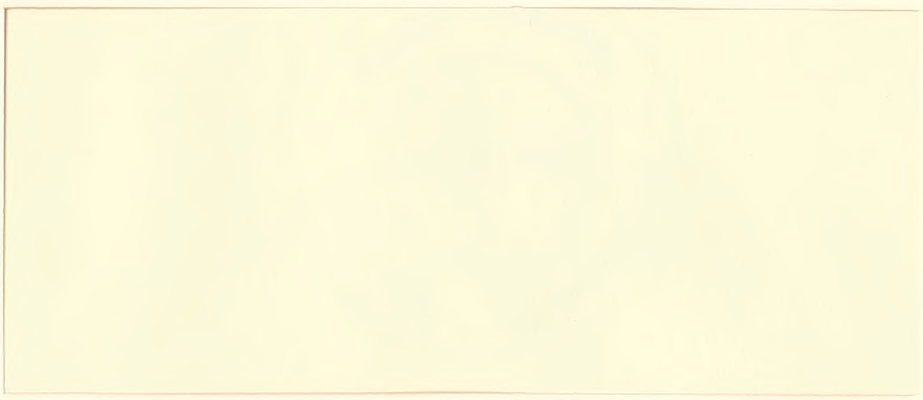
WORKING PAPER
ALFRED P. SLOAN SCHOOL OF MANAGEMENT

The Hypercube of Innovation.

Allan N. Afuah and Nik Bahram

MIT Sloan School WP #3481-92 October, 1992

MASSACHUSETTS
INSTITUTE OF TECHNOLOGY
50 MEMORIAL DRIVE
CAMBRIDGE, MASSACHUSETTS 02139



The Hypercube of Innovation.

Allan N. Afuah and Nik Bahram

MIT Sloan School WP #3481-92 October, 1992

The Hypercube of Innovation.

Abstract:

Innovation has frequently been categorized as either radical, incremental, architectural, modular or niche, based on the effects which it has on the competence, other products, and investment decisions of the innovating entity. Often, however, an innovation which is, say, architectural at the innovator/manufacturer, may turn out to be radical to customers, incremental to suppliers of components and equipment, and something else to suppliers of critical complementary innovations. These various faces of one innovation at different stages of the innovation value-adding chain are what we call the hypercube of innovation. For many high-technology products, a technology strategy that neglects these various faces of an innovation and dwells only on the effects of the innovation at the innovator/manufacturer, can have disastrous effects. This is especially so for innovations whose success depends on complementary innovations, whose use involves learning and where positive network externalities exist at customers.

We describe the hypercube of innovation model and use it to examine RISC and CISC semiconductor chips, and supercomputers, and suggest how firms can better manage the relationships along the innovation value-adding chain using the model. The model forces innovation managers to think in terms of their customers, suppliers and complementary innovators.

INTRODUCTION

Ever since Schumpeter, scholars of innovation, in an effort to better understand how to manage the process of innovation, have tried to categorize innovations as a function of what the innovations do to the skills, knowledge, capabilities, and existing products of the innovating entity. Schumpeter himself described innovation as the "a historic and irreversible change in the way of doing things" and "creative destruction" [Schumpeter, 1947]. Abernathy and Utterback [1978] examined the different stages of an innovation from a fluid state through a transition state to a specific state and what the implications are for the innovating firm. Abernathy and Clark [1985] grouped innovations into four categories based on whether the innovation enhanced or destroyed the innovating firm's technology or market competence. Henderson and Clark [1990] also grouped innovations into four categories based on if the innovation reinforced or overturned knowledge of the components, key concepts or the linkages between the components and concepts for the innovating entity. Tushman and Anderson [1986] classified innovations as "competence destroying" or "competence enhancing" for the innovating firm.

One thing these enlightening categorizations of innovations have in common is that they are based on the effects of the innovation on the innovating entity's capabilities and existing products, and neglect the effects of the innovation on the competence and assets of suppliers of key components and equipment, customers, and suppliers of complementary innovations. The fact is that for many high-technology products, an innovation which is, say, architectural at the innovator/manufacturer, may turn out to be radical to customers, incremental to suppliers of components

and equipment, and something else to suppliers of critical complementary innovations. These various faces of one innovation at different stages of the innovation value-adding chain are what we call the hypercube of innovation.

Stated differently, the studies just cited have the innovating entity asking the question: "What is the impact of this innovation on my organizational capabilities, competence, existing products, knowledge of components, key concepts and linkages between them". In the hypercube of innovation approach we are suggesting that in addition to probing what the innovation will do to its competence and assets, the innovating entity must also ask the question: "What will my innovation do to the competence and products of my suppliers, OEM customers, end-user customers, and suppliers (some of which are competitors) of key complementary innovations -- i.e. what is the impact of the innovation at the various stages of its value-added chain?" For products whose design and manufacture depend on critical components and equipment from suppliers, products whose diffusion depends on complementary innovations or which offer customers positive network externalities, and involve a high level of learning, categorizations that neglect what the innovation does to this supporting network along the innovation value-added chain can have disastrous consequences. The hypercube model forces innovation managers to think in terms of what the impact of their innovation is going to be on customers, suppliers of critical components, equipment, and complementary innovations. Since customers and users can also be future innovators (von Hippel, 1988), the hypercube may also help the innovating entity tract potential competitors and complementary innovators.

In this paper we describe and illustrate the innovation hypercube model using anecdotal examples from different industries and then use it to

examine the RISC (Reduced Instruction Set Computers), CISC (Complex Instruction Set Computers) semiconductor microchips, and supercomputers industries. From the examination, we suggest some measures that the innovating firm has to take to avoid getting lost in the cube. RISC, CISC and supercomputers are particularly interesting examples for various reasons: They depend on complementary innovations for market success, exhibit positive network externalities at customers, and require complex equipment and components from suppliers.

The paper is arranged as follows. In Section I, we detail the hypercube model with anecdotal examples from different industries. In Section II, we briefly describe what RISC, CISC, and supercomputers are all about, and apply the hypercube model to these industries. In Section III we outline, following a summary of where we have been, what the implications of the hypercube are to firms and industries. Section IV is the conclusions and suggestions for further work.

Section I: The Hypercube Model.

Categorizing Innovations:

Abernathy and Utterback (1978) found that as innovation evolves, product innovation gives way to process innovation making it difficult for the innovating entity to revert to new product innovations; i.e. the competence of the innovating entity is effectively destroyed. The innovation goes from a fluid stage to transition stage to specific stage. They outlined the properties of each stage and the implications for the innovating firm's strategy. Using the automobile industry, Abernathy and Clark(1986) grouped innovations into four categories depending on the impact of the innovation on the innovating

firm's capabilities and knowledge of its technology or market. They didn't address the impact of each of the innovations on the capabilities and assets of their suppliers of components, customers, and suppliers of complementary products. Using extensive data from the photolithography industry, Henderson and Clark (1990) classified innovations according to whether the innovation overturned the existing knowledge of core concepts and components, and the linkages between them. They classified an innovation as radical if the core concepts of the innovation as well as the linkages between them overturned existing ones; architectural if the core concepts were being reinforced while the linkages between these core concepts and componer.. ·- the product were changed; incremental if the core concepts were reinforced while the linkages between them were unchanged; modular if the core concepts were overturned while the linkages between the concepts were unchanged. As was the case with the automobile industry, the impact of the innovations on the capabilities and assets of suppliers, customers and suppliers of complementary products was not considered. Given the nature of both industries, the conclusions arrived at by these authors in those studies shouldn't change that much with an analysis that uses the hypercube model.

However, for industries where at least one of the following is true : complementary innovations are critical to the diffusion and success of products; where learning by customers is critical, expensive and often results in lock-in; where positive network externalities at customers are common and where equipment and critical components (that go into the innovation) from suppliers can be innovations in their own right; an examination of the effects of an innovation cannot be limited to the impact on the capabilities, competence and assets of the innovating entity. One must also look at the impact of the innovation on the capabilities of suppliers of components,

customers, and complementary innovators. The hypercube of innovation addresses this.

The Model:

For this model, we will focus on product innovation as the unit of analysis. The product –the final output of the innovating entity – needs critical components or high tech equipment as inputs and can be sold directly to end-users or sold to an OEM (original equipment manufacturer) who adds value to it and then resells it to end-users. The product also possess some subset of the following: 1) it requires some considerable skill or knowledge to use or maintain. This can be obtained by learning. 2) The product is more valuable to an owner the more people that own one. It possesses positive network externalities. 3) Complementary innovations are critical to diffusion and use of the innovation.

As pointed out earlier, an innovation that is architectural to the innovating entity may be radical to customers and suppliers, and incremental to complementary innovators. The hypercube of innovation model examines these different faces which an innovation assumes at the different stages of the innovation value-added chain, i.e. at the innovating entity, suppliers, customers and complementary innovators, and suggests how the innovating entity can best deal with them. It looks at the impact that an innovation has, not only at the innovating entity, but also at suppliers of components, OEM customers, end-users, and complementary innovators. It depicts relationships that are multidimensional in nature. In particular, the hypercube is a 4-dimensional cube with each of the stages of the innovation value-added chain representing a dimension, and the location of any innovation in this 4-dimensional space being determined by the "intensity" of the innovation

along each of these dimensions; where intensity is a measure of how radical the innovation is, using an ordinal scale, say, of incremental = 1, modular = 2, architectural = 3, and radical = 4, with intensity increasing from incremental to radical. Because of the difficulties in visualizing things in four dimensions, however, we have transformed the 4-dimensional hypercube to the 3-dimensional cube of Figure 1 (and later, to the two dimensional GREEN-RED zone map). In Figure 1, the transformed hypercube (shown as a parallelepiped) has a cross-section (X and Y axes) that categorizes innovations according to their impact on the capabilities, competence, assets, and products of the actor in question, and a length on which are located the different actors on the innovation value-adding chain, viz.: suppliers of critical components that go into the innovation, customers [OEM (original equipment manufacturer), and end-user], and suppliers of complementary innovations. Figure 2 further explodes these stages of the chain for better visualization. We emphasize the fact that the innovating entity can use any criteria for categorizing innovations in the X and Y axis.

Tables 1a,1b, 1c and 1d list the range of possible impact of an innovation at the innovating entity, suppliers, customers, and complementary innovators, and we briefly describe what is possible at each stage.

The Innovating Entity:

The focus of most innovation literature has been on the impact of an innovation on the capabilities and assets of its innovator. At the innovator, the primary concern has been what the impact of the innovation is going to be on its organizational competence –whether it enhances or destroys it [Abernathy and Utterback, 1978; Tushman and Anderson, 1986], on the core

concepts and linkages between those core concepts of the product [Henderson and Clark, 1990], on existing innovations, and on the willingness of management to invest in the innovation [Henderson, 1992]. Any categorization framework can be used but for this paper we choose the Henderson and Clark [1990] model and classify an innovation as radical if the core concepts of the innovation as well as the linkages between them have overturned existing ones; architectural if the core concepts are being reinforced while the linkages between these core concepts of the product are changed; incremental if the core concepts are reinforced while the linkages between them are unchanged; modular if the core concepts are overturned while the linkages between the concepts are unchanged.

As detailed in Table 1a below, the innovating entity has to recognize and take the necessary corrective action if the innovation obsoletes or enhances previous designs, if it destroys or enhances knowledge gained in previous designs, if the innovation cannibalizes older products, and if it can be used with previous complementary products or not.

