrates the following theoretical principles:

- Modularization of the bus and processor to facilitate upgrading;
- Encouraging adoption of the new standard by working with key industry members.

Intel's sponsorship of the "universal serial bus" (USB) illustrates the same theoretical principles, and for this reason, I do not consider it in detail. However, from the start, USB required more distributed effort, thus Intel, while supportive, did not exercise

[17] In response to patent infringement suits by DEC, Compaq and Intergraph, Intel threatened to stop sharing advanced technical information with these companies and halt the delivery of chips to their factories. The FTC then filed an antitrust complaint alleging abuse of monopoly. Intel settled without admitting liability. Valentine (1999).

[18] MarketResearch.com (2001).

[19] McLellan (2017).

direct control over the evolution of this standard. Internally, Intel appears to have prioritized the PCI family of buses over the USB family.[20]

Modularization to Facilitate Upgrading

By definition, all platform components are essential, but they will not all change at the same rate. Different modular components have different trajectories and can be upgraded at different times.[21] Consistent with its partition of PC buses (see above), Intel's handling the Peripheral Component Interconnect (PCI) bus and its successors differed from its treatment of the internal buses and chipsets. Here Intel did not integrate components, *but opted for increased modularity by making the PCI bus design independent of the microprocessor*.

The PCI bus connects third-party hardware devices, such as network cards, video cards, sound cards, modems, disk controllers, to the microporcessor via the Southbridge chip (see Figure 17-3). The high-speed graphics buses, AGP and PCI Express, connecting graphics cards to the Northbridge chip, were also part of the PCI family. (AGP was dedicated to graphics, while PCI Express was a more general bus.)

On inspecting existing bus designs in the early 1990s, Intel engineers found that all of them depended on the specific processor in use. As a result, introducing a new generation of microprocessors meant redesigning the buses as well. *This technological requirement gave large OEMs such as IBM and Compaq strong incentives to delay the introduction of new and faster processors*. Yet, Intel's profits from the PC platform depended on introducing the new generations of processors at a rapid pace consistent with Moore's Law.

To encourage adoption of new processors, Intel had to make the upgrade path easier for OEMs and expansion card designers. It needed to allow complementors to change the processor, chipset and internal buses without changing the rest of the system. This was a new technical bottleneck for the system as a whole.

In classic fashion, Intel engineers solved the technical bottleneck by creating a new set of design rules binding on both the processor and peripheral devices.[22] The new design rules allowed the operating system to assign new addresses to each peripheral device every time the system restarted.[23] (Previously the addresses had been set via manual switches when the peripheral device was installed.)

---

[20] See, for example, Vilches (2009); Cunningham (2014)

[21] Dosi (1982).

[22] The technique of replacing lateral interdependencies with hierarchical design rules to create new modules is described in Baldwin and Clark (2000) pp. 259-264.

[23] OS Dev.org "PCI" http://wiki.osdev.org/PCI#Base_Address_Registers (viewed 2/24/16); "PCI Configuration Space," Wikipedia, https://en.wikipedia.org/wiki/PCI_configuration_space (viewed 2/24/16).

The processor and the PCI bus thus became separate modules in the design of the PC: changing one did not require changing the other. The new design for a PC was capable of gracefully evolving because the processor and the external bus were independent of each other.[24]

Encouraging adoption of the new standard by working with industry members in a standards-setting organization

In 1993, Intel published a new standard, PCI 2.0, containing its processor-independent bus solution. However, OEMs and designers of expansion cards had to adopt the new bus standard in order for it to be successful. Incumbents (including the company's own chipset division) were not eager to build machines for a non-existent market.[25] For its part, Intel could not guarantee that the demand for PCI-compliant machines would materialize. IBM's Micro Channel and the industry-sponsored EISA and VESA buses had not been widely adopted. Despite being very slow, the old ISA bus was still in common use.

IAL engineers used a two-pronged strategy to spur adoption of the new standard. First, as recounted above, they persuaded the chipset division to create a PCI-compliant chipset. The chipset and motherboard divisions then received funding from senior management to ramp up chipset and motherboard manufacturing and marketing to mass production levels. In this fashion, Intel provided a "proof of concept" that the new bus architecture would work. It also spurred competition in the OEM market by allowing smaller OEMs, which did not have the ability to design chipsets and motherboards, to offer products that were competitive with those designed by Compaq and IBM.[26]

However, Intel needed the cooperation of the largest OEMs if the PCI standard was to succeed. Thus, in 1992, they recruited four large OEMs—Digital Equipment Corporation, Compaq, IBM, and NCR— to join Intel engineers in a Special Interest Group (SIG) to vet decisions about the standard and give it credibility.[27] The SIG was part of a "compliance program," that sponsored workshops known as "plugfests." At a plugfest, engineers from various hardware vendors would show up with prototypes and test them on different platforms (computers and other equipment) to see if the products would seamlessly operate on those platforms.

---

[24] Mueller (2005). An identical rationale lay behind the creation of System/360 in 1965: if some parts of the system (the hardware) "needed" to change quickly, then switching costs for those components must be low. Those components became modules in the larger system. Baldwin and Clark (2000) Chapter 7.

