A Strategic Study of Disruption, Dis-integration, and Modularity in the Microprocessor Industry

by

Michael J. Bass

B.S. Computer Engineering, University of Illinois at Urbana

Submitted to the Alfred P. Sloan School of Management and the School of Engineering in Partial Fulfillment of the Requirements for the Degree of

Master of Science in the Management of Technology

at the

Massachusetts Institute of Technology

June, 2000

© 2000 Michael J. Bass

ALL RIGHTS RESERVED

The author hereby grants to MIT permission to reproduce and to distribute publicly copies of this thesis document in whole or in part.

Signature of Author:__________________________________________

MIT Alfred P. Sloan School of Management
May 19, 2000

Certified by:__________________________________________________

Rebecca Henderson
Eastman Kodak LFM Professor of Management, MIT-Sloan
Thesis Supervisor

Certified by:__________________________________________________

Clayton Christensen
Professor of Management, Harvard Business School
Thesis Advisor

Accepted by:__________________________________________________

David A. Weber
Director, MIT-Sloan Management of Technology Program
A Strategic Study of Disruption, Dis-integration, and Modularity in the Microprocessor Industry

by Michael J. Bass

Submitted to the Alfred P. Sloan School of Management and the School of Engineering in Partial Fulfillment of the Requirements for the Degree of Master of Science in Management of Technology at the Massachusetts Institute of Technology May 19, 2000

Smashing microprocessors

History has shown, and much management research has pointed out, that the structure of industries is not static. In some cases, striking shifts in industry structure have led to value creation explosions, the entry of new firms, and dramatic shifts in the value capture capability of firms within the industry. Perhaps the most oft-cited example of this phenomenon is the dis-integration of the personal computer industry into a horizontally organized cluster of hundreds of firms, and the emergence of Intel and Microsoft as dominant value-capture powerhouses within that industry.

A large and ever-growing body of research studies the structure of industries. That is, the composition and structure of products created by firms in the industry, the structure of the design processes that generate these products, and the markets in which components and design services are offered to the industry. Of great concern to research in this area are the concepts of modularity and value chain structure. Some argue that the degree of modularity inherent in both an industry's products and the design processes that produce them have a profound effect on industry structure. They argue that modularity affects the structure of design and production value chains within the industry, the markets in which components and design services are made available, on the firms that participate in those markets, and the ability of those firms to create and capture value.

Motivated by the striking example of the PC industry itself and this growing body of research, this thesis studies possible shifts in industry structure in the field of microprocessors, or, more generally, very-large-scale-integrated-circuits (VLSICs) for computing. High-end microprocessor design, like PC design in the early 80's, is currently vertically integrated within a few dominant firms. But a fundamental trend is at work: the rate at which transistors are becoming available exceeds the rate at which the designers and design processes of incumbent firms are able to use them. This situation creates pressures on the boundaries of design and production that may lead to shifts in value chain boundaries that have previously not been considered by the industry. Will the microprocessor industry itself dis-integrate? If these shifts were to manifest, they may consequently impact the value-capture capability of new and incumbent firms, improving some firms' profitability at the expense of others.

This thesis surveys and summarizes the existing body of research on the relationship between modularity and industry structure, and applies it to the field of computing VLSICs. It analyzes the microprocessor industry using those mechanisms proposed in the body of research, suggesting scenarios for how the industry structure may evolve in the near future. It studies and suggests how firms might influence this evolution. This leads to recommendations for firms currently in the industry, and for firms considering entry.

Thesis Advisor: Rebecca Henderson Eastman Kodak LFM Professor of Management, MIT-Sloan
Thesis Reader: Clayton Christensen Professor of Management, Harvard Business School

For my wife Dorothy,

who encouraged me to come here.
CONTENTS

1. MOTIVATION: MODULARITY, VALUE CHAINS, AND INDUSTRY STRUCTURE .......................... 6
   KEY QUESTIONS FOR MICROPROCESSOR AND COMPUTING VLSIC VENDORS .......................... 7

2. STRUCTURE AND METHOD ................................................................................................. 9
   AUDIENCE ....................................................................................................................... 9
   STRUCTURE OF THE THESIS ....................................................................................... 9

3. SURVEY OF RELEVANT RESEARCH ............................................................................... 10
   DOMINANT DESIGNS ................................................................................................... 10
   TECHNOLOGY S-CURVES ........................................................................................... 10
   INNOVATION ............................................................................................................... 11
   DESIGN VALUE CHAINS ............................................................................................. 12
   MODULARITY ............................................................................................................... 13
   PLATFORMS AND OPTION VALUE ............................................................................. 14

4. CONCEPTS AND DEFINITIONS ....................................................................................... 15
   DESIGN DOMAIN VS. PRODUCTION DOMAIN .......................................................... 15
   VERTICAL VS. HORIZONTAL INDUSTRY STRUCTURE .............................................. 18
   DESIGNS VS. ARTIFACTS ............................................................................................ 19
   MODULARITY IN DESIGN, PRODUCTION, AND USE ............................................... 20
   FABRICATED VS. ASSEMBLED .................................................................................. 22
   MODULES VS. SYSTEMS .............................................................................................. 22
   IN-HOUSE VS. OUTSOURCED DESIGN AND/OR PRODUCTION .................................. 22
   MODULE HIERARCHIES ............................................................................................... 23
   INTERMEDIATE (DERIVED) MARKETS FOR HIDDEN MODULES ............................ 24
   PERFORMANCE VS. UTILITY ...................................................................................... 24
   OPTION VALUE ............................................................................................................. 24

5. GENERAL MECHANISMS FOR SHIFTS IN VALUE CHAIN BOUNDARIES .................... 26
   PERFORMANCE SUPPLIED TO MARKET MISMATCHES PERFORMANCE DEMANDED BY MARKET 26
   STICKY INFORMATION AND AGENCY ISSUES ....................................................... 30
   GROWTH OPTIONS ...................................................................................................... 31
   MECHANISM SUMMARY ............................................................................................. 34
6. ANALYSIS - APPLICATION TO MICROPROCESSOR INDUSTRY .......................................................... 35
   DESIGN PROCESS DESCRIPTION ................................................................................................. 35
   DESIGN & PRODUCTION VALUE CHAINS ................................................................................ 37
   VARIANT DESIGN PROCESSES AND APPLICATIONS .............................................................. 38
   CURRENT MARKET AND PRODUCT CLASS DESCRIPTIONS ...................................................... 42
   ANALYSIS TOOLS ...................................................................................................................... 48
      Porter's forces .......................................................................................................................... 48
      Relevant technology S-curves, performance measures, and current position .......................... 58
      Network effects ....................................................................................................................... 63
      Sources of appropriability and complementary assets .......................................................... 64
   STRATEGY ASSESSMENT OF INCUMBENT FIRMS .................................................................. 66
   STRATEGY ASSESSMENT OF NOTABLE ENTRANTS ................................................................. 68
   APPLICATION OF "GENERAL MECHANISMS" FOR INDUSTRY VALUE CHAIN EVOLUTION ........... 72
   EVOLUTION SCENARIOS ........................................................................................................ 77
      Market scenarios .................................................................................................................... 77
      Design process scenarios ....................................................................................................... 79
      Required capabilities shared across scenarios ....................................................................... 81

7. RECOMMENDATIONS ............................................................................................................. 82

8. CONCLUSION .......................................................................................................................... 87

9. APPENDICES .......................................................................................................................... 88
   APPENDIX A – EXAMPLES OF THE PERFORMANCE MISMATCH MECHANISM ....................... 88
   APPENDIX B – EXAMPLES OF HARDWARE ↔ SOFTWARE TRANSFORMATION TECHNOLOGIES .... 93
   APPENDIX C – KEY INTERFACES IN MPU PRODUCT STRUCTURE ........................................ 95

10. REFERENCES ......................................................................................................................... 96
1. Motivation: modularity, value chains, and industry structure

The structure of industry value chains and the architectures, interfaces, and standards that enable them have important consequences for the firms that comprise the industry. Firms make several strategic choices in this area. For example: Where in the value chain should we invest in developing competencies? Which portions of the value chain should we develop within the firm? Which portions should we outsource? And, independent of which firms compete for each piece, where should the boundaries within the value chain lie? How many, “links” should comprise the chain, and how should each link add value? That is, what are the interfaces and handoffs within the value chain, and what standards enable those interfaces?

Such decisions have important strategic consequences for the firms that take them. We can observe that the value created within the chain is unequally distributed across it. Some firms are able to capture the lion’s share of the profits while others struggle to retain any profits. Clearly, firms can benefit from understanding, before investments are made, the distribution of profits across the value chain.

But even this understanding does not guarantee success for firms, for the structure of product value chains is far from static. As evidenced in the personal computer industry, value chains that were once vertically integrated within one or a few firms can dis-integrate to a horizontal structure, with many different firms competing within each value-added link in the chain for a dominant share of that link’s value. Firms could benefit from understanding why and when the structure of their product value chains is about to shift, so they can better modulate their investment decisions towards profitable portions of the chain, or shift the focus of their relationships with partners as appropriate.

Such a situation is evident in today’s microprocessor industry. ITRS(1999) succinctly summarizes the issue:

"The objective of the semiconductor business is to produce chips that create profit for integrated circuit manufacturers, their customers, and their suppliers and strategic partners. The increased difficulty in designing, verifying, and testing these chips has become a larger barrier to achieving this objective than providing the technology for manufacturing them."

Clearly, with industry profitability threatened, it would be odd for the industry structure, the boundaries in the value chains that comprise the industry, and the firms that compete within each portion of the value chain to remain static. Both incumbent and entering firms have an interest in understanding how the boundaries of the value chain will shift, why these shifts occur, and when it is likely to happen.
Disruption, Dis-integration, and Modularity the Microprocessor Industry

An even stronger capability for firms would be the ability to understand why these shifts in the value chain occur, to such an extent that they could actually stimulate such changes. That is, firms could invest in activities, technologies, partnerships, or standards that lead to a re-structuring of the links in the chain in a way that would be advantageous to them.

We can also hypothesize that the technology choices made by individual firms influence the universe of possible resulting product value chains. That is, the technologies (means of gathering inputs, processing and transforming them, and producing output) involved in any particular link of the chain tend to define the boundaries and interfaces by which the links themselves are described. Stated another way, the technology choices made by firms in the value chain – including strategic choices, such as whether to outsource technology development or to develop technology in house, or whether or not to support a technology standard – may influence over time the degree of modularity within the value chain, and the interfaces and standards at the boundaries between the modules. This would be an important outcome, because the degree of modularity within the chain, and the definitions and interfaces at the boundaries of each link, would affect not only the capacity for each link to capture value within the chain, but also the applicability of each link to other chains, perhaps in other markets.

The technology choices made by firms in the value-adding process within any link of the chain also affect the performance of that link relative to the performance of others in the chain. Technology choices impact all of: performance (as measured by the immediately downstream customer), resources required to achieve that performance, and time to market. Again, these characteristics may have an impact on the structure of upstream and downstream links in the chain over time, and firms could benefit by better understanding these impacts.

**Key Questions for microprocessor and computing VLSIC vendors**

This thesis addresses a few key questions that are relevant for vendors of microprocessors and integrated-circuit-based computing systems. These questions assess the potential impact of modularity on the future of the general-purpose computing industry. Specifically:

- What are the key interfaces & boundaries in today’s computing VLSIC design & production value chains?
- What pressures are exerted on these boundaries?
- How might the design & production value chains evolve in the near future? What are the likely scenarios?
- Are there specific actions that entering or incumbent firms can take to influence this evolution?
Disruption, Dis-Integration, and Modularity the Microprocessor Industry

- How might each scenario affect the ability of entering or incumbent firms to create and capture value?
- What should today's microprocessor vendors do? That is:
  - In what technologies should they invest?
  - What portions of the value chain should they seek to own or control?
  - How should they organize to make it happen?
2. Structure and method

**Audience**

This thesis is targeted at managers in incumbent microprocessor design organizations, and at managers who are contemplating entry to some point of the computing VLSIC design value chain. The author hopes to credibly describe the possible evolution of the design and production value chains and make recommendations that guide technology investment, standards creation, development of organizational capabilities, and decisions regarding standards support, technology alliances, and marketing alliances.

**Structure of the thesis**

Section 1 introduces the thesis and the motivation its creation. Section 2 describes the audience and structure. Since the thesis is targeted to a managerial audience, it surveys and summarizes past research before applying it within the computing VLSIC industry. Section 3 surveys past work on modularity, industry evolution, and value chain structure. Section 4 highlights the key concepts to be used in the analysis and, to avoid confusion, offers definitions and explanation. Section 5 extracts from the past research a list of general mechanisms leading to shifts in value chain boundaries.

Section 6 then analyzes the computing VLSIC industry by examining each of these general mechanisms in the context of microprocessors and computing VLSICs. The analysis also considers other relevant mechanisms drawn from the research. The analysis produces scenarios for evolution of the value chain. Section 7 makes recommendations based on the desirability of each scenario. It suggests actions that firms might take to encourage or discourage each scenario from actually manifesting, and to maximize its value capture capability in each scenario.

Section 8 summarizes and concludes the thesis.
3. Survey of relevant research

This thesis applies the findings of a wide body of research to its analysis of the microprocessor industry. This section briefly summarizes each piece of prior research that will have a bearing on the analysis.

Dominant Designs

Utterback (1994) summarizes the earlier work of Abernathy and Utterback and introduces the concept of dominant designs with clarity and numerous interesting and compelling examples. Utterback defines a dominant design as that design which wins allegiance in the marketplace, which defines how a product is supposed to look and operate, and defines a set of implicit features and performance requirements.

Tushman & Murmann (1998) presents a comprehensive review of work until that date regarding dominant designs and technology cycles, including concise summaries of the conclusions of prior work. Their work viewed systems as a collection of subsystems and linking mechanisms, and proposed an innovation model consisting of a nested hierarchy of technology cycles. Technology cycles consist of discontinuous innovation, leading to renewed periods of ferment, resulting in emergence of a new dominant design. Technology cycles within subsystems can result in technical substitutions in the containing system. The paper concludes with a series of observations and hypotheses based upon the prior work, including a key hypothesis “Sustained competitive advantage is gained by shaping dominant designs in core subsystems over time”.

Technology S-Curves

Foster (1986) introduced the concept of technology S-curves, and the concept of discontinuous technological innovation. Technologies move from a period of ferment, during which performance improvements require much effort, to a period of takeoff, during which performance improvements require relatively smaller efforts. Eventually, the technology shifts into a period of maturity, where increasing amounts of effort are required to achieve even small performance improvements. Technology discontinuities occur when new technologies emerge from ferment to takeoff. Such discontinuities present an advantage to firms using technologies lower on their S-curves, requiring relatively little effort.
to improve relative to competing firms, whose mature technologies have little prospects for continued improvement.

**Innovation**

Henderson and Clark (1990) distinguished among various types of innovation. *Incremental* and *Modular* innovations innovate in component technology while leaving the linking mechanisms among components unchanged. Incremental innovation “refines and extends an established design”, while modular innovation overturns core design concepts within the component. *Architectural* and *Radical* innovations both change the way that components are linked together. Architectural innovation leaves the product components unchanged while connecting them in a new way. Radical innovation changes both the connections and the components, and establishes a new dominant design.

Christensen (1992a) found that the flattening of the cumulative effort vs. performance technology S-curve may be the result of individual firms’ expectations about the remaining life of a technology, rather than inherent limits in the technology itself. When threatened with competition, firms were able to find new ways to revive the technology and launch it into renewed takeoff. Christensen includes a useful methodology for correlating component performance with system performance, demonstrating a quantitative technique to distinguish how much of a product’s performance derives from its architecture vs. its components.

Christensen (1992b) found that for architectural technologies, innovation had a significant market aspect as well as a technical aspect. Thus, the attackers advantage stems from the adopting firm’s ability to target and enter new markets in addition to the firm’s choice of technology.

Christensen (1997) crystallized the findings of Christensen (1992b), and explained that the rate of improvement in technology performance may exceed the rate at which the targeted market can consume the provided performance. Such technologies may be overtaken from below by lower-performing technologies introduced to markets with different performance demands. As these lower-performing technologies race up the technology S-curve, however, their performance soon becomes acceptable to both markets. This creates a dilemma for incumbent firms targeting those markets – further innovation in the current technology benefits neither the firm nor its current-market customers, and switching to the new technology initially provides no benefit (and in fact, may provide negative benefit) to the incumbent firms’ existing customers.
Christensen (2000) applies the concepts of performance surplus and performance shortage to industry structure, and proposes that industries supplying end-products that provide less performance than the market demands will tend to integrate vertically in an attempt to make up for the shortage of performance. Industries supplying end-products that provide more performance than the market demands will tend to stratify horizontally, with many different firms competing within each link of the value chain.

Von Hippel (1988) found that the source of innovation shifts between user and manufacturer based upon the presence of an economic incentive to innovate, the availability of non-tacit information, and agency issues.

Von Hippel (1994) examines attributes of the problem-solving process based on the location of relevant information, the cost of transporting the information, and the location of problem-solving ability. Von Hippel defines “sticky information” as having high transport costs, and finds that when information is sticky and resides at one location, problem-solving tends to occur at the site of the information. When information is sticky and resides at multiple locations, problem-solving may tend to iterate across the sites. When iteration is not possible or desired, firms may invest in either “un-sticking” the information (i.e. codifying it in such a way that it is economically transportable), or in decomposing the problem along the boundaries of the sticky information.

Von Hippel (1998) further explores problem de-composition by examining toolkits for user innovation. Von Hippel observes that when need information is sticky, firms may abandon the effort to completely understand user needs and outsource key need-related innovation tasks to their customers. They do so by re-partitioning their development process to concentrate need-related problem solving within a few tasks, and assigning those tasks to those who hold the “sticky information” (typically the firm’s customers). Customers are aided in this task by an “innovation toolkit” provided by the firm. Von Hippel observes four characteristics of good user toolkits: they enable complete cycles of trial and error learning; they require little in the way of additional user skills or training; they focus effort on the design elements unique to the user; and their output can be used directly by the firm and its production process without subjective translation or information loss.

**Design Value Chains**

Iansiti (1998) introduced the concepts of “technological potential” and “technological yield”, establishing another measurement methodology for determining the relative contributions of component vs. architectural technologies. Technological potential defines a quantitative upper bound on the
achievable product performance given the physical properties of the chosen component technologies. Technological yield relates the actual product performance to the product’s technological potential, and is a measure of the effectiveness of the product architecture. Iansiti also distinguished between product performance and project performance. Product performance describes the performance of the resulting physical product. Project performance describes the performance of the development team responsible for producing the product design, in terms of the technological yield that the product design team was able to produce in a given amount of time or with a certain amount of engineering effort.

Fine (1998) provides a description and examples of industry shifts from vertical integration to horizontal stratification and back, and proposes a “Double Helix” cycle as well as forces that drive the cycle. Fine also discusses the importance of supply chain design as a part of the overall product design process and as a core competence of the firm.

Ulrich and Eppinger (1995) is a product development textbook that provides overviews of the product design process, product architecture, and design task interactions.

Whitney (1995) summarizes how Nippondenso Co. Ltd. has met its lead customer’s (Toyota’s) demand for high variety in its production system primarily by adjusting the product development process to produce modular products. Production flexibility is then achieved through automated assembly of the modules.