THE INNOVATING ENTITY	
Asset or Activity	***Possible Impact (Range -- Best to worst)***
Core Concepts	Enhances or obsoletes core concepts from previous innovations
Linkages between core concepts and components	Enhances or destroys previous knowledge of linkages between core concepts and components
Product Components	Components remain the same or change
Competence	Enhances or destroys other competencies (skills & knowledge from previous innovation)
Existing products	Enhances use of previous products or cannibalizes them.
Complementary innovations from previous products	Can use or not use complementary innovations from older products
Institutional support	Any government or other institutional research subsidies?

Table 1a: The effects of the Innovation at the Innovating Entity

Customers:

The impact of an innovation on the capabilities and assets of the innovator's customers has very important implications for the market success of the innovator. Unfortunately, most innovation studies have focused on the impact of the technology on the innovator's knowledge of technology and market, while ignoring the impact on customers' capabilities and assets. There are at least four areas where the impact of an innovation at a customer can have serious effects: Learning, positive network externalities, compatibility with complementary or old products and continued use of old products.

Learning:

Many complex high-technology products require that users invest time and money in learning how to operate and maintain the products. An innovation that destroys the knowledge that the customer has acquired, has a smaller chance of being adapted than one that enhances this knowledge and skills. Thus we expect a person who buys a computer and learns the computer's operating system, to be less willing to buy another computer with a different operating system than one with the same operating system; unless there is another program that can make the new operating system transparent to the old user.

Positive network externalities:

A product or skill is said to possess positive network externalities if it is more valuable to an owner the more people that have it. Positive network externality has its origins from the telephone network where one's telephone is more valuable the more people are connected to one's network. The more friends you have that own a computer that is compatible with yours, the more valuable your computer is to you because you can share software and

innovative ways of using the computer. An innovation that destroys this positive network externality does not stand a good chance of being adapted by customers.

Compatibility with complementary products:

Using the computer example again, a personal computer user who invested in a Lotus 123 spreadsheet would prefer not to switch to a new computer that requires him to buy a new spreadsheet.

Built-up assets:

An airline that has built maintenance facilities for Boeing 737s but has to change to a fleet of Airbus A320s, will have problems with the new parts inventory and maintenance procedures that must now replace the old one.

A user who has written her own applications programs to run a Macintosh will not be easily convinced to switch to an IBM personal computer, if his Macintosh programs cannot run on the new machine.

From all these, it is evident that the innovating entity must make sure that the innovation is 1) not going to destroy the skills and knowledge that its customers learned with previous innovations, 2) not going to destroy any positive network externalities that previous innovations may have created at customers, 3) customer's complementary products can still be used with the new innovations, 4) built-up assets don't have to be destroyed.

CUSTOMER	
Asset or Activity	***Possible Impact (Range -- Best to worst)***
Learning	Enhances or destroys skills & knowledge acquired from previous product.
Built-up assets	Enhances or destroys use of assets built around previous innovations
Positive Network externalities	Enhances or destroys positive network externalities
Complementary innovations from previous products	Can use or not use complementary innovations from older products
Product Design	Enhances or obsoletes previous design.
Design knowledge	Enhances or destroys previous design knowledge
Product Components	Components remain the same or change

Table 1b: The effects of Customers

Complementary Innovators:

The huge success of personal computers since their introduction in the late seventies would not be as phenomenal were it not for complementary innovations like spreadsheet and word-processing software. Innovators not only have to watch out for the inertia of older innovations and the momentum of newer ones, but may also have to cooperate (via, e.g., strategic alliances) with the complementary innovators to produce complementary innovations.

COMPLEMENTARY INNOVATORS	
Asset or Activity	*Possible Impact (Range – Best to worst)*
Inertia of old complementary products	Keeps up with inertia of old complementary innovations
Momentum of new products	Keeps up with the momentum of new complementary innovations
Product Design	Enhances or obsoletes previous design (of complementary product).
Design knowledge	Enhances or destroys previous design & mfg knowledge of complementary product
Product Components	Components remain the same or change
Competence	Enhances or destroys skills & Knowledge from previous products
Existing products	Enhances use of previous products
Positive Network externalities	Enhances or destroys positive network externalities for complementary products
Complementary innovations from previous products	Can use or not use complementary innovations from older products

Table 1c: The effects of the Innovation at Complementary Innovators

Suppliers of components and equipment:

Some high technology product innovations depend heavily on component and equipment innovations from their suppliers. Examples are aircraft and supercomputers. The former cannot move into supersonic flight without the right innovations in engine technology. Most of the gains that we have seen in computer performance have come from innovations in the semiconductor chips that go into them.

SUPPLIERS of COMPONENTS and EQUIPMENT	
Asset or Activity	*Possible Impact (Range -- Best to worst)*
Component and/or equipment Design	Enhances or obsoletes previous design of component or equipment supplied for previous innovation
Design Knowledge of components and/or equipment	Enhances or destroys previous design and mfg knowledge of components or equipment supplier for previous innovation
Competence	Enhances or destroys skills & knowledge used to supply components or equipment for pervious innovation.
Old products	Enhances or destroys use of previous components or equipment.

Table 1d: The effects of the Innovation at Suppliers of Key components or equipment.

THE GREEN-RED ZONE map:

From the hypercube, successful innovations are those that reinforce core concepts and linkages at all stages of the innovation value-added chain. This is best illustrated by the *Green -Red Zone* map of Figure 3, which is a two-dimensional further simplified version of the hypercube. It is a map of the different faces that an innovation assumes at the different stages of the innovation value-added chain. The green zone is where innovations reinforce core concepts, skills and knowledge, and an innovation that falls in this zone at the innovator, supplier, customer and complementary innovators, should be pursued rigorously.

The red zone covers the area where previous core concepts are overturned, and competence destroyed at the various stages of the chain. This is the zone for radical innovation. Any innovation whose map passes through this zone, especially at the customer stage, should be avoided unless a subset of the following is true: 1) The price/performance ratio of the innovation, as viewed by the customer, outweighs any losses incurred as a result of competence or positive network externality destruction. This happens for example when the physical limit of an older technological trajectory has been reached and the only way to overcome this physical limitation is to move to a new technological trajectory -- a move that often means destruction of competence acquired during the evolution along the

older trajectory but great improvement in some key parameter. 2) New markets where customers have not yet had time to build any innovation-specific skills and knowledge, and competence destruction is not an issue. 3) Complementary innovations, that allow customers to keep their competence and positive network externalities, exist. An example of such a complementary innovation would be a software package designed to allow PC users who are only familiar with DOS to be able to sit at a Macintosh and use DOS as they would on a PC, making the Macintosh operating system transparent to the user so that customers' competence is not destroyed when a customer moves from one machine the other. 4) When institutional requirements mandate the innovation. Electric cars for LA are an example.

If the map passes through the yellow zone, the innovator should proceed with caution.

Referring again to Figure 3, innovation A may present the innovator with more problems than innovation B since A's map through the innovation value-added chain passes through the RED zone while B's doesn't.

Some examples:

The hypercube of innovation concept is best illustrated with some examples. We will use the cases of the SDK (Dvorak Simplified Keyboard) keyboard, the electric car, IBM's OS-2 operating system and Microsoft's Windows. The DSK (Dvorak Simplified Keyboard) is an example of an innovation that was architectural to the innovating entity, incremental to suppliers of components and complementary products but radical to customers. Figure 4a shows the GREEN-RED zone map of DSK. DSK is a keyboard arrangement that by many estimates allowed people to type 20-40%

faster than with the QWERTY arrangement that most of today's keyboards have. But by the time the DSK innovation was being marketed, the QWERTY keyboard had been adapted by many customers who learned how to type with it [David 1985]. Switching from QWERTY to DSK meant two things to the customer who had already learnt to type with the former: 1) He/she would have to learn how to type again effectively abandoning the old skills and knowledge of QWERTY. 2) He or she would have a smaller market for his/her skills since more potential employers needed people with QWERTY typing skills. Customers who didn't know how to type at all realized that the QWERTY skills would be more valuable to them since more people and more places of employment use the QWERTY keyboard arrangement. This phenomenon where a product or skill is more valuable the more people that have it, is called positive network externality. So, to potential customers, adoption of the DSK would destroy their competence and/or positive network externalities, and therefore constitutes a radical innovation.

To the innovators of DSK this was an architectural innovation since the core concepts and components for the keyboard had not changed but the linkages between them had changed since the keys were arranged different.

To suppliers of components or complementary products, DSK had no impact on their skills, and products.

There may be other reasons why the DSK keyboard failed to displace QWERTY despite the former's superior performance, but the fact that this innovation was radical to customers has to be a key one.
Figure 4a shows the map through the value-added chain for the DSK keyboard.

The next example, the electric car, is still under development. But we can speculate, for illustrative purposes, on what the innovation's impact is

on the innovation value-added chain. This is a radical innovation to the car companies, to suppliers of key components like the power train, and to suppliers of the key complementary innovation–gasoline. But to customers, it will be an incremental innovation. The GREEN_ZONE map is shown in Figure 4b. What we know as the power train – engine, transmission, fuel injection, and exhaust system – of the gasoline-powered automobile is being replaced by the an electric motor, battery and electric motor controller in the electric car [see for example, Pratt, 1992]. Thus, not only are the key components and design concepts for the electric car different from those of the gasoline-powered car, the linkages between them are also different. For gasoline-powered car manufacturers, development of the electric car is a radical innovation. To suppliers of the power train components for gasoline-powered cars, the electric car destroys a lot of their competence, and is also a radical innovation to them. The electric car also runs on electricity, not gasoline, and so to gasoline companies, the electric car is also a radical innovation. To customers, however, it is an incremental innovation, since drivers of gasoline-powered cars can keep their driving skills, and other knowledge of operating cars, but get a car that emits less pollution. They may have to throw away that old container for gasoline.

The third and fourth examples are from the world of computers that we will explore more later. When IBM, and Microsoft, two of the largest beneficiaries of the PC and PC compatible market, found out how popular the "look and feel" of the Apple Macintosh personal computer was becoming, they decided to develop an operating system with a similar "look and feel" called OS-2 for IBM PCs and PC-compatibles. OS-2 would offer many advantages over DOS including multitasking (have the computer run more than one applications program at any one time). To both firms this was a

radical innovation as core concepts would have to be changed to support multitasking and other key factors. To most customers of the IBM PC and PC compatibles who had already learnt to use the DOS operating system and had seen the advantages of Icon and windows based user interface of the Apple Macintosh, OS-2 would be an incremental innovation. This was particularly true since all DOS applications would run under OS-2.

Faced with the daunting challenge of a radical innovation Microsoft and IBM parted ways with IBM advocating the investment in making the OS-2 radical innovation and Microsoft favoring a more incremental innovation. Microsoft's strategy led to the creation of Microsoft Windows several years in advance of IBM's introduction of OS-2. Although OS-2 is a technically better product, the revenues generated by Microsoft Windows and the ensuing increase in shareholder value suggests that Microsoft's approach may have been the more successful of the two. Microsoft has since put plans in place to enhance Microsoft Windows to Microsoft Windows NT which will have similar functionality as IBM's OS-2. Meanwhile, Microsoft Windows has a 10 to 1 advantage over OS-2 in installed base. The map of both innovations through the innovation value-added chain is shown in Figure 5.

Figure 6 explodes the hypercube to show cross-sectional slices of the cube at each stage of the innovation value-added chain. It shows where the innovation of the DSK keyboard by Dvorak, a keyboard designer/manufacturer, and of the electric car by gasoline-powered car designer/manufacturer, fit on the innovation hypercube. The DSK innovation is architectural to its innovator, incremental to suppliers of components and complementary innovations but radical to customers. Each face of the hypercube or stage of the value added chain is shown in figure 6 with the kind of innovation as perceived at that stage. The electric car

innovation is radical to the gasoline-powered car manufacturers, suppliers, and complementary innovation suppliers, but incremental to customers of the cars. OS-2 was radical to IBM, Microsoft Windows was incremental to Microsoft and both were architectural innovations to complementary innovators like Lotus and most importantly, incremental to customers.