[25] Gawer and Cusumano (2002) p. 37.

[26] Ibid. quoting Andy Grove, p. 33. As recounted in Chapter 15, the transition from the 286 to 386 generation was accelerated by Compaq's adoption of the 386 chip to gain an edge on IBM. In 1994, the shoe was on the other foot. The transition from 486 to Pentium was led by Dell and Gateway, with Compaq mostly sticking with the older 486 line and purchasing reverse engineered chips from AMD. Kirkpatrick, (1994).

[27] Crothers (1994).

In 2000, PCI-SIG was incorporated as a standards-setting organization supported by dues-paying members and fees for conferences and events. Control of the standard thus passed to a non-profit consortium. In 2019, PCI-SIG had over 700 members, and its Board of Directors included representatives from AMD, Dell/EMC, Intel, NVIDIA, Keysight Technologies, Synopsys, Qualcomm, and IBM.[28]

Standards owned by a for-profit enterprise place users at risk of *ex post* hold up by the owner. By placing PCI (and later the USB standard) in the hands of an industry consortium, Intel renounced its rights to profit from the standard. Today, a company can use the PCI bus architecture and all its variants for a flat fee of $4000 per year.[29] This nominal amount stands in stark contrast to the 1%-5% of revenue IBM attempted to charge for using the MicroChannel bus architecture.

To hightlight the comparison, ($4,000 dues x 700 members) = $2.8 million in revenue. This is an upper bound on PCI-SIG's revenue from dues in 2000, when Intel transferred ownership of the standard to the SIG. Total revenue from all Intel-based PCs in that year was approximately $120 billion. One to five percent of that amount would be $1.2 – 6 billion. Intel's actual revenue in 2000 was $33 billion—approximately ¼ of its customers' revenue. Clearly Intel did not need to PCI bus to claim a big chunk of the system's surplus.[30]

Intel already controlled a well-protected strategic bottleneck in the form of the processor, chipset, and internal buses. An important fact about strategic bottlenecks is that they are not additive. As long as it is protected against disintermediation, one strategic bottleneck is all that is needed to claim a share of the system surplus. Thus controlling yet another unique component of the platform would not change Intel's strategic position very much.

It *was* in Intel's interest to address flow bottlenecks in the instruction paths of computers and peripheral devices made by complementors. In the early 1990s, there was great concern among Intel engineers that sales of Pentium chips (first released in 1993) were being held back because the ISA bus was "too stinkingly slow" to take advantage of Pentium's speed.[31] If putting an industry association in charge of PCI could allay the OEM's suspicions and accelerate adoption of the new bus standard, Intel had good reason to follow that course of action.

17.4 Microsoft and the Problem of Subsidiary Platforms

Microsoft was the second sponsor of the IBM-compatible PC platform system. Its approach to technical and strategic bottlenecks paralleled Intel's in many ways. It worked

---

[28] PCI-SIG Membership https://pcisig.com/membership (viewed 10/21/19)

[29] Ibid.

[30] Calculations by the author based on on IDC, WW Quarterly Personal Computer Device Tracker, 2016Q4 Historical Release, Publication Date: February 10, 2017 and Intel financial reports.

[31] Bill Miller of Intel, quoted by Gawer and Cusumano (2002) p. 37.

with OEMs on system integration, especially in the transition to Windows 3.0 and to Internet-enabled PCs. It solved system-wide technical bottlenecks, most notably by creating tools to support software development (e.g. Visual Basic) and transfer and link objects across applications (e.g. COM and OLE).

However, for purposes of building *new* theory, what is most interesting about Microsoft was the way it dealt with the changing scope of the platform itself. As noted in Chapter 16, new functional components often take the form of subsidiary platforms with their own APIs and options. Initially modular experimentation may generate a large number of variant designs with different features, but as users become more aware of their own needs and desires, the variants will often converge to a common technical architecture. If the new technical architecture is controlled by a for-profit enterprise, the subsidiary platform can be the basis of a new strategic bottleneck.

17.5 Productivity Software

This section looks at the emergence of productivity software in IBM-compatible PCs. It illustrates the following theoretical principles:

- Integration to enable seamless transfers of content between software applications and a consistent user experience across applications;
- (Partial) platform "envelopment".

Integration to facilitate seamless transfers of content between software applications and a consistent "look and feel"

A "killer app" is a piece of software so essential to users, that they buy the platform for the purpose of running the application. An application that is essential to most or all users perforce becomes, not an option, but part of the platform itself. Often such applications are installed by OEMs as part of the original system configuration.

The first killer app for a microcomputer was Visicalc, the spreadsheet program written for the Apple II. Lotus 1-2-3, written in x86 assembly language, became the dominant spreadsheet application for the IBM PC. WordPerfect 3.0 was the dominant application for word processing, while Harvard Graphics and Lotus Freelance vied for position in the nascent presentation market. Spreadsheets, word processing and presentation software came to define what office workers needed to do and home users wanted to do on their IBM-compatible PCs.