Whitney (1996) offers excellent summaries of the VLSI design process and mechanical design process, and points out key differences between them (most notably: primarily single-function design for VLSI vs. primarily multi-function design for mechanical systems). Whitney argues that these differences explain observed differences in management metrics between mechanical and VLSI design. For example, for projects having similar part counts, mechanical design exceeds VLSI design by:

- About 1.7 times, for calendar duration of the design process
- About 25 times, for total effort expended in the design process

Whitney further argues that such differences will persist because of the fundamental design process differences between the two fields.

**Modularity**

Ulrich (1995) describes the role of product architecture as it relates to several important areas of importance to managers: product change, product variety, component standardization, product performance, and product development management. In particular, Ulrich discusses how modular
product architectures enable product change, within the life of an artifact, and across generations of the product. Modular product architectures can enable a high degree of product variety to be economically produced. This can occur because variety is achieved through assembly rather than through flexible component production equipment. Modular architectures make standardization possible, because standardization can only occur when a component implements commonly useful functions and when the interface to the component is identical across multiple different products.

Ulrich (1995) also relates modularity to product performance. Some performance characteristics, which Ulrich calls local performance characteristics, arise only from the physical properties of a local region of the product (for example, the brightness of a headlight). Other, global performance characteristics, arise from the physical properties of most or all of the product’s components (for example, the mass of an automobile). In general, modular architectures tend to optimize local performance characteristics, while integral architectures tend to optimize global performance characteristics.

Baldwin & Clark (2000) offer a theory that builds links among dominant modular product architectures, the economic systems that surround them, and the industry structure that is created as a result. Baldwin & Clark introduce the concept of the “option value of modularity” — that modular architectures create design options that have value independent from the particular products that currently demonstrate the architecture.

**Platforms and Option Value**

Meyer & DeTore (1999) describes the product development effort for services, as opposed to physical products. Increasingly in an information based world, what historically has been dealt with only as physical product may now be dealt with in the information realm, as a design to be manufactured. Design services, therefore, are an important class of service that are distinct from the products that derive from them. The article presents a framework for mapping competencies into a modular product platform that uses common design rules and tools, parts and assemblies. From the platform, derivative products or services can be defined.

Amram (1999) presents a primer with many examples of how to apply “options thinking” to reveal the option value embedded within the projects that your firm has undertaken, and how to arrive at a quantitative, economic value, for such options.
4. Concepts and definitions

There are several related but distinct concepts in the area of value chain analysis that can be easily confused. Since this thesis targets a managerial audience and applies concepts pulled from diverse prior research, definitions and explanations will be helpful. These key concepts will be the tools and building blocks used in the analysis section. This section points out the key concepts, provides a definition for each, and attempts to clarify the relationships, similarities, and differences among them. They are:

- Design Domain vs. Production Domain
- Vertical vs. Horizontal Industry Structure
- Designs vs. Artifacts
- Modularity in Design Modularity in Production Modularity in Use
- Fabricated vs. Assembled
- Assemblies vs. Systems
- In-House vs. Outsourced Design and/or Production
- Intermediate (derived) markets
- Performance vs. Utility
- Option value

**Design domain vs. production domain**

The design of a product is distinct from its production, and value chains exist for both. Let’s define a *design value chain* as the series of value-adding steps that result in a fully specified, designed, and manufacturable product. Alternatively, a *production value chain*, is the series of value-adding steps that physically produces the designed product. The design process consists of exercising the design value chain, and produces a manufacturing value chain. The production process consists of exercising the manufacturing value chain, and produces physical product.

We can see clearly that the design value chain is separate and distinct from the production value chain by considering examples. Cyrix, for example, engaged in the microprocessor design process, but not in the microprocessor manufacturing process. Consider also enterprise computing. The design cycle for an enterprise computing product is on the order of 2-7 years. The design value chain involves design and creation of: photolithography equipment, IC processes, integrated circuits, circuit boards, backplanes, and system software. The tools in the design process include CAD software, circuit and logic simulators, and compilers. On the other hand, the manufacturing cycle for the same product is only 4-6 months. The manufacturing value chain involves fabrication of ICs, assembly of IC’s onto printed circuit boards, and assembly of boards and other components into a completed system. The tools used during the
manufacturing process include steppers, pick-and-place machines, wave solder stations, and IC testers. The design and production value chains are clearly distinct.\footnote{Baldwin and Clark (2000) point out that design and production have not always been distinct in this way. Artisans in the past designed their work as they were producing it, perhaps with no clear idea of exactly what the resulting product would look like. The advent of mass-production introduced the need to distinguish the design processes from production processes, and indeed led to the point where production processes themselves need to be designed.}

Firms create and exercise design value chains through their product development processes and their interactions with supplier and partner firms. In some cases we observe that some firms play a disproportionate role in the creation of a design value chain in which many other firms will participate (for example, Intel in the design value chain for personal computers). The output of the design value chain is a set of designs and one or more production value chains. When we speak of value chains and the pressures exerted upon firms to change the structure of those value chains in the design domain, we simply mean that we are discussing design value chains as opposed to production value chains.

Firms also face pressures to optimize their production value chains. There is a robust set of science in the production domain that describes what can be done to increase throughput, reduce work in process, and increase resource utilization, and describes the tradeoffs inherent among such goals.

A relationship exists between value chains in the design domain and value chains in the production domain. New product development can be visualized as two production value chains, separated in time and effort by the exercise of a design value chain, as shown in the figure on the following page.
Disruption, Dis-Integration, and Modularity the Microprocessor Industry

Existing manufacturing value chain ⇒ Current product & product capabilities

Collect need information

Specify new product attributes
  New product performance demands (many attributes)

Specify product design
  Product performance targets (many attributes)
  Product architecture
  Component specifications

Construct remaining design value chain
  Decisions:
  Design new vs. Use existing components
  In-house design vs. Outsourced design

Design and prototype
  Including:
  Design of the production value chain

Turn on the production value chain

New manufacturing value chain ⇒ Tomorrow’s product & product capabilities

Relationship between Manufacturing and Design Value Chains
Vertical vs. horizontal industry structure

An industry is said to have a *vertical structure* when competition occurs among several firms for share of some end-product, the design and production of which is performed or controlled primarily within each firm. In vertically structured industries, product architectures are generally non-standard and proprietary. Design and/or manufacturing interfaces may be loosely defined, and are not typically shared across multiple firms. This situation is depicted graphically in the following figure.

![Diagram showing vertical structure with firms X, Y, and Z.

In a *horizontally structured industry*, many firms compete for share within one or more segments of an industry standard design and/or manufacturing value chain. Design and production of the end-product may be shared by many firms, some of which produce components of the product, and others which integrate the components into subassemblies or an entire end-product. Product architecture is necessarily standard and often non-proprietary. The following figure shows a horizontally structured industry.

Note that the concept of vertical vs. horizontal structure can be applied independently to value chains in either the production domain or the design domain. Value chains characterized by one type of structure in the manufacturing domain need not exhibit the same type of structure in the design domain.
Disruption, Dis-Integration, and Modularity the Microprocessor Industry

For example, the PC industry has dis-integrated to a horizontal structure in the production domain. However, in an effort to reduce the time required to develop and introduce new microprocessor products, Intel has integrated forward in the design domain, producing a largely vertical design value chain. Intel designs chipsets complementary to their microprocessor products, designs motherboards that contain them both, and introduces new standards for PC hardware connectivity that are largely unrelated to the microprocessor (e.g. USB). In fact, it may be this tightly coupled, vertically integrated design process that allows Intel to continually position itself to participate in the most profitable portions of the resulting production value chains for PC components and systems.

---

**Designs vs. artifacts**

The clear distinction made above between design processes and production processes further allows us to distinguish between a design, or detailed set of attributes describing the characteristics of a product, and an artifact, the resulting (usually physical) product itself\(^2\). Baldwin and Clark (2000) point out that artifacts have both structure and function, and are directly usable by some market. The value of an artifact is determined primarily by its intended function, the extent to which the structure of the artifact...
realizes the intended function, and the importance of that function to the market. Designs, on the other hand, are not directly usable by the market, but describe the artifact completely, in such a way that it can be produced. The value of a design derives from the option to produce it, the cost of production, and the value of the resulting artifacts in the market.

For computing VLSICs, a design consists of hundreds to thousands of gigabytes of information that specify the location, characteristics, and interconnections of the transistors that will comprise the resulting artifacts. Artifacts are the output of the VLSI fabrication process – die or packaged parts that can be physically integrated into computing products.

Artifacts are often, but not always, physical products. Software is a general class of non-physical artifact. In particular, note that software that aids or embodies a design process is itself an artifact. VLSI design is full of such artifacts – for example: place and route tools, timing analysis tools, circuit simulation tools.

**Modularity in design, production, and use**

Various definitions of modularity exist in the literature. Ulrich (1995) first proposed definitions for both product architecture and modular architecture. Ulrich defines *product architecture* as:

1. the arrangement of *functional elements* of the product
2. the mapping from *functional elements* to *physical components* of the product
3. the specification of the *interfaces* among the interacting physical components

Ulrich further defines a *modular architecture* as:

- having a one-to-one mapping from functional elements to physical components
- specifying de-coupled interfaces between physical components. Two components are coupled if a change made to one component requires a change to the other component in order for the overall product to work correctly.

Such an architecture guarantees the ability to easily change the way that functionality is implemented, because each function maps to one module, and modules can change without necessitating changes to other modules.

Christensen (2000) clarifies the requirements for a feasible modular product architecture:

- Designers must know which attributes of a module need to be specified, in order to “fit” with the other modules or systems of the overall product.

---

\(^2\) I use the vocabulary introduced by Baldwin and Clark (2000).
Disruption, Dis-integration, and Modularity in the Microprocessor Industry

- Metrics of these attributes must be defined and technology must exist to verify whether a module actually meets these specifications.

- Designers must understand how variations in the critical attributes of a module will interact with the performance of other subsystems and the product or service as a whole, so that various modules can be combined with predictable results.

The first two requirements enable a verifiable abstraction and de-coupled interface to the module, and allow module implementations to change without necessitating changes to other modules. The third requirement relaxes Ulrich's constraint of a one-to-one mapping from functions to physical components, instead requiring only that overall functionality be a comprehensible function of the design parameters of each module. This would allow modules to be combined and substituted to provide novel functionality that was perhaps unforeseen initially.

Baldwin and Clark (2000) offer yet another definition, consistent with that of Christensen:

- A module is a unit whose structural elements are powerfully connected among themselves and relatively weakly connected to elements in other units.

- Modules isolate complexity by defining an abstraction that has a simple interface. The abstraction hides information and complexity within the module. The interface indicates how the element interacts with the larger system.

Baldwin and Clark (1997) make an important distinction among different arenas in which modularity occurs, which Sako and Murray (1999) expand greatly. These arenas are consistent with the distinction made previously between the design domain and production domain:

- Modularity in design focuses on managing complexity in the design process, easing introduction of new technologies into the design process, and creating option value through potential future designs by allowing modules to be combined and designs to be changed in novel ways useful to consuming markets.

- Modularity in production focuses on optimizing production processes with respect to throughput, flexibility, and/or cost.

- Modularity in use focuses on providing wide product variety to end users by allowing users to “mix and match” modules with different attributes to meet their tastes.

Sako and Murray make two important points. First, the optimal module boundaries may differ in each of these three arenas. Second, the driving force for modularization typically differs for each of the arenas, with different individuals or groups providing the impetus for the introduction of modularity and/or for shifts in module boundaries.
Fabricated vs. assembled

Ulrich (1995) distinguishes between fabricated and assembled components, and points out their relationship to product variety and production process flexibility. Product variety in modular architectures can be achieved through assembly of various interchangeable modules. Variety through assembly usually requires little in the way of flexible production equipment. Product variety in integral architectures can be achieved by increasing the number of custom-fabricated, integral components. This can often be done economically only when production equipment is able to flexibly vary the desired component attributes during component fabrication.

This distinction is important in the context of VLSICs because, while IP cores may be considered modules in the design domain, current production processes require full mask sets flattened across all components. While IP core “modules” may be easily assembled in the design domain, the natural boundary of the production “module” is currently the die, and VLSI production processes offer little production flexibility beneath this level.

Modules vs. systems

Sako and Murray (1999) point out the potential tradeoffs between modular design and system integration. Modules optimized for one taxonomy (for example, ease of physical assembly) may obscure or hinder optimization in an alternative taxonomy (for example, overall weight). In VLSI design, modules tend to be organized primarily by the chip area that they occupy. Systems tend to span the entire die (for example: power consumption, global timing, test).

Designers can combat this problem by making explicit requirements for system integration and driving them back into the abstraction and interface specifications for the modules themselves. Thus, a VLSI module may have one design deliverable that specifies the physical layout of its transistors, and another that specifies the timing relationships among its inputs and outputs. The first facilitates physical layout of the chip (i.e. module assembly), the second facilitates global timing analysis (i.e. cross-module system integration). Both are part of the abstraction and interface of the module in the design domain.

In-house vs. outsourced design and/or production

Firms face choices about whether to engage in design themselves, to purchase design services from outside providers (guided by some set of “design guidelines”), or simply to purchase completed designs from those available in the marketplace. Firms face similar choices in the production domain. They can
Disruption, Dis-integration, and Modularity the Microprocessor Industry

themselves produce, outsource production, or purchase commodity products from established markets. Further, firms face each of these sets of choices at each level of the product architecture: whole product, modules, components.

These decisions are typically independent. That is, firms may decide to outsource production but not design, and vice versa. Also, firms can create modular architectures independent from outsourcing decisions. A firm may decide to create a modular production process simply to lower its costs or increase flexibility, with no eye towards outsourcing. It may create a modular architecture to enable selective outsourcing of either design or production. Or, the design or production of an integral product may be completely outsourced, without concern for the path of any potential future modularization.

Sako and Murray (1999) describe alternative paths that firms can take to arrive at outsourced modules (modularize, then outsource; outsource, then modularize, simultaneously modularize and outsource), and point out important consequences to the firm’s ability to control the future modular architecture and capture value.

Module hierarchies

In complex systems, the product decomposition may consist of many “layers” of modules. One of the benefits of modular architectures is that the details within modules are hidden behind the module abstraction and interface. On the surface, then, a product may consist of the integration of a dozen modules. Each of those modules, though, may in turn be either integral or modular. One firm’s modular product may be used by another firm as a component in a higher-level integral product, which may be used by a third as a module in an even higher-level modular system.

While the specific components, architecture, implementation, and interfaces within a module may be abstracted away, it is important to remember that they continue to exist, albeit at a lower level of analysis. They are created by firms and served by markets; these firms and markets may exert pressure along the value chain, and these pressures can sometimes span multiple levels of analysis.

More specifically, modules that comprise a product at one level are themselves lower-level products, and may themselves have an either modular or integral product architecture. Innovation within component technologies may make product modularity feasible when, before, only integral architectures made sense. Alternatively, a drive towards modularity within a product may create a performance deficit within one of its comprising modules, leading the module to swing away from a previously modular architecture back towards integrality.
Intermediate (derived) markets for hidden modules

The existence of module and component hierarchies lead to the creation of intermediate, or derived, markets. Baldwin and Clark (2000) call these markets for hidden modules. Hidden modules are products such as disk drives, DRAM, or software that have essentially no value as stand-alone products, but are useful and valuable parts of larger systems.

These are new markets in their own right. Firms enter to initially create them, and firms compete within them on the basis of some set of parameters valued by the market. But those valued parameters are not decided arbitrarily by customers in the market. They are derived from the value that they deliver within the context of the larger system. Thus the evolving structure of the larger system affects demand in the intermediate markets. Firms that control or influence the structure of the larger system have an impact on the products and product attributes that are demanded in intermediate markets. Likewise, firms that compete in intermediate markets may introduce product features in explicit attempts to influence the evolution of the larger system.

Performance vs. utility

Because demand in intermediate markets is derived from the value that products deliver in the context of a larger system, customers will value improvements in some product parameters more highly than others. That is, increases in performance do not necessarily equate to increases in utility or market value. Christensen (1997) demonstrates that technology development may fuel performance improvements (as measured by some parameter of performance) at a rate that exceeds the market's ability to accept the performance increase. When this occurs, the market stops valuing marginal performance increases on this attribute. Instead, the market seeks improvements in some other area. In end-product or consumer markets, the rate of required performance increase is driven by the characteristics and tastes of the customer. In a derived market, it is driven by both the characteristics of the system in which the intermediate product is used, and by the characteristics and tastes of the system's customer.

Option value

Baldwin and Clark (2000) point out another source of value above and beyond that affixed to modules by derived markets: option value embedded in the design itself. Because modular design allows modules to change independent from one another, options are multiplied. Because changes are guided by
design rules, creation of this option value can be decentralized and spread across many firms. Baldwin and Clark then use the “Real Option” theory summarized by Amram (1999) to link these mechanisms of design change to valuations in the capital markets.

Firms may, as an objective in and of itself, seek to influence the structure of the system design in such a way as to maximize the option value embedded in the design. Firms may seek to position themselves optimally to extract and capture this value in the future. Such objectives are independent from and in addition to seeking to maximize value capture through sales of existing products in existing markets.

\footnote{Here I use the term “utility” in the micro-economic sense.}
5. General Mechanisms for Shifts in Value Chain Boundaries

From the prior research I now extract and summarize three “strong mechanisms” that may lead to shifts in value chain boundaries. Managers can view these as “forces” acting on the boundaries of the value chain. Each mechanism represents a general class of value creation and value capture opportunities. A force is strong from a particular mechanism when stakeholders along the value chain, either individually or collectively, recognize an opportunity to apply the pattern of value creation and capture represented by the mechanism to their particular situation, at a particular boundary in a value chain in which they have an interest.

When multiple forces align, or when forces are particularly strong, value chain boundaries move in response. Such shifts may be accompanied by changes in the degree of modularity or integrality on each side of the value chain boundary, and by changes in the supplying and consuming firms around the boundary. That is, the shift in the value chain boundary leads to a shift in industry structure.

Based on prior research, I propose that the following general mechanisms lead to forces that exert pressure on value chain boundaries:

- Mismatch in performance supplied vs. performance demanded
- Sticky information and agency issues
- Growth options

**Performance supplied to market mismatches performance demanded by market**

The defining characteristic of this mechanism is that, on at least one attribute of performance (e.g. speed, functionality, reliability, etc.), the design process delivering a design to some boundary in the design value chain either over-delivers or under-delivers. That is, the performance delivered for some attribute mismatches the performance required by downstream designers to create, through exercising the remainder of the design value chain, an end product whose utility (and, therefore, value) in the marketplace is maximized.

This mechanism derives primarily from the work of Christensen and Verlinden (2000). I have extended their framework to take into account design boundaries within a design value chain. I will then illustrate the extended framework with examples from the real world.
In some cases an upstream design process produces designs having more performance on some attribute than downstream designers require or are able to use. When this happens, the value of the marginal performance delivered (for that attribute) becomes small. The marginal utility to downstream designers of increases in this performance attribute is small. This creates three opportunities for resolution. First, the design supplier can modify the upstream design process, trading away the excess performance in exchange for an improvement in some other performance attribute that is valued more highly by the downstream designers. Second, the consuming designer can modify the downstream design process to transform the “excess” performance into improvements in some other performance attribute that is valued more highly by designers further downstream, or by the end consumer. Third, this design value chain interface can be substituted, or eliminated completely.