Summary of the model:

An innovation that is incremental to the innovating entity may be radical to customers, and something else to complementary innovators and suppliers of critical components for the innovation, and a technology strategy that dwells only on the impact of an innovation on the innovating entity, may be in for disastrous consequences. The hypercube model forces managers of the innovating entity to examine the impact of their innovation at all the stages of the innovation value-added chain. The model suggests that innovations that destroy competence, positive network externalities, and assets at any stage of the chain, especially at customers, should be avoided except under special circumstances. The map of such an innovation passes through the red zone (see figure 3). Those that reinforce core concepts, and enhance competencies (the green zone) should be pursued. Somewhere between these two extremes is the yellow zone. Any innovation whose map passes through this zone should be pursued with a lot of precaution.

Finally, the innovator should monitor the inertia of older innovations and the momentum of newer ones, to take advantage of them.

We are now ready to apply the model to RISC, CISC and supercomputers.

Section: II: The Hypercube: The cases of RISC, CISC, and Supercomputers.

RISC & CISC Computing Architectures

CISC:

In 1970 Intel Corporation invented the first microprocessor, a microchip implementation of the Central Processing Unit (CPU) of a computer. This was a 4-bit microprocessor, which means that it could process four bits of information at any one time. As time went on, Intel and its competitors like Motorola made incremental improvements to the components and linkages between the components of the microprocessor. They introduced 8-bit machines followed by a 16 bit and finally by today's 32-bit microprocessors. There were numerous other improvements than increasing the word width (i.e. going from 16-bit to 32-bits). These included increasing frequencies at which data could be processed, increasing the functions through integration and increasing the number of available microprocessor instructions using microcode techniques. The large number of available microprocessor instructions came to characterize this class of microprocessors as Complex Instruction Set Computers (CISC).

Complex Instruction Sets have their roots in the demand for increased software productivity. Even before the advent of the microprocessor, software development had become the dominating cost component of computer ownership. Software developers, particularly end-user application developers, had to relate to the computer hardware in languages that were specific to a given computer hardware and somewhat remotely related to human, natural, languages. The software world responded by creating languages that were more easily used to convey the programmers intent to

the computer. The COBOL and FORTRAN languages were developed and became very popular.

The hardware industry including the microprocessor industry responded in its own way to the software developers productivity challenge. Computer hardware implementations strived for reducing the semantic gap between machine language and higher level languages such as FORTRAN and COBOL by moving the machine language closer to higher level languages. This philosophy laid the foundation for implementing microprocessors that used Complex Instruction Sets. This meant numerous instructions each performing complex functions. The then prevailing wisdom was that by reducing the semantic gap between machine and programming languages, software productivity would improve. In essence the hardware world had decided to bridge the gap between human and computer by moving the complexity into the hardware and relieving the compiler, which translates the higher level languages into microprocessor instructions, from handling the more difficult translation.

Complex Instruction sets would set the direction for innovations coming from suppliers and complementary innovators. These included semiconductor fabrication machinery (from suppliers), Computer Aided Design (CAD) tools (from suppliers), compiler technology (from complementary innovator), Operating Systems (from complementary innovator), complementary ICs (from complementary innovators) and development tools (from complementary innovator) that were geared towards microprocessors with large instruction sets.

The shortcoming of using a CISC approach was that the large number of relatively complicated instructions would require ever increasing die sizes and would also place a limit on how fast a given operation could be executed.

The die area and speed of operation problems were masked, however, by the rapid increases in semiconductor fabrication capability leading to smaller transistors and larger die sizes at reasonable yields.

RISC:

By the mid 1970s industry researchers and academics had begun to question the efficiency of the CISC approach. In 1975 at IBM's Thomas J Watson Research Centre a team of researchers began the development of the IBM 801. Although this Minicomputer was not implemented using VLSI ICs, it laid the foundations for many RISC concepts. The IBM researchers determined that compiler technology requires a set of simple instructions many more times than the more complicated instructions offered by the microprocessor. On this basis, a far smaller set of simple instructions were chosen for the 801 design.

The principles developed in the IBM 801 such as single cycle execution of instructions, reduced number of instructions and more sophisticated compiler that would bridge the semantic gap would later be largely adopted by RISC microprocessors.

RISC microprocessors would enhance the implementation efficiency of RISC notions as they are implemented using VLSI IC technology. RISC could be implemented in a smaller die than CISC processors reducing cost. It would execute faster because of its smaller size and complexity thereby increasing performance. Successive generations of RISC processors could be designed faster than same generation CISC processors providing RISC with a Time-To-Market (TTM) advantage. These factors would lead to large price/performance ratio advantages over same generation CISC processors.

By 1981, Patterson et al at Berkeley and Hennessy et al at Stanford had implemented the RISC concepts in single VLSI Integrated Circuits. The RISC1 processor at Berkeley and the MIPS processor at Stanford formed the foundation of the two most successful RISC Architectures in the industry today: The SUN Microsystem's SPARC RISC processor and the MIPS Corporation's family of Rxxxx (R2000, R3000, R4000) processors.

Despite pursuing the RISC principles first, IBM has only recently developed on a relatively successful VLSI RISC processor of its own, the RS6000, and plans on co-developing the Power series of RISC processors with Motorola and Apple Computer.

Digital Equipment Corporation, at first bought out the RISC technology by using the MIPS family of processors to power its DECStation products but has recently introduced the Alpha microprocessor which will be used for Digital's products ranging from Desktop to Data Center minis and mainframes.

Hewlett Packard has also moved into adopting RISC for its hardware platforms as well. The PA-RISC architecture is now available across the range of HP's hardware platforms.

In the next two sections the Innovation Hypercube is applied to RISC and CISC microporcessors. Figure 7 illustrates the various areas of knowledge required for implementing and using microprocessor based computer systems. This figure also labels the appropriate areas as being provide a by suppliers or complementary innovators to the firm developing the microprocessor.

The Hypercube model and CISC.

Figure 8 illustrates the hypercube applied to various generations of CISC processors from key innovators.

Innovating Entity:

Putting the CPU of a computer on a single chip in 1970 was a radical innovation. Design and implementation of subsequent generations in the same microprocessor family, Intel's 80xxx family for example, are for the most part incremental innovations.

Until Intel's 4-bit microprocessor, chipmakers had only designed components like decoders, Arithmetic Logic Units, Register files etc. out of discrete ICs. The microprocessors integrated these elements into a programmable device that provided a set of instructions as the medium of programming. Thus, the components as well as the linkages between core concepts that chipmakers used in designing microprocessors had changed. This was radical innovation.

Beginning with the 8-bit generation of microprocessors, CISC suppliers chose to address the investment preservation needs of their customers by offering "upward compatibility" in successive generations of microprocessors. This meant that software written for, say, an 8-bit generation of microprocessors would run on a 16-bit and a 32-bit generation of the same microprocessor. The customer's need to preserve their software investment would essentially lock them into a given family of microprocessors.

This upward compatibility forced microprocessor developers to preserve core concepts and linkages between them leading to incremental innovation in successive generations of microprocessors.

Suppliers:

For the first generation microprocessor, Semiconductor fabrication capability had been pushed to its limits to achieve the desired integration levels, new CAD tools had to be developed in order to allow the designers to deal with large complexity levels. These innovations were radical at the time.

Beyond the first generation of microprocessors, various innovations have occurred that may be classified as incremental or architectural. Fabrication capabilities, in general, have not seen radical innovation for successive generation of microprocessors. CAD tools have a similar innovation profile.

Complementary Innovators:

The first generation of 4-bit microprocessors demanded radical innovations from complementary innovators. Compilers and development systems/software had to be written to support the microprocessor's instruction set and a host of complementary ICs with the same word width had to be developed. Let us examine the complementary ICs in more detail as a means of illustrating the importance of complementary innovations.

For microprocessors to work effectively, they need complementary chips like memory chips, chips that allow the microprocessor to talk to printers, modems or keyboard, and chips that control disk drives. Thus development of any new generation of microprocessors must consider what the effect of the processor is going to be on these complementary devices and vice versa. The microprocessor developer must either cope with the lack of available complementary ICs or induce their creation in a timely fashion.

The case of Intel's 16-bit microprocessor best illustrates this. When Intel designed its 16-bit microprocessor, the 8086, it discovered that a lot of the

less expensive complementary chips in existence then were for the previous generation of microprocessors, i.e. 8-bit microprocessors. Intel had two choices. Wait until the complementary ICs catch up with it's 16-bit micro or innovate again. Intel chose the latter and designed the 8088 whose internal architecture was 16-bit, while those parts of the processor that were connected to complementary chips were 8-bit. This allowed the customers, i.e. system builders, to take advantage of those features 16-bit internal architecture provides while also using the inexpensive, more easily available 8-bit complementary chips. Intel was cognizant of volume manufacturing requirements of its end customers and assured that the end customer will have access to all complementary innovations required to put a system together.

When IBM decided to enter the Personal Computer (PC) market, and had to choose a microprocessor to power its PC, they reviewed various microprocessors. Although, Intel's 8088 product may not have been superior to other microprocessors, specifically the Motorola 68000, the 8088's 8-bit interface was crucial to system builders and gave them access to a supply of relatively inexpensive complementary ICs. IBM chose the Intel product.

Other Complementary innovations include compiler and operating system technology. The CISC dominated architectural philosophy had emphasized Complex Instruction Sets and relatively simple compiler technology. Operating Systems, the programs which manage the hardware resources of the computer and act as the interface to application software, were geared to the demands of specific CISC hardware. Microsoft DOS, for example, was geared for Intel's product whereas the Apple Macintosh Operating System was geared towards the Motorola microprocessors.

End user customer:

The end user interface is the Operating System and the application software. Up to the advent of DOS and UNIX, Operating systems were proprietary and were bundled with hardware from a single hardware manufacturers. These included Operating Systems from IBM, Digital Equipment Corporation, HP and others.

For all non-UNIX Operating Systems, once the end user customer committed to a given hardware/Operating System, they were locked into it for the most part to preserve the investment made in application software. After that any changes to the underlying hardware would not be very visible to the end user were it not for improved speed and support for more users. As such for any given hardware family (i.e. VAX or 80xxx or 68xxx) the end user would require only incremental innovation to support successive generations of microprocessors.

If the end-user customer chose to move to another hardware family, the end user would be forced to port all application software to the new operating system/hardware platform. This is a monumental task for different proprietary operating systems. Although DOS could run on PCs from IBM and clone makers, it too was tied to the Intel 80xxx family. As such porting DOS based applications to, say, Macintosh based applications would require significant effort.

UNIX was designed to address this problem. This operating system can be relatively easily ported to many hardware platforms (i.e. many processor families). As such an application developed on the UNIX Operating System running on Digital Equipment Corporation's hardware could be ported to IBM hardware running UNIX with relative ease.

The Hypercube Model and RISC

Figure 8 illustrates the hypercube applied to various RISC processors from key innovators.

Innovating Entity

RISC processors utilize the same building blocks as CISC processors. The way these blocks are put together, however, have changed in RISC processors. As such, the RISC processors are architectural innovations for the innovating entity.

RISC's salient architectural innovations included first and foremost a reduced number of instructions and the lack of a microcoded approach to decoding those instructions, a heavy use of pipelining that would allow the microprocessor to work on multiple instructions at the same time and a focus on executing each instruction in a single clock cycle or faster. Other innovations such as large windowed register files and cache memories on and off chip are important but were not fundamental to RISC architectures.

