경영학 석사학위논문

Platform Leadership
in the Era of Rising Complexity

2018년 8월

서울대학교 대학원

경영학과 경영학전공

정 혁

ABSTRACT

Platform leadership has drawn substantial attention not only from high technology industries, but also from conventional industries. We examine two cases of platform leadership in semiconductors to shed light on the questions of why some platform leaders do better than others, and why existing platform leaders sometimes fail. As the number of transistors on a chip doubles every two years for more than a half-century, semiconductor companies face rising complexity in designing their chips. When the number of transistors on a chip increased into the tens of millions in the 1990s, chip design process had become extremely difficult. Our study shows that in the face of the rising complexity, successful platform leaders figured out what to overlook, while focusing on what they could do best. They built their ecosystems based on their strengths and encouraged other firms to do business by embracing their ecosystems. Winners were those who understood that the increasing complexity was far beyond the reach of what a single company could manage. Attracting competent allies and allowing them to prosper was a sustainable solution to the ever-increasing complexity in the semiconductor industry. Those who were blinded to or resisted this sea change were driven out.

Keywords: platform leadership, platform, complexity, semiconductor, ARM Holdings, Intel

Student number: 2015-20669

TABLE OF CONTENTS

I. INTRODUCTION

Recently, many high technology industries, including smartphones and servers, have become battlegrounds for platforms (Iansiti & Levien, 2004; Gawer & Cusumano, 2008). It is increasingly becoming more important to understand platform strategy as artificial intelligence (A.I.) and autonomous driving are expected to expand rapidly in the near future. Platform leadership has drawn substantial attention not only from high technology industries, but also from conventional industries. However, the issue has remained less addressed by systematic research, and a gap still remains in the literature for sorting out complex phenomena associated with platforms (Gawer & Cusumano, 2008). In particular, Gawer and Cusumano (2014) has pointed that prior literature "has delivered only limited insight into why such platforms come into existence in the first place." (Gawer & Cusumano, 2014: p. 422). To fill in this gap, we pose the following questions: *why do some platform leaders do better than others, and why do existing platform leaders sometimes fail?*

To answer these questions, we consider high-tech industries, where firms survive by upgrading or adding new features or functionalities to their products or services. For example, in the smartphone industry, Apple, Qualcomm, and Samsung compete by adding new functionalities to their chips. Building on Gawer & Cusumano's (2014) definition, we consider platforms as "products, services, or technologies developed by one or more firms, and which serve as foundations upon which a larger number of firms can build further complementary innovations" (Gawer & Cusumano, 2014: p. 420). In this framework, a platform leader acts as "a catalyst for innovation in the industry." (Gawer & Cusumano, 2014: p. 424). In this paper, we aim to examine why Intel, the world's leading platform leader in the 1990s, missed the golden opportunity in smartphones, while ARM Holdings, a small unknown firm, was able to become a dominant platform leader in the mobile application processor (AP) market.

At the dawn of the smartphone industry, Intel had its chance to establish hegemony in the mobile AP market. Qualcomm and Apple asked Intel to produce chips for their smartphone business. However, Intel killed this golden opportunity by declining their requests. Its first chance was through Qualcomm, as Intel had provided processors for Qualcomm until 1997. However, Intel refused to provide a roadmap update for Qualcomm, seeing risk in both yield and design (Dingee & Nenni, 2015). Intel also missed its second chance in 2007 when Apple asked Intel to design and manufacture chips for the iPhone. Intel refused Apple's request because it was skeptical of

Apple’s volume projections to achieve economies of scale (Madrigal, 2013). Intel’s decision left the company locked out of the exploding smartphone market, and the once-invincible semiconductor giant lost its position in this market. On the other hand, ARM seized the opportunity and became the platform leader.

To understand why Intel missed its golden opportunity and why ARM triumphed in the mobile AP market, we need to comprehend the transformation in the semiconductor industry. The essence of the industry transformation was associated with the rising complexity in chip development due to Moore’s Law, which predicts that the number of transistors on a chip doubles every two years (Moore, 1965). In the 1990s, the number of transistors on a chip increased into the tens of millions. As such, the cost and the period required to develop a chip rose dramatically. In particular, errors in chip design increased exponentially as the industry followed Moore’s Law for decades. Furthermore, every time a new manufacturing process technology was released, chipmakers needed to upgrade their chip designs to benefit from this process innovation. However, the daunting challenge associated with dramatically increasing errors made it extremely difficult for chipmakers to validate new designs. Some of industry experts reached the conclusion that the whole design process could no longer be managed with the existing methods.

On the other hand, Intel did not pay much attention to this emerging view. In the past, Intel used to design and manufacture a few standardized chips all by itself in the PC industry. This is known as the “we-built-it-and-you’ll-like-it” approach (Dingee & Nenni, 2015). Intel maintained its conventional approach in the smartphone business as well. In light of this conventional way of doing things, we could glimpse why Intel might have rejected the requests by Qualcomm and Apple. Given its own version of the standardized architecture Intel developed for the mobile AP market, Intel might had to redesign the new chips substantially to meet the idiosyncratic needs of Qualcomm and Apple. Serving the customers’ needs might not have justified the rise in chip development cost. In this context, Intel’s refusals to the prominent customers’ requests were, perhaps, not unreasonable. However, Intel was blinded to a major transformation in the industry.

To manage the enormous complexity described above, the semiconductor industry witnessed the chip design business being divided into two types of semiconductor firms: some specialized in developing functional blocks for their design libraries, whereas others developed chips by building on these libraries. By promptly embracing this burgeoning change, ARM disclosed its design libraries and encouraged other firms to develop their own chips by building on and modifying some components of the ARM libraries. In particular, ARM offered a radically new approach, which was called the "build-whatever-you-want" approach. That is, ARM provided reusable error-free components and upgraded them every time a new manufacturing process technology was released. By building on ARM's design libraries, chipmakers in the mobile AP market were free from the burden of redesigning and validating components in the face of the rising complexity. By producing ARM-based chips, other chipmakers started to earn money and in return, their successes also made ARM prosper as well. As a result, the ARM architecture became popular in the mobile AP market, eventually becoming the dominant platform.

Specializing in providing reusable and error-free components for other chipmakers, ARM successfully managed to reduce the increasing complexity in chip development, not only for itself but also for other chipmakers. ARM flourished as a platform leader by "offering canvas and paint through which designers large and small can express their vision of how things with a chip inside should work – the very essence of what makers do" (Dingee & Nenni, 2015: p.217).

Our historical analysis of ARM Holdings and Intel sheds new light on the research on complexity. Assuming that the complexity is given, much of prior work has investigated how the complexity affects organizational adaptations (Ethiraj & Levinthal, 2004 a,b; Ethiraj, Levinthal, & Roy, 2008; Levinthal, 1997; Levinthal & Posen, 2007; Zhou, 2013). However, little is known about how the complexity actually arises in the first place. In the semiconductor industry, the complexity in chip design has originated from Moore's Law. Following Moore's Law meant that chip companies had to cram more and more transistors into a chip exponentially over time, increasing the likelihood of adding errors in designing integrated circuits. The increasing likelihood of errors provided daunting challenges to chip designers in the semiconductor industry. Managing the rising complexity was a crucial factor that determined the winner and the loser in the semiconductor industry. Intel's failure to manage the complexity led Intel to miss its golden opportunity to establish hegemony in the smartphone business. On the other hand, ARM Holdings successfully coped with the rising complexity by embracing the burgeoning industry transformation, the division of labor between development of design libraries and chip development, and became the winner in the mobile market.

II. PLATFORM STRATEGY IN AN ECOSYSTEM

2.1. What is a Platform?

A platform can be defined as a foundation technology or service upon which other firms can build further complementary innovations (Gawer & Cusumano, 2008; 2014). In the platform strategy, it is of paramount importance to build an ecosystem, which consists of a platform leader and complementors. Complementors are firms which use the platform leader's technology for their own businesses. In the context of the mobile chip market, a platform is characterized by the functional blocks, or components, in design libraries, which are used for mobile chip design. For example, baseband and DSP functional blocks are required to make phone calls. In addition, functional blocks for graphics processing unit (GPU) are used to watch YouTube or play high graphic games such as Angry Birds.

--------------------- *Insert Figure 1 here* ----------------------

----

An ecosystem in the mobile chip market is composed of platform leaders, such as ARM Holdings, and complementors, such as Apple and Qualcomm. The latter use some of ARM's functional blocks or modify them to design their new chips.

To understand why some platform leaders do better than others, and why existing platform leaders sometimes fail, we first need to understand how the platform strategy in an ecosystem differ from Porter's competitive strategy in the traditional supply chain.

2.2. Complementary Relationship in Platform Strategy

In Porter's competitive strategy, the very essence of strategy formulation is to cope with competition (Porter, 1979). Porter's strategy emphasizes strategic maneuvering in a competitive relationship. In order to increase a firm's profit, the firm has to jockey for its position and increase its bargaining power over other players in the traditional supply chain (Porter, 1979). In such a win-lose situation, the firm has to lower its dependency on others to pass on cost and squeeze profitability out of an industry (Porter, 2008).

Unlike Porter's competitive strategy, the key success factor in the platform strategy is to increase the potential size of the pie for everyone

(Cusumano & Gawer, 2002). The success of platform leadership depends on building a viable ecosystem, which is characterized by complementary relationship. That is, a platform will attract customers only when there are a myriad of complementors. In high-tech industries, where innovation is the name of the game in survival, complementors use the platform leader's technology for their own businesses. Thus, a platform leader must act as "a catalyst for innovation" to facilitate complementors' innovation. In such a win-win situation, increasing rather than decreasing dependence is a recipe for success. Based on the distinct characteristics of the platform strategy, we now analyze the cases of ARM Holdings and Intel in the semiconductor industry.

III. THE RISE OF ARM HOLDINGS IN THE MOBIEL AP MARKET

How did David, *ARM Holdings*, beat Goliath, *Intel*, and became the winner in the mobile AP market? Intel has been the dominant platform leader in the semiconductor industry for a long time. In 2017, Intel had a 95% market share in the PC market and a 99% market share in the server market.[1] However, ARM Holdings has defeated Intel in the mobile AP market, with ARM-based chips dominating the whole market. Intel retreated from the market, recording a loss of $5 billion from its mobile investment during the year 2015 alone.[2] To understand how Intel failed so badly, we first investigate the rise of complexity in chip development.

3.1. Implications of Moore's Law for Platform Strategy

The semiconductor industry has been following the well-known Moore's Law since 1965, when Gordon Moore predicted that the number of transistors on a chip would double approximately every two years. For example, a typical chip in 1965 contained 64 transistors. In 2017, Intel released a new microprocessor, Broadwell-EP Xeon E5-2600 V4, containing about 7.2 billion transistors inside a 456mm^2 die.[3]

--------------------- *Insert Figure 2 here* ----------------------

----

[1] DRAMeXchange report (2017)

[2] Intel 10-K report (2016)

[3] http://wccftech.com/intel-broadwell-ep-xeon-e5-v4/

Following Moore's Law implies that chip performance increases exponentially over time (Kang, 2010). Based on his observation, Gordon Moore predicted that "Integrated circuits will lead to such wonders as home computers, automatic controls for automobiles, and personal portable communications equipment" (Moore, 1965: p. 114). Quite surprisingly, most of his predictions have come true, just with different names: PCs, autonomous driving, and smartphones respectively.