In other cases an upstream design process produces designs having less performance on some attribute than downstream designers require or are able to use. When this happens, the value of the marginal performance delivered (for that attribute) remains large. This creates three opportunities. First, the upstream supplier can modify the upstream design process, trading away excess performance in other performance attributes in exchange for additional performance in the deficient attribute. Second, the consuming designer can modify the downstream design process to do the same. Third, a combination of these two could occur, which could potentially shift the design boundary, or even remove it from the design value chain.

An example. Later we will see that an important part of both the design and production value chain is packaging raw silicon die into some carrier suitable for mounting on a printed circuit board or otherwise interfacing to the rest of the system. Important performance metrics for the packaging task include number of pins and cost per pin. An important business metric as relates to both package design and production is package cost as a percent of total VLSIC product cost.

Examination of these metrics reveals a performance mismatch. The combination of number of pins and cost per pin using current technologies mismatches the requirements for package cost as a percent of product cost required by the business. These creates pressure for on value chain boundaries, for example to reach across traditional boundaries and subsume more and more functionality onto a single die:

“Future semiconductor component products must be targeted to maintain constant or decreasing prices and the average number of pins per unit increases at 11% while the average cost per pin decreases at only 5%, then the following will occur:

1) the average packaging share of total product cost will double over the 15-year roadmap period
Disruption, Dis-Integration, and Modularity the Microprocessor Industry

2) the ultimate result will be greatly reduced gross profit margins and limited ability to invest in R&D and factory capacity

This conclusion is one of the drivers behind the industry trends to reduce the overall system pin requirements by combining functionality into Systems-on-Chip (SoC) and through the use of multi-chip modules, bumped chip-on-board (COB), and other creative solutions.”

Note that these pressures will create winners and losers. Firms competing in areas of the value chain that become more tightly integrated with and subsumed by other parts of the chain must change, or face extinction.

The following table summarizes an example for each of the performance mismatch cases. More detailed descriptions of each of these examples are included in the appendix.

<table>
<thead>
<tr>
<th>PERFORMANCE SURPLUS</th>
<th>PERFORMANCE DEFICIT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Addressed UPSTREAM</strong></td>
<td><strong>Addressed DOWNSTREAM</strong></td>
</tr>
<tr>
<td>“I could deliver more performance, but no one would pay...”</td>
<td>“I just bought performance that I don’t need. How can I use it anyway?”</td>
</tr>
<tr>
<td>e.g. web search speed → result relevance directhit.com</td>
<td>e.g. multimedia instructions, applications Intel MMX, RealNetworks RealPlayer</td>
</tr>
<tr>
<td><strong>Addressed at BOUNDARY</strong></td>
<td><strong>Addressed at BOUNDARY</strong></td>
</tr>
<tr>
<td>“We need to focus on other types of performance. Can we simplify away this interface while maintaining performance?”</td>
<td>“We need to deliver extra performance. Has this interface become a limiter?”</td>
</tr>
<tr>
<td>e.g. StarOffice, ASPs, Compute Utility</td>
<td>e.g. IP on SONET</td>
</tr>
</tbody>
</table>

Note: The table above exemplifies the mismatch between supply and demand in the context of the microprocessor industry.
Notes about the Performance Mismatch Mechanism

First, note that these forces give rise to opportunity both for incumbent firms and for new entrants. In vertically integrated industries, these forces will likely lead to design process changes within the one or two incumbent firms. However, left unaddressed, they may lead to opportunities significant enough for firm entry to become viable. In horizontally structured industries, such forces may lead either to changes in the design process of incumbent firms, or to firm entry.

Second, note that the firm or firms that actually invest in changing their design process, and the resulting changes to the structure of their intermediate products, may be removed by several stages in the value chain from the customer that “feels the pain” most acutely. The forces that arise may initiate at the location in the value chain where the pain is most acute. But, they may be transmitted forward and backward along the chain in search of that part of the overall design process that can be most efficiently modified to address the problem. Alternatively, they may be transmitted forward and backward along the chain to firms having relatively fewer sources of power than the customer that “feels the pain” most acutely.

Third, note that in each case the general sequence of the mechanism’s operation is as follows. A performance mismatch is discovered at some boundary in the value chain. This leads to design process changes that attempt to resolve the mismatch. The design process changes occur either upstream from, downstream from, or around the boundary where the performance mismatch occurred. These design process changes result in shifts to existing design process handoffs, additions to the value chain (that is, new design process handoffs), or deletions from the value chain (that is, obsolete design processes and handoffs).

Fourth, note that these design process changes might incorporate any of the types of innovation described by Henderson and Clark (1990), as appropriate to address the performance mismatch that has been discovered.

Summary. The performance mismatch mechanism increases efficiency by trading away design methods that produce unnecessary product attributes in favor for those that do not. This occurs through individual stakeholders in the current value chain. Those near a value chain boundary recognize the inefficiency inherent in a performance mismatch, and either communicate it upstream and/or downstream, negotiate directly for ways to resolve it, or opportunistically change their own design flow to be more congruent with the demands of the market.

**Sticky information and agency issues**

The second general mechanism for shifts in value chain boundaries that I propose derives from the work of Eric VonHippel. While the performance mismatch mechanism deals with mismatches in the performance of the design itself, or in the design process, this mechanism addresses an inability to explicitly comprehend and/or efficiently communicate which performance attributes should be considered, and what levels of performance on each attribute are truly required. Sometimes participants at some stage in the value chain have a limited and developing explicit understanding of which product and/or design process performance attributes matter, and of the mapping between those performance attributes and the utility of the intermediate product (and, indirectly, the utility of the end product). Alternatively, this mapping may be clear but only tacitly understood, such that communicating the mapping to others along the value chain is expensive and inefficient. Stated another way, participants along the value chain sometimes have a limited and developing understanding of how they will value a design and/or the design process that produces it based upon its features or lack of features, and the performance of each feature.

Remember that Utterback (1994) made precisely this observation – that innovation begins with product feature experimentation, then shifts to process and cost improvements – in the context of end products. Tushman and Murmann (1998) extended his thinking to include product structure.

I propose that the general sequence of this mechanism’s operation is as follows. A stakeholder at some point along the value chain notices that a seemingly disproportionate amount of resources are being spent on specification of product features and performance attributes, or on design rework as a consequence of incorrect or incomplete specifications. Alternatively, an outside observer might notice the same. This individual investigates the structure of information flow along the relevant portions of the value chain and constructs a map describing information content and flow within the value chain. This map describes where in the design process information resides, its cost of codification and transport, and the necessity of transport given the current design process. This map might be created explicitly, but it may also be only tacitly understood. Based upon the map, the individual either proposes a change to the design process to shift design boundaries such that expensive information transfer is reduced, or proposes an addition to the design process that codifies information in such a way that its cost of transport is reduced.

Note that this mechanism inherently modularizes the design process by necessarily creating module abstractions and interfaces to the modules. The process of “unsticking” information that VonHippel
Disruption, Dis-integration, and Modularity the Microprocessor Industry

describes is consistent with the process of modularization described by Baldwin and Clark (2000). The author suspects that they are describing parts of the same phenomenon using different terminology. Localization of information at a particular site so that problem solving can occur (VonHippel’s terminology) produces the result of “powerfully connected within and weakly connected across” modules (Baldwin and Clark’s terminology). “Unsticking” information, that is codifying it for easy transport, (VonHippel) amounts to explicitly identifying those portions of the information that have a large impact on performance attributes that matter to customers. This is the same as “defining an abstraction that has a simple interface” to simplify the design task at an acceptable cost in terms of performance (Baldwin and Clark).

The basic activity is one of creating abstractions and interfaces that remove portions of the design space while highlighting the performance metrics that matter to customers and retaining the portions of the design space that are most likely to deliver on those metrics.

Baldwin and Clark (2000) point out that:

“This kind of rationalization of a design and its corresponding task structure is the first step toward modularization. Rationalization can go on for a long time – indeed, some designs never get out of this phase. The process takes time, because every move needs to be based on knowledge gleaned from experience with previous designs or scientific understanding of the underlying parameter interdependencies.” (p. 69)

To clarify the design boundaries and/or to help explicitly codify information, the concerned stakeholders may introduce user toolkits that strongly facilitate the new design process.

**Summary.** The sticky information mechanism increases efficiency by realigning who does what. Either the design process is partitioned more strongly along the boundaries of sticky information, or the design process is changed to inexpensively codify and reduce the stickiness of this information.

**Growth options**

The sticky information mechanism ensures that valued performance attributes are understood and clearly communicated along the value chain. The performance mismatch mechanism ensures that resources along the value chain are appropriately applied to maximizing these valued performance attributes, and not engaged in non-value producing tasks. Both of these mechanisms attempt to maximize the value of a given end product. The growth options mechanism, on the other hand, creates value by modularizing the design in such a way that design resources can be more strongly leveraged across several products or markets. The design modularization creates this resource leverage by shifting value

Disruption, Dis-Integration, and Modularity the Microprocessor Industry

chain boundaries towards intermediate products that are usable and valuable to a wide audience, perhaps outside the immediate firm, value chain, or industry. Modularization can also enable the concurrent application of multiple sets of design resources, from the same firm or from different firms.

I borrow this mechanism almost completely from Baldwin and Clark (2000). From their work I distill two primary opportunities for growth through the modularization of designs: the opportunity to build scale, and the opportunity to bridge markets.

The opportunity to build scale. Because taking action to modularize a design necessarily creates clearly defined abstractions for and interfaces to its components, modularization allows design teams to become decoupled from one another, both in space and in time. Modularizing a product has a profound impact on the processes used to design it. Once the module abstractions and interfaces are clearly defined, design teams need not communicate extensively with one another, nor be co-located. Their work can even be separated in time. At boundaries in the design value chain, this phenomenon changes the design process. Without modularization, design tasks occur serially, and active communication and negotiation regarding design parameter interdependencies is required. After modularization, design tasks can occur in parallel, or they can be completely independent in time. In this latter case, the design task at value chain boundaries reverts to selecting the most appropriate incoming design from a market of designs that have already been created.

Such a transition to market-based design processes have important implications in terms of network effects, and can have important implications for the scale of the industry. The market for designs to be used at some stage of the design value chain may be initiated within just one or a few firms. Once established, however, the definitions of the design abstractions and interfaces allow input to the marketplace to become decentralized and spread across many firms. Network benefits may drive the value of the marketplace to exceed that which could be created by any single firm.

By modularizing design processes, then, firms can create a web of industry stakeholders, each contributing resources to create designs that populate a series of markets along the value chain. The industry grows to a much larger scale than could ever be achieved by just a few firms.

The final opportunity is then for value capture within this enlarged industry. Firms which are able to dominate portions of the overall value chain that scale directly with the size of the industry will capture a disproportionate share of industry value. Firms might establish and maintain such dominance by maintaining control over the structure and terms for participating in the marketplaces for intermediate designs along the value chain. For example, in Microsoft's case (anti-trust issues aside) the windows
Disruption, Dis-integration, and Modularity the Microprocessor Industry

platform itself defined the interfaces to which firms must conform in order to participate in the marketplace for windows-compliant applications. Microsoft's control of this platform (and consequent control of the terms of entry into the marketplaces that it created and enabled) has allowed Microsoft to long maintain a dominant position in both PC operating systems and office application suites.

Structuring designs to establish market and build scale is also consistent with Meyer and DeTore (1999), although their argument focuses on service offerings.

The opportunity to bridge markets. The creation of modular designs creates another opportunity— that of choosing module boundaries which broaden the applicability of intermediate designs to other products and value chains. Appropriately chosen, design or design process boundaries can sometimes enable designs or design processes to be applied to additional markets, perhaps even within other industries.

For example, Advanced RISC Machines (ARM) drew its design boundaries around its RISC microprocessor cores, and partners with other firms who integrate its design into VLSI systems tailored for market-specific applications. The approach has propelled ARM into several markets including cellphone handsets (where ARM owns a 70% share), set-top boxes, personal digital assistants, and internet appliances. The licensing approach that ARM has taken allows ARM to focus its competencies on microprocessor core design, while the array of partners that it has assembled— including 38 semiconductor manufacturing partners, 25 design partners, and 45 software partners⁴— allow ARM cores to be relevant and valued in more applications and markets than ARM could ever take on itself.

This approach is also taken by new entrant Tensilica. Tensilica has modularized the microprocessor design process to such an extent that they are able to programmatically create a custom microprocessor core whose instruction set is tailored to optimize the application performance required for each particular customer. Notably, Tensilica has not sought out or partnered to establish a dominant “killer app” for its product. Instead, the firm ensures that the boundaries around its design process enable multiple customers in diverse markets to easily create and apply Tensilica-designed cores.⁵

---

⁵ Interview with Chris Rowan, Tensilica CEO, 01/12/2000.
**Mechanism Summary**

The following table summarizes the three general mechanisms for shifts in value chain boundaries that I have proposed.

<table>
<thead>
<tr>
<th>Mechanism</th>
<th>Objective</th>
<th>Action</th>
<th>Desired Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sticky Information</td>
<td>Remove inefficiency by increasing understanding of <em>which types of performance are valued</em>, reducing the cost of communicating this information, and removing design cycles.</td>
<td>Re-partitioning of design activities among resources.</td>
<td>Optimal translation from design effort expenditure into captured design and product value.</td>
</tr>
<tr>
<td>Performance Mismatch</td>
<td>Remove inefficiency by shifting design resources from less valued performance-generating tasks to more valued performance-generating tasks.</td>
<td>Re-structuring, addition, or deletion of design activities.</td>
<td>Optimal translation from design effort expenditure into captured design and product value.</td>
</tr>
<tr>
<td>Growth Options</td>
<td>Leverage resources</td>
<td>Partition design boundaries for wide applicability.</td>
<td>Design activities relevant and valuable to a wider market.</td>
</tr>
</tbody>
</table>

There appears to be a natural sequence for application of the mechanisms. First, when a dominant design, or perhaps a dominant *design process* has not yet been established, the sticky information mechanism improves the understanding along the value chain of which product and design process performance attributes are important. Once this understanding is clear and information regarding performance expectations is flowing up and down the chain as needed, the performance mismatch mechanism ensures that resources are optimally allocated on value-producing design tasks. As business within each link of the design value chain matures, entities begin to apply the growth options mechanism in an attempt to create and capture even more value from their limited design resources. Such application may in turn create a new performance mismatch that needs to be resolved elsewhere along the value chain.

Entities might choose to apply technology (leap S-curves, introduce a wholly new technology, or pursue an architectural innovation) in the course of exercising *any one of* these general mechanisms. This process may itself create the need to cycle back through the application of the other mechanisms.
6. Analysis - application to microprocessor industry

Now, let's apply the concepts and general mechanisms discussed above to the microprocessor industry. The subsequent analysis is structured as follows. First, I describe the design process and the design and production value chains typical for microprocessors and similar related VLSI. I distinguish among several common variants of the design and production process, pointing out the key differences and areas of application for each. This sets the stage to describe the markets and classes of microprocessor products that currently comprise the industry.

Second, I apply common strategy analysis tools at the industry level. I assess the overall attractiveness of the industry using a Porter's forces model. I describe the relevant technology S-curves, performance measures for each, and the current position of the industry on each S-curve. I examine network effects in each of the major markets, and discuss sources of appropriability and complementary assets.

Third, I assess the demonstrated strategy of selected industry incumbents using a three-pronged model of value creation, value capture, and firm competency and organization. I also assess the strategy of notable entrants to the industry. For each entrant I discuss whether entry might evidence application of one of the "general mechanisms" for shifts in value chain boundaries discussed previously.

Fourth, I explore each of the "general mechanisms" for value chain evolution in the context of VLSI value chains. To do so, I describe the key boundaries (interfaces) within the VLSI design and production processes. Then I test applicability of each of the three general mechanisms at the key design and production process boundaries.

Finally, I propose some scenarios for both market evolution and design process evolution that are consistent with this analysis.

Design process description

In this section I distill the microprocessor design process to its essence. For an alternative description see Whitney (1996). Even when distilled, the design process is quite complex. The sometimes loosely defined boundaries between these steps, and the number of iterations required within the design process, make it easy to understand why the microprocessor industry is currently dominated by vertically integrated incumbents. The steps required to create a microprocessor design are currently:
These steps can be partitioned into four major sets of design activity: platform selection and design, circuit design and implementation, prototype value chain design, and production value chain design.

In platform selection and design, the design team creates a stable set of resources and design rules upon which many different variants of microprocessor products could conceivably be created. The resources that comprise the platform include: the instruction set architecture, the IC process and associated design rules, design libraries, and CAD tools.

The requirements that drive platform selection may include market-driven technical performance requirements, such as integer performance, floating point performance, and backward-compatibility with an installed base of software, as well as business-driven requirements such as target packaged part cost and production volume ramps. The instruction set architecture defines the interface between the resulting microprocessor and the software that it will process. The IC process and the associated process design rules define the interface between the IC design and IC production processes. The functional decomposition defines the key components or modules that the microprocessor will need to contain in order to meet the market and business requirements while conforming to the process design rules and the instruction set architecture. Since these three sets of design choices are so tightly coupled, they frequently are taken together, or iterate repeatedly.

Once the functional decomposition is understood, design libraries of common gates, cells, or modules are designed to aid in the downstream design process. Such libraries then become resources of the design platform, and can drive alternative functional decompositions in the future. CAD tools are also
created or tailored to meet any particular needs that arise as consequences of the selections made within the design platform (i.e. IC process, instruction set architecture or microarchitecture, libraries, etc.)

In circuit design and implementation, each of the modules in the functional decomposition is implemented at decreasing levels of abstraction, and the resulting modules are integrated and verified to form a complete IC-process-level design. Module implementation at each of the levels may include work supportive of the integration and verification efforts (for example, local timing simulations that are supportive of global timing analysis). Module implementation usually leverages the previously created resources of the design platform, including cell libraries and CAD tools. The output of this stage completely specifies the initial, simulation-verified design of the microprocessor IC.

Prototype value chain design provides a means for the initial design to be transformed into initial physical product that can be further tested through use in systems. The production test suite provides a means to distinguish parts manufactured without defects ("good" parts) from those manufactured with defects ("bad" parts). The packaging solution defines the physical interface between the microprocessor IC and the system. The design of prototype value chain execution processes includes the prototype value chain specification (What are the steps? Which vendors perform which steps?), as well as the management processes required to exercise the value chain and produce parts across all the vendors involved.

In Production value chain design, the design team verifies and corrects its design based upon feedback from the output of the prototype value chain, and creates the means to ramp production to levels specified by business requirements. The masks are the interface between the design process and the production process, and encode the design output in a format known to both the design team and the production team. Mask generation is the first step in exercising the value chain; once created, masks can typically be re-used to produce many physical parts. To fabricate wafers, the prototype value chain is exercised. The resulting ICs are packaged, physically verified in systems, and corrected if necessary. New masks are created and the process is repeated until the design is perfected. Finally, the value chain design is augmented or replaced as necessary to support the production ramp.