Microsoft entered the market for productivity software early, developing spreadsheet and word processing programs for Macintosh computers and IBM PCs in the early 1980s. Under Apple's tutelage, Microsoft developers designed graphical user interfaces for these programs. Microsoft went on to leverage its developers' knowledge when designing Windows, a DOS-compatible graphical operating system for the PC.[32]

Microsoft's original spreadsheet and word processing programs were not leaders in either market at any time during the 1980s. There was, however, an emerging technical bottleneck. Users often wanted to incorporate content created in one productivity application in a document created by another. For example, a user might want to embed a spreadsheet in a memo; text in a spreadsheet; or a chart in a presentation. At the same time, productivity applications had a number of functional components—file management, fonts, graphics, and printing—that were implemented differently in each program and appeared different to the user. Hence there was latent demand for an integrated set of productivity applications providing an easy transfer and linking of contents and a consistent user interface.

(Partial) Platform Envelopment

Microsoft addressed this technical bottleneck with the Microsoft Office™ suite, introduced in 1989 for the Macintosh and 1990 for Windows 3.0. In addition to the Excel spreadsheet program and Word word processing program, Office included Powerpoint, a program that produced graphical presentation slides for overhead projectors.[33]

Like Intel's integration of the microprocessor and chipset, the introduction of Microsoft Office was a variation on the strategy of "platform envelopment."[34] However, instead of combining the functionality of a stand-alone rival with a pre-existing platform, Office took three stand-alone subsidiary platforms and combined them into an integrated whole, with a similar user interface and the ability to easily transfer material between the different applications.

Importantly, the category leaders in spreadsheets, word processing, and presentation software at the time were all stand-alone products. In addition, they were all based on the DOS operating system, and were slow to make the transition to the Windows graphical user interface (GUI). As a supermodular complement of Windows, the Office suite had the potential to increase demand for the new operating system and vice versa.

Like Intel, Microsoft did not take its stand-alone application programs off the market. Instead it priced Office at approximately a 30% discount to the cost of acquiring

---

[32] Apple sued Microsoft, claiming that the "look and feel" of the Macintosh operating system was protected by copyright, but the suit was not successful. A similar suit by Xerox against Apple was dismissed. Apple Computer, Inc. v. Microsoft Corp., 35 F.3d 1435 (9th Cir. 1994); Xerox Corp. v. Apple Computer, Inc., 734 F. Supp. 1542 (N.D. Cal. 1990).

[33] Powerpoint was developed by Robert Gaskins and Dennis Austin at Forethought, Inc. and released in April 1987. Microsoft acquired the company in July of that year for $14 million. It was Microsoft's first "significant" acquisition, but its early sales were disappointing. "Microsoft Buys Software Unit," The New York Times (July 31, 1987);. Raikes (2010).

[34] Eisenmann, Parker and Van Alstyne (2011).

the components separately.[35] The Office suite quickly became dominant in all three submarkets. In 1997, Excel and Word had over 90% market share in the spreadsheet and word processing markets, while Powerpoint controlled 85% of the presentation software market.[36]

17.6 Internet Browser

This section considers the competition between Microsoft and Netscape in the early market for Internet browsers. It illustrates the following theoretical principles:

- The threat of disintermediation from platform-independent complements; and
- Full platform envelopment.

In the case of Internet browser software, Microsoft did not have as easy a road as in productivity software. In the mid-1990s, commercial and social use of the Internet exploded, in part because of the low cost and ease of use of the Navigator browser, provided by a startup firm, Netscape. By mid-1995, Netscape had a 70% market share of the browser market, and its momentum was growing, as the use of other browsers declined.[37]

In 1995, Bill Gates came to see the Internet as an inescapable "tidal wave," arguing that "virtually every PC will be used to connect to the Internet and … the Internet will keep PC purchasing very healthy for many years to come."[38] In other words, Internet access was on its way to becoming an essential component of the PC platform as well as a supermodular complement of the operating system and the microprocessor.

Gates proceeded to make Internet interoperability the highest priority for all Microsoft products. He also identified Netscape as a new competitor whose market share allowed its managers to determine which new features in browsers and servers would catch on.[39]

The threat of disintermediation from platform-independent complements

In fact, there were several aspects of Netscape's strategy that were problematic for Microsoft. First, the WorldwideWeb itself was defined as a platform-independent set of protocols. Its three elements, the uniform resource locators (URLs), the Hyper Text Markup Language (HTML) and the Hypertext Transfer Protocol (HTTP), could be

---

[35] "The Microsoft Office Bundles 4 Programs," *Infoworld* (June 19, 1989); "Office for Windows Bundles Popular Microsoft Applications" *Infoworld* (October 1, 1990).

[36] Liebowitz and Margolis (2001); Ziff Davis Market Intelligence (September 1998).

[37] Gates (1995); Kwak and Yoffie (1998) p. 12.

[38] Gates (1995).

[39] It is rumored that Microsoft attempted to buy Netscape in 1994, but the offer was turned down because the price was too low. Rosoff, M. (2011).

implemented on any computer regardless of the microprocessor or operating system. Netscape's browser (Navigator) was also designed to be independent of the operating system, although the company had to partially give up on this goal in order to achieve acceptably fast page loads.[40]

Second, new categories of software developers and IT professionals—web page designers and web administrators—were emerging. Netscape was actively working to supply them with tools in the form of APIs that could be accessed from its browser and server software. These instructions were a new type of visible information, which developers would build into their webpages and server software. Thus the browser itself was a subsidiary platform that could be the basis for a third strategic bottleneck in the PC system (see Chapter 16, Equation 9).