Complexity in chip manufacturing process

Following Moore's Law also implies that the chip development cost increases exponentially. To achieve the exponential growth in transistor counts on a chip, semiconductor companies had to shrink the transistor size. To manufacture a smaller transistor, they had to upgrade the manufacturing process technology. Developing sophisticated process technology increased complexity in the chip manufacturing process. The intensified complexity increased the cost in manufacturing chips. For instance, the cost of building a new manufacturing facility (commonly called 'fab') increased exponentially by following the development of the process technology. In 1966, the cost of building a new fab started from $ 14 million, and rose to $ 1.5 billion in 1995.[4] In 2013, $10 billion dollars was required to build a state-of-the-art fab (Nenni & McLellan, 2013). Now, building a new fab costs a couple of times as much as a nuclear power station. (McLellan, 2010).

--------------------- *Insert Figure 3 here* ----------------------

----

Until the 1980s, integrated device manufacturers (IDMs), such as Intel, Texas Instruments, and IBM, carried out the whole chip development process, from chip design to manufacturing and the final commercialization, inside the firm boundary. However, the increasing cost in the chip manufacturing process became highly burdensome for IDMs and triggered the birth of a new business model for specializing in manufacturing chips for other fabless semiconductor companies, which design chips without having their own manufacturing facilities (Iansiti & Strojwas, 2003). With the emergence of these manufacturing services, hundreds of fabless firms entered the market and solely concentrated on designing chips (Hobday, 1991).

[4] http://jimgray.azurewebsites.net/moore_law.html

--------------------- *Insert Figure 4 here* ----------------------

----

Complexity in chip design process

Chipmakers faced another challenge, namely the rising complexity in chip design process. In designing a chip, it was essential for a semiconductor company to eliminate any errors, because even a small error in chip design meant that the whole wafer mask had to be thrown away, resulting in a $1 million loss for a single mask (Nenni & Mclellan, 2013). Every time a new manufacturing process technology was released, chipmakers needed to upgrade their chip designs to benefit from this process innovation. However, the exponentially increasing number of transistors on a chip meant that the likelihood of errors in chip design also increased exponentially. To check errors in chip design, simulations in multiple levels (e.g. gate-level simulation, post simulation, etc.) were carried out.

--------------------- *Insert Figure 5 here* ----------------------

----

Rising complexity in chip design after the 1990s

In the 1990s, the number of transistors on a chip increased into the tens of millions. As the number of transistor on a chip increases, the likelihood of errors in a chip design increases exponentially. Hence, it was no longer possible to simulate every nook and cranny of the integrated circuits with simulation techniques (Shih, Shih, & Chien, 2009). Moreover, chip design process was delayed for many years, or it may fail eventually with the existing approaches. Soon, chipmakers realized that most of their limited resources were spent on validating new designs, barely keeping up with the pace of the process innovation (Collet & Pyle, 2013). Exponential increases in errors made it extremely difficult for a single company to manage the whole design process alone. Some industry participants began to realize that the whole design process could no longer be managed with the existing methods.

3.2. The Industry Transformation in Response to the Rising Complexity

The separation between suppliers of design libraries and chip designers

To manage enormous complexity in a chip design process, there was a major industry transformation in response to the rising complexity. If reliable and tested functional blocks could be reused, the whole design process could be simplified (Shih et al., 2009). Before the 1990s, chip designers had to design chips from scratch every time a new manufacturing technology was released. In the 1990s, the industry witnessed division of labor in the chip design process. Some suppliers of design libraries began to provide error-free functional blocks which were reusable for chip designers. By relying on these error-free functional blocks, chip designers could develop their own chips much faster without worrying about the ever-increasing errors. Specializing in developing and licensing the design libraries, ARM Holdings quickly seized the opportunity from this major industry transformation.

--------------------- *Insert Figure 6 here* ----------------------

----

3.3. Intel vs. ARM: Different Responses to the Industry Transformation

Intel's failure in the mobile market

Why did Intel fail to seize the opportunity in the mobile market? Intel failed in the mobile market because it seemed to be blinded to the major industry transformation, which led to the inevitability of division of labor between developing design libraries and designing chips. Even when complexity in chip design was high, Intel maintained its conventional approach in the PC industry. Renee James, former Intel president, explains Intel's conventional approach as follows:

> At the end of the day our core strategy has remained unchanged for the last 47 years. Be the number one semiconductor company, lead in process technology, achieve economies of scale. (Burgelman & Schifrin, 2015: p. 1)

Intel developed its conventional approach when the complexity in chip design was relatively low. In the PC industry, Intel used to design and manufacture a few standardized chips all by itself. Then, Intel closely guarded its top secret, design libraries for its chips. Intel tried to lower its dependency toward others to monopolize its profit by internalizing the entire chip development process. In the design part, Intel had to squeeze in exponentially more transistors on a chip to follow the Moore's Law. In the manufacturing part, Intel had to build a state-of-the-art fab to reduce chip sizes and to improve chip performance. Failure in either of the design part or the manufacturing part would lead to a major delay or failure in the chip development.

---------------------- *Insert Figure 7 here* -----------------------

----

When customers, such as Qualcomm and Apple, asked Intel to develop chips that were incompatible with Intel's ATOM architecture, Intel might have to redesign the new chips, which might not have justified the rise in chip development cost, both in the manufacturing and design process. Paul Otellini, former Intel CEO, explained Intel's decision to refuse Apple's request:

> The thing you have to remember is that this was before the iPhone was introduced and no one knew what the iPhone would do… At the end of the day, there was a chip [Apple was] interested in that they wanted to pay a certain price for and not a nickel more, and that price was below our forecasted cost. I couldn't see it. It wasn't one of these things you can make up on volume. (Madrigal, 2013: p. 4)

In this context, Intel's reactions to the prominent customers' requests were, perhaps, not unreasonable.

ARM's success in the mobile market

How, then, was ARM able to succeed in the mobile AP market? Unlike Intel, ARM Holdings more effectively adapted to the sea change in the mobile market by embracing the industry transformation associated with the separation between suppliers of design libraries and chip designers. ARM CEO Simon Segars described how ARM responded to the increasing complexity in chip development:

> With ARM designing the architecture once and licensing many times, ARM is able to cover its own R&D costs and also reduce the cost for each semiconductor company. ARM's partners are then able to invest in the complementary technologies that go into a System-on-Chip. This lead to more choice in digital electronics for OEMs and consumers. (ARM 2014 Strategic Report: p. 21)

In response to the rising complexity in chip design, ARM positioned itself as a supplier of design libraries and disclosed its design libraries, which were most famous for the low-power, high-efficiency microprocessors. More specifically, ARM specialized in providing reusable, error-free functional blocks and upgrading them every time a new manufacturing process technology was released. Then ARM encouraged other chipmakers to develop diverse kinds of chips by building on and modifying some components of the ARM libraries. Chip design became just like playing with Lego blocks. ARM provided basic Lego blocks to play with, and chipmakers built on those blocks to develop new chips. If chipmakers did not like some parts of the basic building blocks in the ARM libraries, chipmakers were certainly allowed to modify them.

Attracting Complementors to the ARM platform

Initially, ARM Holdings was not a successful company.[5] Before the 1990s, ARM Holdings failed in almost every business it was involved in (Dingee & Nenni, 2015). People even joked about ARM as follows: "ARM design team had two distinct advantages over others – no money and no people" (Nenni & Mclellan, 2013: p. 164). Until the 1990s, chips were mostly used in PCs or servers, where high performance was required. The lack of resources for chip development made ARM include only essential elements for its chip. As a consequence, ARM chips consumed extremely low power. Ironically, this minimalism was what attracted Nokia to the ARM platform at

[5] ARM Holdings was structured as a joint venture between Acorn Computers, Apple Computer (now Apple Inc.) and VLSI Technology in 1990. It is currently owned by SoftBank Group.

the dawn of the mobile market. Back in the 1990s, battery life was critical for a mobile phone; high computing power was of no use if its battery lasted for only an hour (Kang, 2010). Instead of high computing power, Nokia was looking for a chip with low power consumption for its mobile phone. Although Texas Instruments developed its own chip for Nokia, Texas Instruments abandoned it and switched to the ARM architecture to satisfy Nokia's need. As Nokia, the top phone vendor in the 1990s, became a complementor to the ARM platform, the ARM platform started drawing attention and subsequently grew rapidly.

--------------------- *Insert Figure 8 here* -----------------------

----

At the dawn of the smartphone industry, prominent participants in the ARM ecosystem included Qualcomm and Apple. Until 1997, Qualcomm used Intel 80C186 processor for its phone business. However, Intel refused to accommodate Qualcomm's requests on the radio frequency quadrupole function. As a consequence, Qualcomm had little choice but to choose the ARM platform. In 1998, Qualcomm released its next chip, MSM3000, based on the ARM platform. Another prominent customer, Apple, came to Intel to design and manufacture chips for the iPhone. As mentioned before, Intel was skeptical about Apple's volume projections and refused Apple's request. Apple also turned to the ARM platform in designing its own chips for the first iPhone and iPod in 2007.

Complementors in the ARM ecosystem also included small companies. Because ARM provided basic components that could be incorporated into chip designs, even small companies in China, such as Rockchip and Allwinner, were able to develop their chips by building on components from ARM's design libraries. ARM was able to quickly attract many complementors because it completely disclosed all the functional blocks in its design libraries. In the midst of rising complexity, however, Intel top managers were proud of maintaining its half-century old core strategy, which include closely guarding the top secret, the Intel design libraries.

Building the ecosystem around the ARM architecture

ARM continued to expand its ecosystem by providing more functional blocks which could be used in a chip design, so that chipmakers could more easily build on the ARM architecture. For example, ARM acquired Artisan Components that provided functional blocks for memory compilers and interface components in 2004. In addition, ARM released Mali graphics processing unit architecture in 2008. Components of ARM's design libraries now includes processor, graphics and multimedia, physical components such as embedded memory, interface, development tools for applications, and even security service.

With ARM covering the development cost for design libraries, even small firms could afford to develop chips. Thus, hundreds of chipmakers participated in the ARM ecosystem, providing various ARM-based chips in the market.

Intel's utter defeat in the mobile AP market

With the mobile device market exploding day by day, it was hard for Intel to ignore the fast-growing market. Intel decided to enter the mobile AP market, only to see a failure in building an alternative platform against the ARM platform. In 2015, Intel released its new ATOM chips – "SoFIA" for low-cost devices and "Broxton" for high-end devices – in the Chinese tablet market. Intel invested in Chinese chip manufacturers, such as Rockchip and Spreadtrum, to promote ATOM chips for the Chinese tablet market. Intel also asked many small Chinese firms to switch from ARM-based chips to Intel chips.