**Design & production value chains**

Typical design and production value chains, and the relationship between them, are shown in the following schematic diagram.
Note the handoffs and clear distinction between the design process and the production process. Some looping between the two processes is usually required, because microprocessors typically cannot be fully verified using only simulation (some physical verification of electrical and functional behavior is required). This requires a prototype value chain (which must be designed). At some point, either the prototype capacity is scaled up or an alternative production value chain is turned on to accommodate the ramp-up to production volumes.

**Variant design processes and applications**

**ASIC vs. custom VLSI Design.** Whitney (1996) contrasts the ASIC design process with that of custom design. In the mid 1970’s, Carver Mead and Lynn Conway demonstrated, in the Mead-Conway method, how to separate element design from system design. This enabled a top-down VLSI design approach in which components, after being initially designed by electrical engineers, could be combined algorithmically based upon the specifications of a logic designer. Prior to this time, VLSI design required
the full attention of electrical engineers (EEs) who understood the electrical behavior of each circuit element and designed each device anew in order to build up the integrated circuit. After Mead-Conway, it was possible for a single non-EE to specify an IC at a high level, and for a suite of software tools to build up the chip by combining pre-designed “standard cells”.

Von Hippel (1999) describes LSI Logic’s move to offer a top-down IC design tool suite to its customers as an example of a toolkit for user innovation. The move allowed customers to cycle on their design definition, test it, and ensure that critical design specification information was captured correctly in the design. This eliminated costly errors late in the design cycle. Customers were able to model their designs using boolean logic, or even higher level behavioral languages with which they were already familiar. After finishing the design, LSI’s set of tools would translate it automatically into the set of data required by the IC manufacturer to implement the design.

However, this flexibility comes at a price. For several technical reasons, such designs tend to take up more silicon area and run at lower frequencies than designs overseen more completely by teams of electrical engineers. Users of LSI’s approach were trading away the performance of the resulting design (as measured by die size, cost, and operating frequency), in exchange for time-to-market, and the ability to specify and test the design themselves. For designs requiring only low to moderate performance with respect to state-of-the-art VLSI manufacturing processes, this tradeoff made sense and created much value for LSI and its customers.

For certain classes of design, however, this tradeoff did not make sense. The performance demands of customers of high-end compute servers and technical workstations mandated a different design style for the microprocessors used in such products. To meet these performance demands while retaining as much design efficiency and time-to-market advantage as possible, microprocessor design firms used a “custom design” methodology. This methodology blended LSI’s fully automated (and correspondingly constrained) approach with the ability for teams of EEs to specify new classes of underlying devices, and to hand-tailor specific circuits to enhance operating frequency or minimize area. For example, many circuits were separated into well-defined datapaths and separate control logic. Underlying datapath and control cell libraries were optimized for their particular use. If a library cell did not completely meet a designer’s needs, a skilled EE could simply copy and modify it as appropriate. All of these approaches helped to optimize performance, but increased time to market.

Whitney (1996) points out that for today’s large and complex systems, the user toolkit approach is reaching its breaking point. This occurs for two reasons. First, today’s processors have become so large
and complex that the “simple” approach of using unoptimized library elements leads to an intolerable loss of space. Designs created in this manner simply do not fit on the maximum die area that can be processed today. Second, the underlying electrical elements in today’s advanced semiconductor processes have become so small, and the wires that connect them have become so long and closely spaced, that the electrical effects introduced by combining individual elements into a system can no longer be ignored. Therefore, successful design requires either vastly more sophisticated automated tops-down software, simplification of the design tasks to automate, or direct involvement of skilled EEs in the design process.

This observation is mirrored in ITRS(1999):

“The increasing complexity of design requires a corps of highly skilled and more broadly trained designers, and computer-aided design tools that take into account factors that could be ignored in previous technology generations.” (page 35)

However, subject to the performance constraints described above, and especially for applications that do not require leading edge gate counts or performance, ASICs offer the freedom of fairly automated discrete design into an integrated, monolithic chip. Unit costs (as compared to non-integrated designs) and quality-related problems are thus dramatically reduced. ASIC design has historically involved significant fixed costs: costs of the CAD (Computer-Aided Design) tools used to design the chip, costs of the masks to manufacture the chip, and costs of the design itself, usually taking months and even years of dedicated engineering effort. These costs have traditionally made ASICs cost-effective only for relatively high production volumes, normally in the range of several hundred thousands to millions.

Semi-custom and gate array. To reduce the high fixed costs associated with ASICs, and especially to improve their time-to-market (prototype lead times often are larger than 10 weeks), semi-custom design and production processes were invented. These chips consist of an array of pre-defined blocks that can be customized for the user's particular function without undergoing the whole normal ASIC manufacturing process. In the case of the so-called gate arrays, only the one or two last metal layers are used to appropriately connect the base cells. This means that a large supply of customer-independent product can be partially manufactured. The fixed array of gates are processed, leaving only the top level interconnect process steps to vary per customer. This reduces prototype cost by about 80% and the prototype manufacturing cycles by more than 50% compared to normal ASICs. Again, such improvements do not come without tradeoffs. Since the arrays of gates are fixed and customer independent, the customer’s logic must be mapped to existing gates in their fixed positions. This sometimes leads to sub-optimal gate placement and inefficient routing between the gates, which in turn
leads to lower operating frequencies and performance. Further, the types of gates are fixed and the mixture of gates required in the customer’s solution may not exactly match the mixture of gates available in the fixed array. Thus, gates must sometimes be combined or used sub-optimally in order to implement the customer’s logic. Alternatively, a larger fixed array may be used, allowing a better mapping between the customer’s logic and the available gates, but leaving some gates unused. A final tradeoff is a consequence of the space inefficiencies of gate array implementations relative to ASICs and full custom designs. The maximum die size that can be produced by a particular IC process will hold less functionality in a gate array than in an ASIC or full custom implementation. Therefore, gate array implementations often must be partitioned across multiple chips, when an alternative ASIC or full custom implementation fits on a single chip.

Gate arrays are typically used where required performance in terms of functions per chip and operating frequency are low relative to the state of the art, where volumes are low or moderate, and where there is high pressure for fast time to market and low part cost.

**Field programmable logic.** The great breakthrough during the 1990s in chip customization was the introduction of the Field Programmable Gate Arrays (FPGAs), which basically are gate arrays in which the local customization is done by the user herself and stored in memory cells. With this technology, the cost of the prototypes is dramatically reduced to the cost of the chips (starting in $10 apiece), and the time to obtain a working prototype is reduced to seconds (the time to download the configuration into the programmable chip).

Obviously, programmability is not for free. First, the programmable resources take chip space which could be used otherwise, thus increasing the chip area and its cost accordingly; and second, the programmable interconnections between the pre-defined logic cells are considerably slower than the fixed ones in their gate arrays or full ASICs counterparts. Therefore, FPGAs are only an alternative for production in the case of prototyping, low or medium production volumes, or fast time-to-market applications. However, as technologies get smaller and faster they start to surpass the current market needs, making FPGAs cost effective for increasingly larger production volumes.

**System on a Chip.** The ever-increasing number of transistors that can be integrated onto a single die has made it possible in recent years to integrate all of the electronics of an entire general-purpose computing or application-specific system onto a single chip. SOC capabilities promise to reduce costs by eliminating or drastically reducing board-level integration and test.
This has consequently led to pressures to expand IC process and VLSI design capabilities to include multiple circuit classes and types of components on a single die including: analog (e.g. amplifiers, sensors), digital logic, radio-frequency (RF) circuits, digital signal processors (DSPs), field programmable logic, and microprocessor cores. Including capabilities to support such diverse circuit disciplines within a single IC process and VLSI design platform (libraries, CAD tools) has proved to be a significant technical challenge that no firm to date has completely addressed.

Current market and product class descriptions

The value of semiconductors consumed worldwide in 1999 was $149.4 bn\textsuperscript{6}. About $130 bn corresponded to integrated circuits (ICs) as opposed to discrete components. The figure on the following page details revenue partitioning within the semiconductor industry.

About half of world semiconductor consumption by revenue derives from embedded systems\textsuperscript{7}. An embedded system consists of "hardware and software which forms a component of some larger system and which is expected to function without human intervention. A typical embedded system consists of a single-board microcomputer with software in ROM, which starts running some special purpose application program as soon as it is turned on and will not stop until it is turned off (if ever)."\textsuperscript{8}

The Semiconductor Industry Association distinguishes between MPUs (Microprocessor Units) and MCUs (Microcontroller Units). MCUs are targeted at the embedded market and have some form of non-volatile on-board memory that can be programmed to store the embedded software application. MPUs are (initially) targeted at the general purpose computing market. They may contain cache memory, but typically do not contain non-volatile storage.

\textsuperscript{6} Semiconductor Industry Association
\textsuperscript{7} HALFHILL (1999), page 5.
\textsuperscript{8} Definition from The Free On-line Dictionary of Computing, http://foldoc.doc.ic.ac.uk/, Editor Denis Howe
Disruption, Dis-Integration, and Modularity the Microprocessor Industry

Total processor revenues (MPU & MCU) for 1999 were $36.9 bn. Of this, just under three-quarters ($27.2 bn) derived from sale of some 328 million MPUs, while about a quarter ($9.7 bn) derived from sale of 4.4 billion MCUs. Half of MPU units (about 164 million) are 32-bit CISC processors destined for the PC market. 27% of MPU units (88 million) are 32- or 64-bit RISC processors used in PCs or servers. The remaining 23% (75 million) are 8- and 16-bit processors used in embedded applications.

1999 ASIC revenues were $10.2 bn. Over 50% of ASICs include IP (Intellectual Property) blocks, most of which are microprocessor cores, or are used together with a microprocessor in the final application.

1999 Field Programmable Logic Device revenues were $2.9 bn.

---

9 Source: Integrated Circuit Engineering's ASIC Status 1999
Management of Technology Thesis 44/99

The dominant market for semiconductors is data processing electronics, which consumed $64 \text{ bn}$ of semiconductors in 1998, just under 51% of worldwide production. Of this, most (71%) is attributable to computers (PCs and PC motherboards, workstations, and servers), which consumed $45 \text{ bn}$ of semiconductors, or about 35% of worldwide production. Other major semiconductor consumers in 1998 were communications electronics (e.g. digital cellular handsets, mobile telecom infrastructure) at $31 \text{ bn}$, and consumer electronics (e.g. televisions, VCRs, video game controllers, and digital set-top boxes) at $22 \text{ bn}$.

This figure depicts the major markets in which semiconductors are consumed.

The following classes of semiconductor products are relevant for the purpose of this thesis. They are adapted from ITRS(1999), which extracts the categories from observations of actual dynamics of microprocessor product lifecycles in high-end and entry-level general purpose computing markets as well as embedded computing markets.

**High-Performance MPU.** High-performance MPUs integrate leading-edge processor cores with large amounts of cache (either on-chip or nearby, such as on a multi-chip processor module) in a state-of-
the-art IC process. They are initially used primarily in low volume markets (servers and workstations) for high-performance, general purpose computing systems. Examples of high-performance MPUs are Intel’s Pentium Pro and HP’s PA-8700 microprocessors.

**Cost-Performance MPU.** Cost-performance MPUs retain a leading-edge processor core and a state-of-the-art IC process, but sacrifices large amounts of cache. This reduces cost at the expense of lower general-purpose computing performance. Cost-performance MPUs are used primarily in the high-volume entry-level PC market. An example of a cost-performance MPU is Intel’s Celeron-brand “Covington” microprocessor.

Sometimes microprocessor designers include features in the processor microarchitecture to allow the processor core to be used in either the high-performance or cost-performance markets. A good example is Intel’s “Deschutes” processor core, which could be configured to use any of an expensive and high-performance level 2 cache implemented within the processor module from custom SRAMs, a cheap low-performance level 2 cache implemented outside the processor module from commodity SRAM, or no level 2 cache at all.

**Embedded MPU/MCU.** Embedded MPUs are extremely cost-optimized processors used primarily in the embedded systems market. They fall into two categories: slightly altered versions of historic cost-performance MPUs that have been shrunk in size (and, correspondingly, cost) into a newer IC process, and microprocessor cores targeted from the outset at the embedded market. Examples of the former include Intel’s 486 and almost all of MIPs’ product history (for example the R4000). Examples of the latter include ARM’s StrongARM core, Hitachi’s SH product line, and NEC’s V8xx products. This latter category of 32-bit and, in some cases, 64-bit microcontrollers was small in 1999 ($824 million), but experienced extremely strong growth (91%).

Embedded MPUs and MCUs enter markets as either packaged, commodity parts, or as Silicon Intellectual Property (SIP) in the form of design information for the microprocessor core that can be integrated into a custom ASIC (e.g. ARMs StrongArm core).

**ASIC.** At over a third as large as 1999 worldwide MPU revenues, and slightly larger than the entire value of 1999 worldwide MCU production, the size of the ASIC market is significant. 1999 growth

---

10 Pentium Pro integrated a Pentium Pro CPU die with a custom level 2 cache chip, containing either 256K or 512K of SRAM, within a single ceramic processor module. PA-8700 integrates 2.25MB of cache memory with the processor core on a single die.

11 Covington is simply a Pentium-II processor with all level 2 cache removed from the processor module.
in ASICs, at 35.4%, was over 3 times as strong as MPU’s 9.7% growth. ASICs remain a critical product class that either complements standalone microprocessors (when the ASIC implements differentiated and application-specific logic, protocols, and/or interfaces), or substitutes for them (when the microprocessor core is integrated onto the ASIC itself). As a product class, ASICs face a fundamental tension between the scale economies provided by general-purpose computing products and the differentiation provided by ASICs. As functions per chip continue to increase, the magnitude of this tension will necessarily increase.

Note that the dominant product class in this market is not the ASICs themselves (which are by definition highly differentiated, and fragmented), but the design platform that enables their efficient design and production. That is, a class of ASIC products is defined by design tools, the design processes that are embodied within such tools, and the SIP represented by the design platform’s cell libraries and compatible functional cores.

**System-on-a-Chip (SoC).** As stated earlier, SoC products take the ASIC design process to the extreme by including multiple circuit classes, types of components, and even IC processes within the same design, targeted for production on a single chip. At this time, SoC is an emerging market. Examples of firms enabling ASIC and SoC product classes are Virtual Silicon\(^\text{12}\) and Cadence\(^\text{13}\).

\(^{12}\) See [http://www.virtual-silicon.com](http://www.virtual-silicon.com)
Analysis tools

Porter's forces

Michael Porter offers a “five forces” model for analyzing the attractiveness of an industry based on the ability of firms to influence prices and earn supernormal profits. The model is based upon the forces on price that derive from the relative power of suppliers, rivals, buyers, barriers to entry, and substitutes. To these I have added two additional commonly considered factors: the strength of complementary products, and the degree of growth in the industry. The following table summarizes the results of the detailed analysis to follow. The analysis indicates that VLSIC design is an attractive industry.

<table>
<thead>
<tr>
<th>Force</th>
<th>Factors</th>
<th>Overall Influence on Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supplier power</td>
<td>• Scale requirements</td>
<td>Moderate negative force</td>
</tr>
<tr>
<td></td>
<td>• Incremental entry precluded</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• High firm concentration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• VLSIC firms integrate backwards to design platforms</td>
<td></td>
</tr>
<tr>
<td>Rivalry among firms</td>
<td>• Intense, but based on factors other than price (except in low-end markets)</td>
<td>Moderate negative force</td>
</tr>
<tr>
<td>Buyer power</td>
<td>• Fragmented</td>
<td>Weak</td>
</tr>
<tr>
<td></td>
<td>• Cyclical market; limited power through futures contracts</td>
<td></td>
</tr>
<tr>
<td>Barriers to entry</td>
<td>• Massive for integrated design and foundry</td>
<td>Weak</td>
</tr>
<tr>
<td></td>
<td>• High entry costs even for fabless design firms</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Resource constrained – difficulty in sourcing design teams</td>
<td></td>
</tr>
<tr>
<td>Availability of substitutes</td>
<td>• Limited substitutes</td>
<td>Weak</td>
</tr>
<tr>
<td></td>
<td>• VLSIC content of products growing over time</td>
<td></td>
</tr>
<tr>
<td>Complementary products</td>
<td>• Strong complements: internet infrastructure, digital media, appliances,</td>
<td>Strong positive force</td>
</tr>
<tr>
<td>Industry growth</td>
<td>• ~20% CAGR</td>
<td>Strong positive force</td>
</tr>
</tbody>
</table>

Overall Attractiveness          Attractive

Suppliers. There are three main classes of suppliers to firms engaged in VLSIC design: wafer foundries, package and test firms, and suppliers of design platforms, including CAD tool suites, design methodologies, and silicon intellectual property (SIP). We will see that supplier power is strong among...
consolidating foundry suppliers, but that the power of design platform suppliers is limited by their ability to cleanly integrate with the design process of their customers, and by their customers’ willingness and ability to integrate backward into self-supplying design platforms. Overall, suppliers will exert a moderate negative force on VLSIC firms’ pricing and their ability to earn supernormal profits.

Wafer foundries manufacture the design, and supply the resulting physical artifacts (wafers or die) to the design firm, or directly to their customers. In some cases the design firm has integrated backward into wafer foundry (or, perhaps more accurately in some cases, the wafer foundry has integrated forward into design). Three models for wafer foundries exist today.

First, pure components firms hold as their primary business the manufacture and, if necessary, design of VLSIC components that they then supply to a fragmented downstream market. This type of firm forecasts, builds, and themselves supply all the foundry capacity required to support the range of products that they have designed themselves or licensed from others. Intel is an example of a pure-components foundry.

Second, the so-called independent device manufacturers (IDMs) hold as their primary business the design and production of high-volume products that have a high degree of VLSIC content. Examples of IDMs are Motorola, in cellphones, Lucent in internet infrastructure equipment, and Hitachi in consumer electronics. These firms may also manufacture pure VLSIC components that they sell directly, but their primary motivation for holding IC foundries as assets is to support their dominant business, in systems that consume VLSICs. These firms take on pure component businesses only when their foundries have excess capacity. In fact, capacity demanded for system components has of late been so strong that the IDMs have begun to outsource the manufacture of a portion (currently around 10%) of their system components to the third type of foundry, the pure play wafer foundry.

Third, the pure play wafer foundry undertakes solely the contract manufacture of designs created by other firms. They explicitly do not compete with their customers in the design or use of the components that they produce. Increasingly, pure play wafer foundries also supply IC design platforms and packaging & test solutions to their customers. In 2001, the top three pure play wafer foundries in the world are projected to be Taiwan Semiconductor Manufacturing Company (“TSMC”) with production of 4.7 million 8-inch wafer equivalents, Taiwan’s United Microelectronics Corporation (“UMC”) with 3.0 million, and Singapore’s Chartered Semiconductor Manufacturing Company with 1.4 million. As of June
Disruption, Dis-Integration, and Modularity: the Microprocessor Industry

1999, TSMC and UMC comprised 80% of the $7bn pure play foundry market\textsuperscript{14}. Another notable supplier is IBM microelectronics division, which recently entered the commercial foundry business.

Of the three types of foundries, TSMC forecasts that in 2001 the “big three” pure play foundries will account for 16% of the industry’s output of 58 million 8-inch equivalent wafers. Of late, demand for wafers has been on the rise, with “fabless” design firms facing a squeeze in wafer starts as the IDMs outsource an increasing portion of their production to the pure play foundries. This has been driving the above percentage higher in recent years.