Finally, given high-speed Internet connections, computation of all types could be moved from local PCs to remote servers. When this happened (today we call it "cloud computing"), much of the processing power, memory, and storage capacity of a standard PC would become unnecessary, as would the operating system that managed these resources. A less expensive device with a stripped-down operating system might then satisfy the needs of most users. To Gates, this was a "scary possibility."[41]

Gates' dealt with the Netscape threat by initiating an accelerated program to create an Internet browser capable of displacing Netscape's browser in the marketplace. He also set a goal of achieving a 30% market share for the Microsoft browser.[42] With two browsers in widespread use, *the Netscape browser would not be unique, hence could not become a strategic bottleneck on the Web or within the PC platform system*. Windows would not be disintermediated by a platform-independent browser.

Full Platform Envelopment

By August 1996, fourteen months after Gate's "Tidal Wave" memo, Microsoft developers had created a product (Internet Explorer 3), which was almost as good as Netscape Navigator. The company then used full (not partial) platform envelopment to drive Netscape from the market.

Netscape, which had no other source of revenue, priced its client software at around $50 per download with a 3-month free trial.[43] Microsoft, with revenue from Windows and application software, gave away its browser to OEMs and Internet Service Providers and included it for free in the Windows 95 operating system.

---

[40] Cusumano and Yoffie (1998) pp. 159-174.

[41] Gates (1995). Today, a Chromebook running Google's Chrome OS with the Chrome browser fits this definition.

[42] Cusumano and Yoffie (1998) p. 111.

[43] Kwak and Yoffie (1998) pp. 13-14.

Microsoft also took steps to make sure that the browser code could not be separated from the rest of the operating system. Beginning in 1996, Microsoft developers began to place code used by Internet Explorer into the same files that supported Windows 95.[44] These actions in effect demodularized the operating system and the browser, making it essentially impossible for Microsoft licensees to remove browser code from the larger system. In Windows 98, every system window was in effect a browser: thus users could access the web from any system window without opening a separate application.[45]

It is impossible at this juncture to know what was going on in the code itself. It is clear that IE code was highly dispersed in the directory structure (thus hard to find). It may have been functionally entangled as well (thus hard to separate).

From a purely technological perspective, however, the browser did not need to be intertwined with the operating system. Netscape Navigator was a separate module designed to sit on top of different operating systems.[46] Internet Explorer itself was available as a stand-alone product for Macintosh computers.

Nevertheless, the technical interdependence of the operating system and the browser lay at the heart of Microsoft's defense in the antitrust suit brought by the Department of Justice in 1997.[47] Microsoft had earlier agreed not to tie the sale of one stand-alone software product to another, but the company was allowed to sell integrated products, which combined different functionalities in a single package. However, the difference between stand-alone and integrated software products had never been defined, much less tested in a court of law. There was plenty of scope for legal arguments.

Microsoft managers took three further actions that affected users directly. First, they removed the option to "uninstall" the browser. This was rather disingenuous because one can "uninstall" a function while leaving the supporting code in place. Those portions of the code needed by other parts of the system will be activated when called upon, while those not needed will simply lie dormant in memory. Thomas Jackson, the judge in the Microsoft antitrust case, famously "uninstalled" the IE icon with a few keystrokes in December 1997.[48]

Second and more significant, Windows 98 required the user "to employ Internet Explorer in numerous situations that, from the user's perspective, are entirely unexpected." [49] As one example, Microsoft made IE the gateway to the help system in Windows 98.[50] This meant that a user needing help with any part of the operating system

---

[44] Findings of Fact, *United States v. Microsoft Corporation*, 253 F.3d 34 (D.C. Cir. 2001) p. 81.

[45] Randall (1998).

[46] Cusumano and Yoffie (1998) pp. 159-174.

[47] U.S. vs. Microsoft Timeline (2002).

[48] Ibid.

[49] Findings of Fact, *United States v. Microsoft Corporation* p. 85.

[50] Lea, G. (1999).

would need to use Internet Explorer. Having to switch from another browser to IE to get help is a bit of an inconvenience, and thus users had incentives to make IE their default browser.

Finally, Microsoft threatened to terminate the Windows licenses of OEMs that removed IE icons and programs from the Windows desktop or added their own software to the initial startup sequence. For example, in late 1995, Compaq removed IE and Microsoft Network (MSN) icons from the desktop of its Presario products. On May 31, 1996, Microsoft sent Compaq a letter stating its intention to terminate Compaq's license for Windows 95, if the icons were not restored. Compaq quickly complied.[51]

Following the release of Internet Explorer 3.0 in August 1996, Microsoft's share of the browser market began to rise rapidly and Netscape's began to decline.[52] Six months later (January 1997), IE 3.0 was approaching the magic 30% threshold and by mid-1998 the two companies were neck and neck, according to most surveys.[53] Subsequently Netscapes's market share continued to decline, and company was sold to AOL in 1999.