However, most Chinese smartphones and tablets had already adopted the ARM platform. To switch to the Intel platform, Chinese chipmakers had to readjust compilers, operating systems, as well as many other interfaces that surrounded the ARM architecture (Nenni & McLellan, 2013). Chinese chip manufacturers could not afford all the switching costs for using the Intel platform. To persuade them to use its chips, Intel had to subsidize a huge amount of capital to chip manufacturers.

As a result, Intel lost about $5 billion in its mobile investment by January 2015. Even with its enormous investment including the subsidization, Intel had virtually no presence in the smartphone market. In April 2016, Intel officially announced that Intel cancelled the development of new platforms for phones and tablets, practically exiting itself from the mobile market.[6]

ARM current status as a platform leader

A small British firm that started as a team of 12 people has now become a dominant platform leader in the smartphone industry, with over 75 billion ARM based chips shipped around the world (15 billion chips in 2015). As of 2015, ARM has 1,348 active licenses and 425 firms are ARM processor

[6] http://www.anandtech.com/show/10288/intel-broxton-sofia-smartphone-socs-cancelled

licensees (Dingee & Nenni, 2015). ARM-based chips held a 95% market share in the mobile AP market. ARM yielded a revenue of 968.3 million euro (1,489 million dollar) with a net profit margin of 35.08% in 2015.[7]

IV. THE LITERATURE ON COMPLEXITY AND PLATFORM

4.1. Complexity Theory

The source of rising complexity

Research on complexity has studied how complexity affects organizational adaptations and innovative performance (Ethiraj & Levinthal, 2004 a,b; Ethiraj, Levinthal, & Roy, 2008; Levinthal, 1997; Levinthal & Posen, 2007; Zhou, 2013). However, the literature has paid less attention to where the complexity actually arises from, assuming that the complexity is given. Our historical analysis of the semiconductor industry sheds new light

[7] Net profit margin is particularly high, compared to Intel's 17.90%. This is because of the special business model of ARM that licenses design IP cores, which has an almost zero marginal cost.

on this issue.

For more than a half-century, the semiconductor industry has been following Moore's Law, which predicts that the number of transistors on a chip doubles every two years (Moore, 1965). Following Moore's Law meant firms had to bear the rising cost in chip development.

Why, then, did the industry follow Moore's Law? In the semiconductor industry, chip development process has evolved to include more and more functions to satisfy the insatiable needs of demanding customers. For example, Apple was able to differentiate its products and enjoy the largest profitability in the smartphone industry by continuously adding various functions to its chips: 1) a security function to protect customers' payment and biometric data, 2) a graphic processor unit to power graphics for intensive tasks like gaming, and 3) Neural Engine to improve A.I. and machine learning applications (Gurman, 2018). To add more functionalities on a chip, industry participants had to shrink the transistor size for higher energy efficiency. Smaller transistors allowed the participants to cram more and more transistors on an integrated circuit, leading to increased processing power. In this regard, Intel (2018) noted:

> The insight, known as Moore's Law, became the golden rule for the electronics industry, and a springboard for innovation. …

> Performance—aka power—and cost are two key drivers of technological development. … This development not only enhanced existing industries and increased productivity, but it has spawned whole new industries empowered by cheap and powerful computing.[8]

With the ever-increasing number of transistors as well as functionalities, the chip development process became more complex. For example, Paul McLellan, an industry expert, explained the rising complexity in chip development with the following analogy:

> Designing an integrated circuit is like designing the Boeing 787 except doing it in 12 months using a manufacturing technology that has never been used before. … expect it to take off first time, using engines that have never run before and flight surfaces that have never flown before. … Next Christmas the 797 will be required, even bigger and more complex. But it will need to fly first time too. (McLellan, 2010: p. 72)

The upshot is that, the complexity in the semiconductor industry arose from the fact that the industry had followed Moore's Law for more than

---

[8] Intel (2018), https://www.intel.com/content/www/us/en/silicon-innovations/moores-law-technology.html

a half-century.

"Complexity catastrophe" in the semiconductor industry

As mentioned above, the chip development process has evolved to include more functions. Multiple functions in a system mean that improving one function can hinder improving the performance of other functions. As the number of functions in a system increases, conflicts between different functions are more likely to occur. With the increasing number of conflicts between functions, delays in the chip development process were not unusual, and some development projects ended up in failure (McLellan, 2010; Nenni & McLellan, 2013).

Kauffman (1993, 1995) referred to this type of complexity as "complexity catastrophe". It occurs when optimal solutions to one part heavily conflict with optimal solutions to other parts in the overall system, resulting in exponentially slowing rate of improvement for innovation attempts (Kauffman, 1993). For example, Kauffman (1995) illustrates the complexity catastrophe in a supersonic transport as follows:

> Suppose we are designing a supersonic transport and have to place the fuel tanks somewhere, but also have to design strong

> but flexible wings to carry the load, install controls to the flight surfaces, place seating, hydraulics, and so forth. Optimal solutions to one part of the overall design problem conflict with optimal solutions to other parts of the overall design. (Kauffman, 1995: p. 179)

Much of prior work on complexity in the management field has focused on this form of complexity – as the level of interdependence between components in a complex system increases, it becomes extremely difficult for organizations to adapt or to improve performance through search activities or innovation (Ethiraj & Levinthal, 2004 a,b; Levinthal, 1997; Zhou, 2013).

For example, Ethiraj and Levinthal (2004b) illustrates problems inherent in Intel's development of Itanium chip. Intel started to develop Itanium chip since 1994. But the development was delayed for many years; eventually it took eight years to launch the new chip in 2002, resulting in a commercial failure. Intel originally aimed to develop the chip to be compatible with multiple operating environments, such as Linux, HP UX, NT, Monterey, together with multiple databases made by IBM, Microsoft, Sybase, and Oracle. Consequently, Itanium chip had to operate on various software applications that were optimized on different operating systems (Leamon & Hardymon, 2000; Restivo, 1999).

However, Intel soon found out that their goal of embracing various environments was too ambitious. The optimal chip design for one operating environment conflicted with optimal designs to other environments, resulting in incompatibilities as well as much lower overall performance. As Ethiraj and Levinthal (2004b) noted, it is extremely difficult to increase the overall performance when there are numerous interactions between different parts of the system. Intel's ambitious goal resulted in a chaotic experience in developing its Itanium chip:

> The team found itself in a nightmarish world where a change to one module would ripple through the work of several hundred other people, leaving more problems in its wake. If engineers couldn't balance their signals, the only solution would be to slow down the entire chip – unacceptable for what was supposed to be a groundbreaking design. By mid-1998, the problem had grown so serious that Intel announced the chip would be delayed at least six months beyond its planned late-1999 launch. (Hamilton, 2001: p. 4)

In addition to complexity catastrophe, our analysis of the semiconductor industry reveals that error catastrophe in chip design, which has been less known in the management literature, also had detrimental effects organizational adaptations. Error catastrophe refers to dysfunctional effects of high mutation rates on adaptation in evolutionary biology (Eigen & Schuster, 1977). Kauffman (1995: p. 185) explains the term: "[A]ll accumulated useful traits (through natural selection) melt away at a high enough mutation rate." In the chip development process, an error catastrophe occurs when random errors in chip design result in malfunctioning of a chip. When designing a chip, eliminating any errors were essential, because even a small error in chip design meant the whole wafer mask had to be thrown away, resulting in $ 1 million loss just for a single mask (Dingee & Nenni, 2015).

In the 1990s, the number of transistors on a chip increased into the tens of millions as the industry follows Moore's Law. It was the critical point where the chip design became too complex. It was no longer possible to simulate every nook and cranny of the integrated circuits to eliminate the possibility of any errors (Shih et al., 2009). Because of the increasing errors in chip design, the chip design processes were frequently delayed for many years, and some of them failed (McLellan, 2010; Nenni & McLellan, 2013). The likelihood of increasing errors in chip design also led semiconductor companies to suffer from making improvements in innovative performance.

Managing the complexity in platform-based industries

The existing literature suggests how the complexity discussed above can be managed. Levinthal and March (1993) argue that organizations can manage complexity by decomposing tasks into subtasks and developing independent units for different subtasks. This is known as the "divide-and-conquer" approach. Siggelkow and Levinthal (2003) numerically showed that the choice of "divide-and-conquer" allows organizations to make better adaptations and to yield superior performance.

Our historical analysis of the semiconductor industry reveals that the adoption of the "divide-and-conquer" approach, indeed, decided who succeeded and who failed in the era of rising complexity. ARM Holdings became the winner by actively embracing the "divide-and-conquer" approach (i.e., the company was specialized in providing design libraries, while letting chipmakers develop their own chips by building on ARM's libraries). On the other hand, Intel became the loser as it ignored the inevitability of separation between suppliers of design libraries and chipmakers, sticking to its old approach in the PC industry, where Intel managed the entire development process in-house – from library development, chip design, to manufacturing.

4.2. Platform Literature

Research on platform has investigated various strategies for a platform to build its ecosystem effectively (Adner & Kapoor, 2010; Eisenmann, Parker, & Van Alstyne, 2009; Schilling, 2009; Kapoor, 2013; Kapoor & Lee, 2013). However, Gawer and Cusumano (2014: p. 422) noted that the platform literature "takes for granted the existence of the markets that transact through the platform." Given this taken-for-granted quality, Gawer and Cusumano (2014: p. 422) argued: "this literature has delivered only limited insight into why such platforms come into existence in the first place: the drivers of platform emergence and evolution." Our paper fills this gap by highlighting how the rising complexity in chip development process played the role of the driver of platform emergence in the semiconductor industry.

Had the number of transistors on a chip not exceeded the tens of millions in the 1990s, Intel might not have missed the opportunity in the mobile market. At the dawn of the smartphone industry, Intel possessed numerous advantages to lead the market. First of all, Intel had an unchallenged market position with a deep pocket as well as the most talented engineers. Intel also had the potential to achieve economies of scale in developing mobile chips by leveraging design and manufacturing capabilities from its PC business. Furthermore, Intel could leverage its huge installed base in the PC market to penetrate into the mobile market. Indeed, the most prominent complementors in the mobile market, Apple and Qualcomm, recognized Intel's unsurpassed advantages at time and wished to work with Intel.

Unfortunately, the fact was that the level of complexity at the dawn of the smartphone industry was beyond what a single company could manage. To fulfill the would-be complementors' requests, Intel had to upgrade its design libraries every time a new manufacturing process technology was released. At the same time, Intel had to add various functionalities that the would-be complementors demanded. As the cost in chip design soared along with the level of complexity, Intel could not risk developing chips with uncertain volume projections. Satisfying the would-be complementors at a reasonable cost was, in Paul Otellini's word, "not one of these things you can make up on volume" (Madrigal, 2013: p. 4). However, Intel's decision to decline Apple and Qualcomm's requests turned out to be tantamount to giving up the role of innovation catalyst in the burgeoning mobile market.

ARM Holdings was able to seize this opportunity by focusing on providing error-free design libraries, while encouraging its complementors to design their own chips based on ARM design libraries. Licensing ARM design libraries at a reasonable cost gave complementors full of choices to develop chips on their own by adding whatever they like. The “divide-and-conquer” approach that ARM Holdings embraced stimulated complementors to innovate by building on ARM platform. In sum, by offering “canvas and paint through which designers large and small can express their vision” (Dingee & Nenni, 2015: p.217), ARM Holdings has become the catalyst for the industry innovations, which is essential in platform leadership.