A useful proxy for supplier output is capital investment by supplier. The following table\textsuperscript{15} shows those firms making the largest capital investment in semiconductor production capability in 1998:

<table>
<thead>
<tr>
<th>Rank</th>
<th>Firm</th>
<th>Capital Investment ($bn)</th>
<th>Rank</th>
<th>Firm</th>
<th>Capital Investment ($bn)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Intel</td>
<td>3.70</td>
<td>6</td>
<td>AMD</td>
<td>1.20</td>
</tr>
<tr>
<td>2</td>
<td>Motorola</td>
<td>1.55</td>
<td>7</td>
<td>IBM</td>
<td>1.20</td>
</tr>
<tr>
<td>3</td>
<td>Siemens</td>
<td>1.35</td>
<td>8</td>
<td>TSMC Group</td>
<td>1.09</td>
</tr>
<tr>
<td>4</td>
<td>UMC Group</td>
<td>1.31</td>
<td>9</td>
<td>Winbond Group</td>
<td>1.08</td>
</tr>
<tr>
<td>5</td>
<td>NEC</td>
<td>1.25</td>
<td>10</td>
<td>Toshiba</td>
<td>1.03</td>
</tr>
</tbody>
</table>

Intel, a giant in the semiconductor industry, in 1999 accounted for about 20% of worldwide semiconductor revenue, and in 2000 will produce 5.25 million 8-inch wafer equivalents, or about 78% as many as the big three foundries combined output of 6.77 million 8-inch wafer equivalents\textsuperscript{16}. Intel invested more than the combined investment of the “big three” pure-play foundries 1998, and is on track to invest $5 billion in capital expenditures in 2000.

The increasing amount of capital investment required to maintain competitive scale in the foundry business suggests that consolidation of suppliers will occur over time. This is evidenced by TSMC’s

\textsuperscript{14} CHINA NEWS (2000).
\textsuperscript{15} Data from LAMMERS (1999), originally from Dataquest.
\textsuperscript{16} CHASE(2000) and LINEBACK(2000).

Disruption, Dis-Integration, and Modularity the Microprocessor Industry

recent acquisitions of Worldwide Semiconductor Manufacturing Company and TSMC-Acer Manufacturing Company. In 1999, UMC also merged with four of its affiliates.

The factors above indicate that supplier power is strong for foundry suppliers to firms undertaking VLSIC design. To summarize, the factors that contribute to this strength are:

- The scale required to compete as a VLSIC foundry
- the relatively small number of firms that currently comprise the industry
- the trend towards increasing demand for semiconductors
- the trend towards further consolidation in the industry

Indeed, the following quote typifies the situation:

"TSMC and UMC are expected to raise their prices on selected products by 5 percent to 15 percent this year after similar moves in 1999. "Nobody can avoid paying more for their chips to be made," said Frank Jiang, a spokesman for Taiwan's Via Technologies Inc. "But if you have long-term relations with foundries, you may be charged less [than] those who don't." Via is a major customer of TSMC and a rival of Intel Corp. in making computer chip-sets."\(^{17}\)

This supplier strength is mitigated to some degree by the cyclical nature of the semiconductor industry. The huge capital investments required in IC foundries make it a feast or famine business. Fab owners will continue to operate so long as marginal revenues from operations exceed marginal cost – and the marginal cost to operate a fab is orders of magnitude less than the fixed costs required to build it. This means that in times of low aggregate semiconductor demand, foundry suppliers may compete for customers, rather than the other way around. Because of this, VLSIC design firms can reduce supplier power by offering to suppliers futures contracts that lock in demand, in exchange for better per-part prices.

Package and test firms are the second main type of supplier to VLSIC design firms. These firms implement the “back end processes” of VLSIC manufacture. After wafers are manufactured, they must be tested, then sawed, and the individual die installed into packages or multi-chip-modules and then tested again. Only then is the VLSIC product complete and ready for use in a system or sale to a customer.

Increasingly, the dominant foundries are offering these services themselves, and the trend towards consolidation may squeeze out smaller players offering these services.

\(^{17}\) CHINA NEWS (2000).
Disruption, Dis-integration, and Modularity the Microprocessor Industry

Package and test cost as a percentage of total VLSIC cost will grow as functions per chip and chip size increase. This will tend to increase the influence that package and test vendors can have on the pricing of VLSIC end-products.

Design platform suppliers are the third main class of supplier to VLSIC design firms. I argue that design platform suppliers’ power to influence prices set by VLSIC design firms is moderate. These suppliers offer VLSIC design firms the tools, methodologies, design libraries, functional cores, and other Silicon Intellectual Property (SIP) that they need to complete their designs while minimizing time to market. Increasingly, design platforms are tightly coupled with the characteristics of the IC process. It is not surprising to observe the pure-play foundries offering design platforms to their customers. Of special interest is the trend for foundries to offer SIP that has already been ported, produced, and physically verified in a particular IC process offered by the foundry. These so-called “golden” cores are an important part of the design platform that the foundry can offer VLSIC design customers, because they not only accelerate time to market, but they reduce the verification risk that the design firm must bear.

Because design productivity is so central to the success of a VLSIC design firm, design firms are often reluctant to give up control over this important component of success, and integrate backwards into self-supplying design platforms to enable their designs. Because of the fundamental gap between growth in IC process capability and growth in VLSIC design productivity, the onus currently falls to design firms to increase their design productivity to keep pace with advances in IC process capabilities. This means that design platform suppliers that offer significant design productivity improvements have a lot of leverage over their potential customers. But it also means that incumbent VLSIC design firms are likely to innovate themselves in an attempt to resolve the mismatch. Further, in the high-performance design space, design processes are sufficiently complex and integral that it is difficult for design platform suppliers to cleanly offer solutions that are easily incorporated into their customers’ design flow. Often, significant customization and integration effort is required to realize the benefits of the solution. This reduces the supplier’s leverage over the customer.

Overall, foundry, package & test, and design suppliers exert a moderate negative force on VLSIC design firms’ ability to extract supernormal profits for their designs.

Rivals. Rivalry in VLSIC design is fierce, but is not at its core based upon competition around price. The semiconductor industry is somewhat unique in that functions per chip continues to increase over time, and cost per function continues to decrease over time. To ensure that this continues to occur,
participants along the value chain have locked on to Moore’s law as a tangible guideline that modulates investment in incremental or radical technologies that augment performance. Since an entire industry uses Moore’s law as a guideline in this way, Moore’s law itself is really a self-fulfilling prophecy. Moore’s law holds not because of any property of nature, but because the industry makes it so.

This means that firms that lag behind the pack will face price pressure in the face of their rivals. Their products will either be late or will under-perform relative to products of their competitors. Either of these will mandate lower prices than products that remain “on the curve”, and firms that find themselves in this situation may be pressured from the industry. But this is not a classic price war, with rival firms driving prices down until no profit remains for anyone, because costs are also decreasing. Firms can gain supernormal profit in two ways. Either they can deliver designs that perform above Moore’s law curve while holding their costs consistent with the rest of the industry. Or they can reduce their costs faster than the industry, while still delivering designs whose performance remains on the curve. In both of these cases, the competitive imperative is for efficiency in design. Such efficiency could manifest through either architectural efficiency, or through efficiency and leverage of the design team itself.

The idea that product performance and price/performance improve consistently with time is one of the fundamentals of competition within the VLSIC industry, but it is not, by far, the only basis for competition. Competition for customers and design wins is based on many factors, including:

- the degree to which the functionality and performance offered in the design matches that required by the customer’s application
- the date when the design will be available, and alignment with the design and production schedule of the customer’s application
- the degree to which the design can be believed to be compatible with any previous installed base of hardware and/or software
- the degree to which the customer perceives that they will be able to maintain independence and avoid “lock-in” from the VLSIC design firm
- the reliability of the design
- the price per unit performance delivered by end-applications implemented using the product
- and, only then, the overall price.

Another non-price based mode of VLSIC design firm rivalry is competition for the pool of available design resources (people). Yet another is the extent to which each individual firm is able to partner with and/or invest in firms that either offer technology to help stay ahead of the Moore’s law curve or that stimulate demand and create new markets for its products. For example, Intel maintains an equity investment fund (currently valued at $8 billion) for this purpose.
While overall, rivalry is not price-based, in the low-end of the MPU market this is no longer true. Intel’s initiative to brand VLSIC components is an example of an effort to prevent erosion to pure price competition. Such brands are in part an effort to promote visibility of the non-price aspects of competition, and in part an effort to project the image of the VLSIC manufacturer further downstream in the value chain. “Intel Inside” and the various MPU brands (e.g. Celeron) summarize to the customer that the MPU is optimized for use with the system that contains it, that it will be compatible with the customer’s existing software, and that it will work reliably for as long as the customer owns it. In short, it brings to mind all the non-price competitive differentiators.

Overall, competition is fierce, but is based primarily on factors other than price. Rivalry among firms, then, exhibits a moderate negative force on VLSIC prices.

**Buyers.** Buyers of VLSIC products are incredibly fragmented, and consequently have little power to influence prices. Because the semiconductor industry is cyclical in nature, buyers of large quantities of standard components can influence prices to some degree by offering long-term or futures-based contracts that remove demand uncertainty. Overall, though buyers can only exert a weak negative force on VLSIC prices.

**Entry Barriers.** Entry barriers are huge to firms seeking to enter the VLSIC design business, especially in the high-performance MPU/MCU segment. It is nearly impossible to enter as an integrated foundry / MPU design firm.

It is true that, if the potential returns are sufficiently high, raising the capital to effect entry should not be a barrier even if the amount of capital required is enormous. In reality, however, accessing such large amounts of capital is beyond the reach of many firms. Further, it requires more than just capital to enter. Constructing an IC foundry and assembling a proficient design team is a tremendously difficult task, and one that must be executed flawlessly. Massive scale is required for profitable entry, and this scale increases the complexity of the entry task, thereby reducing investor confidence that successful

---

18 Keep in mind that over 75% of the IC process equipment in Intel’s latest generation of IC fabs will remain at the technology forefront for three years or less, after which it will need to refitted. Delaying the ramp-up of a foundry because of execution errors therefore has a devastating financial impact. See Johnson (1997).
entry is actually possible. The fact that *incremental entry is precluded* is what makes the entry barriers so high.

Even entry as a fabless design firm is extremely expensive. The difficulty is in acquiring an adequate design team and a critical mass of intellectual property. An illustrative recent example of the costs involved comes from Via, a Taiwanese PC chipset maker who is attempting entry to the x86 MPU market. Via is orchestrating its entry through acquisitions and additional investment. It acquired two fabless x86 MPU design groups – IDT’s Centaur Technology microprocessor division (~60 designers), for $51 million, and National Semiconductor’s Cyrix Corporation (~160 designers), for $167 million – which it merged and will now spin out into a separate company. Via will bootstrap the spinout with an additional investment between $100 million and $300 million, for a total entry cost somewhere between $300 million and $500 million.

In 1996, National decided to take a run at the x86 MPU business and acquired Cyrix for $388 million, only to sell it Via a year and a half later for less than half that amount.

Even after surrendering such amounts for entry, success is not assured. Says one analyst:

"Designing an MPU is much harder than designing core logic... Look, even Advanced Micro Devices Inc. can't do it well and they have their own fab and 20 years of experience."19

Entry barriers are about an order of magnitude smaller for VLSIC design firms targeting the embedded market. Tensilica provides application-specific tailored processor cores to its customers, and entered the design platform business in July 1997. Cumulative funding through November 1999 was only $33 million, which Tensilica was able to raise as risk capital from the VC community. Transmeta has created a VLIW processor for the x86 laptop computer and appliance market. It got started using $15 million in funding from deep-pocket investors George Soros and Paul Allen. The firm recently closed an additional funding round of $88 million. Its price tag for entry is likely to end up at around $150 million.

All in all, though, *entry barriers are high*, making entry barriers a weak negative force.

**Substitutes.** In today’s world, there are few substitutes for most VLSIC products. Systems can certainly be designed using discrete components, but integrated circuits offer compelling cost advantages, and are the only viable solution for systems that are large in scope (more than a few hundred thousand
transistors). Certain classes of VLSIC products can substitute for other types. For example, an MPU of sufficient performance and the requisite software can substitute for dedicated digital signal processing hardware. In some cases, standards and “lock-in” to those standards may reduce the viability of substitutes, even to other classes of VLSIC products. This was the case with x86 processors up until a few years ago.

Indications are that VLSICs will continue to substitute for other portions of application systems, rather than the other way around. The average semiconductor content for all electronics products was up 5.1% from 1998 to 1999\(^{20}\).

Substitutes, offering little influence on VLSIC pricing, constitute a weak negative force.

Complements. There are many complements to VLSICs. A few notable classes of products have sprung up in recent years that strongly complement VLSIC designs and artifacts, and that fuel growth in VLSICs overall. These include products dealing with information storage & retrieval, the internet, and mobility.

Information storage and retrieval products are products, such as servers, that manage storage and retrieval of large amounts of data. New types of electronic services involving digital media (photos, articles, music, and video) are fueling consumption of storage and retrieval, and the creation of new products that facilitate it. The internet, and packet switching and the internet protocol in particular, is becoming a de facto worldwide standard for device connectivity. This fuels creation of products to deal with building out the global packet-switched network, as well as products that facilitate connectivity via internet protocol. Mobile information access (for example, via GSM and wireless access protocol) builds demand for terminal products to access information services. These so-called “information appliances”, evolved from today’s cellular telephones, will fuel VLSIC consumption for years to come.

This variety makes complements a strong positive force.

\(^{19}\) CARROLL(1999).
\(^{20}\) DATAQUEST(1999).
Growth. Growth in VLSICs is strong, overall. 1999 saw over 20% average growth for all types of Digital ICs\textsuperscript{21}. Growth was more moderate for MPUs and MCUs, which grew an average 10.4% in 1999. Growth was very strong for ASICs and DSPs, with 28.4% growth in 1999.

I expect growth in SOC products to be extremely bullish as traditional ASIC and DSP solutions are combined with MPUs and/or MCUs onto single-chip solutions.

Attractiveness of the Industry. Suppliers and rivals exert a moderate downward influence on prices, buyers exert only a weak influence, high entry barriers prevent anything but a weak downward pressure on prices, complements offer a strong positive force, and growth is solid.

These factors combine to make VLSIC design an attractive business, especially in high-performance MPUs & MCUs, and in systems-on-a-chip that integrate high-performance MPUs & MCUs. The degree of price competition at the low-end of the MPU market makes it increasingly worrisome and unattractive for VLSIC design firms.

\textsuperscript{21} Source for all growth quotations is Semiconductor Industry Association.
Relevant technology S-curves, performance measures, and current position

Several key technologies contribute to the ability of microprocessors to deliver end-customer performance. A key component of technology strategy is developing an understanding of what these technologies are, what the important performance measures for each technology are, and where each technology lies on its technology “S-curve” of performance vs. effort expended. This understanding can then guide the firm’s actions going forward.

I will review and discuss 3 key technology drivers in the high-performance VLSIC arena. I argue that these technology drivers are critical to VLSIC design firms competing in the MPU/MCU/SoC product categories. They are:

- semiconductor IC process and packaging technology
- design platform effectiveness, including:
  - CAD Tool effectiveness
  - SIP reusability and integration costs
  - Integration costs of multidisciplinary circuits
- transformation effectiveness (hardware ↔ hardware, hardware ↔ software)

We will see that semiconductor IC process technologies may be nearing the top of their S-curve, but that their remaining life, while still delivering performance improvements as per Moore’s law, is measured in decades. Because of this, the author does not see IC process as a limiter in the short term. Design platform effectiveness, on the other hand, is currently limited by CAD-supported custom design technology, which the author argues is at the top of its S-curve. VLSIC design firms face a transition to re-use oriented technologies; re-use can occur both in design, as with technologies for integration of SIP, and in use, as in technologies that transform functionality from hardware to software.

The essential S-curves are shown on the next page. We will examine each of these in more detail in the remainder of this section.

Semiconductor IC process and packaging technology is a fundamental determinant of product performance whose impact is felt throughout the entire VLSIC design and production value chains. Historically, DRAM products were the driving force behind the development of the entire semiconductor industry. In the past few years, this role has shifted to be shared with microprocessors, and we have seen an increase in the rate at which new technologies are applied to microprocessor manufacturing.
Integrated Circuit Process Technologies

Technologies for design platform effectiveness: the leap from custom design to Modularity & Re-use
Disruption, Dis-Integration, and Modularity the Microprocessor Industry

Improvements in IC process technology affect many end-product performance metrics, including: Functions per chip, maximum chip size, chip density (functions per cm²), chip frequency, minimum mask count and mask set creation cost, and overall chip cost. Packaging and test technology affects end-product performance metrics such as: maximum number of chip inputs and outputs (I/Os), cost per I/O, package cost, and overall chip cost (packaged, tested).

Since IC process and packaging technology is such a key driver of end-product performance, firms undertaking VLSIC design must ensure that they either own or have access to them. Access to IC process is required to support both the design value chain and the production value chain. Firms must take care that access is available early enough to support both.

An enormous amount of energy is expended by firms across the semiconductor industry to continually deliver performance increases consistent with Moore’s law, and to understand whether and when IC process is nearing the top of its performance S-curve. In fact, this is the express purpose of the Semiconductor Industry Association’s “International Technology Roadmap for Semiconductors” (ITRS), which notes that

“...this challenge has become so formidable that more and more of the development effort has been shared in a pre-competitive environment including consortia and collaboration with suppliers.”

There is growing concern that IC process is in fact near the top of its S-curve. Silicon and photolithography are not capable of delivering lower cost and higher performance forever. The ITRS points out certain areas in the 5-15 year timeframe where known solutions do not yet exist to enable continued performance increase as per Moore’s law. This has led to short-term investment in those areas, and long-term investment and exploration in some new radical technology improvements. Molecular computing, for example, holds the potential to carry on when silicon and photolithography reach their inevitable limits.

For at least the next 10 years, however, it appears extremely likely that the industry will continue to deliver performance improvements as Moore’s law predicts²². And such improvements may continue for much longer, in no small part as a result of efforts like the ITRS to identify holes in the roadmap ahead.

---
²² Author’s analysis of ITRS, and also MSDW(1999) p. 10.
Design platform effectiveness. While the semiconductor industry as a whole is succeeding in keeping up with the demands of Moore’s law in the production domain, VLSIC design firms are having more difficulty keeping pace in the design domain.

"While the industry is clearly evolving towards more complex chips ... the transition has been slowed by a growing "design gap" between what companies can manufacture and what they can economically afford to design."23

While logic transistors per chip are increasing at a compounded annual growth rate of 58% per year, transistors per staff-month is increasing at only 21% CAGR24. This mismatch is only partially addressed by re-use strategies, and as a result, the average size of design teams is increasing.

To make matters worse, VLSIC design firms face pressure not just to increase the growth rate of transistors per staff-month, but also to do so more efficiently in terms of use of silicon area. To stay on-target with Moore’s law,

"For MPUs, the allowable [preserving Moore’s law while maintaining economical manufacturability] chip size growth is flat through 2001, then grows at 1.2x every four years. To add only 20% in area every four years, while quadrupling functionality, requires ... design productivity which further reduces chip size by an additional minus 7-8% per year. This design-related area reduction is in addition to the basic lithography-provided area reduction of –11% per year."25

What this means is that the “create design platform” portion of the design value chain is increasingly more important within the industry. It is the design platform that provides leverage for the VLSI design staff, and has the potential to increase the rate at which design firms can grow transistors per staff-month. It can do so either by helping designers be more effective in creating new designs, or by increasing re-use and/or lowering the integration costs of existing designs.