Judge Thomas Jackson ruled that Microsoft had violated federal antitrust law and engaged in predatory practices targeted at Netscape, Apple, Sun Microsystems, Lotus Software, Real Networks, Linux and others. He ordered the company to be broken up into two separate companies. An appeals court overturned this judgment, but did not dispute Jackson's Findings of Fact. The DOJ and Microsoft reached a settlement in November 2001, and Microsoft continued as one company.[54]

17.7 Shared Platform Sponsorship

This section considers how Intel and Microsoft shared sponsorship of the Wintel standards-based platform from the late 1980s until the present day. The section illustrates the following theoretical principles:

- Distributed supermodular complementarity (DSMC) supporting shared sponsorship; and
- The threat of disintermediation by substitution.

Distributed supermodular complementarity (DSMC) supporting shared sponsorship

During the period of split sponsorship, Intel controlled the microprocessor instruction set while Microsoft controlled APIs and user interfaces of the operating

[51] Findings of Fact, *United States v. Microsoft Corporation*, pp. 98-101.

[52] Kwak and Yoffie, p. 12.

[53] Browser market shares during this period were mostly estimated from small surveys. The estimates were highly inconsistent. See CNN (1998).

[54] "DOJ, Microsoft Settle," *CNNMoney* (2001); Final Judgment, *United States v. Microsoft Corporation*, Civil Action No. 98-1232 (November 12, 2002).

system and the main productivity applications. Each company took charge of improving its own platform component. Thus Intel introduced ever faster generations of microprocessors, each with new instructions. Microsoft brought new functions into the operating system, increasing the range of programs users could run and activities they could engage in.

A platform with shared sponsorship can be analyzed using the analytic methods developed in Chapter 16. Using "bottleneck notation", we can characterize the value structure of the platform system with two system-wide strategic bottlenecks as follows:

$$V_{Shared} = P_1^{*INTC} \cdot P_2^{*MSFT} \cdot P_3 \cdot (1 + \overline{a^{*INTC}} + \overline{b^{*MSFT}} + c^?) \cdot \sum_{i=1}^{N} O_i. \quad (1)$$

Here the P terms are binary variables indicating the presence of each essential platform component. $\sum_{i=1}^{N} O_i$ denotes the value of the platform's original portfolio of options. $\overline{a^{*INTC}}$ represents stand-alone percentage contribution of Intel's stream of improvements while $\overline{b^{*MSFT}}$ represents the stand-alone percentage contribution of Microsofts. The term $c^?$ denotes the supermodular value created by both streams together.

By the arguments on value capture presented in Chapter 16, Intel would be able to claim a fraction of the surplus generated by $\overline{a^*}$ and Microsoft a fraction of the surplus generated by $\overline{b^*}$. However, the split of $c^?$'s value contribution is indeterminate.

The shared platform will be an equilibrium if each firm's share of *c* is greater than what the firm could obtain with another partner or on its own. Put bluntly, removing either party must put a gaping hole in the profit stream of the other party. That is the first condition for equilibrium.

In addition, high costs of integration are also needed to support ongoing separation of the two parties. The supermodular surplus, *c*, must be substantially diminished by a merger of the two partners. Neither side can think it may swallow up the other.

Finally, for the shared platform to be a long-term dynamic equilibrium, prospective *c* must continue to be large as time passes. This in turn means that the system as a whole must be on a trajectory of rapid improvement through complementary investments by both sides.

These conditions require that each partner must have unique capabilities which the other cannot imitate or acquire elsewhere. In addition, their products must be strong complements, each needing the other, and having no good substitutes. At the same time, the products must be sufficiently modular so that the tasks contributing to one or the other can be performed by groups working at different companies.

These requirements are generally satisfied by hardware and software. Each has no value without the other, and strong complementarity (co-specialization) often leads to superior performance. Still, the underlying technology and steps needed to produce these two types of artifacts are very different.

In the 1990s, Microsoft made a successful transition to a graphics-intensive user interface (GUI) for both the operating system and its key applications. As a result, the demand for processing power increased, which in turn led to higher demand for Intel's high-end processors and high-bandwidth buses to support the microprocessor. In effect, the GUI and the microprocessor were supermodular complements: more of one enhanced the value of the other.

Although they were separate companies, Intel and Microsoft's interests were aligned by virtue of this supermodular value function. Each benefited from the investments of the other. Furthermore, each party could "refresh" its revenue stream by offering new products that enhanced the performance of the other party's products. They were joined by the interdependence of their instruction sets and APIs: Microsoft still wrote portions of its code in Intel assembly language to speed up performance.[55] However, each company had separate revenue and profit streams to cover its costs and finance its future growth.

Despite their strong incentives to cooperate, the two companies struggled throughout the 1990s and 2000s to coordinate product releases and product specifications.[56]

The threat of disintermediation by substitution

Of course, each had to protect itself against threats of disintermediation by third parties *assisted by its partner*. Thus the two sponsors sought to define separate spheres of influence. Microsoft was especially disapproving of the Intel Architecture Lab's forays into software.