V. THE RISE OF INTEL IN THE SERVER PROCESSOR MARKET

As mentioned previously, the purpose of this paper is to shed some light on the question of why some platform leaders do better than others and why platform leaders sometime fail. To address this central question, we now examine how Intel established hegemony as a platform leader in the server industry.

How did Intel, a mere PC chip manufacturer, surpass traditional server vendors and became the champion in the server processor market? When Intel initially entered the server market in the 1990s, everyone laughed at Intel. With a low reputation and unproven quality, Intel had to start its business in the low-end server market, such as the mailing service server. Currently, Intel is the dominant ruler of the server processor market, commanding a 99% market share.[9] To understand how Intel became the platform leader in the server processor market, we first discuss the rising complexity in the server processor development.

5.1. Rising Complexity in Server Chip Development

Server machines performed complex operations, such as data-mining and Internet hosting, which required millions of transactions to be processed in a minute (Hardymon & Leamon, 2000). Therefore, server machines required chips with a much higher performance compared to that of PC chips. In addition, enterprises, the main customers of the servers, emphasized the reliability of the server. The use of unreliable servers could result in a fatal damage to enterprises' operations, as Ghemawat, Subirana, & Pham (2004) pointed out:

> [I]f the network crashes at Yankee Candle, the candles ship out

---

[9] https://www.idc.com/getdoc.jsp?containerId=prUS41419716

> a day late. If the network crashes at Goldman Sachs, the firm's quarter may be in jeopardy. (Ghemawat et al., 2004: p. 8)

To achieve high performance and reliability at the same time, server chip design evolved into an extremely complex system. It became extremely difficult to fully understand the magnitude of interactions between different components in a chip (Ethiraj & Levinthal, 2004b).

Intel's entry into the server processor market

Intel entered the server processor market in the 1990s. Because of its low reputation and unproven quality of its chips, Intel started its business in mailing service server or file servers. These servers were less mission-critical, in the sense that the low price of Intel's Xeon chips could compensate for the low functionality.[10] Intel's position in the market was very weak. Victor Na, Intel architecture manager in the server sector, explained Intel's position as follows:

> Everybody laughed when Intel announced its entry into the server market. How can a mere low-quality desktop chip

[10] From the interview with Victor Na, Intel architecture manager in the server sector (17.10.30)

producer make a high-quality server processor? It was compared to the battle of David and Goliath. Of course, Intel was David at the time.[11]

Intel did not develop server software applications that could operate on Intel's chips unlike other server vendors. Thus, there were few software applications that could properly operate on Intel chips. Since most software applications were not compatible with Intel's chip, using Intel's chip could result in serious errors, including server failure (Hardymon & Leamon, 2000). Without reputation on server chips, enterprises saw a huge risk in using Intel's chips. The unproven quality of Intel's processor may result in critical damage to firms, including attacks from hackers.

Traditional server vendors' response to the rising complexity

Traditional server vendors like Sun Microsystems developed in-house most core components of server machines, which ranged from server chips to operating systems, as well as applications, to maintain high performance and functionality. For instance, Sun Microsystems developed its

[11] Ibid.

SPARC server processor and Solaris operating systems.[12] Since the mid-1990s, most server vendors used 64-bit server chips, which showed much faster operation speed, compared to previous 32-bit chips (Zietsma, Mark, & Mitchell, 2004). However, advancing chip performance through increasing clock frequencies became extremely difficult and required enormous development costs (Geer, 2005).

Until 2000, increasing chip performance was achieved by increasing the clock rate, the frequency at which a chip was running. However, the clock rate could not be increased forever. Eventually the rate at which performance increased began to slow down (Geer, 2005). Moreover, developing operating systems and application software that operated on chips further exacerbated the problem (Ghemawat et al., 2004). The rising costs in both hardware and software dramatically increased the risk of developing new servers for traditional server vendors.

[12] from SPARC webpage (http://sparc.org/)

5.2. Three Major Steps of Intel in Managing the Rising Complexity

In contrast to the incumbents, Intel undertook three major steps in response to the increasing complexity in server chip development. First, Intel developed chips with multiple computing cores inside to carry out many operations simultaneously.[13] By spreading workloads over multiple cores in parallel, Intel achieved higher chip performance without improving clock frequencies.[14] Second, Intel realized the economies of scale in developing server chips by leveraging design and manufacturing capabilities from its PC business. Third, Intel embraced burgeoning Linux and Linux-based applications for its server software.

Intel's first step was to develop a multicore processor by applying parallel computing technology. After increasing clock frequencies of a chip encountered its limitation, chip manufacturers started to manufacture chips with multiple energy-efficient cores instead of one increasingly powerful core to increase performance (Geer, 2005). Parallel computing was a type of computation in which many calculations were carried out simultaneously by

---

[13] https://www.intel.com/content/www/us/en/architecture-and-technology/many-integrated-core/intel-many-integrated-core-architecture.html

[14] https://itpeernetwork.intel.com/are-you-realizing-the-payoff-of-parallel-processing/

connecting multiple cores. Victor Na explained how the parallel computing helped Intel:

> The development of the parallel computing technology also helped Intel. Parallel computing, or scale-out, means that handling operations in multiple low-performing cores can be faster than in one high-performing core. This meant that if one computer shuts down, the other computers can handle the task, which enormously lowered the mission-criticalness in the server market.[15]

Intel's second step was to realize the economies of scale in developing server chips by leveraging design and manufacturing capabilities from its PC business. By sharing core architecture with PC chips, Intel could design and manufacture server chips in a much lower cost. This led to lower prices compared to that of traditional server vendors. Victor Na described Intel's strategy as below:

> Intel manufactured processors for both desktops and servers. The core architecture was the same for both chips, so many

[15] From the interview with Victor Na, Intel architecture manager in the server sector (17.10.30)

parts of the manufacturing process was shared. Thus Intel could realize the economies of scale in producing server chips, which made Intel's chip price much cheaper than those of others, about the price of 1/10.[16]

Intel's third step was embracing Linux and open source development as compatible software with Intel's chips. In the server market, there were four main environments: mainframe, UNIX, Linux and Intel Architecture (Mark, Kernan, & Faber, 2004). Until the 1990s, UNIX-based servers, including that of Sun Microsystems, IBM, and HP had been a mainstay for running large databases and enterprise servers.[17] However, UNIX environments turned out to be inefficient because UNIX was often modified by server vendors. Incompatible software with similar solutions was developed, leading to considerable inefficiencies in the market (Moody, 2009).

In the 1990s, Linux started to attract thousands of developers. First released in 1991, Linux started to grow at an annual rate of 24% with an army of Linux developers worldwide. In 2000, Linux was installed on 27% of network server market (Burgleman & Meza, 2001). Software for server that

---

16 Ibid.

17 https://www.itworld.com/article/2785695/operating-systems/intel-or-unix--server-customers-face-a-tough-choice.html

had been previously developed solely by server vendors was now developed in open source. This included Apache, the most popular web server on the Internet since 1996, and MySQL, the most popular open source relational database management system (RDBMS).[18] For the purpose of source code redistribution, Linux was designed to work well on many types of hardware, enabling greater interoperability between Linux and other architectures. The rise of Linux was an excellent opportunity for Intel to gain ground in the server market. Intel decided to use Linux-based open source software, which could fill in the gap in complements for its processor.

To further enhance Linux-based software compatibility for Intel processor, Intel invested a large amount of capital for the development of Linux (Burgleman & Meza, 2001). Patrick Gelsinger, the chief technology officer of Intel Architecture Group, mentioned: "Intel has been committed to making sure that all important software runs on Intel architecture … Our job is to make sure that Linux runs best on Intel." (Burgelman & Meza, 2011: p. 16)

Intel also recruited Linux lieutenants. For example, in 2011, Intel hired Alan Cox, a top Linux kernel contributor who was often described as

---

[18] http://www.linfo.org/appslist.html

Linus Torvald's 'second in command'.[19] A 2012 report by the Linux Foundation listed Cox as the 18th most active individual contributor to the kernel, having submitted 1,703 code changes over a 5.5-year period.[20] Hiring Linux's main lieutenants meant that Intel could actively influence the development of Linux, optimizing its processors to operate perfectly on Linux-based software.

By 2000, big server vendors supported Linux. IBM, Compaq, Dell, and Hewlett-Packard all sold Linux preinstalled on Intel-based boxes and offered technical support for the platform. Major enterprise software vendors created Linux versions of their applications. Intel-based servers running on Linux software, particularly when grouped into clusters, started to challenge midrange servers in the $100,000 to $1 million price range.[21] Intel-based servers were generally much cheaper than Unix machines, allowing customers to scale out by adding additional servers to a cluster as the need arose.

---

[19] https://www.theregister.co.uk/2013/01/24/alan_cox_quits_linux_development/

[20] Ibid.

[21] https://www.itworld.com/article/2785695/operating-systems/intel-or-unix--server-customers-face-a-tough-choice.html

5.3. Dominance of the Intel Architecture in the Server Processor Market

The Intel architecture became dominant in the server processor market. To strengthen the ecosystem built around the Intel architecture, Intel invested a large amount of capital for software. For example, to enhance security software development, Intel announced that it would purchase McAfee for $48 a share in a deal valued at $7.68 billion on August 19, 2010.[22]

Intel's competitors from the desktop and mobile processor market entered the server processor market recently, but their market shares were insignificant. Although AMD and Qualcomm transitioned to the more cutting-edge manufacturing technologies such as the 14nm and the 10nm processes for their server solutions towards the end of 2017, they may not be able to expand their market shares significantly in the short term. The key problem for Intel's competitors is that they are still behind in developing third-party hardware and software support.[23]

[22] "Intel to Acquire McAfee". Intel Corporation. August 19, 2010. Retrieved August 19, 2010.

[23] http://press.trendforce.com/node/view/2855.html

Intel's current status and future in the server market

In 2017, enterprises made up the largest application segment in the server market, with 60% of the total demand. Data center covers about 35% of the total demand. However, the portion of data center is expected to surpass 50% in 2020.[24] Top three players in the data center, Amazon.com, Microsoft Corp., and Alphabet Inc., spent a combined $ 31.5 billion on data center.[25] Intel predicted that the data center, or cloud, segment is growing by 20% annually.[26] For example, Amazon operates at least 30 data centers in its global network, with another 10 to 15 on the drawing board (Amazon does not disclose the full scope of its infrastructure). More than 1 million customers are using Amazon.[27]

---------------------- *Insert Figure 9 here* -----------------------

----

One may wonder whether the dominance of Intel could persist when

[24] http://press.trendforce.com/node/view/2855.html

[25] https://www.bisnow.com/national/news/data-center/top-three-cloud-firms-spent-315b-on-data-center-leases-capital-expenses-last-year-73081

[26] From the interview with Victor Na, Intel architecture manager in the server sector (17.10.30)

[27] https://datacenterfrontier.com/inside-amazon-cloud-computing-infrastructure/

data centers, not enterprises, are the main customers in the server processor market. In November, 2017, Qualcomm released Centriq 2400, its new server processor. Qualcomm's new chip was based on the ARM architecture, while relying on Samsung's 10 nanometer FinFET process.[28] However, Intel's dominance may be solidified if its share in the data center segment rises even more, as Victor Na pointed out:

> We are always aware of the possibility that a strong competitor will enter the server market with a highly competitive product, as Intel has done in the 1990s. … However, data center is a completely different story. Data centers have thousands of servers that are all connected. Those servers share the same software and solutions, which are optimized on Intel's Xeon processors. I do not believe there is even a small chance that Qualcomm can gain a foothold on the data center segment, where the whole ecosystem is built around Intel.[29]

Unlike the 1990s, when Intel had no complementary software, almost all software applications for the server system recently are based on the Intel

---

[28] https://www.qualcomm.com/news/onq/2017/11/08/qualcomm-centriq-2400-worlds-first-10nm-server-processor

[29] From the interview with Victor Na, Intel architecture manager in the server sector (17.10.30)

architecture. Intel has successfully build its platform in the server market, with numerous complements surrounding the Intel platform.