It is important to note that many RISC innovations such as pipelining have been adopted by CISC processors in the latest generation of their families. RISC concepts are relatively easily appropriated.

Complementary Innovators:

The first generation of complementary innovations would for the most part be radical in nature as wholesale changes would be required to available products used with CISC microprocessors.

Fundamental to RISC hardware simplicity is more sophisticated compiler technology that must translate sophisticated programming languages such as C into a reduced and simplified set of microprocessor instructions. The availability of optimizing compilers is a critical

complementary innovation. This innovation in Compiler technology was radical in nature.

Development Systems would also undergo radical innovation to support RISC's higher speeds of operation as well as RISC's pipelined nature. Non-pipelined CISC processors would execute one instruction at a time. The development system could easily determine which instruction was being processed when, say, an interrupt occurred. It would not be so easy to determine the same for RISC processors where up to 5 instructions were being processed at various stages in the pipeline.

With the advent of RISC, the gap between the microprocessor and complementary IC speeds widened. Complementary ICs were hard pressed to keep up with the processor speed. A solution in the memory subsystem area was off-processor cache technology. Cache memory subsystems would allow a decoupling of very fast processors from relatively slow main memory. Architectural Innovation in this area would promote the use of fast Static RAM (SRAM) technology as external cache (as opposed to slower DRAM for main memory) and various cache controllers that would manage the cache subsystem and its interface to the processor. Once again, this innovation was adopted by CISC processors and has become standard design practice for CPU subsystems.

Suppliers:

Suppliers of Silicon fabrication technology and CAD tools would not see much difference between the demands of CISC and RISC microprocessors. This does not mean that there have been no radical or architectural changes in these fields. There have been many. The point is that these innovations

have been independent of RISC and CISC architectures. The impact of these architectures has been to create incremental innovation in the supplier base.

End Users:

RISC users have in general adopted the UNIX Operating System. By doing so the advantages of a non-proprietary, very low cost, high performance and portable operating system were added to the superior price/performance of the RISC microprocessor.

To the end user, porting application software from non-UNIX Operating Systems to UNIX is a major task and will demand an overturning of core concepts as well as linkages between them.

The innovation required at the outset of adopting the RISC/UNIX combination is therefore radical. After the initial investment, however, the user can benefit from the advantages of hardware independence and application software portability. The promise of RISC/UNIX has outweighed the demands of making the necessary radical innovations. A new class of computer hardware, the workstation, has been developed around RISC/UNIX. Furthermore, most minicomputer and mainframe manufacturers have also adopted the RISC/UNIX combination over or in addition to their CISC/proprietary operating system platforms. RISC/UNIX, however, has not been able to supplant the 80xxx/DOS and in PCs and 68xxx/Macintosh Operating System in the Apple Computers.

SUPERCOMPUTERS:

Supercomputers are generally described as the most powerful computational systems available at any given time. This would mean that the first supercomputer dates back to Charles Babbage's mid 1800s "analytical engine". Our discussion here, however, starts with the Seymour Cray era. Most of today's installed base of supercomputers can be attributed to Seymour Cray who, in 1975, left Control Data Corporation (CDC) where he had designed supercomputers, to start his own supercomputer company, Cray Research Inc.. At CDC, Cray had designed the CDC 7600 supercomputer, a so-called scalar supercomputer because it had a scalar processor (the "engine" or brain of the computer). Scalar processors have to issue an instruction for every single operation (e.g addition of two numbers) so that even vector data would have to be broken down and an instruction issued for operation on each element of the vector. The 7600 was also of the traditional Von Neumann architecture[1]. In 1976, Cray Research shipped its first supercomputer, the Cray-1, the first commercially available vector supercomputer. Vector computers, for the most part, need only one instruction to execute each operation on vectors, and this greatly improves processing time (for applications that lend themselves to lists) compared to scalar processors. Vector processing was a key innovation in supercomputers especially since a lot of data on which supercomputers operate are either vector-like or could be vectorized. The first vector supercomputer was actually the CDC Star-100 but was not

[1]The architecture used in most of today's computers is often attributed to John Von Neumann's mid-1940s architecture. In that architecture, the Central Processing Unit (CPU) of the computer fetches an instruction (data) from a central store (main memory), operates on it (for example, add or subtract), and returns the results into the main memory. Only one CPU is used, and that one CPU can do only one thing at a time.

commercially available until after the Cray-1 when it was released as the Cyber 205.

One thing which the Cray-1, CDC7600, Cyber 205, and previous supercomputers (vector or scalar) had in common was that they each had only one processor that could be put on any one processing job at any one time. Cray Research changed all that in 1982 when it introduced its multiprocessor Cray X-MP, the first commercially successful supercomputer to apply more than one processor to the same problem at any one time (The ILLIAC IV was the first parallel supercomputer). In the years that followed, Cray Research introduced many other multiprocessor supecomputers with the Cray Y-MP C90 its latest with 16 processors in 1991 that delivers 16 GFLOPS (gigaFLOPS = billion floating point operations per second) compared to the Cray-1's XX MFLOPS (million floating point operations per second). In 1992, NEC introduced its 4-processor SX3 that gives 25 GFLOPS. Table 10 lists some of the key supercomputers that have been introduced over the years.

Most of the gains in supercomputer performance have come as a result of innovations in semiconductor technology, from the transistor to very large scale integrated circuits. NEC's four-processor supercomputer, for example, was able to deliver the 25 GFLOP primarily because of its advanced ECL (emitter-coupled Logic) semiconductor technology and premier packaging techniques.

A key goal of these traditional Cray supercomputer designs that use few (1-16) processors is to make each processor as fast as possible. But despite all the dramatic improvements in microchip and packaging technology, these kinds of supercomputer designs are reaching a physical limit -- the speed of light. Computer signals travel through the computer's

electrical circuitry at the speed of light. And no matter how much these computers with 1-16 processors speed up each processor, they would never attain some of the speeds that many compute-intensive jobs need [for example, supercomputers still cannot synthesize a protein from its gene] because of the physical limit imposed by the speed of light. This is where massively parallel computers (MPC) come in.

In massively parallel computers, hundreds or thousands of processors are put on one job, with each processor simultaneously tackling an assigned stage of the job to get the whole job done faster than one processor operating serially – the structure of the job permitting. So rather than trying to speed up one or a few processors to do the job, massively parallel computers (MPCs) put very many processors on the job to perform it in parallel. Now, the speed of light is no longer the physical limit, and execution of inherently parallel jobs can be speeded up considerably. Thinking Machines' CM5 uses hundreds of 32-bit SPARC CMOS (Complementary Metal Oxide Semiconductor) microprocessors and runs at 128 GFLOPS peak. The physical limit to the speed of MPCs will eventually be the ability of the processors to communicate with each other.

MPCs use readily available CMOS (a proven technology) chips that consume less power than the ECL (Emitter Couple Logic) chips used in conventional (Cray-like) supercomputers, and these CMOS chips don't have to be as fast as ECL chips since it is not the speed of each one that matters (at this stage of the technology) in MPCs but their combination. And because they consume less power, they are air-cooled and don't need the elaborate liquid cooling systems of ECL-based systems.

MPCs can be divided into two groups: multiprocessors and multicomputers. Multiprocessor MPCs like Kendal Square Research's KSR

1 have numerous processors that share one main memory. The KSR 1 has 1088 64-bit microprocessors that share one memory. Multicomputer MPCs are interconnected microprocessors, each with its own memory, that communicate via message passing. Examples are Thinking Machines CM5, Intel's Paragon, and supercomputer MPCs from Ncube, Ametek, and Transputer.

Most of the manufacturers of traditional supercomputers (those with 16 or fewer very fast processors, and elaborate cooling systems) like Cray Research Inc., IBM, etc. have either already started MPC programs or announced that they will do so. But their quest to improve traditional supercomputers has not stopped. When, in 1989, Seymour Cray left Cray Research to start Cray Computer, his answer to getting a faster supercomputer was to use Gallium Arsenide chips which are two and half times as fast as conventional silicon chips and also consume a lot less power. Gallium Arsenide is a relatively new technology that is still in its infancy compared to the silicon semiconductor technology that now provides chips for computers. The introduction of the Cray-3 has been delayed primarily because of the difficulties in getting GaAs chips to work. Supercomputer Systems Inc. (SSI), another supercomputer start-up is also having difficulties delivering its first supercomputer because it was banking on GaAs chips. Steve Chen, the founder of SSI was a supercomputer designer at Cray Research and SSI is backed by IBM.

Another viable set of computers are the so-called minisupercomputers. They utilize the same vector processing of traditional supercomputers, but with some important differences: They are cheaper, provide 25 to 35 percent the performance of traditional supercomputers [Kelley 1988] , offer lower price for the performance provided and lend themselves to those low-end

applications that don't need the higher performance of higher power supercomputers, let alone their prices. They use proven CMOS (Complementary Metal Oxide Semiconductor) chips that are less expensive and consume less power than the power-demanding but faster ECL chips used in traditional designs. This results in cheaper systems that are air-cooled.

The Hypercube Model and Supercomputers

In this section we use the hypercube of innovation model to examine the supercomputer industry that we have just described. The examples we use are only the tip of the iceberg of the number of innovations in supercomputers. This is not a comprehensive treatment of innovations in supercomputers. We do not attempt to look at innovations in microchip technology although they have been responsible for most of the speed improvements in supercomputers.

We will look first at the supecomputers industry as a whole and then Cray Research in particular since the latter has played such a critical role in supercomputers.

Table 12 lists key supercomputer innovations, while their impact on the capabilities and assets of their innovators, suppliers, customers, and complementary innovators is shown in Table 13 and figure 9. Using the Henderson and Clark categorization criteria, vector processing was an architectural innovation to CDC (Star-100) and Cray Research (Cray-1) when they designed these systems. The main components of the supercomputer--memory, CPU, I/O -- and core design concepts had not changed radically; the key change was the provision of vector processing. But the linkages between these components and core concepts were being altered. To many customers and suppliers of applications software, however, this was a radical innovation

because they had to learn how to program with vector processors. Luckily, for the Cray, most users of supercomputers then were scientists who wrote their own software and were more interested in a number-crunching engine than a complete data processing solution. More importantly, a complementary innovation, the vectorizer was developed that could convert some of the old software written for scalar machines to forms in which vector machines could crunch. So the impact of the radicalness of the innovation on customers was not that important. Cray Research's multiprocessor, Cray X-MP, can also be considered an architectural innovation for reasons similar to those just listed above. At customers, its impact was more incremental than radical for several reasons: It still used the same Cray Operating. System (COS),

The Cray-3 and SSI's machine are examples of machines that are radical to suppliers and facing problems because of it. Both machines are multiprocessor but with no more than 16 processors and not radically different from previous designs. They are, however, depending on GaAs chips to make major contributions to speed gains. But GaAs technology is still in its infancy compared to the proven silicon technology that other computers use and is thus a radical innovation to any computer. Cray Computer's solution to reducing this uncertainty was to acquire Gigabit Logic, a GaAs chip manufacturer. That has still not worked. While the problems with the Cray-3 and SSI's machine may not be entirely due to GaAs chips, it is true that GaAs, a radical innovation to most suppliers of chips, has contributed to the problems of the two machines.