For example, in the mid-1990s, Intel sought to develop a "Native Signal Processor," a software layer inserted between the operating system and the processor that supported realtime audio, video and 3-D graphics. At the 1998 Microsoft antitrust trial, Steven McGeady, an Intel vice president, testified that Microsoft "believed they owned software to the metal" (i.e., to the chip itself) and convinced Intel's top management to abandon this initiative.[57]

Similarly, to help developers take advantage of hardware peripherals with multimedia capabilities, Intel wrote "device driver interfaces" (DDIs) that allowed software developers to give instructions to the device without going through the operating system. Microsoft (quite rightly) viewed DDIs as substitutes for its own visible information (APIs), and urged OEMs not to adopt Intel's multimedia enabled processors. Microsoft also discouraged Intel from working with Netscape to make their Internet

---

[55] Cusumano and Selby (1995).

[56] Tedlow (2007) pp. 311-317.

[57] *United States vs. Microsoft: Trial Summaries*; "Intel, Microsoft Reach Truce on Native Signal Processing," *CBR* (1998); Tedlow p. 316.

server product run fast on Intel systems and from developing a virtual machine (VM) and class libraries for Sun's Java language.[58]

The theme across all of Microsoft's objections to Intel's software projects was the threat of a platform-independent software layer. This is the most classic form of disintermediation by substitution—an intermediary is simply replaced by a direct connection. Just as when buyers and sellers on a transaction platform decide to deal with each other directly, software developers could send instructions directly to Intel processors without going through Windows. Web application developers and Java programmers could use Intel instructions, optimized for Intel processors, to write their Web-based and multimedia programs. As the set of Intel-specific instructions grew, programmers and users would rely less and less on Windows APIs. The number of Intel's complements would increase, while Microsoft's complements would diminish.

It was an ambitious idea, but in the end not one that Intel was willing to pursue. Faced with Microsoft's fierce opposition, Grove's biographer quotes him as saying, "We caved. Introducing a Windows-based software initiative that Microsoft doesn't support … well life is too short for that."[59]

A way of interpreting Grove's inconsistent actions is that he was torn between the competing lures of $c^?$ and $\overline{a^{*INTC}}$, in other wordes, the value that might be captured via continuing the complementary relationship vs. disintermediating Windows and Microsoft. In effect, the judgment of Grove and Intel's top managers was that such disintermediation was not in Intel's interest at the time. In a world where complementors had the ultimate vote, Intel could not displace Microsoft's power or expertise.[60] Thus Intel withdrew from its software initiatives, to the chagrin of McGeady and others in the IAL.

17.8 Conclusion—How Technology Shapes Organizations

The technical architecture of an open platform characterized by distributed supermodular complementarity can never be static. Each new round of complementary investments changes the architecture of the system and the incentives and rewards to members of the ecosystem.

In an open platform, the sponsors cannot rely on unified governance, hierarchy, and direct authority to solve emerging flow bottlenecks and encourage technical improvement across the entire platform system. A platform sponsor nevertheless has a great deal of latitude in deciding which specific activities to bring within its own zone of authority and how much integration and interdependency to adopt in its own products. It

[58] *United States vs. Microsoft: Trial Summaries..*

[59] Tedlow p. 316.

[60] Ibid. p. 315.

can also deal with its complementors in various ways: competing with them in some arenas, collaborating in others, and coercing them when the stakes are sufficiently high.

In this chapter, we saw that Intel integrated the microprocessor and chip set, but kept the motherboard and other hardware as separate modules. It competed in the chip set and motherboard markets, displacing Chips and Technologies from its position as market leader. It collaborated with other large, influential firms in standard-setting organizations to get the PCI and USB bus standards adopted.

Microsoft used a similar set of tactics in the software arena. It competed with makers of productivity software, and displaced the market leaders with its integrated Office suite of applications. It also competed in the browser market against Netscape. In this encounter, it used coercive measures to prevent OEMs from removing the Internet Explorer icon from the desktop screen or changing the startup sequence of a Windows PC. And it was quick to discourage Intel from providing developers with the means of distintermediating the operating system.

In contrast to Intel, which worked closely with many standard-setting organizations, Microsoft did not go out of its way to collaborate with other firms. It also strongly opposed open source communities.[61] It did however support complementary software developers by creating tools including Visual Basic (an easy-to-learn programming language and environment used to create Windows applications), OLE (a set of APIs allowing objects to be transferred and linked across documents), and COM (a more general embedding and linking technology).

Intel's approach to buses and Microsoft's to "killer" applications were not too different from the methods of systematic management developed for flow-based production at the turn of the 20th Century. Intel "solved" the problem of buses by creating an integrated system made up of the processor, chipset, buses, and motherboard. It drew on its existing expertise in the underlying step processes used to design and manufacture both chips and systems. For its part, Microsoft created an integrated set of software applications, including the operating system, brower, and productivity applications. Internally both companies were classic corporations with unified governance, exercising direct authority within formal managerial hierarchies.