Currently, Intel has a 99% market share in the server processor market.[30] Unlike the incumbents in the traditional server market, Intel has successfully managed the rising complexity of developing server platform in three ways. First, Intel developed multicore chips to advance chip performance without increasing clock rate. Second, Intel realized the economies of scale in developing server chips by leveraging design and manufacturing capabilities from its PC business. Third, Intel embraced Linux and Linux-based applications for its server software.

VI. CONCLUSION

In this paper, we analyze two cases of platform leadership in the semiconductor industry: ARM Holdings in the mobile AP market and Intel in the server processor market.

In the mobile AP market, many people expected Intel would establish

[30] https://www.idc.com/getdoc.jsp?containerId=prUS41419716

hegemony by leveraging its platform leadership in the PC industry. However, Intel killed its opportunity by declining the requests of Qualcomm and Apple. Intel believed that it could not make money by fulfilling their requests, which were incompatible with the semiconductor architecture Intel envisioned for smartphones. To meet the idiosyncratic needs of customers, Intel might have designed new chips and would have increased the cost of chip design and manufacturing to the point where Intel could not make any profit. From this perspective of rising complexity, Intel's reactions to the prominent customers' requests was not unreasonable. However, Intel appeared to be blinded to the sea change in the industry associated with the inevitability of the separation between IP suppliers and chip designers in the midst of rising complexity.

It is quite ironic that Intel missed the deeper implications of Moore's Law articulated by its cofounder, Gordon Moore. As a consequence, Intel missed the new strategic inflection point. Andrew Grove articulated the term as follows:

> An inflection point occurs where the old strategic picture dissolves and gives way to the new, allowing the business to ascend to new heights. … These signals may have been out there all along but you may have ignored them. The strategic inflection point is the time to wake up and listen. (Grove, 1997:

p. 35)

Intel was aware of the inflection point at the beginning of the PC industry, which radically departed from the mainframe industry in terms of developing chips, operating systems and software applications. Intel woke up and listened to the major changes in the computer industry as a whole. Intel seize the opportunity and became the platform leader in the PC market and the server market. However, Intel failed to foresee another inflection point in the mobile chip market. What Intel ignored was the inevitability of division of labor between building design libraries and developing chips in the midst of the rising complexity.

Unlike Intel, ARM quickly seized this opportunity at the moment of the new inflection point, by actively embracing the inevitability of separation between IP suppliers and chip designers. ARM disclosed its design libraries and encouraged other firms to develop their own chips by building on ARM's libraries and modifying some of their components. Other chipmakers started to earn money by producing diverse kinds of ARM-based chips, which made ARM prosper as well. By providing other chipmakers with reusable error-free components, ARM became the undisputed platform champion in the mobile AP market. The ARM platform became a sustainable solution to the ever-increasing complexity in the semiconductor industry by allowing its complementors to get access to its design libraries and develop new chips without worrying about exponentially growing errors in the design process.

In the server processor market, major players such as Sun Microsystems, IBM, and HP developed most core components for their servers, from server chips to operating systems. They also increased clock frequencies to achieve higher chip performance. Since the 1990s, increasing clock frequencies became extremely difficult and required enormous development costs (Geer, 2005). Moreover, developing operating systems and application software that properly operated on chips exacerbated the problem (Ghemawat et al., 2004). The rising costs in both hardware and software dramatically increased the risk of developing new servers for traditional server vendors. Unlike these incumbents, Intel did not try to produce every core component of a server machine. To cope with the rising complexity, Intel undertook three major steps. First, Intel developed chips with multiple computing cores, which carried out multiple operations simultaneously. By spreading workloads over multiple cores in parallel, Intel achieved advanced chip performance without improving clock frequencies. To reduce the costs of designing and manufacturing these new types of server chips, Intel relied on the economies of scale by leveraging design and manufacturing capabilities in its PC business. In addition, Intel embraced burgeoning Linux (operating system) and Linux-based applications for software complements to its server chips. With these new steps to reduce rising complexity, Intel gradually became the dominant leader in the server platform market, commanding a 99% market share.

By analyzing the ARM and Intel cases, we shed some light on the questions of why some platform leaders do better than others, and why existing platform leaders sometimes fail. According to William James, *"The art of being wise is knowing what to overlook."* In the face of the rising complexity, winners figured out what to overlook, while focusing on what they could do best. They built their ecosystems based on their assets and encouraged other firms to do business by embracing their ecosystems. Winners were those who understood that the increasing complexity was far beyond the reach of what a single company could manage. Attracting competent allies and allowing them to prosper was a sustainable solution to ever-increasing complexity in the semiconductor industry. Those who were blinded to or resisted this sea change were driven out. Intel was not an exception in the tectonic shifts in the computer industry.

Successful platform leaders' strategy in semiconductors is similar to the Roman Empire's symbiotic approach. The Romans never believed they could be the best in every field and instead relied on others to thrive. For example, the Romans encouraged the Greeks to promote commerce and entrusted the Etruscans with civil engineering. Had Romans tried to do commerce and civil engineering on their own, as Intel did in the mobile market, the Roman Empire might not have prospered for such a long time.

VII. APPENDIX

7.1. Computational Analysis

7.1.1. Basic Model

Based on the analyses of ARM Holdings and Intel, I modeled the dynamics of the platform competition where there are two incompatible platforms with different architectures: open vs. proprietary architecture. The purpose of this computational modeling is to analyze the effect of complexity on platform competition.

In the supply side, two platforms compete with each other by developing complements. The difference between the open architecture and the proprietary architecture is that in the open architecture, 3rd party complementors independently develop functionalities of the complement. In the proprietary architecture, platform leader develops all functionalities of the complement. Specifically, the proprietary architecture invests equally in each functionality, to balance the quality of each functionality (Burgelman & Schifrin, 2015). In the open architecture, complementors only invest their revenue to their own developing functionalities, with no coordination in developing different functionalities. In the demand side, consumers adopt complements at every time step. Consumers choose complements with a higher quality. The basic model for platform competition can be simplified as below.

---------------------- *Insert Figure 10 here* ----------------------

----

There are two platform leaders, L_O and L_P, each choosing an open architecture (denoted as *platform O*) and a proprietary architecture (denoted as *platform P*). Each platform *k* has two functionalities that can be developed, that is functions ${f^k}_1$ and ${f^k}_2$. In platform *P,* platform leader L_P develops two functionalities ${f^P}_1$ & ${f^P}_2$ and releases complement M_P. In platform *O*, two complementors, complementor *A* and *B*, develop functionality ${f^O}_1$ and ${f^O}_2$ respectively. Platform *O* releases complement M_O that includes function ${f^O}_1$ and ${f^O}_2$. The price for both complements, M_P and M_O, is fixed to *p*.

7.1.2. Supply side assumptions

Firm growth & exit

Complementor A and B in platform O each starts with initial capital of K_o. Platform leader L_P in platform P starts with initial capital of $2K_o$. Thus, the total capital amount is the same for each platform. Firm *j*'s capital *K* at time *t+1* is decided as the following equation:

$$K_{jt+1} = K_{jt} - C_{jt} + R_{jt}$$

, where K_{jt} is firm *j*'s capital at time *t*, C_{jt} is firm *j*'s R&D cost at time *t*, R_{jt} is firm *j*'s revenue at time *t*

Cost assumptions

To stay in the R&D race, firm *j* has to invest R&D cost C_{jt} to improve the quality of its complement functionality. To develop one functionality, every firm has to invest *at least* one unit of R&D cost, *c*. The larger the firm, the more new complement functionality development trials it can carry out, increasing its chance of developing functionalities with higher quality. Let d_{jt} be the number of firm *j*'s functionality development trials. Assuming indivisibility in organizing R&D, d_{jt} becomes a step function of capital K_{jt-1} with a discrete unit τ.

$$d_{jt}(K_{jt-1}) = \begin{cases} 1 & if \quad c \leq K_{jt-1} \leq \tau \\ h+1 & if \quad h\tau < K_{jt-1} \leq (h+1)\tau \end{cases}$$

, where h = 1, 2, … .

Thus, firm j's R&D cost at time t, C_{jt}, is decided by how many complement functionality development trials the firm is investing in. That is, C_{jt} increases as a multiple of c.The total cost is decided by how many types of functionality a firm develops. Since complementor A and B in platform O develop only one functionality each, while platform leader L_P develops two functionalities, R&D cost of firm j is decided as follows:

$$C_{jt} = \begin{cases} c \times d_{jt} & if\ firm\ j\ is\ a\ complementor\ in\ platform\ O \\ 2c \times d_{jt} & if\ firm\ j\ is\ a\ platform\ leader\ in\ platform\ P \end{cases}$$

If a firm's capital stock becomes smaller than the minimum R&D cost for developing complements (that is, c for complementor *A & B* and 2c for platform leader L_P), the firm exits the market.

Complement development assumptions

$q^j{}_{lt}$ represents the quality of firm j's complement functionality *l (l = 1 or 2)* as a result of innovation activities at time *t*. $q^j{}_{lt}$ is decided by three decision rules:

1. At each time step, $q^j{}_{lt}$ is a random variable from [1, 100] that follows a power-law distribution, $p(q) \sim q^{-\gamma}$. *(γ = 3)*
2. $q^j{}_{lt}$ is compared with $q^j{}_{lt-1}$ and the variable with a higher value is chosen; thus the quality is monotonically increasing in time:
3. $q^j{}_{lt} = max\,(q^j{}_{lt},\ \ q^j{}_{lt-1})$

If firm *j*'s capital grows, it can invest in more trials, increasing the number of random quality draws. If a firm didn't invest in functionality *l*, the quality of functionality *l* is treated as 0.

Revenue assumptions

A firm's revenue is decided by how many customers buy its complement released. n_{jt} denotes the number of customers that buy firm *j*'s complement in time *t*. Firm *j*'s revenue at time *t* is decided as follows:

$$R_{jt} = n_{jt} \times p$$

Customer adoption dynamics

Consumers consider whether to adopt or switch to a platform at every time step. The choice depends on utility gained in using complements; the quality of complement functionality that the customer prefers affects the choice.