Massively parallel computers (MPCs) are a radical innovation for all members of the innovation value-added chain except suppliers. Their design is conceptually very different from that of traditional supercomputer designs. Writing software for them is even trickier. Users of the huge installed base of traditional Cray-type designs would prefer machines that allow them to keep some of the skills and knowledge acquired with the Cray-like machines, and especially the any applications programs that they may have written. Their operating systems are also different. Applications, as well as systems programmers for the new machines are also not easy to find. Hardcore supercomputer users (scientists and academics) can write their own software. But for MPCs to diffuse into the commercial applications that will greatly increase their diffusion, they need lots of software.

Cray Research:

As of 1990, more thar. 80% of the installed supercomputers in the United States were Cray Research's. So it is only appropriate that we look at the product innovations that came from Cray. These innovations are listed in Table 12 and their hypercube map given in Figure 8.

Section IV: Summary and Implications of the Hypercube.

Using several examples, we have shown that an innovating entity that only looks at the impact of its innovation on its competence and existing products, and does not critically examine the impact of that innovation on the competence and capabilities of its suppliers, customers, competitors and complementary innovators, may be making a mistake. We anecdotally used

the cases of 1) Dvorak's DSK keyboard that failed to diffuse because although it was an architectural innovation to Dvorak, it was a radical innovation to its customers. 2) the case of OS-2 which is a radical innovation to IBM but a incremental innovation to DOS users vs. Microsoft Windows which is an incremental innovation to Microsoft and an incremental innovation to users of DOS, allowing Microsoft to enter the market early. So far, Microsoft Windows is winning. 3) the case of Lotus that didn't pay attention to the momentum of Windows software and lost some ground in its spreadsheet market share to Microsoft's Excel, and 4) the case of the electric car which is a radical innovation to the innovating firms, suppliers of components and complementary products, but an incremental innovation to users.

The model forces innovation managers to look at their innovations not only in terms of what the impact of the innovation will be on the innovating entity's capabilities and assets, but also those of suppliers, customers, and complementary innovators. We then suggested that innovators should think twice about innovations that destroy skills, knowledge and positive network externalities at any of the stages of the value-added chain, especially at customers. They should avoid the Red Zone (of the mapping of innovations along the innovation value-added chain) and go with innovations that reinforce key concept and linkages all along the value-added chain (innovations that fall in the green or yellow zone).

We also noted some exceptions to avoiding the the Red Zone. Specifically, it should be avoided unless a subset of the following is true: 1) The price/performance ratio of the innovation, as viewed by the customer, outweighs any losses incurred as a result of competence or positive network externality destruction. This happens for example when the physical limit of an older technological trajectory has been reached and the only way to

overcome this physical limitation is to move to a new technological trajectory – a move that often means destruction of competence acquired during the evolution along the older trajectory but great improvement in some key parameter. 2) New markets where customers have not yet had time to build any innovation-specific skills and knowledge, and competence destruction is not an issue. 3) Complementary innovations, that allow customers (or other members of the innovation value added chain) to keep their competence and positive network externalities, exist. 4) When institutional requirements mandate the innovation.

We then turned to the detailed analysis of RISC and CISC chips, and supercomputers using the model. In particular, we analyzed the impact of key innovations in CISC (Complex Instruction Set Computers) chips, RISC (Reduced Instruction Set Computers) chips, and supercomputers on the capabilities of suppliers, customers, and complementary innovators. In CISC, we suggested that Intel's foresight in designing the 8088 microprocessor in response to the inertia of complementary 8-bit chips, may have contributed to its being chosen by IBM over competitors to provide the microprocessor architecture for the now very popular IBM PC and PC compatibles. We also found that although RISC is an architectural innovation as far as chipmakers like Motorola and Intel are concerned, it is a radical innovation to OEM customers who have been using CISC chips to design personal computers and sell to end users. This is because with RISC, these OEMs have to learn new assembly languages, establish new development systems, retrain their engineers on how design systems with the RISC chips. To personal computer end-users who have learned to use DOS, acquired or written their own applications programs, and established positive network externalities on CISC-based machines, RISC is a radical innovation since in its present form, it

destroys the competence, capabilities and positive network externalities of these customers. The promise of speed alone is not enough to dislodge CISC in this particular market. We also suggested that it is the realization of the inertia of CISC vis-a-vis RISC that made firms like Compaq pull out of ACE consortium. All that could change if a complementary innovation (e.g software) could be developed that allows all DOS users to preserve their skills and old applications software when they use RISC machines. Microsoft NT is intended to be this innovation.

In newer markets like Workstations where the capabilities, competencies, and positive network externalities have not been well-established yet, RISC is doing very well. In the embedded control market where speed is critical and the end-user is not locked into CISC as in the PC market, RISC is also doing well.

In the Minicomputer and Mainframe markets the price/performance advantages of RISC have been sufficiently compelling that manufacturers and customers of this class of computers have adopted RISC/UNIX technology instead of the mostly CISC/proprietary operating system solutions of the past.

In supercomputers, we saw how Cray Computer Corp. and SSI are having difficulties introducing their new gallium Arsenide (GaAs) chip-based supercomputers partly because GaAs is a radical innovation to chip suppliers relative to the mature silicon technology.

Earlier versions of supercomputer innovations that were radical innovations at customers didn't have the disastrous consequences predicted by the hypercube model because many of those early users were scientists and academics who wrote their own programs, and could trade the program writing for a more powerful computing engine.

Massively parallel computers, despite being faster than the traditional Cray-like supercomputers may not be diffusing as fast as one would expect because they are a radical innovation not only to the innovating entities but also to customers and suppliers of complementary innovations like software. It is, however, an incremental innovation for suppliers of hardware components like microchips and disk drives.

The real breakthrough in supercomputer diffusion will come when the machines penetrate the commercial businesses that could use their compute power. This will come only if the software is there.

Implications:

An innovating entity must pay attention not only to the impact of the innovation on its organizational competence and products, but also to what the impact will be on the competence and products of its suppliers, complementary innovators, and customers.

The innovating firm must study the impact of its innovation on the capabilities and assets of its customers. In particular it must look at what the innovation will do to the skills and knowledge acquired by customers with previous products, to add-on products, and to any positive network externalities. In short, the innovator firm should reconsider innovations that are competence- or positive network externalities-destroying for customers. For some products, the radicalness of the product at customers can be reduced if 1) the innovation is functionally compatible with previous products, 2) a complementary innovation comes along that makes customers utilize their old skills on the new product.

The innovator must watch out for the inertia of old complementary innovations and the momentum of new ones.

Suppliers of critical components can also be a big gating factor to an innovation, and can't be neglected.

Conclusion;

The common practice of classifying innovations only according to the impact of the innovation on the innovating entity's capabilities vis-a-vis its existing technology and markets is not adequate for high technology products that require critical input components and equipment from suppliers, depend on complementary innovations for success, require high levels of learning by customers before use, and that lend themselves to positive network externalities. For such products, the impact of the innovation on the capabilities and assets of suppliers, customers and complementary innovators is just as critical as that on the innovating entity's competence and assets.

Using anecdotal examples we built the hypercube model that forces managers at the innovating entity to evaluate their innovations in terms of the impact of those innovations on the competence and assets of all the members of the innovation value-added chain. We then used it to examine the CISC, RISC and supercomputer industries. The innovator should pursue innovations that reinforce core concepts and competence along the innovation value-added chain, while avoiding those that destroy them. We noted some key exceptions to the latter. We also suggested that the innovator watch out for the inertia of older complementary innovations and the momentum of newer ones, and take advantage of them.

References:

Abernathy, W. J and Clark, K. (1986) : Mapping the Winds of change. Research Policy.

Abernathy, W. J., and J. M Utterback,: Patterns of innovation in technology. *Technology Review* 80:7(June-July):40-47. 1978.

Afuah, A. N. and J. M Utterback,: The Emergence of a New Supercomputer Architecture. Technology Forecasting and Social Change 40, 315-328 (1991) Cray Research Inc. company financial statements and other company documents.

David, P. A: (1985)Clio and the Economics of QWERTY. Economic History. vol 75 No. 2

Els, Sharon A., "Configuration and Competitiveness in the Supercomputer Industry." Master's Thesis at MIT. 1989

Fernbach, Sidney "Supercomputers--Past, Present, Prospects." *Future Generation Computer Systems*, July, 1984.

Foster, Foster. "*Innovation : The attacker's Advantage.*" 1986

Freeman, Christopher, "The Economics of Industrial Innovation." MIT Press. Cambridge 1986. Second Edition.

Henderson, Rebecca and CLark, K (1990). Architectural Innovation: "The Reconfiguration of existing product technologies and the failure of established firms". Administrative Sciences Quarterly, 35, 9-30

Pratt, G. A (1992): Electric Cars: On the Road again. Technology Review, August/September 1992.

Schumpeter, J. A, *"The Creative Response in Economic History"*. Journal of Economic History, 7 (1947), 149-159. Reprinted in *Essays of J. A. Schumpeter*, ed. by R. V. Clemence, Cambridge, MA.: Addison-Wesley, 1951.

Teece, David J., "Profiting from Technological Innovation: Implications for Integration, Collaboration, Licensing and Public Policy." *The Competitive Challenge.*, edited by David Teece. Ballinger. Cambridge 1987.

Tushman, Michael L., and Anderson, Phillip, "Technological Discontinuities and Organizational Environments." *Administrative Science quarterly*, 31 (1986): 439-465

Utterback, James M. "Innovation and Industrial Evolution in Manufacturing Industries." in *Technology and Global Industry: Companies and Nations in World Economy* edited by Bruce R. Guile and Harvey Brooks. National Academy Press. Washington D. C. 1987. To be elaborated on in *The Dynamics of Innovation in Industry*: Chapter 5. Harvard Business School Press, 1991

Utterback, James M., and Kim, Linsu "Invasion of a Stable Business by Radical innovation." *The Management of Productivity and Technology in Manufacturing*, by Kleindorfer, P R. Plenum Press 1986.

von Hippel, Eric (1988): The Sources of Innovation, New York. Oxford University Press.

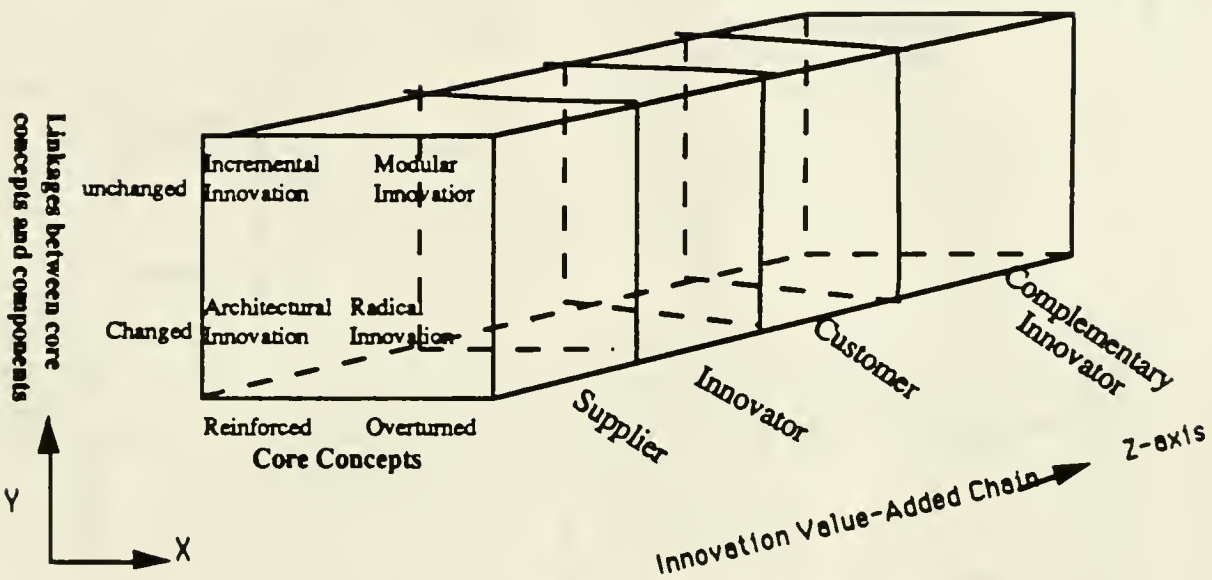


Figure 1: The Hypercube of Innovation. The X and Y axes are the innovation-classifying factors. The Z-axis is the innovation value-adding chain of supplier of key components, innovator, Customer and supplier of complementary innovators