However, unlike the flow-oriented companies in the early 20th Century, Intel and Microsoft were consciously and strategically part of an *open* platform ecosystem. If the number of new things a user could do with more powerful PCs stopped growing, then demand for new processors and larger operating systems would flatten out and possibly decline. *The new options created by complementors in the open system were thus critical to the sponsors' continued prosperity.*

One important group of complementors were the companies that designed, manufactured and delivered whole computer systems to end users. After the cloning of

[61] Halloween Documents.

the BIOS, none of these companies controlled system-level visible information. However, each created a subsidiary logistical platform that integrated the many step processes needed to create a fully functional computer. For many years, the most successful systems integrator was Dell Computer Corporation. The design of Dell's logistical platform is the focus of the next chapter.

References

"Microsoft Buys Software Unit," *The New York Times* (July 31, 1987); http://www.nytimes.com/1987/07/31/business/company-news-microsoft-buys-software-unit.html (accessed 3/5/18).

"DOJ, Microsoft Settle," *CNNMoney* (November 2, 2001) https://money.cnn.com/2001/11/02/technology/microsoft/ (viewed 10/21/19).

"Intel, Microsoft Reach Truce on Native Signal Processing," *CBR;* (November 24, 1998) https://www.cbronline.com/news/intel_microsoft_reach_truce_on_native_signal_processing (viewed 3/30/18).

"Office for Windows Bundles Popular Microsoft Applications," *Infoworld* (October 1, 1990) https://books.google.com/books?id=VTwEAAAAMBAJ&pg=PT17#v=onepage&q&f=false (viewed 3/5/18).

Apple Computer, Inc. v. Microsoft Corp., 35 F.3d 1435 (9th Cir. 1994);

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

Benschop, Lennart (2011) "A Brief History of the PC," https://lennartb.home.xs4all.nl/coreboot/col2.html (viewed 10/26/19).

Casadesus-Masanell, R., Yoffie, D. and Mattu S. (2010) "Intel Corporation: 1968-2003," Harvard Business School Publishing 9-703-427.

Chiapetta, M. (2018) "All of AMD's 7nm Processors to be Manufactured at TSMC" *Forbes* (Aug 27, 1018: https://www.forbes.com/sites/marcochiappetta/2018/08/27/all-of-amds-7nm-processors-to-be-manufactured-at-tsmc/#4b1cd1c46423 (viewed 10/17/)

CNN (1998) "Behind the Numbers: Browser Market Share," (October 8, 1998) http://www.cnn.com/TECH/computing/9810/08/browser.idg/ (viewed 3/7/18).

Copeland, R. (1989) "Chip Makes Cheaper AT Clones Possible," *Info World* (October 9, 1989) https://books.google.com/books?id=pjAEAAAAMBAJ&pg=PA1#v=onepage&q&f=false (viewed 2/23/18)

Crothers, B. (1994) "Intel PCI bus benefits appear to outweigh drawbacks," *InfoWorld* (October 3, 1994) pp. 36; available at https://books.google.com/books?id=aDgEAAAAMBAJ&pg=PA36#v=onepage&q&f=false (viewed 10/14/19).

Cunningham, A. (2014) "A brief history of USB, what it replaced, and what has failed to replace it," https://arstechnica.com/gadgets/2014/08/a-brief-history-of-usb-what-it-replaced-and-what-has-failed-to-replace-it/2/ (viewed 2/21/18).

Cusumano, Michael A. and Richard W. Selby (1995) *Microsoft Secrets*, New York: Free Press.

Cusumano, M. A. and D. B. Yoffie (1998) *Competing on Internet Time* (New York: Free Press)

Dosi, G. (1982). Technological paradigms and technological trajectories: a suggested interpretation of the determinants and directions of technical change. *Research Policy*, *11*(3), 147-162.

Eisenmann, T., Parker, G., & Van Alstyne, M. (2011). Platform envelopment. *Strategic Management Journal*, *32*(12), 1270-1285.

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.

Final Judgment, *United States v. Microsoft Corporation*, Civil Action No. 98-1232 (November 12, 2002) https://www.webcitation.org/query?id=1298667420478033 (viewed 3/12/18).

Findings of Fact, *United States v. Microsoft Corporation*, 253 F.3d 34 (D.C. Cir. 2001) p. 81; https://www.justice.gov/atr/us-v-microsoft-courts-findings-fact#vf (viewed 3/11/18)

Gates (1995) "The Internet Tidal Wave, http://www.lettersofnote.com/2011/07/internet-tidal-wave.html (viewed 3/7/18);

Gawer, Annabelle and Michael A. Cusumano (2002) *Platform Leadership: How Intel, Microsoft and Cisco Drive Industry Innovation*, Harvard Business School Press, Boston, MA.

Halloween Documents; http://www.catb.org/esr/halloween/index.html (viewed 3/13/18).

Johnston, S.J. (1990)"The Microsoft Office Bundles 4 Programs," *Infoworld* (June 19, 1989) https://books.google.com/books?id=lzAEAAAAMBAJ&pg=PA17#v=onepage&q&f=false (viewed 3/5/18).

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

Intel Timeline 1980-1989, *Intel Vintage*, http://www.intel-vintage.com/timeline19801989.htm (viewed 2/23/17).