7.1.3. The level of interdependency between platform components

Let us first assume the simplest case with two functionalities. First, we assume that there is no interdependency between components. For simplicity, assume that a complement is composed of two components that have different functionalities. All consumers have the following payoff function for complements, where they have equal preferences toward both functionalities.

$$u_{it} = \frac{1}{2}(q_{1t} + q_{2t})$$

Where $u^j{}_{it}$ is utility of platform *j*'s complement for customer *i* in time *t*,

$q^j{}_{kt}$ is the quality of platform *j*'s component functionality *k* in time *t*

Since all complements have the same price, we ignore the price from now on.

Now, we relax the assumption of no interdependency and add the interdependency between components to the model. With full interdependency, we assume that each components acts as a bottleneck to others. That is, the quality of its functionality $\boldsymbol{f_{1t}}$ and $\boldsymbol{f_{2t}}$ at time t is as follows:

$$q(f_{1t}; f_{2t}) = min(q_{1t},\ q_{2t})$$

$$q(f_{2t}; f_{1t}) = min(q_{1t},\ q_{2t})$$

In each case, the payoff function of consumers is as below:

$$u_{it} = \frac{1}{2}(min(q_{1t},\ q_{2t}) + min(q_{1t},\ q_{2t})) = min(q_{1t},\ q_{2t})$$

We ran a simulation model with / without interdependency. Each case was run for 1,000 trials. As the graph below shows, proprietary architecture has a much higher rate of winning probability in the environment where the interdependency between components exists. The proprietary architecture has a winning probability of 55.1% in the environment with no interdependency; the probability increases to 75.8% in the environment with full interdependency.

---------------------- *Insert Figure 11 here* ----------------------

----

To understand the driving factor of the result, we ran a simulation model where a fixed amount of capital (k = 100) is given to each platform and each platform develops a complement every time step, without any competition. The proprietary architecture invests an equal amount of capital in developing each component, to balance the quality of components (feature parity). Complementors in the open architecture don't coordinate on developing components; each complementor independently invests in R&D.

We now compare the average payoff level to consumers for each architecture in period 100, with different R&D investment allocation level for complementor A in the open architecture. All results are averaged over 1,000 simulation trials.

In the environment with no interdependency, the payoff function is as below:

$$u_{it} = \frac{1}{2}(q_{1t} + q_{2t})$$

That is, the quality of one component isn't affected by the quality of other components. As the R&D allocation to complementor A gets lower, component A results in a lower quality. However, component B from complementor B results in a higher quality at the same time. Deficiency in one component from a low R&D investment is compensated by the superior performance of other components, thus maintaining the total quality of the complement. Thus, the open architecture has a fair chance to compete with the proprietary architecture.

---------------------- *Insert Figure 12 here* ----------------------

----

In the environment with full interdependency, the payoff function is as below:

$$u_{it} = min(q_{1t},\ q_{2t})$$

That is, one functionality with a low quality limits the functionality of the whole platform. As the R&D allocation level for complementor A gets lower, component A results in a lower quality.

This limits the total performance of the open architecture, as the below graph shows. End users will avoid the adoption of a platform if any of its components suffers from a deficiency in its quality because "improvements in performance in one part are of limited significance without simultaneous improvements in other parts" (Rosenberg, 1979: p. 30). This was the very reason why Intel's 32-bit microprocessor (the 386) waited 10 years before Microsoft developed a 32-bit operating system (Windows 95) (Casadesus-Masanell, Yoffie, & Mattu, 2003).

---------------------- *Insert Figure 13 here* ----------------------

----

7.2. ARM business model

ARM business model consists of three types of licensing: architectural license, implementation license, and foundry license ('per use' royalty model). Numerous participants in the ARM ecosystem chose the appropriate licenses according to their usages (Shih, Shih, & Chien, 2009).

First, an architectural license provides licensees the right to develop its own derivatives based on the ARM architecture, providing only the abstract architectures and algorithms.[31] It offers high flexibility for adaptation, allowing licensees to design their own cores faster, slower, or more efficiently.[32] Only large players such as Apple, Qualcomm, Samsung, and Nvidia with enough design capabilities can manage it. Qualcomm's Scorpion core and Apple's Typhoon core are from ARM architectural licenses. Architectural licenses are the most expensive, ranging up to $10 million.[33]

Second, implementation license provides a complete package of chip design and manufacturing with physical layout information, usually fixed for a specific process technology. It permits fast implementation, with

---

[31] https://semiaccurate.com/2013/08/07/a-long-look-at-how-arm-licenses-chips/

[32] https://semiaccurate.com/2013/08/07/a-long-look-at-how-arm-licenses-chips/

[33] https://www.anandtech.com/show/7112/the-arm-diaries-part-1-how-arms-business-model-works/3

performance and timing already verified in advance (Linden & Somaya, 2003). The license types can vary depending on the usage time (single / multi use), the number of cores (single / multiple cores), and the license period (term / perpetual). For the multi-product license, the fee was \$ 5 – 7 million (Iansiti & Strojwas, 2003).

Lastly, foundry license, or 'per use' royalty model, collects fee per use of wafer in foundries. ARM established partnership agreements with foundries so that chipmakers could use ARM IPs in foundries that they use, without directly contacting ARM to license design libraries.

REFERENCES

Adner, R., & Kapoor, R. (2010). Value creation in innovation ecosystems: How the structure of technological interdependence affects firm performance in new technology generations. ***Strategic management journal***, 31(3), 306-333.

Burgelman, R., & Philip Meza, P. (2001). The Open Source Software Challenge. Case No.SM85. ***Stanford Graduate School of Business***.

Burgelman, R., & Schifrin, D. (2015). Intel's Mobile Strategy in 2015 and Beyond. Case No.SM236. ***Stanford Graduate School of Business***.

Borkar, S. (2005). Designing reliable systems from unreliable components: the challenges of transistor variability and degradation. ***Ieee Micro***, 25(6), 10-16.

Charlene Zietsma, Ken Mark, & Jordan Mitchell. (2004). Hewlett-Packard: Sun Microsystems in 2001. 9B04M085. ***Ivey Business School***.

Clarke, P. (2003). MIPS slides as semiconductor IP market grows. ***Electronic Times***, Vol. 06/18/2003.

Collet, R., & Pyle, D. (2013). McKinsey on Semiconductors: What happens when chip-design complexity outpaces development productivity. ***McKinsey & Company***.

Connor. (2001). Linux slips slowly into the enterprise realm. ***Network World***.

Cusumano, M. A., & Gawer, A. (2002). The elements of platform leadership. ***MIT Sloan management review***, 43(3), 51.

Dingee, D., & Nenni, D. (2015). Mobile Unleashed: The Origin and Evolution of ARM Processors in Our Devices. ***SemiWiki LLC***.

Eigen, M., & Schuster, P. (1977). A principle of natural self-organization. ***Naturwissenschaften***, 64(11), 541-565.

Eisenmann, T. R., Parker, G., & Van Alstyne, M. (2009). Opening platforms:

how, when and why?. ***Platforms, markets and innovation***, 131-162.

Ethiraj, S. K., & Levinthal, D. (2004a). Bounded rationality and the search for organizational architecture: An evolutionary perspective on the design of organizations and their evolvability. ***Administrative Science Quarterly***, 49(3), 404-437.

Ethiraj, S. K., & Levinthal, D. (2004b). Modularity and innovation in complex systems. ***Management science***, *50*(2), 159-173.

Ethiraj, S. K., Levinthal, D., & Roy, R. R. (2008). The dual role of modularity: innovation and imitation. ***Management Science***, 54(5), 939-955.

Gawer, A., & Cusumano, M. A. (2002). Platform leadership: How Intel, Microsoft, and Cisco drive industry innovation (Vol. 5, pp. 29-30). ***Boston: Harvard Business School Press***.

Gawer, A., & Cusumano, M. A. (2008). How companies become platform leaders. ***MIT Sloan management review***, 49(2), 28.

Gawer, A., & Cusumano, M. A. (2014). Industry platforms and ecosystem innovation. ***Journal of Product Innovation Management***, 31(3), 417-433.

Geer, D. (2005). Chip makers turn to multicore processors. ***Computer***, *38*(5), 11-13.

Ghemawat, Subirana, & Pham. (2004). Linux in 2004. ***Harvard Business Review***. # 705407-PDF-ENG.

Grove, A. (1997). Only the Paranoid Survive: How to Exploit the Crisis Points that Challenge Every Company. ***Crown Business***.

Gurman, M. (2018). How Apple Built a Chip Powerhouse to Threaten Qualcomm and Intel. ***Bloomberg***.

Hamilton, D. P. (2001). Intel gambles it can move beyond the PC with new microprocessor. ***The Wall Street Journal***.

Hardymon, F., & Ann Leamon, A. (2000). INTEL 64 FUND. 800351-PDF-ENG. ***Harvard Business School***.

Heck, S., Kaza, S., & Pinner, D. (2011). Creating value in the semiconductor industry. ***McKinsey & Company***.

Hobday, M. (1991). Semiconductor technology and the newly industrializing countries: the diffusion of ASICs (Application Specific Integrated Circuits). ***World Development***, 19(4), 375-397.

Iansiti, M., & Levien, R. (2004). Strategy as ecology. ***Harvard business review***, 82(3), 68-81.

Iansiti, M., & Strojwas, M. (2003). Taiwan semiconductor manufacturing company: Building a platform for distributed innovation. ***HBS case***, 604-044.

Kang, G. (2010). Understanding the Semiconductor Business. ***Jisungsabook***.

Kapoor, R. (2013). Persistence of integration in the face of specialization: How firms navigated the winds of disintegration and shaped the architecture of the semiconductor industry. ***Organization Science***, *24*(4), 1195-1213.

Kapoor, R., & Lee, J. M. (2013). Coordinating and competing in ecosystems: How organizational forms shape new technology investments. ***Strategic management journal***, 34(3), 274-296.

Kauffman, S. A. (1993). The Origins of Order. ***Oxford University Press***. New York.

Kauffman, S. (1995). At home in the universe: The search for the laws of self-organization and complexity. ***Oxford university press***.

Kirby, Chen, & Wong. (2015). Taiwan Semiconductor Manufacturing Company Limited: A Global Company's China Strategy. ***Harvard Business School***. 9-308-057.

Restivo, K. (1999). "Intel looks to jump start IA-64," ***Computer Dealer News***. pg. 10.

Leamon, A., & Hardymon, G. F. (2000). Intel64 Fund. ***Harvard Business School Case***, 9-800.

Levinthal, D. A. (1997). Adaptation on rugged landscapes. ***Management science***, 43(7), 934-950.

Levinthal, D. A., & March, J. G. (1993). The myopia of learning. ***Strategic management journal***, 14(S2), 95-112.

Levinthal, D., & Posen, H. E. (2007). Myopia of selection: Does organizational adaptation limit the efficacy of population selection?. ***Administrative science quarterly***, 52(4), 586-620.

Linden, G., & Somaya, D. (2003). System-on-a-chip integration in the semiconductor industry: industry structure and firm strategies. ***Industrial and Corporate Change***, *12*(3), 545-576.