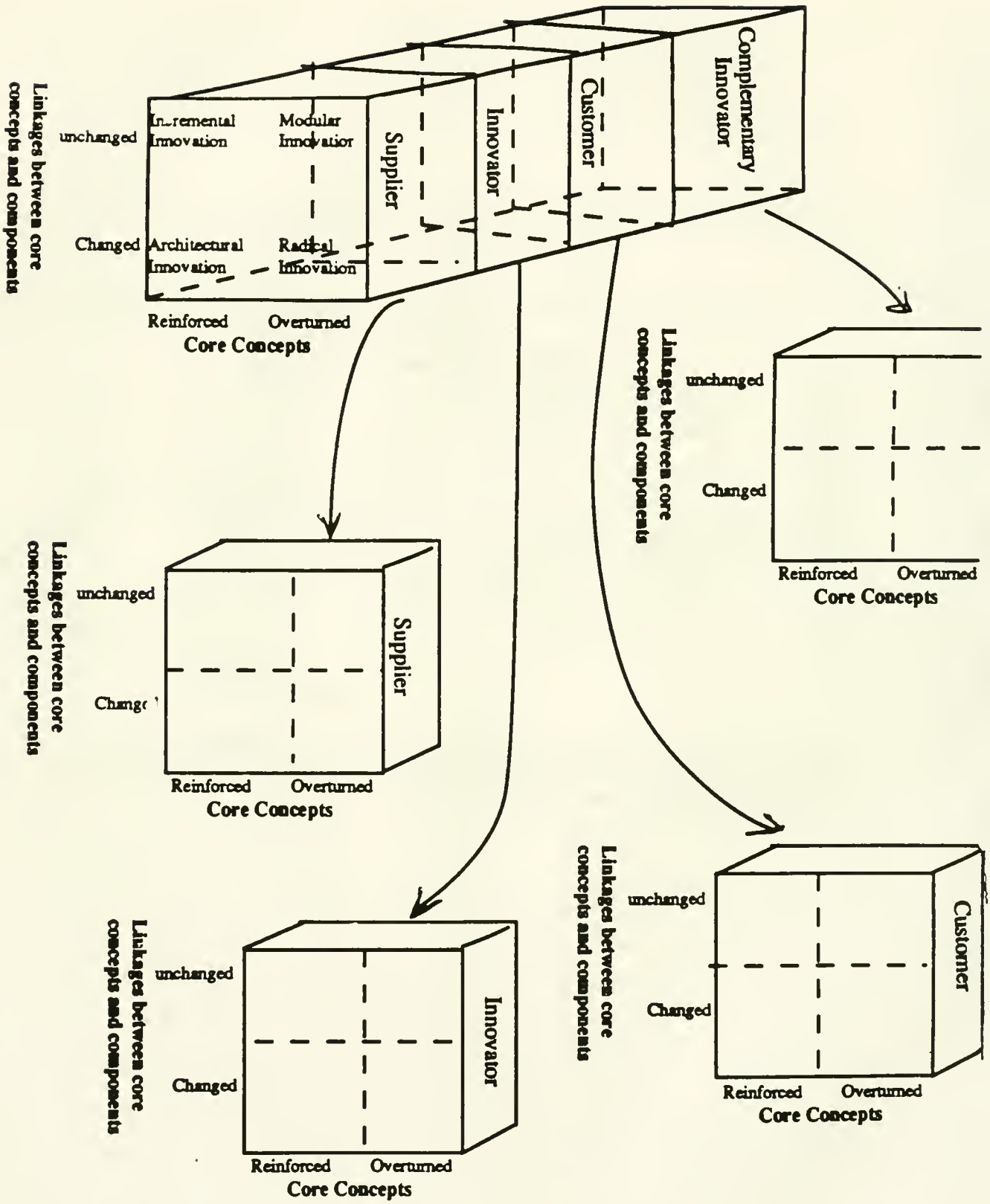


Figure 2: The hypercube of Innovation exploded to show the various faces that an innov can assume along the innovation value-added chain.

FIGURE 3): The GREEN-RED zone map

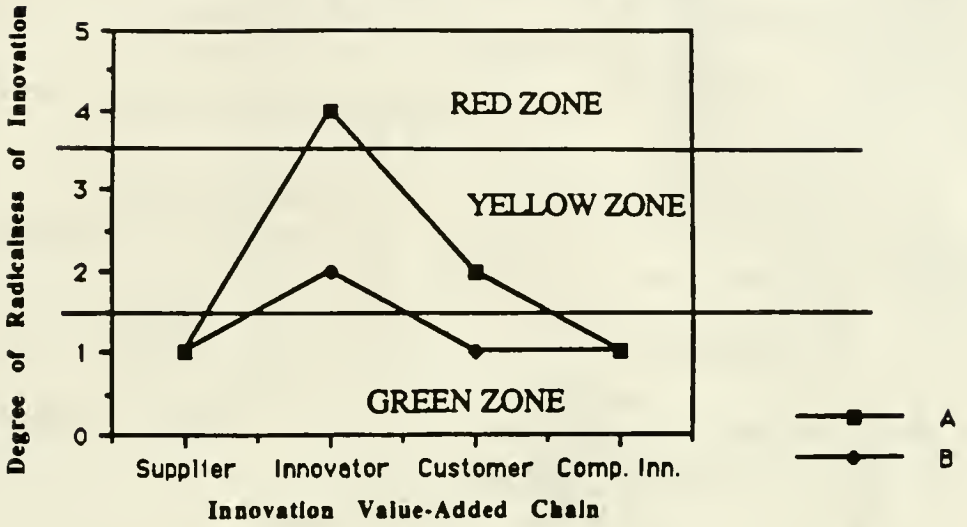


FIGURE 4a): The GREEN-RED zone map for the DSK keyboard.

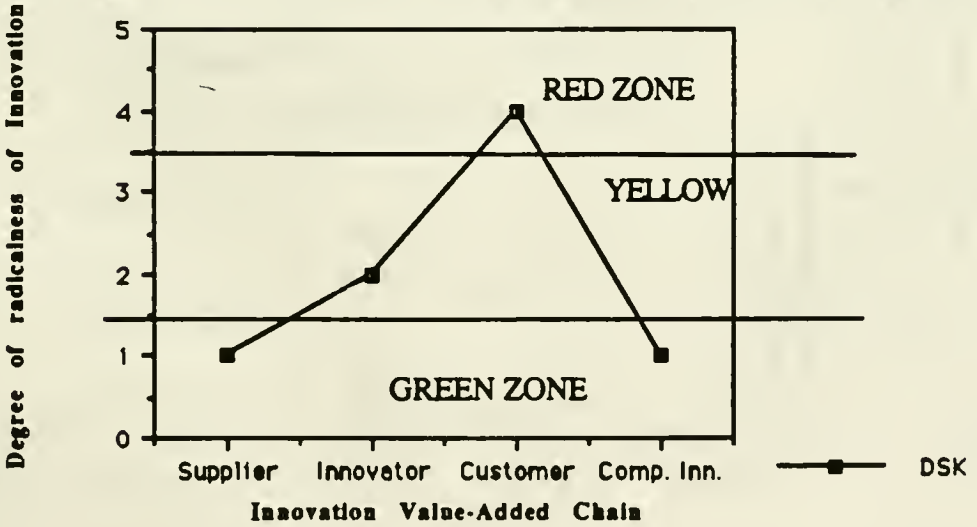


FIGURE 4b): The GREEN-RED zone map for the Electric Car.

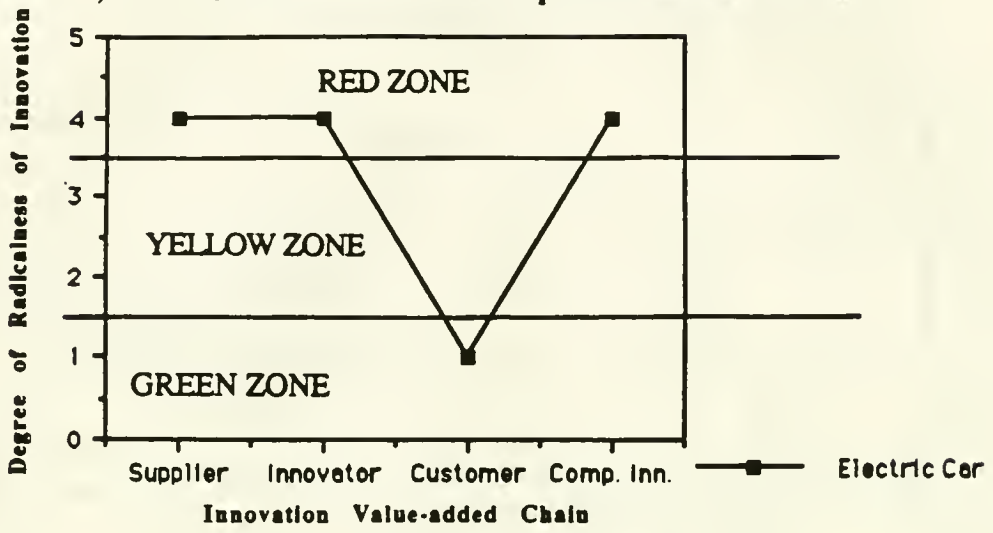
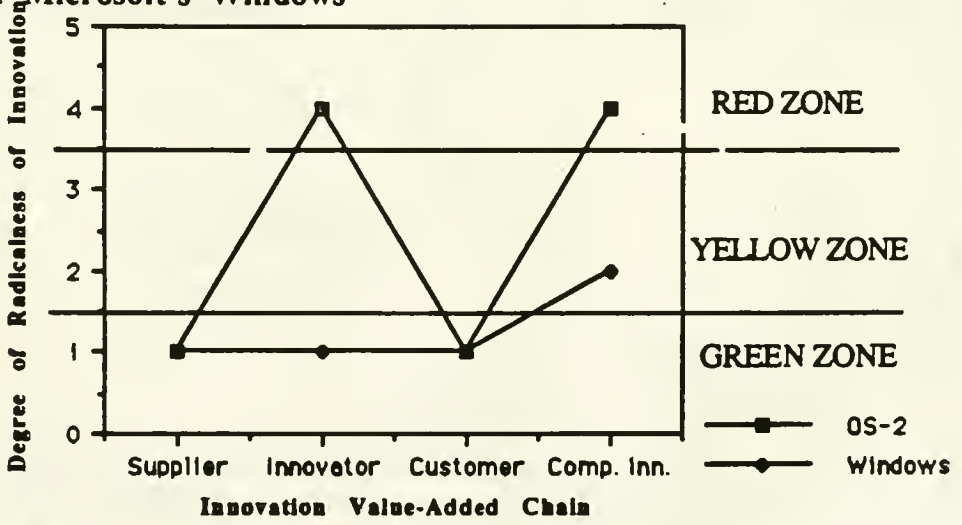