Kirkpatrick, D. (1994) "Why Compaq is mad at Intel; Compaq CEO Eckhard Pfeiffer's public attack on his top supplier is just the start of a war." *Fortune* (Oct 31, 1994) https://archive.fortune.com/magazines/fortune/fortune_archive/1994/10/31/79893/index.htm (viewed 10/17/19).

Kwak, M. and D.B. Yoffie (1998) "The Browser Wars, 1994-1998," HBS Case 9-798-094 (Rev. Jan. 22, 2001),

Lea, G. (1999) "DOJ's 'Punch' Beats MS' 'Judy': The Felten Deposition," *The Register* https://www.theregister.co.uk/1999/06/01/dojs_punch_beats_ms_judy/ (viewed 3/11/18).

Lewis, P. H. "Introducing the First PS/2 Clones," *The New York Times* (April 24, 1988) https://www.nytimes.com/1988/04/24/business/the-executive-computer-introducing-the-first-ps-2-clones.html (viewed 10/26/19).

Liebowitz, S., & Margolis, S. (2001). Network effects and the Microsoft case. *Dynamic Competition and Public Policy: Technology, Innovation, and Antitrust Issues*, (J. Ellig, ed.) Cambridge University Press. https://personal.utdallas.edu/~liebowit/netwext/ellig%20paper/ellig.htm (viewed 10/14/19);

Mackie-Mason, J.K. and J.S. Netz (2007) "Manipulating Interface Standards as an Anti-competitive Strategy," in *Standards and Public Policy* (S. Greenstein and V. Stango, eds.) Cambridge University Press: Cambridge UK; pp. 231-259.

Manners, D. (2010) "Top 25 Fabless Companies," *Electronics Weekly* (Jan 19, 2010): https://www.electronicsweekly.com/news/business/manufacturing/top-25-fabless-companies-2010-01/ (viewed 10/17/19);

MarketResearch.com (2001) "Acer, Inc. – A Competitive Assessment Report," pp. 39 - 42. http://www.marketresearch.com/Vendors/SampleViewer/unlock.ncp?vendorid=1039 (viewed 2/23/17).

McLellan, P. (2017) "Chips and Technologies: The First Fabless Company," https://community.cadence.com/cadence_blogs_8/b/breakfast-bytes/posts/candt (viewed 10/26/19).

Mueller, Scott (2005), *Upgrading and Repairing PCs*, 17th ed., Indianapolis: Que Publishing.

Raikes, J. (2010) Interview, available at https://www.webcitation.org/6Yp0CXRBx?url=https://filetea.me/t1s6cpDUdxcROysEP9aBVOGHQ (viewed 10/14/19).

Randall, N. (1998) "So What's an Operating System, Anyway?" *PC Magazine* (May 5, 1998) p. 299. https://books.google.com/books?id=Law6OJQT1sQC (viewed 3/11/18).

Rosoff, M. (2011) "Worst Miss Ever? Microsoft Tried To Buy Netscape in 1994," *Business Insider* (October 28, 2011) http://www.businessinsider.com/worst-miss-ever-microsoft-tried-to-buy-netscape-in-1994-2011-10 (viewed 3/7/18).

Tedlow, R. S. (2007). *Andy Grove: The life and times of an American business icon*. Penguin.

U.S. vs. Microsoft Timeline (2002) *Wired*; https://www.wired.com/2002/11/u-s-v-microsoft-timeline/ (viewed 3/11/18).

United States vs. Microsoft: Trial Summaries; https://cyber.harvard.edu/msdoj/transcript/summaries2.html (viewed 3/30/18).

Valentine, D. A. (1999) "Abuse of Dominance in Relation to Intellectual Property: U.S. Perspectives and the Intel Cases," Paper presented at the Israel International Antitrust Conference, Tel Aviv, Israel, November 15, 1999; Available at https://www.ftc.gov/public-statements/1999/11/abuse-dominance-relation-intellectual-property-us-perspectives-and-intel (viewed 3/1/17).

Valentine, D. A. (1999) "Abuse of Dominance in Relation to Intellectual Property: U.S. Perspectives and the Intel Cases," Paper presented at the Israel International Antitrust Conference, Tel Aviv, Israel, November 15, 1999; Available at https://www.ftc.gov/public-statements/1999/11/abuse-dominance-relation-intellectual-property-us-perspectives-and-intel (viewed 3/1/17).

Venditto, G. (1989) "First EISA Chips Delivered," PC Magazine (September 26, 1989) https://www.nytimes.com/1988/04/24/business/the-executive-computer-introducing-the-first-ps-2-clones.html (viewed 10/26/19).

Vilches, J. (2009) "Intel delays USB 3.0 support until 2011?" https://www.techspot.com/news/36673-intel-delays-usb-30-support-until-2011.html (viewed 2/21/18);

Xerox Corp. v. Apple Computer, Inc., 734 F. Supp. 1542 (N.D. Cal. 1990).

Ziff Davis Market Intelligence (September 1998). "The 800-Pound Gorilla of the Presentation Market". *Mobile Computing and Communications* [later, *Mobile Office*]. 9 (9): 95; archived at https://www.webcitation.org/6bxj2eryp?url=https://filetea.me/t1sEVBHlotISPCAVUKpeg2F5A (viewed 10/21/19).