Madrigal, A. C. (2013). Paul Otellini's Intel: Can the company that built the future survive it?'. ***The Atlantic***, 16.

Mark, Kernan, & Faber. (2004). Hewlett-Packard: Sun Microsystems in 2001. ***Ivey Management Services***.

McLellan, P. (2010). EDAgraffiti. ***Booksurge Llc***.

Moody, D. (2009). The "physics" of notations: toward a scientific basis for constructing visual notations in software engineering. ***IEEE Transactions on Software Engineering***, *35*(6), 756-779.

Moore, G. E. (1965). Cramming More Components Onto Integrated Circuits, ***Electronics***,(38) 8.

Nenni, D. (2014). Semiconductor IP Information Flow***. Semiwiki Llc***.

Nenni, D., & McLellan, P. (2013). FABLESS: the transformation of the semiconductor industry. ***SemiWiki. Com***.

Or-Bach, Z. (2012). Is the cost reduction associated with IC scaling over?. ***EETimes***.

Pankaj Ghemawat, Brian Subirana, & Christina Pham (2004). LINUX IN 2004. 705407-PDF-ENG. ***Harvard Business School***.

Porter, M. E. (1979). How competitive forces shape strategy. ***Strategic Planning: Readings***, 102-117.

Porter, M. E. (2008). The five competitive forces that shape strategy. ***Harvard business review***, 86(1), 25-40.

Roser, M. & Ritchie, H. (2018). "Technological Progress". Published online at ***OurWorldInData.org***. (https://ourworldindata.org/technological-progress)

Schilling, M. A. (2009). Protecting or diffusing a technology platform: tradeoffs in appropriability, network externalities, and architectural control. ***Chapters.***

Shih, W., Shih, C., & Chien, C. F. (2009). Horizontal specialization and modularity in the semiconductor industry. ***Harvard Business School Case Study***, 9-609.

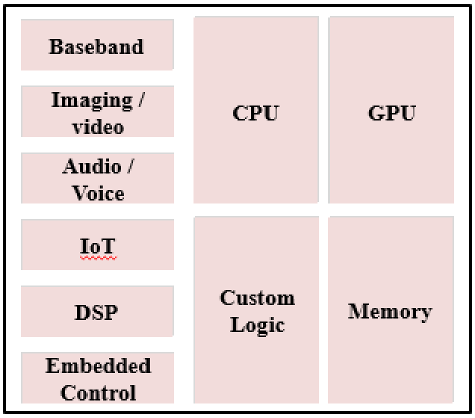
Siggelkow, N., & Levinthal, D. A. (2003). Temporarily divide to conquer: Centralized, decentralized, and reintegrated organizational approaches to exploration and adaptation. ***Organization Science***, 14(6), 650-669.

Zietsma, C., Mark, K. & Mitchell, J. (2004). Hewlett Packard/Compaq Competitive Dynamics Case Set: Sun Microsystems in 2001. ***Ivey Publishing***.

Zhou, Y. M. (2013). Designing for complexity: Using divisions and hierarchy to manage complex tasks. ***Organization Science***, 24(2), 339-355.

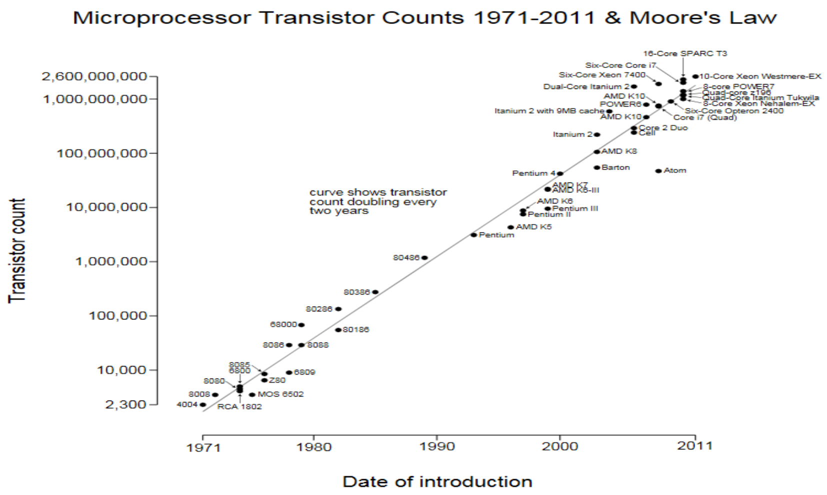
FIGURES

Figure 1. Typical mobile chip design layout with basic functional blocks



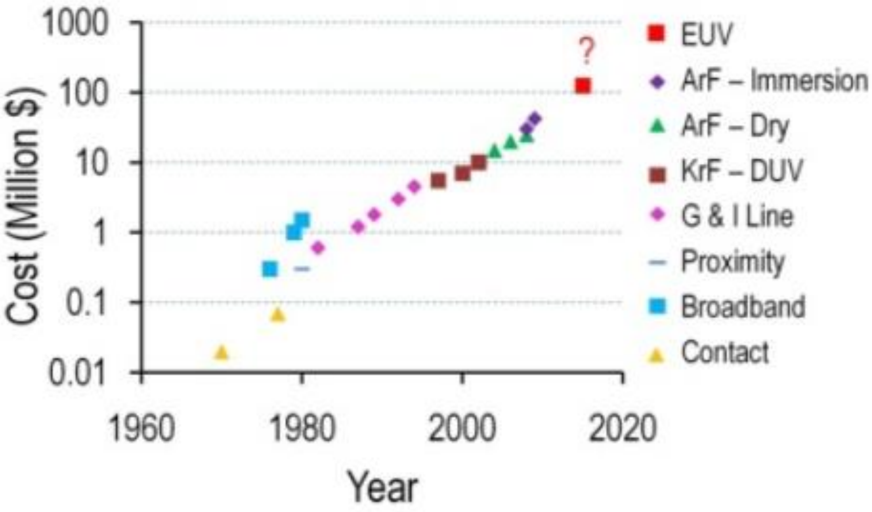
Source: Kang, G. (2010). Understanding the Semiconductor Business. Jisungsabook.

Figure 2. Exponentially increasing number of transistors on an integrated circuit



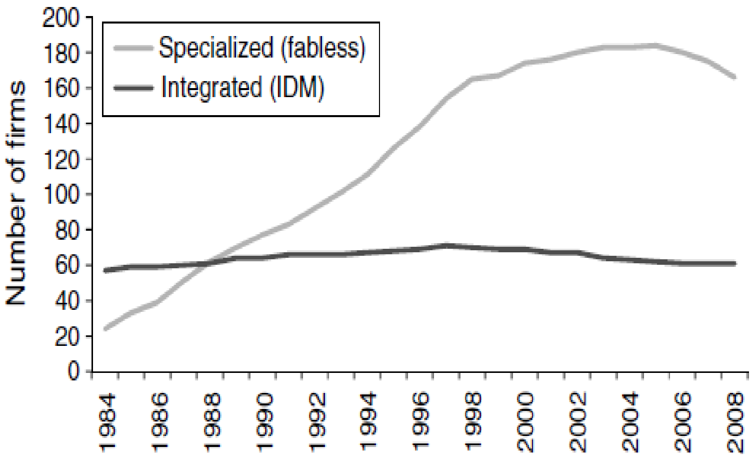
Source: Roser, M. & Ritchie, H. (2018). "Technological Progress". Published online at OurWorldInData.org. (https://ourworldindata.org/technological-progress)

Figure 3. Exponentially increasing cost for the capital equipment in a leading-edge fab



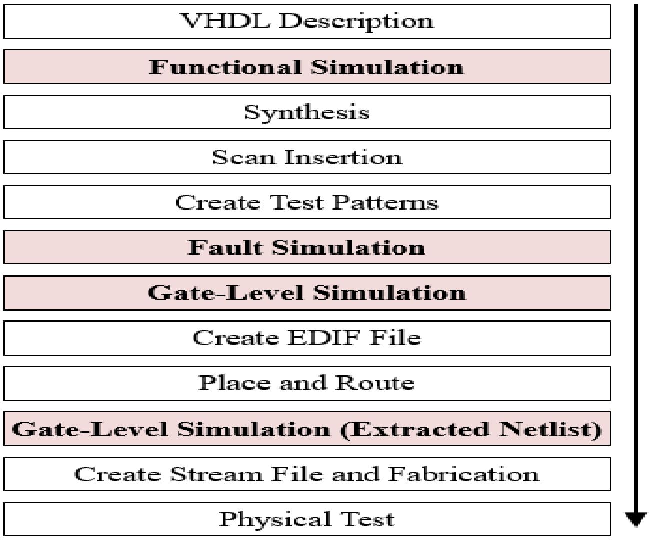
Source: Or-Bach, Z. (2012). Is the cost reduction associated with IC scaling over? (https://www.eetimes.com/author.asp?section_id=36&doc_id=1286363)

Figure 4. Number of fabless and IDM firms in the semiconductor



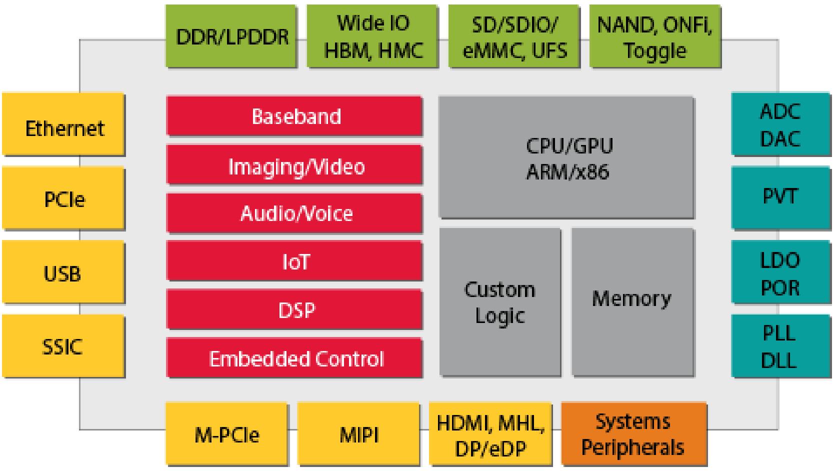
Source: Kapoor, R. (2013). Persistence of integration in the face of specialization: How firms navigated the winds of disintegration and shaped the architecture of the semiconductor industry. Organization Science, 24(4), pp.1195-1213.

Figure 5. A typical chip design process



Source: Kang, G. (2010). Understanding the Semiconductor Business. Jisungsabook.