FIGURE 5): The GREEN-RED zone map for IBM's OS-2 and Microsoft's Windows



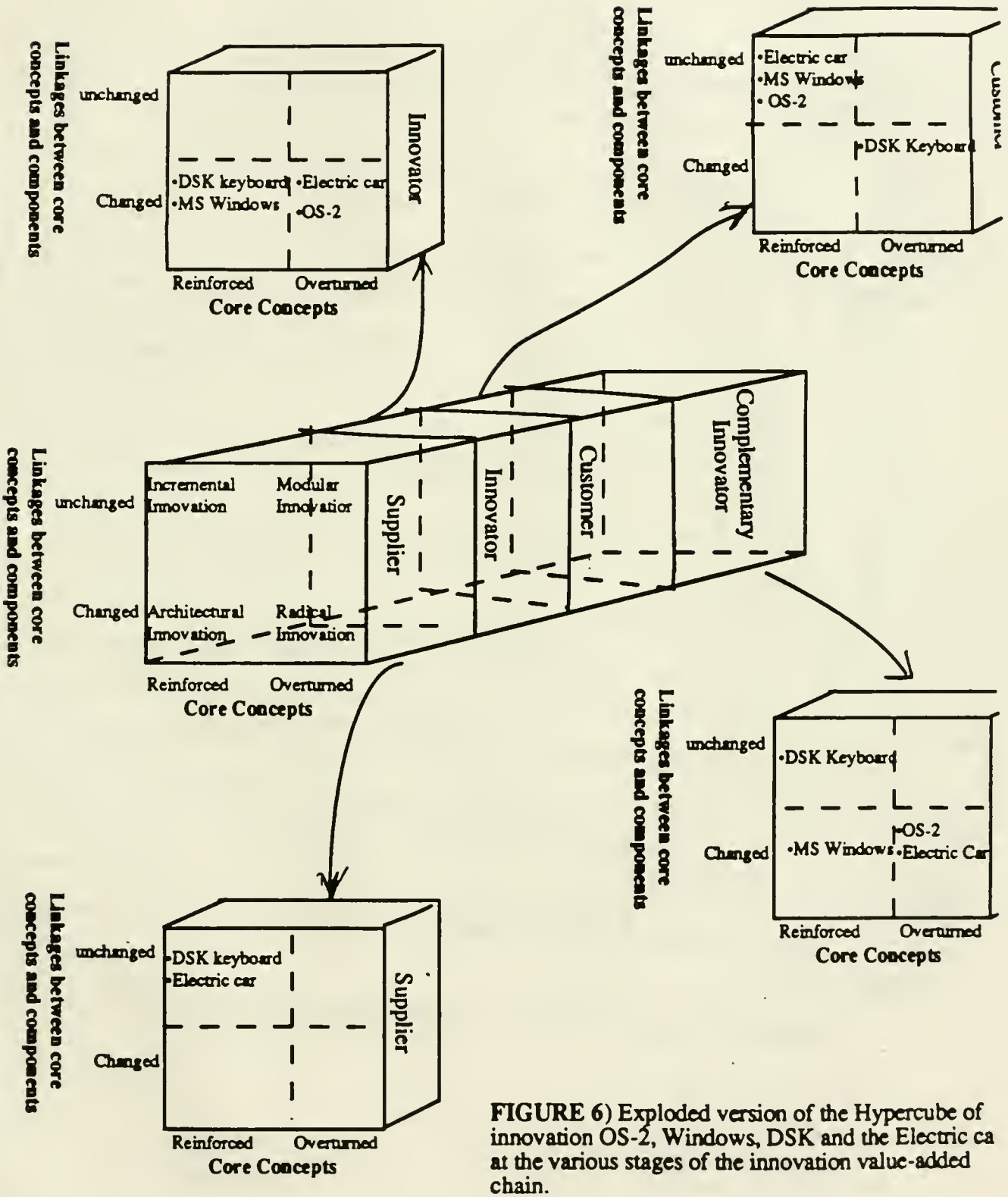


FIGURE 6) Exploded version of the Hypercube of innovation OS-2, Windows, DSK and the Electric ca at the various stages of the innovation value-added chain.

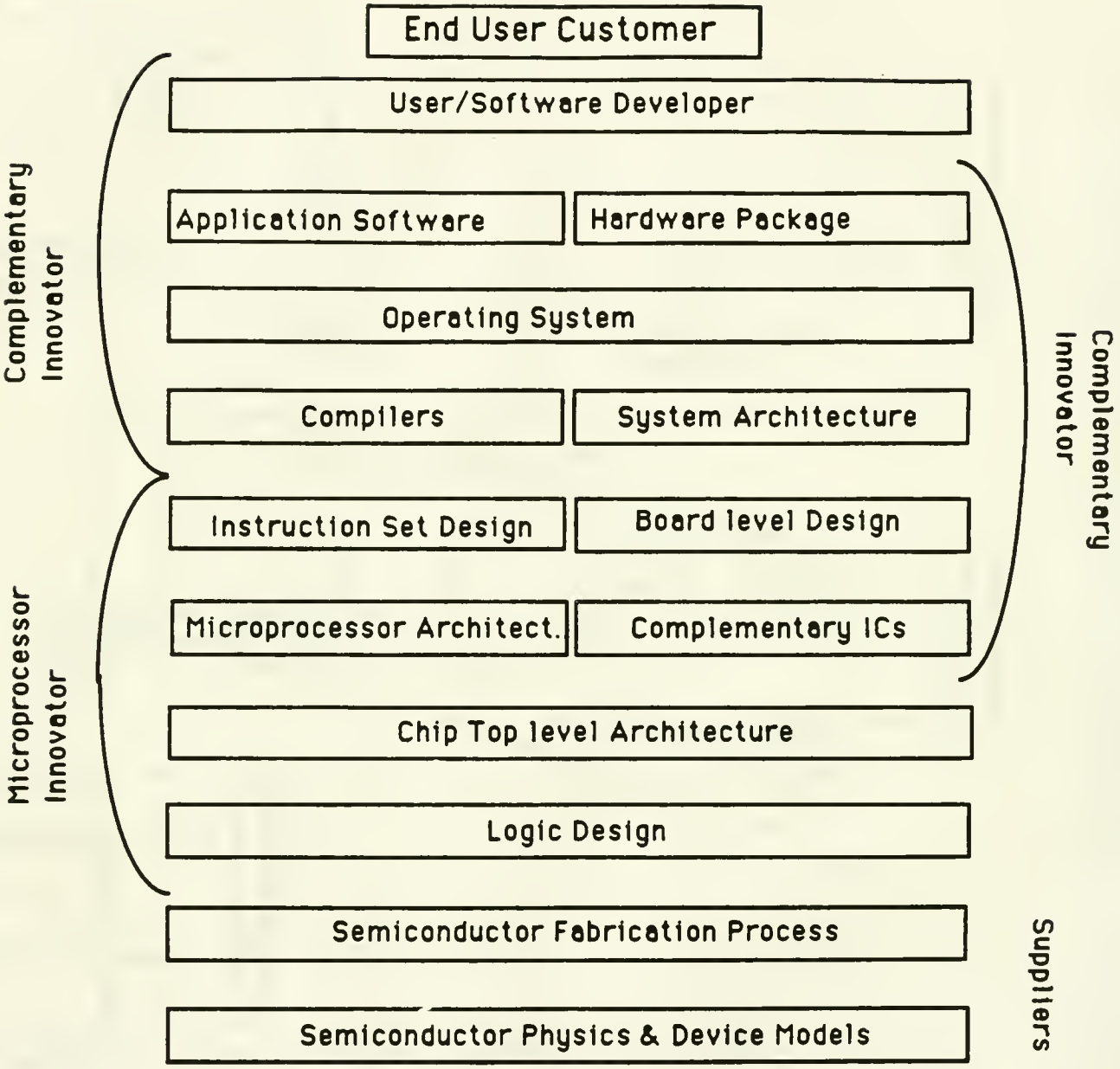


Figure 7. Computer System knowledge Areas and their providers .

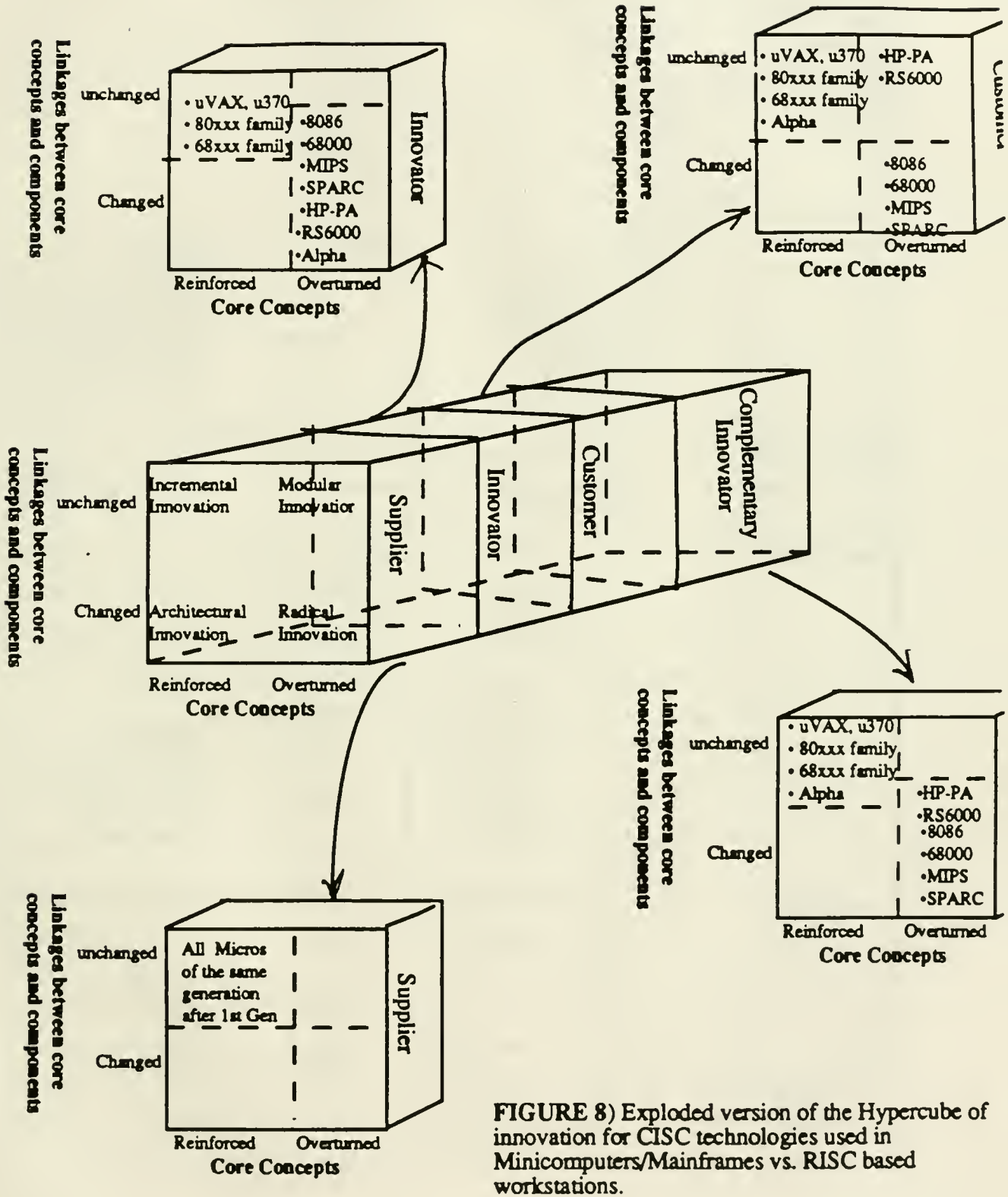


FIGURE 8) Exploded version of the Hypercube of innovation for CISC technologies used in Minicomputers/Mainframes vs. RISC based workstations.