Overall measures of design platform effectiveness are transistors per staff-month on new designs, and the rate of change of this effectiveness from design to design. I can partition technologies that increase design platform effectiveness into three classes that address: CAD tool effectiveness, SIP reusability & integration costs, and integration costs of multidisciplinary circuits.

CAD tool effectiveness. One way to increase design platform effectiveness is to increase the efficiency of the CAD tools within the platform. Examples of technologies in this area are CAD tools

---

that increasingly consider and link, in Murray and Sako’s terminology, the *systems* of the resulting designs (e.g. timing, power, testability). These tools behave as agents that act on behalf of design engineers, that the engineers can oversee and guide intelligently. The author could locate no data to indicate where such technologies lie on their effort vs. performance S-curve. However, the need for such sophisticated designer-focused CAD support is driven by the increasing number of dependencies and system-level interactions in the design process that could previously be ignored. On one hand, since the industry has only just begun to implement CAD tools to deal with such complexity, there is likely to be significant progress before further improvements become prohibitive. On the other hand, the breakdown of the “traditional” ASIC design process and the necessary introduction of such sophisticated CAD suites is evidence that designer-focused CAD tools may be nearing the top of its S-curve.

*SIP reusability and integration costs.* As mentioned above, design firms can at least partially address the design productivity gap through re-use. Technologies that reduce the cost of re-using past designs and reduce the cost of integrating many design “modules” into a working system contribute to overall design platform effectiveness. Examples include technologies that:

- reduce the cost of porting designs from one generation of IC process to the next
- selectively use programmable logic to increase SIP generality and reduce re-use costs
- define protocols for module discovery and arbitration
- provide standard module integration and verification frameworks
- provide libraries of modules already produced, tested, and verified in the target IC process
- reduce physical prototyping time and/or cost

Note that such technologies tend to optimize or drive *modularity in design* – their explicit goal is to increase the productivity of the design process.

*Integration costs of multidisciplinary circuits.* The ever-increasing maximum chip size and functions per chip have increased the need to integrate different types of circuits onto the same die (e.g. analog, RF, DSP). Historic design platforms do not address this need, and integration and verification mechanisms for such circuits have, until recently, been ad-hoc. CAD suites that understand and accommodate these various circuit disciplines will be increasingly important as more and more of the overall system becomes integrated onto a single die.
**Transformation effectiveness (hardware ↔ hardware, hardware ↔ software).**

Transformation effectiveness refers to a broad class of technologies that shift traditional product structure boundaries by substituting hardware for software, or by substituting hardware of one type for hardware of another. Often such technologies represent architectural innovations that improve the capability of a particular product structure to deliver performance. Examples of such innovations are VLIW computing architectures, dynamic object code translation, dynamic mapping of programmable logic, and "sea of processor" approaches. These approaches are described in Appendix B.

Note that transformation technologies tend to optimize or drive *modularity in use* — their goal is to allow general-purpose components address application-specific needs by mapping, tailoring, and/or transforming one sort of component (and its performance attributes) into another.

These techniques are particularly interesting because they each significantly shift boundaries in the design value chain and the relative importance of the firm's competencies. These shifts lead to either performance improvements in the end product, a reduction in time to market, or both. Further, the shifts of functionality from module to module can have important consequences in terms of network externalities seen by some of the modules. Transformation techniques are all very new to the VLSIC industry. The author perceives such substitution techniques to be low on their performance vs. effort S-curves.

**Network effects**

Network effects are significant for MPUs. Backward compatibility with the installed base of software is an imperative, as a processor is valued not just based on the performance it can achieve, but by the amount of software that it can run. The installed base of existing software and existing customers form a virtuous cycle, because author's of new software are more likely to make their solutions available first on those platforms which have the most customers and the largest amount of complementary software.

Network effects are limited in the application-specific, embedded space. Standard interfaces (e.g. internet protocol, wireless access protocol) render the choice of embedded architecture irrelevant from the perspective of network externalities. VLSIC design firms therefore optimize embedded architectures around performance, and are not hesitant to change them or reinvent them. Vendors announced more new
Disruption, Dis-integration, and Modularity the Microprocessor Industry

architectures for the embedded market in 1999 than they have for the PC market in the past 15 years. Some analysts count more than 100 different 32-bit embedded CPUs on the market. The primary source of network benefits in the embedded marketplace is in the design domain, where standard hardware & software design tools can reduce product development costs. New architectures that necessitate a change in design tool suite face a disadvantage over those that do not.

Network externalities have a significant impact on the value of SIP libraries and the value capture potential of distribution channels for SIP. A silicon design “module” contributing to a library of SIP has value commensurate with the number of other modules in the library. This means that standards for SIP interoperability can create value, as can firms that simply aggregate SIP, but do not undertake SIP design. It also means that firms that create channels for SIP distribution hold potentially potent value capture mechanisms, especially if they are able to achieve exclusive rights to SIP distribution, or by some other means become the sole source for a broad class of SIP.

Sources of appropriability and complementary assets

The previous sections discussed some forms of value creation that are widely applicable within the MPU/MCU/SoC arenas. Here we will look at some generally applicable mechanisms for value capture. These value capture mechanisms present VLSIC design firms with an array of options for building a competitive strategy. They are:

- Ownership or access to IC foundry
- Design platform technology patents
- Design platform channel ownership
- Design teams themselves
- SIP ownership
- SIP channel ownership

Ownership or access to large-scale IC foundry. Failure to secure access (through ownership or other means) to adequate IC foundry capacity would be catastrophic for most VLSIC design firms. In a long-term view, foundry capacity is not a complementary asset – while far from straightforward, any firm in the industry can create capacity by making the required capital investments and assembling the appropriate IC processing equipment. In the short-term, however, access to capacity can be a

26 HALFHILL(1999).
27 MAYER(1998), Electronic Buyer’s News
complementary asset. Those firms that have previously invested in foundry capacity get to decide how and whether to use it themselves, and/or who to sell it to. Since the lead time to construct new foundry capacity is long with respect to changing demand conditions, the industry is highly cyclical.

However, design firms may be able to appropriate some value through creation of differentiated foundry-related capabilities – for example, design for high yield and/or low cost, or design for absolutely leading-edge performance. These competencies may themselves be appropriable, leading to a protected source of advantage even if access to foundry cannot be protected.

VLSIC design firms may be able to generate some complementary assets in the form of ongoing relationships with foundries. These relationships could involve long-term contracts (perhaps with some form of exclusivity) and/or joint development efforts.

*Design platform technology patents and/or secrecy.* Certainly VLSIC design firms can capture value by appropriating technology contributing to design platform efficiency, either through developing a strong patent portfolio of techniques that promote design platform efficiency, or through secrecy – simply keeping the software implementations of such design platform improvements proprietary and in-house.

*Design platform channel ownership.* A less obvious source of value capture is through strong control of dominant channels for design platforms. Such channels, if dominant in the industry and strongly controlled, would become complementary assets because any firm wishing to commercialize compelling new design platform technology would need to provide the technology through the dominant channel. This is the value capture strategy pointed out in Meyer & DeTore (1999).

*Design teams themselves.* VLSI designers are themselves in such short supply that an assembled team, with a demonstrated ability to work well together and actually produce designs, is itself a complementary asset. Assembling such a team often takes years of dedicated recruiting and retention. The prior quote regarding AMD’s entry into x86 microprocessors is a testament to the difficulty of assembling capable design teams.

*SIP ownership.* Once designs exist, the designs themselves are typically easy to appropriate through either patents or secrecy. Reverse engineering complex VLSI is definitely possible, but especially where architecture standards are involved, quite difficult. Intel successfully used this lead time for several product generations in remaining more than one full design cycle ahead of their reverse-engineering competitors. As soon as the competitors succeeded in getting their design to market, Intel would release their higher-performing next generation at a price that severely depressed their competitors’
Disruption, Dis-integration, and Modularity the Microprocessor Industry

Margins. Legal defense of silicon SIP is quite common, as is licensing SIP for inclusion in others' designs (e.g. ARM).

SIP channel ownership. Once again, value can be captured not only by appropriating the SIP itself, but by garnering strong control over a dominant SIP channel. Both foundries and design platform vendors are well positioned to attempt to establish such a dominant channel for SIP.

Strategy assessment of incumbent firms

Thus far we have reviewed some of the potential ways in which VLSIC design firms might attempt to create and capture value. Now let's take a quick look at some incumbent firms in the industry and analyze their technology strategy. How does each firm differentiate itself in terms of creating value, capturing value, creating competencies, and organizing the enterprise? This will set the stage for us to examine the strategy of notable entrants, and to look for instances of the "general mechanisms" previously described that lead to shifts in value chain boundaries and, potentially, to either more modular or more integral design processes and product structures. I will examine Intel, the powerhouse in MPUs (with a unit share around 80%), and AMD, their main competitor.

Intel. Intel's value creation formula has been simple and consistent for most of the past 15 years: create value through the network benefits created in an ever-increasing installed base of software compatible with their proprietary stream of x86 microprocessor architectures. Then appropriate the architecture through a combination of a strong patent portfolio, aggressive legal action to defend their intellectual property positions, secrecy, and a product development pace that evolves the architecture more quickly than the competition can comprehend and reverse-engineer it. Build competency in the creation of efficient foundries (that is, having high yields and consequently low per-part cost). Optimize the design value chain to accommodate the foundry business. To ensure that demand exists for the increasing foundry capacity that accompanies larger scale and lower costs, stimulate demand by entering and/or supporting downstream portions of the value chain. If necessary, reduce prices to maintain share and keep the foundry full, albeit at lower margins. Protect share through branding.

28 For an excellent in-depth discussion of Intel's strategy see Afuah (1999).
29 For example, see Thryft (1995), which describes how Intel supported its core foundry business by entering the motherboard business.
30 Johnson (1997) explores the success of Intel's brand campaign.
Disruption, Dis-integration, and Modularity in the Microprocessor Industry

It is important to note that, in order to protect and extract value from its capital investments in foundry, Intel is highly incented to optimize around production. Modularity at Intel, if introduced at all, would most likely be targeted towards modularity in production. Intel's overriding concern is to keep its foundry fully utilized, and to maximize the gross margin of the products that the foundry produces. Design costs for the components to fill the foundry are likely to be inconsequential when compared to the costs of creating the foundry itself, especially when amortized across the volume of parts that Intel has in the past been successful in selling. For this reason, Intel is likely to be unconcerned with design platform effectiveness, as long as some way exists to successfully complete the design.

The advent of sub-$1000 PCs has to some degree stymied Intel's traditional strategy. Because consumers in this segment are extremely price sensitive and care less about leading-edge performance and brand, PC-makers are willing to consider using (and, in fact, prefer to have a choice among) non-Intel processors. This has led to the introduction of an extremely price-competitive low-end segment of the market, and Intel's competitors have succeeded in growing their share within this segment. For example, AMD or Cyrix processors were used in 60 percent of the sub-$1,000 desktop PCs sold through the U.S. retail channel in June 1999\(^1\). Once legitimized at the low end, these competitors are likely to march upward into Intel's core market of business PC's.

This has led Intel to reinforce the low end with its Celeron processor brand, and to aim higher, at the server market, as a source of future growth and continued high margins. If necessary, Intel would then be able to use margins in the high-end segment to subsidize lower price points at the low end, protect its share, and maintain control of the design value chain. This will require that Intel grow its competencies to include those required for design at the high end -- reliability, stronger design verification, server-appropriate chipsets, and a focus on higher performance (and not necessarily low cost). Already, Intel has initiated a "server farm" compute-utility business as a means to learn and understand the requirements of the server market as they relate to MPU and chipset design.

**AMD.** AMD's strategy has relied primarily on entry through price competition to break Intel's stranglehold on the market. AMD prices their MPUs at least 25% lower than comparably performing Intel products. AMD has catered to the sub-$1000 PC market, enabling that market and fueling the demand for low-end MPUs.

---

\(^1\) Slater (1999).

Management of Technology Thesis 67/99

Mick Bass, 19 May 2000
This strategy is to a large degree reliant on Intel’s willingness to cede share rather than lower its margins on low-end products. If MPU demand is strong and Intel’s foundry capacity is oversubscribed, then when AMD offers low price points at the low end, Intel must, in the margin, choose between filling its foundry with low-end products or products from elsewhere in the product line. Choosing the latter would maximize Intel’s short-term profit, but would also open the door for AMD to gain share in the low end. To date, Intel has not chosen that course.

If AMD is able to gain entry and legitimize themselves as a supplier in the sub-$1000 space, they would then be able to move upward into the business market with higher-margin differentiated products.

The strength of Intel’s IP position has recently forced AMD to design new MPU module interfaces that differ from those established by Intel as the dominant interface for the PC industry. This is both an opportunity and a threat for AMD – the stakes have become higher, for AMD must now integrate forward and provide chipsets compatible with its MPU interface. If they succeed, however, they are well positioned to provide PC manufacturers with differentiated MPU and chipset products, which may in the long term provide AMD with the market share and profit margins that they require to be successful.

In terms of competencies, this means that AMD should be focusing on two areas: design for low cost, and creation of foundry capacity at a scale that enables continued low cost. Even more than Intel, AMD is likely to focus on reducing the price/performance of their processors at low price points. To do so, they must build both foundry and design expertise that optimizes per-part cost.

**Strategy assessment of notable entrants**

Now let’s examine some recent entrants to the MPU/MCU/SoC design scene. These firms are interesting because their entry may in fact evidence one of the previously described “general mechanisms” for shifts in value chain boundaries. In this light, I will examine the strategies of ARM, Tensilica, and Transmeta.

**ARM.** Advance RISC Machines (ARM) is a fabless design firm of MPU cores. ARM is a SIP firm; they earn revenues by licensing their MPU core designs for production by others (e.g. Intel), or for inclusion in ASIC or SoC designs that are produced by others.

---

32 Slater (1999). In exchange for continuing its x86 cross-licensing agreement with AMD, Intel was able to gain the concession that AMD would not be able to use Intel’s physical processor module interface in the future.
Disruption, Dis-integration, and Modularity the Microprocessor Industry

ARM creates MPU cores for the embedded market – these cores dominate certain segments (for example, digital cellular telephones). ARM’s primary value creation mechanisms are increased design platform effectiveness through SIP re-use (you don’t have to design their core yourself) and hardware to software transformation (their core offers sufficient performance that many functions can be implemented in software, reducing the complexity and time to market of the required hardware). Their primary value capture mechanism is SIP ownership, which enables their licensing revenue stream.

I chose ARM as a notable entrant because it is a concrete example of a microprocessor core as SIP, and validates the economic viability of the model. In ARM we see evidence of the “performance mismatch” mechanism leading to a shift in modularity and in design value chain boundaries. Because the number of transistors per die that were available so greatly exceeded the number required to implement a microprocessor for the embedded market, pressure existed within the design value chain to use the extra transistors for something else. This required a modular boundary to be drawn around the MPU core so it could be easily instantiated among multiple solutions, each of which optimizes the use of the remaining transistors on the die to optimize their particular problem. Further, the design value chain needed to be augmented – tools needed to exist to easily instantiate and verify ARM’s core with the remainder of the chip, and the core needed to be ported into multiple target IC processes.

ARM also exhibits the “growth options” general mechanism by bridging markets. Because ARM creates value through hardware to software transformation, and is integrated into a surrounding ASIC or SoC, the ARM core is applicable in a wide array of applications in many different end-product markets.

These phenomena occurred first in the embedded MPU market, because it was there that transistor availability first outpaced optimal core size. As transistor availability continues to grow, we can expect these mechanisms to apply as well to the mainstream cost-performance MPU market, and finally to the high-performance MPU market.

_Tensilica._ Tensilica differentiates itself from ARM by offering a means to not simply include an off-the-shelf embedded MPU core, but to include at similar cost an MPU core that is tailored to perform particularly well within the application environment that it will be included. Tensilica offers a web-hosted automated design platform that generates entire embedded MPU cores according to customer specifications.

Because the created MPU core is application-specific, Tensilica can offer tremendous value through hardware/software transformation. For example, a tailored Tensilica core is about 100 times more effective at video decompression than a comparable non-tailored core. Tensilica captures this value
Disruption, Dis-integration, and Modularity the Microprocessor Industry

through SIP ownership, and appropriable design platform technology (through both patents and secrecy). Tensilica is well positioned to attempt to create a dominant design platform channel.

Tensilica is a notable entrant because they offer a modular interface for their MPU core, and also because they have modularized the MPU itself to such a degree that end-to-end MPU customization, guided completely by automated software tools, is possible. In Tensilica we see the potential of intra-MPU modularity, and design platforms that take strong advantage of it. Tensilica stresses modularity in design — its design process is optimized to enable turnkey design of an entire customized MPU.

Tensilica evidences all three of the “general mechanisms”. The core-tailoring website is a textbook example of a “user toolkit” approach to reduce the impact of sticky information. Tensilica places responsibility for specifying which performance attributes matter where it belongs — with the system designer — and the toolkit churns out an MPU core that is consistent with these performance expectations. It is an example of the performance mismatch mechanism in the same ways as ARM. From a transistor count perspective, Tensilica can afford the overhead of the intra-MPU modularization required to make the performance-tailoring toolkit possible. And from a performance perspective, the performance overhead of modularization is more than paid for by the application-specific customizations that the modularity enables. An integrated (non-modular) MPU may be able to outperform a Tensilica produced core on a particular application, but would not be able to achieve similar time-to-market (Tensilica provides most of the performance benefit without the time-to-market cost). Finally, it is an example of the growth options mechanism. To an even greater extent than ARM, Tensilica cores are capable of bridging a wide array of markets.

*Transmeta.* Transmeta has produced a VLIW MPU core surrounded by dynamic code translation software to form a “virtual processor” that is backward compatible with the installed base of x86 software. But, since the underlying VLIW engine is small and power-optimized, Transmeta’s MPU, when used in a laptop application, offers battery life four times longer than Intel’s current offering.

Transmeta is an example of the performance mismatch mechanism. In this case the mismatch was in application performance (oversupplied). Laptop consumers instead desire longer battery life (undersupplied). The mismatch drove modularity that enables easy transformation from one type of performance to another. The underlying modularity, Transmeta argues, also reduces time to market:

“Transmeta’s Code Morphing technology changes the entire approach to designing microprocessors. By demonstrating that practical microprocessors can be implemented as hardware-software hybrids, Transmeta has dramatically expanded the design space that microprocessor designers can explore for optimum solutions. Microprocessor development teams
Disruption, Dis-integration, and Modularity in the Microprocessor Industry

may now enlist software experts and expertise, working largely in parallel with hardware engineers to bring products to market faster. Upgrades to the software portion of a microprocessor can be rolled out independently from the chip. Finally, decoupling the hardware design from the system and application software that use it frees hardware designers to evolve and eventually replace their designs without perturbing legacy software.\textsuperscript{33}

Transmeta appears to also have been driven by the growth options mechanism. The underlying modular architecture can be used to create x86-compatible MPUs optimized for a wide array of applications in a wide array of markets, and can even be used to create MPUs that are compatible with the software installed base of other architectures (e.g. Sparc, PA-RISC).