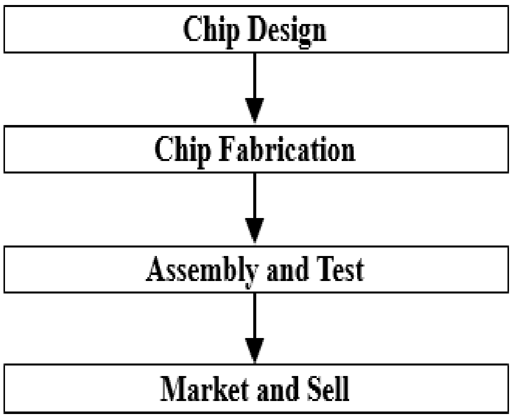
Figure 6. A typical chip layout design with various design components



Source: Nenni, D. (2014). Semiconductor IP Information Flow. Semiwiki

(https://www.semiwiki.com/forum/content/4031-semiconductor-ip-information-flow-e.html?new_comment=)

Figure 7. Semiconductor chip value chain



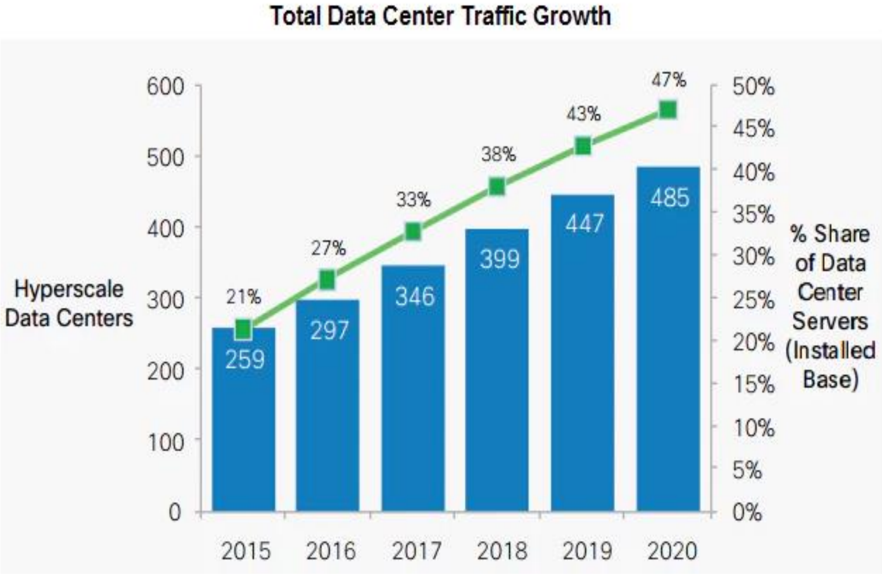
Source: Kirby, Chen, & Wong. (2015). Taiwan Semiconductor Manufacturing Company Limited: A Global Company's China Strategy. Harvard Business School. 9-308-057.

Figure 8. Phone vendor rank in 1999

Phone vendor	M/S
Nokia	**26.9%**
Motorola	16.9%
Ericsson / Sony	10.5%
Siemens	6.2%
Panasonic	5.5%
Samsung	4.6%

Source: Statista (https://www.statista.com/statistics/271574/global-market-share-held-by-mobile-phone-manufacturers-since-2009/)

Figure 9. Total data center traffic growth



Source: Cisco Global Cloud Index, Synergy Research (http://marketrealist.com/2016/12/oracle-amazon-microsoft-heavily-investing-data-centers/)

Figure 10. Basic model for platform architecture competition

Platform (background)	Platform *O* (open architecture)		Platform *P* (proprietary architecture)
Complementary firms	*Complementor A*	*Complementor B*	*Platform leader* L_P
Complements	*Complement* M_A	*Complement* M_B	*Complement* M_P
	contains $f^O{}_1$	contains $f^O{}_2$	contains $f^P{}_1$ & $f^P{}_2$

Figure 11. Proprietary architecture's winning probability with different levels of interdependency between components

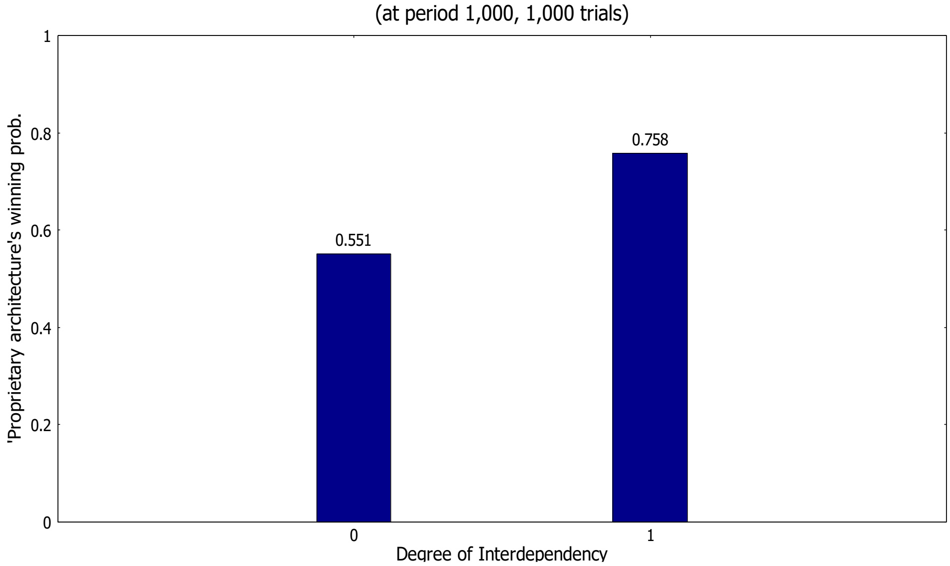


Figure 12. Average payoff level to consumers in the environment with no interdependency

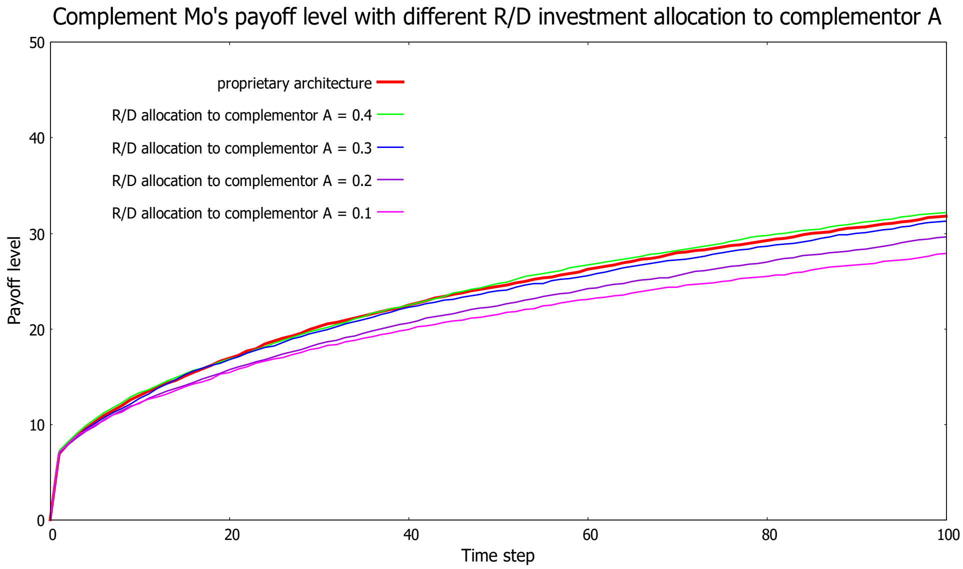
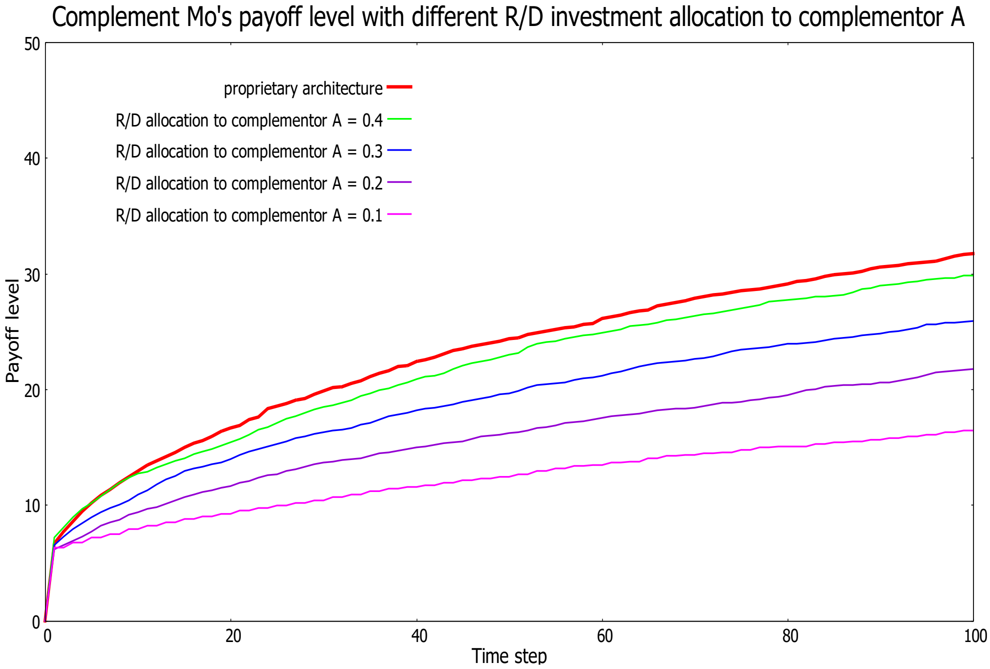


Figure 13. Average payoff level to consumers in the environment with full interdependency



국문초록

복잡성의 시대 속 플랫폼 리더십에 관한 연구

정 혁

경영학과 경영학전공

서울대학교 대학원

플랫폼 리더십은 전통 산업에서 하이테크 산업에 이르기까지 많은 관심을 불러일으켜왔다. 본 연구는 플랫폼 리더십과 관련하여 반도체 산업 내 2가지 사례를 분석함으로써 어떤 플랫폼 리더가 성공하는지, 그리고 기존의 플랫폼 리더는 왜 실패하는지에 대해 답을 하고자 한다. 무려 반세기 동안 반도체 칩에 들어가는 트랜지스터 숫자가 2년마다 2배씩 증가하면서, 반도체 회사들은 칩 설계의 치솟는 복잡성을 직면하게 되었다. 1990년대에 들어서 반도체 칩 내의 트랜지스터 숫자가 1,000만 개를 돌파하였고, 이를 계기로 칩 설계 산업은 새로운 국면을 맞이하였다. 본 연구에 따르면, 치솟는 복잡성 속에서 성공한 플랫폼 리더들은 무엇을 간과해야 할지를 정확히 파악하고, 그들이 잘할 수 있는 분야에만 집중하였다. 그들은 그들의 강점을 바탕으로 생태계를 구축한 후 다른 기업들이 생태계 내에서 사업을 영위할 수 있도록 도왔다. 승자의 왕관은 무한히 증가하는 복잡성이 더 이상 혼자서 감당할 수 없는 수준에 도달했다는 것을 깨달은 자에게 돌아갔다. 뛰어난 조력자들을 생태계 내로 끌어드리고 그들이 번영할 수 있게 돕는 것이야말로 반도체 산업 내 끝없이 증가하는 복잡성에 대한 해결책이었다. 치솟는 복잡성을 외면하거나 저항하려고 했던 기업들은 이러한 변화의 물결에 모두 쓸려 나갈 수 밖에 없었다.

주요어: 플랫폼 리더십, 플랫폼, 복잡성, 반도체, 암 홀딩스, 인텔

학 번: 2015-20669