Table 2: The Hypercube in tabular form -- CISC & RISC innovations

INNOVATOR		CUSTOMER	
Incremental	***Modular***	***Incremental***	***Modular***
• uVAX, u370 (CISC) • 80xxx family(CISC) • 68xxx family(CISC)		• uVAX,u370 (CISC) • 80xxx family (CISC) • 68xxx family (CISC) • Alpha	
Architectural	***Radical***	***Architectural***	***Radical***
	• 8086(CISC) • 68000(CISC) • MIPS R2000(RISC) • SPARC1(RISC) • HP-PA (RISC) • RS6000(RISC) • ALPHA(RISC)		• 8086(CISC) • 68000(CISC) •MIPSR2000(RISC) • SPARC1(RISC) • HP-PA (RISC) • RS6000(RISC)
SUPPLIER		**COMP. INNOVATOR**	
Increemental	***Modular***	***incremental***	***Modular***
All micros of the same generation		• uVAX,u370 (CISC) • 80xxx family (CISC) • 68xxx family (CISC) • Alpha	
Architectural	***Radical***	***Architectural***	***Radical***
			• 8086(CISC) • 68000(CISC) •MIPSR2000(RISC) • SPARC1(RISC) • HP-PA (RISC) • RS6000(RISC)

Table 10: Some supercomputer product innovations.

Machine	Year	Mfr	Bits	CPU	Technology	O.S
IBM700	1954	IBM	36	Sequential	Vacuum T	
IBM7000	1959	IBM	36	Sequential	Transistor	
CDC6600	1965	CDC	60	Seq. Functional units Scalar		
CDC7600	1969	CDC	60	Scalar	Transistor	
Star-100	1972	CDC	64	Vector Proce	ICs	
ILLIAC IV	1972	Burroughs	64	64 Process	ICs	
Cray-1	1976	CDC	64	Vector/Scalar	LSI	
CDC 205	1976	CDC	64	Vector	LSI	
Univac LARC	1960	Univac				
IBM7030	1960	IBM				
IBM360/195	1971	IBM				
TIASC	1974	TI	32	Vector	LSI	
Denelcor HEP-1	1977					
Cyber205	1981	CDC	64	Vector	LSI	
Hitachi S810 20	1983			Vector/Scalar	LSI	
Fujitsu VP				Vector		
ETA-30						
Cray X-MP	1982	Cray	64	Vector. Ist Multicprocessor		
Denelcor HEP-1		Denelcor		Multiprocessor		
ETA-10		ETA Systems		Vector. 8 processors		
Cray-2	1985	Cray	64	4 CPUs		
Cray Y-MP	1988	Cray		8-16 CPUs Vector		UNICOS, COS, CTSS
Cray C-90	1991	Cray		Vector. 16 processors	ECL	UNICOS, COS, CTSS
Cray-3					GaAs	UNICOS, COS,
IBM3090/600S VF	1988	IBM		Vector. 1-6		MVS, AIX, VM/CMS
Fujitsu VP-2600/20	1991	Fujitsu		Vector		Propr. OS. UTS/M
Fujitsu FACon VP-200	1984	Fujitsu			ECL	
NEC SX-2	1985	NEC			ECL	
Hitachi S820/80	1988	Hitachi		Vector	ECL	Proprietary OS, HIUX
NEC SX-3	1992			Multiprocessor. Vector		UNIX

Table 11: Key innovations in supercomputers

Innovation	Machine	Year	Firm
Vector Processing	Star-100	1973[2]	CDC
Vector procssing	Cray-1	1976	Cray Resea
Multiprocessing (traditional)	Cray X-MP	1982	Cray Research
MPC SIMD	CM-2	1986	Thinking Machines
MPC multiprocessor	KR 1	1992	Kendal Square Research
MPC multicomputer	Intel Paragon	198x	Intel Corp
Minisupercomputers	Convex-2	198x	Convex Computers
	TIASC	1974	TI
	Denelcor HEP-1	1977	Denelcor
	Cyber205	1981	CDC
	Hitachi S810 20	1983	Hitachi
	Fujitsu VP		Fujitsu
	Cray-2	1985	Cray
	Cray Y-MP	1988	Cray
	Cray C-90	1991	Cray
	Cray-3		
	IBM3090/600S VF	1988	IBM
	Fujitsu VP-2600/20	1991	Fujitsu
	Fujitsu FACon VP-200	1984	Fujitsu
	NEC SX-2	1985	NEC
	Hitachi S820/80	1988	Hitachi
	NEC SX-3	1992	NEC

Table 12: Classifications of some of Cray Research's Innovations:

Year	Product	Innovation	Operating System	Microchip Technology	Cray Research
1976	Cray-1	Vector Processing	COS (Cray Op System)	ECL	radical Innovation
1979	Cray-1/S		COS		
1982	Cray-1/M		COS	MOS memory	
1982	Cray X-MP	Multiprocessing	COS		
1985	Cray-2	4-CPUs	UNICOS (instruction set different)		
1988	Cray Y-MP	8-CPUs			
1990	Cray Y-MP 2E	air/water-cooled			
1991	Cray Y-MP 8E				
1991	Cray Y-MP		UNICOS		
199x	Cray-3				

[2] designed in 1969 but became operational in 1973. Never shipped. Features maintained in the Cyber 205. (1982)

Table 13: The Hypercube in tabular form -- Supercomputer innovations

INNOVATOR		CUSTOMER	
Incremental	***Modular***	***Incremental***	***Modular***
• Cray Y-MP 2E • Cray Y-MP 8E		• Cray X-MP • Cray-2 • Convex-2 • Cray Y-MP C90	
Architectural	***Radical***	***Architectural***	***Radical***
• Cray-1 • Star-100 • Cray X-MP • Cray-2 • Convex-2 • Cray Y-MP C90 • Cray-3 • SSI	• Illiac IV • CM-2 • Paragon • KR1 • CM-5		• Illiac IV • Cray-1 • Star-100 • CM-2 • CM-5 • Paragon • KR 1
SUPPLIER		**COMP. INNOVATOR**	
Incremental	***Modular***	***Incremental***	***Modular***
• Illiac IV • Cray-1 • Star-100 • Cray X-MP • Cray-2 • Convex-2 • Cray Y-MP C90 • CM-2 • CM-5 • Paragon • KR 1		• Cray X-MP • Cray-2 • Convex-2 • Cray Y-MP C90	
Architectural	***Radical***	***Architectural***	***Radical***
	• Cray-1 • SSI		• Illiac IV • Cray-1 • Star-100 • CM-2 • CM-5 • Paragon • KR 1

Table 14: The Hypercube in tabular form – Cray Research Inc.

INNOVATOR		CUSTOMER	
Incremental	***Modular***	***Incremental***	***Modular***
• Cray Y-MP 2E • Cray Y-MP 8E		• Cray X-MP • Cray-2 • Cray Y-MP C90 • Cray XMS	
Architectural	***Radical***	***Architectural***	***Radical***
• Cray-1 • Cray X-MP • Cray-2 • Cray Y-MP C90 • Cray XMS			• Cray-1
SUPPLIER		**COMP. INNOVATOR**	
Incremental	***Modular***	***Incremental***	***Modular***
• Cray-1 • Cray X-MP • Cray-2 • Cray Y-MP C90 • Cray XMS		• Cray X-MP • Cray-2 • Cray Y-MP C90 • Cray XMS	
Architectural	***Radical***	***Architectural***	***Radical***
	• Cray-1		• Cray-1

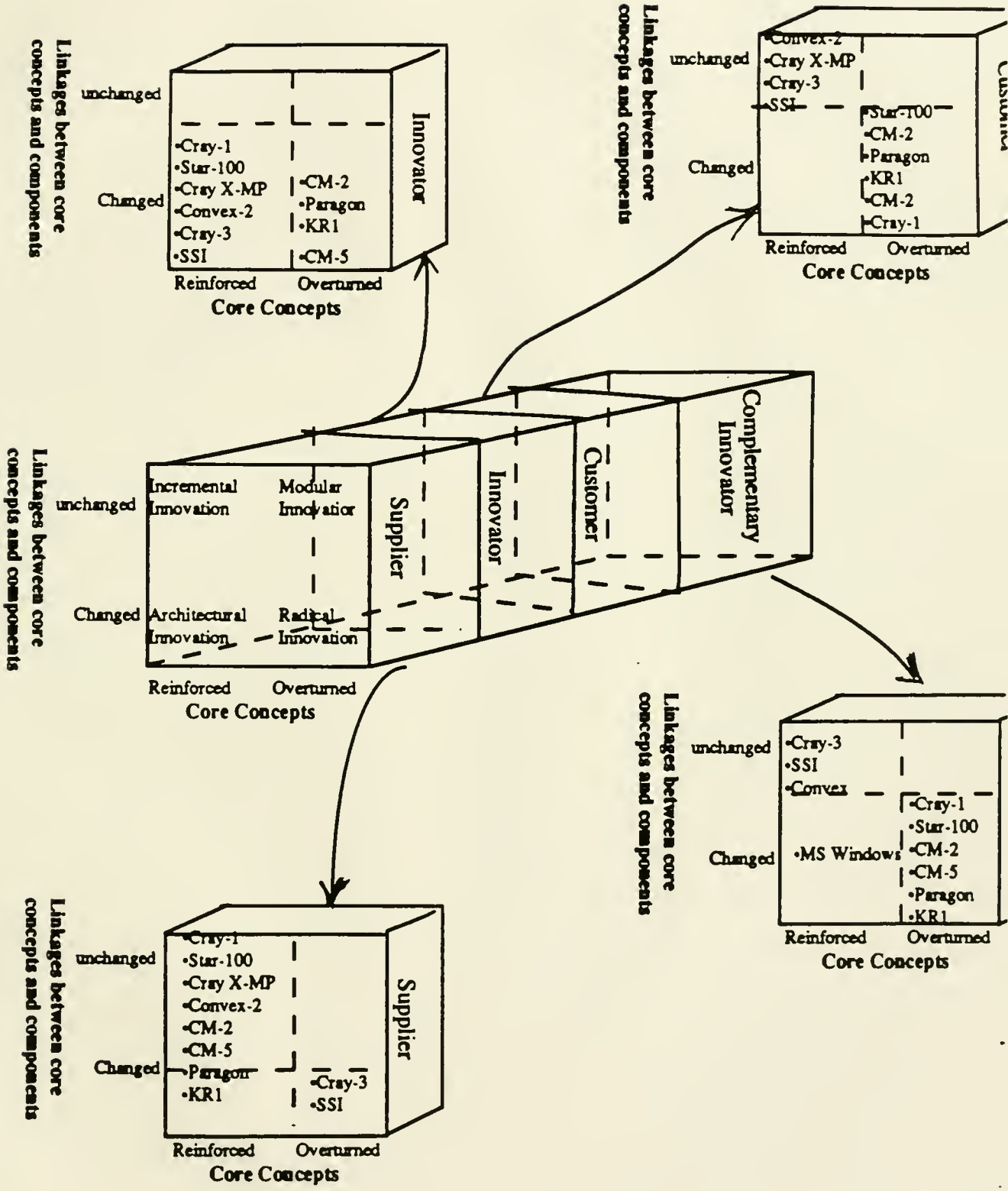


Figure 9 :Supercomputers and the Hypercube