Note that Transmeta’s code-morphing technology is an example of modularity in use – the software layer surrounding the underlying hardware enables optimization for different applications and performance attributes. This can be done while maintaining the legacy x86 interface.

While they have not yet done so, Transmeta appears well positioned to create a user toolkit that, like Tensilica, allows customers to select the performance attributes and tradeoffs that they value, and take delivery of an auto-designed MPU core that meets their requirements.

Transmeta’s entry is notable because it is the first evidence of an intra-MPU modular architecture used to tailor an MPU to a particular segment within the mainstream PC market. Transmeta’s modular intra-CPU architecture thus poses a credible threat to Intel’s dominance.

\textsuperscript{33} KLAIBER(2000). Management of Technology Thesis
Application of “general mechanisms” for industry value chain evolution

Given these examples of incumbent and entrant firms in VLSIC design space, let’s now generalize to conclusions about existing pressures on current value chain boundaries. This will allow us to reasonably speculate about how the design and production value chains might evolve in the future.

Key design and production process boundaries (fundamental interfaces). Certain interfaces in the design and production value chains, and the resulting product structure, are fundamental within MPU and system design. These are the important boundaries to examine in order to see the general mechanisms at work in the value chain. The most important interfaces in physical product structure are\(^{34}\):

- Execution unit
- SIP core / MPU core
- Physical die
- Physical module
- Board
- Box

On the software side of product structure, the critical interfaces are:

- Architectural applications
- Compiler
- Operating system
- Application software

Finally, the key design and production value chain boundaries, previously described, were:

**Design:** Design platform → Circuits → Proto production VC → Volume production VC

**Production:** Masks → Wafers → Physical Module → Board → Box

Now, within this framework for product structure and value chain interfaces, what applications of the general mechanisms do the prior analysis and examples suggest?

Several applications of the general mechanisms, and consequences in terms of the value chain, are apparent. These occur in a rough progression – I observe that some are already well underway, and others

\(^{34}\) Refer to Appendix C for definitions and short descriptions of each of these interfaces
are just beginning. I describe each application in the following paragraphs. The table on the following page summarizes all of them.

First, there is a performance mismatch between the number of transistors per die that foundries can manufacture and the number of transistors that design teams can design into the die. The magnitude of the mismatch is large and growing. This creates a pressure for increased design team efficiency, and strengthens the position of design platforms within the value chain. Value will shift towards design platforms that demonstrate increased design team productivity.

Yet another pressure created by the performance mismatch noted above is to maximize the portion of the design cycle spent in system and circuit design, and reduce or eliminate any serialized design of prototype or production value chains. This creates a pressure for the role of end-to-end value chain management to shift away from VLSIC design entities, and towards foundry entities. Standard, well-established processes for prototype production and ramp to volume are likely to emerge. Value will shift towards design platforms that support these standards.

Another performance mismatch exists with respect to the scale required by certain classes of products. The number of transistors available at the scale that produces cost economies exceed the requirements to implement many types of historical end-products. This creates a pressure to put the remaining transistors to good use, and leads to the emergence of SIP as an intermediate product (witness ARM). Other consequences are the development of standards and interfaces for SIP, the use of multiple SIP cores per die, and design platforms that focus on integration of SIP rather than creation of SIP. Modular systems, implemented on a single die, emerge.

The above scale mismatch creates another pressure – to integrate multidisciplinary circuits onto a single die. This mixes up value chains that were traditionally, say, purely digital or purely analog, and leads to value chains that can deal efficiently with multiple types of SIP. Once again, value shifts towards design platforms that can accommodate various circuit disciplines and integrate SIP of different types.

Growth options motivate the creation of option value by combining existing SIP in innovative ways. This creates a pressure to solidify standards and leads directly to the emergence of multi-firm markets for SIP. This is most strongly observable today in the lower tiers of the market (e.g. components for cellphones). The existence of markets for SIP shifts value towards design platforms that can efficiently access those markets and efficiently obtain and integrate the SIP that populate them. See for
example, the recent HP/Cadence/Flextronics "SpinCircuit" design platform\textsuperscript{35} that addresses the market for discrete ICs. It is reasonable to expect that a similar dynamic will occur for VLSICs as SIP becomes standardized.

Now the industry is at a critical juncture. \textit{Performance mismatch} may exist on key metrics between the application performance delivered \textit{vs. desired}, while the performance on less important metrics is more than sufficient. This creates pressure to structure the value chain such that performance on the various metrics is transformable. Surplus application performance on certain metrics may lead to \textit{intra-MPU modularization}, so that systems can more easily be designed that optimize one type of performance at the expense of another. Execution units within the MPU are themselves viewed as SIP. Value shifts towards design platforms that aid system design through combination of such SIP. Tensilica is one of the first example of this phenomenon, as is (to at least some degree) the emergence of VLIW computing architectures (e.g. Intel IA-64, Transmeta).

Once intra-CPU modularization occurs, \textit{growth options} drive the creation of "architectural applications" that allow the underlying design to be tuned to reflect the customer's specific performance requirements. This in turn allows the intra-MPU modules to be combined to produce products applicable in a wide array of markets. Such architectural apps are a new piece of the VLSIC value chain. (Transmeta's code morphing software is an example.) Architectural verification, and ensuring that existing products are in fact backward compatible with existing software, increases in importance as a competence.

When product tuning capabilities exist, \textit{sticky information} costs will likely lead to turnkey user toolkits that allow system designers to guide the automated creation of system products that meet their performance specifications. Value shifts to design platforms that are capable of automated, turnkey production of performance-tuned systems. Tensilica's web site is an early example of such a design platform.

Eventually, the industry will comprehend the \textit{performance mismatch} between the cost of physically integrating SIP into a flattened mask set that enables wafer production and the cost of "virtually integrating" SIP into a production-ready design. This mismatch will limit the value of SIP re-use because of the cost and time to market associated with mask generation. This will lead to pressures in the value

Disruption, Dis-Integration, and Modularity the Microprocessor Industry

chain to reduce mask generation cost and time to market, or to enable re-use of mask subsets that implement a particular piece of SIP.

Finally, growth options lead to aggregation of SIP at the foundries themselves. The foundries become a natural distribution platform for SIP and design platforms.
Summary – general mechanisms at work in VLSI value chains. This table summarizes the above-described applications of the general mechanisms.

<table>
<thead>
<tr>
<th>Mechanism</th>
<th>Where applied?</th>
<th>Pressure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Performance mismatch</td>
<td>Design process</td>
<td>Need for increased design team efficiency. Value shift towards efficient design platforms. Design platform becomes more integral.</td>
</tr>
<tr>
<td>2. Performance mismatch</td>
<td>Design process</td>
<td>Need to reduce or eliminate time spent in serialized creation of prototype value chains. Value shift towards established foundry paths and design platforms that support them.</td>
</tr>
<tr>
<td>3. Performance mismatch</td>
<td>Design process Product structure</td>
<td>Transistors available exceed requirements to implement many types of historic end-products. Need to put remaining transistors to good use. Pressure towards multiple cores per die, and SIP as an intermediate product. Modular systems on a single die.</td>
</tr>
<tr>
<td>4. Performance mismatch</td>
<td>Design process Production process</td>
<td>Integration of multidisciplinary circuits on single die. Design platform extensions to accommodate multiple circuit disciplines.</td>
</tr>
<tr>
<td>5. Growth options</td>
<td>Design process</td>
<td>Creation of option value by combining existing SIP in innovative ways. Emergence of SIP markets. Value shifts to design platforms that include existing SIP and means to integrate it.</td>
</tr>
<tr>
<td>6. Performance mismatch</td>
<td>Design process Product structure</td>
<td>Surplus application performance leads to intra-MPU modularization. Execution units as SIP. Value shifts to design platforms that automate system design.</td>
</tr>
<tr>
<td>7. Growth options</td>
<td>Design process Product structure</td>
<td>Value shifts to architectural applications that allow underlying design or product to be performance-tuned base on user values. Increasing importance of architectural verification.</td>
</tr>
<tr>
<td>8. Sticky information</td>
<td>Design process</td>
<td>Design platform optimizes design performance based on input from system designers. Design platform as “user toolkit” for system designers.</td>
</tr>
<tr>
<td>9. Performance mismatch (physical integration costs $\gg$ virtual integration costs)</td>
<td>Design process Production process</td>
<td>Physical SIP core integration costs exceed virtual integration costs. Value of SIP reuse limited by mask generation cost and TTM. Pressure to reduce mask generation cost and TTM, and/or to enable re-use of SIP mask subsets. Further design platform integration with foundry.</td>
</tr>
<tr>
<td>10. Growth options</td>
<td>Design process</td>
<td>Aggregation of SIP at foundries. Foundries become a natural channel for SIP and design platforms.</td>
</tr>
</tbody>
</table>
Evolution scenarios

Given that these pressures on the value chain exist, how might the design process evolve? How might the market evolve? How will this evolution change the competencies required by firms to be able to create and capture value? Note that we cannot say (nor can anyone…) with certainty exactly what will happen. But if we observe some competencies that would contribute to success across several scenarios, those would be candidates for active development.

Market scenarios

The market for VLSIC MPUs and the systems that contain them may look quite different in the future based on the outcome of two nascent battles: in the low end between PCs and high-performance appliances, and in the high end between commodity and differentiated server solutions.

The collapse of the middle. The market for computing systems today can be roughly split into four tiers:

- Servers
- Personal computers (PCs)
- High-performance appliances
- Cost-performance appliances

Today, MPUs for PCs are the largest tier (by value). One of the primary drivers of the growth of PCs as a market segment for MPU consumption is that their general purpose nature makes them suitable to solve a wide range of important problems. This has allowed the industry to commoditize PC systems to a large degree. General purpose solutions are consistent with establishing economies of scale, driving costs down, and further stimulating demand.

On the other hand, as PC hardware and software has grown in terms of computing power, it has also grown in terms of complexity. Today’s PC is overpowered and cumbersome to use for many classes of problems. In some cases, a more tailored high-performance appliance simply solves the problem better, and as such is welcomed by the market. Oft-quoted examples are Sony Corporation’s Playstation and Playstation 2000 home entertainment platforms.
A critical question from a market perspective is “to what extent will general-purpose PCs be replaced by application-specific, high-performance appliances?” Will the middle of the market collapse, leaving general purpose computing only in the server tier? Or will the PC’s flexibility allow it to remain dominant?

If the market shifts to prefer high-performance appliances to PCs and “the middle collapses”, VLSIC design firms optimized for large-scale production of general-purpose high-performance MPUs, such as Intel, are likely to be driven upwards into the server tier. Firms optimized for production of application-specific, performance-tuned systems and MPUs would likely gain share in the middle tier of the market.

The compute utility. In the server tier itself, we find another key question. It is clear that the growth of the internet – first as a means of delivering static information, and more recently as a means of delivering application, computation, and storage resources – is driving growth in the server tier. Some have speculated that the availability of high-bandwidth networks will lead to the creation of a “compute utility”, and that the bulk of servers will be sold into such utility environments. Individual business enterprises, the traditional customers in the server market, would no longer buy hardware, but would purchase services from a compute utility. The utility would then optimize its computing assets to minimize total cost of ownership, allow easy addition or removal of capacity, and allow redirection of assets among its various customers.

The question for firms designing VLSIC MPUs and server systems is “to what extent will the compute utility model commoditize server MPUs and related hardware?” Will a few general-purpose solutions come to dominate the utility’s assets? Or, as computing is driven into society’s infrastructure, will competition and cost pressure force tailored computing solutions for even subtly different applications (that is, how different might solutions be that: serve static web pages, vs. run applications, vs. store large files, vs. house critical databases)? If servers are highly differentiated, will their MPUs be differentiated as well?

It is likely that both scenarios will exist to at least some degree. Examples can be found for both scenarios. As mentioned earlier, Intel recently created a “server farm” compute-utility business that employs rack upon rack of identical server hardware. Applications are beginning to crop up for tailored server MPUs, though, especially as server-class performance needs to be embedded into an always-on network infrastructure:
Disruption, Dis-integration, and Modularity the Microprocessor Industry

“There is no classic ‘embedded’ application for the high-SRAM-content MPU, but that may change as future market demand develops for multiple-MPU-per-box internet server and communication processor applications emerge.”

Design process scenarios

Based on the analysis, the major scenarios for evolution of the design process can be tiered according to degree of re-use and targeting:

---

“Sea of Processors”

Architectural Apps, Automated Performance Tuning

SIP integration

CAD-supported custom design, SIP creation

---

Management of Technology Thesis 79/99
Mick Bass, 19 May 2000
Disruption, Dis-integration, and Modularity the Microprocessor Industry

Firms must make decisions about how to evolve their design process, in two dimensions. These decisions are somewhat independent (a desire for re-use over long time horizons usually involve targeting multiple IC processes).

Targeting in the design process involves making designs usable in multiple production processes, incorporating multiple circuit disciplines and/or IC processes into the design, and enabling production of the design at multiple foundries.

Re-use involves increasing design productivity and reducing time to market by various strategies. Let’s examine the tiers of re-use from bottom to top. **CAD-supported custom design** represents the status quo of MPU design today. This tier is characterized by large teams of electrical engineers designing libraries and creating critical building blocks for VLSIC systems. This tier forms the foundation for further evolution of the design process, and improvements here will continue to be valuable regardless of how quickly such evolution proceeds. Characteristics of this tier include: extension of the design platform to enable increased designer productivity on moderately sized custom designs in state-of-the-art IC processes; and design platforms increasing the efficiency of designer-led ports from one IC process to another. Unfortunately, the technologies supporting designers in this tier are at the top of their S-curves, and designer productivity still lags behind that required to fill chips at cost-efficient scale.

**SIP integration** represents design processes that support significant designer-guided re-use of SIP. Characteristics of this tier include: successful establishment of SIP standards, markets, and distribution channels; design platforms supportive of locating, obtaining, and integrating SIP into current designs; and foundry-led ports of SIP into their native IC processes. In general, technologies supporting this tier are at or near the bottom of their S-curves.

**Architectural Apps and Automated Performance Tuning** represents design processes that implement automated selection, integration, and/or optimization of underlying hardware in response to customer-provided performance parameters. Characteristics of this tier include: design platforms capable of turnkey, automated component selection and system integration; extensive parameterization of system performance metrics; and hardware reuse enabled by software-based “architectural re-mapping” techniques (e.g. dynamic code translation). Technologies supporting this tier are at the bottom of their S-curves, some still very much in the “ferment” stage.

The **Sea of Processors** tier represents design processes achieving extreme re-use through mapping solutions onto arrays of standard hardware. Characteristics of this tier include: limited numbers of commodity VLSIC hardware components that can be used to implement any desired system; and design
platforms involving extensive use of software mapping and optimization. Technologies supporting this tier are nearly exclusively in the “ferment” stage.

Note that it is reasonable to expect VLSIC design firms targeting different tiers of the market to make the transition from lower re-use tiers to higher re-use tiers at different times. In general, since the performance overhead associated with reuse is nonzero, I would expect reuse to occur first in the lower market tiers, where the technology is more easily able to meet the performance demands of the market. The examples described previously (Intel, Tensilica, Transmeta) appear to be consistent with this expectation.

**Required capabilities shared across scenarios**

Now, let’s think about these evolution scenarios for the market and design process, and see if there are any capabilities that are applicable across all or most of them. In particular, we’re looking for emerging capabilities that are perhaps not so important today, but will grow in importance in the future.

Inspection reveals the following such capabilities:

- *Software competency* is in general becoming more important…

- *Design platform creation* will compete strongly with design itself in terms of value creation.

- *Integration of existing components*, as opposed to design of new components, will be increasingly important. Technologies supporting re-use and integration are poised to enter the fruitful portion of their S-curve.

- *Managing external vendor relationships*, and defining modular interfaces with vendors, including metrics especially with IC foundries.

- *Intellectual property management* capabilities, including negotiating and administering contracts, licensing arrangements, and selecting components based on IP as well as technical considerations.

- *Marketplace creation and management* for SIP, including creation and definition of standards for SIP interfaces, metrics for SIP performance, and standards for licensing and use.

- *Software ↔ Hardware transformation* capabilities as required to create architectural applications and automated performance tuning, especially for those firms in the PC and low-end server market, and those in the appliance market.

The following capabilities will be required going forward, but only by a subset of firms:

- *Multidisciplinary circuit design* and integration. Since SIP is becoming a major force in the industry, many firms will be able to choose to specialize in a particular circuit discipline. These firms would supply SIP cores to firms specializing in multidisciplinary circuit integration.

- *CAD-supported custom design.* Firms will be able to choose whether to be SIP suppliers, SIP integrators, or both.
7. Recommendations

At the outset, we asked the questions:

- What should today’s microprocessor design entities do? That is:
  - In what technologies should they invest?
  - What portions of the value chain should they seek to own or control?

Having completed the preceding analysis, we are in a good position to make some recommendations to incumbent and entering firms.

1. *Start building a software competency*, if you haven’t already. It will be generally useful no matter where in the value chain you are participating.

2. *Capture the value created through marketplace formation*. Much more value will be created through the formation of SIP marketplaces than the creation of SIP itself. *Foundries* are well positioned to manage and control this process. VLSIC design firms should take steps to ensure that foundries do not end up with a natural monopoly or oligopoly on SIP distribution channels. Design firms that are positioned to establish and support standards for SIP, and to begin to aggregate libraries of SIP, should do so, lest the foundries walk away with the store:

   “TSMC has been working since last summer to make the cores of about a dozen IP suppliers compatible with its manufacturing process”\(^{37}\)

and a TSMC spokesperson says:

"[IP reuse is] a structural change that is going to have benefits [for foundries], but there’s a number of hurdles in order for it to work well,"\(^{38}\)

3. *Create value through SIP re-use and integration*. As much value will be created through design platforms, especially platforms enabling SIP integration, as through component design. If your firm already has a competency in design platforms, your roadmap should address re-use and integration of SIP.

4. *Limit the size of design teams producing SIP*, to further encourage re-use and integration, and to avoid problems with the efficiency of large teams.

\(^{37}\) Arensman (1999).

\(^{38}\) Arensman (1999).
5. Choose between system design and component design. Traditionally, component designers have created chips. In the future, component designers will create SIP and system designers will create chips. Focus on which you will excel at – the competencies required for the two are different. Recognize that the value capture capability of the two may be quite different:

"Of course there will probably be opportunities for those with valuable IP to license their technology to other chip makers. Yet most analysts believe the biggest profit margins will accrue to those who do the integration and manufacturing."

6. Designers of traditional high-performance MPUs have a strategic choice to make regarding their future focus:

MPUs for general-purpose computing systems (e.g. processors for servers) vs.
MPUs for application-specific, high-performance appliances

Choose one. The competencies required for success are sufficiently different across the two choices that it will be difficult for a single entity to succeed at both. Since it is difficult to project the future value of each of these in the market, firms might choose based upon their existing competencies and the fit with other parts of their business.

7. Entities creating VLSIC systems should be sure that they have established competencies in:

- Negotiation, contracts, licensing, and vendor management. These capabilities may be new if the entity is shifting its focus from component design to system design.
- Multidisciplinary circuit design and/or integration. To truly deliver systems on a chip requires that the integrator have access to and experience in each of the circuit disciplines, and access to foundry and IC process conducive to each of the circuit disciplines.

8. Establish a presence in Software ↔ Hardware transformation technologies. These technologies, enabling much larger degrees of hardware reuse by using software mapping techniques, have all the characteristics of a disruptive technology. They are already at work in the lower tiers of the market, and continued improvement in underlying IC process performance will eventually make them applicable upmarket as well. Both system designers and designers of complex components (e.g. MPUs) should consider investing in this capability.
Disruption, Dis-integration, and Modularity the Microprocessor Industry

Organization

Decisions about how to organize while implementing these recommendations will have a substantial impact on their eventual success or failure. This is true because the recommendations entail creating new organizational competencies, in some cases will involve shifting organizational charters, and may conflict with existing organizational culture and norms.

Professor Rebecca Henderson at MIT has established a model for an organization’s “Deep Structure” that contains components at all levels of the organization:

```
Leadership
  Structure
    Incentives
      Culture
```

Let’s examine just one of the recommendations, “create value through SIP re-use and integration”, in the context of this model. The exercise will demonstrate just how difficult such a change can be.

Many of today’s MPU design organizations come from a background of component creation. Historically, this has occurred through the process of custom design explained previously. MPUs were largely handcrafted assemblies of handcrafted libraries, with designer-led application of various productivity-enhancing CAD tools along the way.

Now, in order for a change towards the capture and re-use of SIP to occur in such an organization, this change needs to enabled at all levels of Henderson’s “Deep Structure”. Let’s take a brief look at each of these.

**Leadership.** In order for SIP creation and re-use to occur within this organization, the leaders in the organization must first recognize that this is what they *really want* to occur. Then they must lead the change. They must educate the organization, and clearly explain where they are heading and why it makes sense to go there. They must allocate staff to develop reuse-and-integration-enabled design
Disruption, Dis-integration, and Modularity the Microprocessor Industry

Platforms. To do so, they will likely need to not staff some VLSI design positions. They must manage to complete current products using existing methods while attempting to establish and protect the development of new methods. They must spend energy to see that the organization’s structure and incentives are consistent with the desired outcome of SIP re-use, and change them if they are not. They must have the patience to wait while the organization’s culture changes. Clearly, there are many potential stumbling blocks.

Structure. Further, in order for SIP creation and re-use to occur, the organization must be appropriately structured. For example, should re-use be prototyped in a standalone team, then rolled out to existing design groups? Or perhaps design teams with a re-use charter should be established to compete directly with design teams using traditional methodologies? Should the re-use-enabling platform be developed in-house or purchased from an external vendor? To whom should the team creating the design platform report to? The answers to these questions will vary depending on your particular organization, but an inappropriate or conflicted structure (for example, where the design platform team reports to a component development manager who is currently developing components using traditional methods) can severely impede progress towards the desired outcome.

Incentives. In addition to appropriately structuring the organization, people need to be rewarded for creating re-usable SIP, and then re-using it:

“Steve Svoboda, marketing director for the design-reuse business unit of Synopsys Inc., Mountain View, says companies need to find ways of rewarding engineers for making their designs reusable, rather than for just the success of a particular project. ‘It’s not clear yet how you reward the actual engineers,’ he says, ‘but it’s not primarily a technology problem—it’s a managerial and methodology and strategy problem.’”

Designers historically have been paid based on their ability to design custom solutions to particularly constrained problems. Re-use requires that designers be paid on their ability to solve problems using components that others have designed. These are quite different standards.

Incentive mismatches exist at levels throughout the organization. For example, in some organizations, design managers are compensated based on the size of their team. The more direct reports a manager has, clearly the more he should be paid. But if you want to encourage a manager to promote re-use, you need to reward him for having a smaller team, not a bigger one.

40 Arensman(1999).
Culture. Ultimately, any change needs to be reflected in the tacit assumptions and behaviors of the individuals within the organization. Culture can impede a change even when leadership, structure, and incentives are aligned. Because it involves individuals’ sense of themselves and their role within the organization, it is sometimes slow to change.

Establishing re-use and SIP integration as a dominant mode of design, for example, shifts the designer’s role from “artistic creator” to “guide for component selection and assembly”. Some designers may be threatened by this change, or may be less challenged by the design problem. Therefore, they may (consciously or unconsciously) oppose the change.

Consider another example. In many VLSIC design organizations, designers are trained to optimize performance. Circuit performance may have historically been viewed as paramount in importance to other factors. Performance optimization may have become “hardwired” into the way that designers think about design. Now we want designers to optimize re-use. But re-use sometimes conflicts with performance. It will take time for designers to resist their first, almost instinctual reaction, to add a new custom component (and add time to market) rather than re-use one that is nearly as good (and ship immediately).

Summary. Managers implementing the recommendations in this thesis should also take care to ensure that the changes are enabled at each level of the organization’s “Deep Structure”.
8. Conclusion

This thesis began hoping to answer some critical questions for microprocessor design firms: Will microprocessor design dis-integrate as did the PC industry? If so, where should they focus to create and capture value? What new competencies will be required?

Based upon a review of the existing research literature, I proposed three “general mechanisms” that create pressure to shift existing boundaries within an existing value chain:

<table>
<thead>
<tr>
<th>Sticky information</th>
<th>Improves ability to comprehend and/or communicate performance attributes to be considered, and requirements for each</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance mismatch</td>
<td>Shift design resources from less valued performance-generating tasks to more valued performance-generating tasks</td>
</tr>
<tr>
<td>Growth options</td>
<td>Leverage resources</td>
</tr>
</tbody>
</table>

Then I analyzed the VLSIC design industry, and found it attractive. Upon examining the VLSIC design value chain, I found 10 instances of the above general mechanisms (Table p. 81).

Some fundamental insights were:
- Productivity gap exists between designer output and available transistors
- Compelling force towards SIP, re-use, and design platforms that automate integration
- Dis-integration and performance tailoring is likely to occur. Lower tiers of market go first.
- Value capture will occur through design platforms and SIP marketplaces and channels
- Software ↔ hardware transformation looks like a disruptive technology, is already occurring in the lower tiers of the market, and can be expected to move upmarket.

Key new competencies for VLSIC design firms were:
- Software, in general
- Design platform creation, both for component design and SIP integration
- IP management
- Marketplace creation and management, including distribution channels
- Software ↔ hardware transformation technologies
9. Appendices

Appendix A – Examples of the performance mismatch mechanism

Performance Surplus. When performance is in surplus, suppliers to this design boundary may attempt to modify their own design process or design processes further upstream to emphasize other, more highly valued, performance attributes. Alternatively, firms may enter to augment the value chain at this point with alternative designs that emphasize more highly valued performance attributes. This has recently occurred in the web search space, as evidenced by the entry of directhit.com. Direct Hit realized that their potential customers – portals and large corporate intranets – no longer placed a high incremental value on improvements in the speed with which a large set of documents could be searched. Search engines had become “fast enough”. Instead, Direct Hit realized that the portals (and, indirectly, end users) would value search engines that produced highly relevant search results. Further, they could capture this value by creating intellectual property that was strongly appropriable. From the portal’s perspective, the performance mismatch mechanism worked by creating a force on suppliers to re-organize the upstream design processes and intermediate products to stop focusing solely on search speed, and begin to perform as well on search relevance.

Alternatively, when performance is in surplus and the supplying entity has not yet modified its design and intermediate products to deliver value elsewhere, it then falls to the remainder of the design value chain to attempt to transform the excess performance into something useful. This is what Intel did in creating its MMX instructions, and in creating an environment that fostered the growth of multimedia applications on the PC platform. The performance of Intel’s CPUs on standard integer performance benchmarks had grown sufficiently large that further improvements were no longer highly valued by downstream PC system integrators, IT procurement agents, and end users. The integer performance of Intel’s CPUs was simply fast enough for most common home and office computing tasks (for example, word processing, spreadsheet computation, and web surfing). In response, Intel sought other types of applications to stimulate demand for the performance that it already had. By tweaking its instruction set to include special purpose multimedia instructions, Intel was able to make real-time video viable on the desktop with a software-only approach. This opened the door for new firms, such as RealNetworks, to

---

41 I use “entity” to signify, as appropriate, “team”, “organization”, or “firm”. The supplying entity might be any of: a team within your organization, an organization within your firm, or a different firm.

enter this space and fill out the design value chain required to deliver whole solutions to customers. Intel was able to transform the surplus performance on one performance measure (integer computation) into another type of performance (frames per second achievable with software-only MPEG decompression) that would be more highly valued by intermediate and final markets.

Sometimes when performance is in surplus it may become apparent that a particular boundary in the design value chain no longer makes sense. In some cases there may sufficient surplus performance that a new design approach is possible that obsoletes an entire portion of the design value chain, and significantly changes the structure of the resulting products. In such cases, firms competing within the obsolete segments of the value chain suddenly face competition from a new, alternative value chain. Competition shifts, from competition within a particular portion of the chain, to competition between the two sub-chains. This is arguably what is happening with the emergence of Application Service Providers. Enterprise computing performance has advanced sufficiently on several required performance attributes (networking bandwidth, transactions per second, scalability, clustering capability, and reliability) that further improvements are valuable only to the largest of consuming firms. This is because the performance improvements are often accompanied with corresponding increases in the complexity of configuration and maintenance of the systems, which only the largest organizations, having sufficient scale, can afford. Even in the largest organizations, for a large class of enterprise applications (e.g. Enterprise Resource Planning, e-mail, simple word processing), further performance improvements are not valued highly. This has opened the door to a completely new form of service delivery. Rather than themselves purchasing, installing, configuring, and maintaining hardware and software capable of running an array of applications, consuming firms can simply purchase an application service, an instantiation of a well-defined application that runs with remote computing and storage at the facility of the service provider. The service provider purchases, installs, maintains, and upgrades hardware and software. The consumer simply configures and accesses the application, typically through a web browser. Numerous examples of this arrangement exist today, including Sun’s StarOffice suite of office applications and Microsoft’s Hotmail e-mail services.

The point is that surplus performance has forked the value chain and given rise to a competing sub-chain for value delivery. This fundamentally affects both the design processes and the product structure of entities involved in either fork of the chain. Hardware computing systems optimized for use in an ASP environment will look different from hardware systems optimized for use within an enterprise. Traditional client-server software architectures are very different from the architectures required to make an application deployable as a web-accessible service. The design processes used to create them also
differ. It is possible that both forks will be durable, or that one will eventually dominate. Even if both forks endure, the majority of value may migrate from one fork to the other. Entities that recognize themselves in this situation must actively decide: On which fork(s) will I create value? How will I capture value there? How must my competencies change to be able to do so?

**Performance Deficit.** Now let’s consider the cases where, on some performance attribute, performance is insufficient at a particular boundary in the design value chain.

When performance is at a deficit, suppliers to this design boundary may change their own design process, or design processes further upstream, to focus on improving the performance attribute in question. Several years ago, this was the case in two microprocessor design efforts within HP. The performance delivered by pre-silicon functional simulation methods had become insufficient to adequately verify the functionality of the chips in the schedule that was available. HP chose different responses on the two design efforts. In the first, they changed their design method to employ a field-programmable hardware simulation system provided by Quickturn Systems. In the second, they invested in a proprietary logic simulator that used a novel parallel computing technique to increase simulation throughput by about a factor of 30. In the first case, Quickturn had recognized an opportunity for new firm entry into the design value chain upstream of the functionally verified design handoff. Quickturn then developed customer relationships with incumbent firms in the existing value chain, who in turn changed their design methods to use the new approach. In the second case, the incumbent supplying firm (HP) itself invested to wholly solve the problem within its portion of the upstream value chain.

Alternatively, when performance is at a deficit, the entity downstream from a design boundary may change its design process, or design processes further downstream in an attempt to bolster performance. This is what many web content providers have done to boost the speed at which their pages are displayed to their customers. Many content providers have invested substantially to ensure that upstream performance is optimized: that their servers are sufficiently fast, and their connection to the internet backbone has sufficient bandwidth. However, issues with downstream quality of service still hamper end customers from receiving quick web page updates, especially on sites that are graphically intense. This situation provided an opportunity for Akamai to enter the downstream value chain. Akamai offers a process by which content providers can cache commonly accessed pages and images in Akamai servers strategically placed throughout the internet infrastructure. When a customer accesses an “Akamaiized” page, an Akamai server computes which cached copies of the information would be served most quickly.
to the requesting customer, and routes the information request to those caching servers. Akamai accumulates revenue per page served. From the perspective of the content provider, the upstream performance shortfall (ISP unable to guarantee connections with sufficient quality of service to geographically diverse customers) creates a force leading to changes in the downstream design methods (How are pages authored? How are they served?). The need for change creates an opportunity for firms such as Akamai.

Sometimes a performance shortfall at a design boundary leads stakeholders along the value chain to eliminate the design boundary completely in an attempt to optimize performance. A good example is the transition currently underway in some markets towards eliminating ATM from data transport protocol stacks. Explosive growth in the internet has established Internet Protocol (IP) as a de facto standard for connecting heterogeneous networks. Different networks can have different underlying physical and/or logical implementations, but as long as both run IP at the top layer, the networks will be able to communicate.

Many current networking products implement IP on top of another networking protocol, Asynchronous Transfer Mode (ATM). ATM is a data transport protocol that includes features for flexibly provisioning bandwidth, fault tolerance, and addressing & routing. ATM is a logical encoding layer. Once encoded in the ATM protocol, data is then further encoded into Synchronous Optical Network (SONET) format for physical transport over a fiber optic backbone. ATM has various performance attributes, including the overhead introduced by the protocol, its ability to configure and guarantee connections of various bandwidths, and its cost to configure and maintain. In certain markets (primarily Internet Service Provider networks), point to point network capacity is the primary performance measure, and current products that implement IP over ATM over SONET under-deliver on this metric. The protocol overhead introduced by ATM contributes to this problem. This overhead enables the flexible features of ATM, but these features are not valued highly by the ISPs.

This has created an opportunity for firms to design and deliver new products that eliminate ATM from the protocol stack and implement IP more directly on top of SONET. Such a change increases the effective data rate to 95% of the underlying physical line rate, from the 80% that is achieved when ATM is included in the protocol stack. Once again we have a situation where the value chain has forked, and competition exists not only within each sub-chain (e.g. competition for share of ATM routers), but across competing sub-chains (e.g. share of IP on ATM products vs. IP on SONET products). The design
methods, competencies, and investments required in these two sub-chains differ, and firms must choose the sub-chain(s) in which they will create and capture value.
Appendix B – Examples of Hardware ↔ Software Transformation Technologies

VLIW architectures move a critical portion of processor functionality (instruction dispatch among execution modules) from an integrated hardware implementation to a modular software implementation. Non-VLIW architectures rely on integrated hardware to examine the instruction stream and dispatch instructions to available execution modules. The processor’s microarchitecture (e.g. the number of execution modules, cycle time for each module, and execution dependencies) need be visible only to the integrated dispatch unit, and specifically need not be visible outside the processor. VLIW architectures instead implement the dispatch functionality in software (typically in the compiler that translates application source code to object code executable on the target processor). The approach forces a modular implementation because the microarchitecture must be visible outside the processor, and specified via an interface agreed upon by both the processor team and the compiler team. VLIW architectures offer performance increases by enabling instructions to be scheduled and dispatched to a much larger number of execution units than would be feasible using a pure hardware approach in real-time. The microarchitecture interface enables the compiler to implement the required dispatch and scheduling in software at compile time, incurring a one time cost to enable repeated program execution with higher performance. The technique allows more effective use of silicon and eases the design task by enabling a larger number of modular execution units to be included in the processor core, some of which may be re-used. Note also that the technique makes the compiler much more central to both value creation and value capture, and mandates augmented competencies in compiler design. An example of such a VLIW architecture is Intel’s IA-64 architecture.

Dynamic object code translation (DCT) translates streams of instructions written for one instruction set architecture (ISA) to another ISA and/or processor microarchitecture. This can be done using a similar approach as described above, but by mapping the instructions to available execution units at run time instead of compile time. Hewlett-Packard will use this approach to ensure that IA-64 processors are backward-compatible with existing software that was compiled for execution on PA-RISC processors. Another current example is Transmeta Corporation’s “Crusoe” processor, which uses DCT around a streamlined VLIW processor core to provide a “virtual processor” that is x86 compatible and dramatically reduces power consumption for laptop applications (more on this later...).

Dynamic mapping of programmable logic substitutes hardware implementations of several modules with a set of programmable logic that can be used to reproduce the functionality of each module in turn. The technique increases the generality of the hardware by relying on programmable logic for
certain functions, and reduces time to market by moving implementation of the selected modules into the software domain, where their design can proceed in parallel with production of the hardware. A variant of this technique is to substitute a hardware implementation of a particular module with a software implementation running on an MPU or MCU with sufficiently high performance. The same effects are achieved – increased hardware generality and parallelism between hardware production and software design.

"Sea of Processor" approaches take the above approach to an extreme and recognize that, given sufficient MPU performance, all hardware modules in a system could be substituted by a software implementation running on an MPU. Sea of processor architectures interconnect a number of general-purpose MPUs that, when paired with appropriate software, can emulate each of the system’s requisite modules. This approach makes the hardware completely general-purpose, and completely de-couples system design from system production. That is, the design of the software implementations of the modules that comprise the system can proceed independent from the production of the "sea of processor" hardware on which the system will run. These benefits come at the expense of performance. Such a system would offer less performance than a corresponding optimized hardware implementation. But it would likely come to market more quickly, and have a lower cost. The sea of processors approach might enable huge economies of scale in hardware production, as the hardware would be widely applicable to a broad class of applications. This is the general approach taken, albeit in a scaled down fashion, in many entrant “network processor” firms.
Appendix C – Key Interfaces in MPU Product Structure

Physical Interfaces.

Execution unit – a computation module capable of performing some specified set of computations on a specified data set. For example, a multiply and accumulate unit, an integer adder, etc. MPUs are typically comprised of many execution units along with other modules for storing computation results, and mechanisms for intelligently moving data among them.

SIP core / MPU core – a reusable piece of silicon intellectual property. An entire MPU core (as in the ARM’s product line) is a good example, but more granular SIP could also exist.

Physical die – the piece of silicon on which a system or a portion of the system is implemented.

Physical module – Die must be packaged in some way to effectively interface with the rest of the system. Sometimes several die are packaged together in a single physical module.

Board – one or several modules are sometimes integrated onto a printed circuit board with the remainder of the system. SoC implementations strive to reduce the number of modules on the board, implementing the entire system in one die and/or physical module.

Box – the physical product housing, with which the end customer interacts.

Software Interfaces.

Architectural applications – software layers that map or transform the way that the computing instruction and data streams are presented to the underlying hardware, so as to optimize performance for desired performance attributes.

Compiler – A static architectural application that translates high-level programs into object code that can be executed on a particular hardware platform.

Operating system – Special programs that allocate and manage the use of high-level computing system resources.

Application software – Programs that implement useful work for the end-customer.
10. References

Academic Articles


Disruption, Dis-integration, and Modularity the Microprocessor Industry


Books


Business Press


**Interviews**

Chris Rowan, Chief Executive Officer, Tensilica. 01/12/00.

Management of Technology Thesis 98/99

Mick Bass, 19 May 2000
Trade Publications & Research Reports


