As an electronics professional, you are very much aware of the phenomenal growth that has taken place in our industry in the past decade. The 1980s promise to be equally rewarding as VLSI makes possible new and more complex applications for microelectronics.

The fundamental driving force behind the success of the industry has been the ability to build increasingly complex devices. This growth in complexity will continue in the 1980s. Our task will be to exploit this driving force in a way that will help you solve your problems and meet your needs.

The problem of the 1970s was to reduce the cost of the electronic functions needed to store and process data. Our approach was to integrate more and more *components* on a silicon chip at continually decreasing prices.

The problem of the 1980s is different. Now we must address the task of reducing the cost of electronic solutions; that is, reducing the cost you incur in using our devices to build products. Solving this problem will require a shift from the component integration that was prevalent in the 1970s to concentration on *system level* integration in the 1980s.

The reason for this shift is really inherent in the success of the 1970s. The declining price of LSI devices made it possible to put microelectronics into everything from toys to automobiles to appliances. Concurrently, the capability of the microcomputer was also growing. Used first for simple logic replacement, the microcomputer became powerful enough for dedicated computer applications and then moved on to become a building block for user-programmable computer systems.

We can now talk about putting the power of a mainframe CPU on a single chip. This buys you nothing as a customer, however, unless you can use that power. Hardware is computing *potential*; it must be harnessed and driven by software to be useful.

The proliferation and increasing complexity of microelectronic applications is rewarding in one sense, but troubling in another. When we multiply what it takes to program all the new microcomputer products so they can be applied, and then calculate how many computer programmers we will need, we come out with a requirement for about *one million programmers* by 1990! When we look at the fact that U.S. electronic engineering schools produced only an estimated 17,000 graduates in 1979, the problem looks even worse.

It is clear that our success is leading to what I call the *PROGRAMMER CATASTROPHE*. Unless an answer is found, there simply will not be enough programmers to use the computing power VLSI technology is capable of producing.

There is an answer, and perhaps the best way to consider that answer is by looking at an historical precedent. Ten years ago, all of MOS LSI technology was implemented in custom circuits. When a user wanted to design a product, he came up with specifications for several dozen custom circuits and an MOS manufacturer would design and build them.

If that approach had continued, electronics would be far less prominent than it is today. We would have faced rather quickly a "circuit-designer catastrophe." There wouldn't have been enough designers to keep up with the proliferation of applications.

The answer was the general purpose microcomputer. It may have provided more capabilities than you needed as a customer, but as a standardized product, it was less expensive than a custom approach and therefore more useful.

In the 1980s, we will have to replicate this success by integrating standardized software that will help you develop microcomputer applications less expensively. Figure 1 illustrates how this can work.

The present level of integration is characterized by the integrated central processor such as our iAPX 86 CPU. We are now in the process of developing and introducing additional support processors, such as a math coprocessor and input-output processor. Either of these devices, in conjunction with a CPU, can make up quite a large portion of a math-oriented or input/output-oriented system.

In the future, we will extend the process and begin to integrate system programming and software functions.

For example, we will be integrating the basic functions of an operating system. The benefits of using a standardized operating system should prove to be as significant as the benefits of using standardized microcomputer hardware. Development and programming costs will be reduced substantially, and you will have an upward compatible interface for future products.

Following that, we will integrate the features of high-level languages into the silicon-based computer

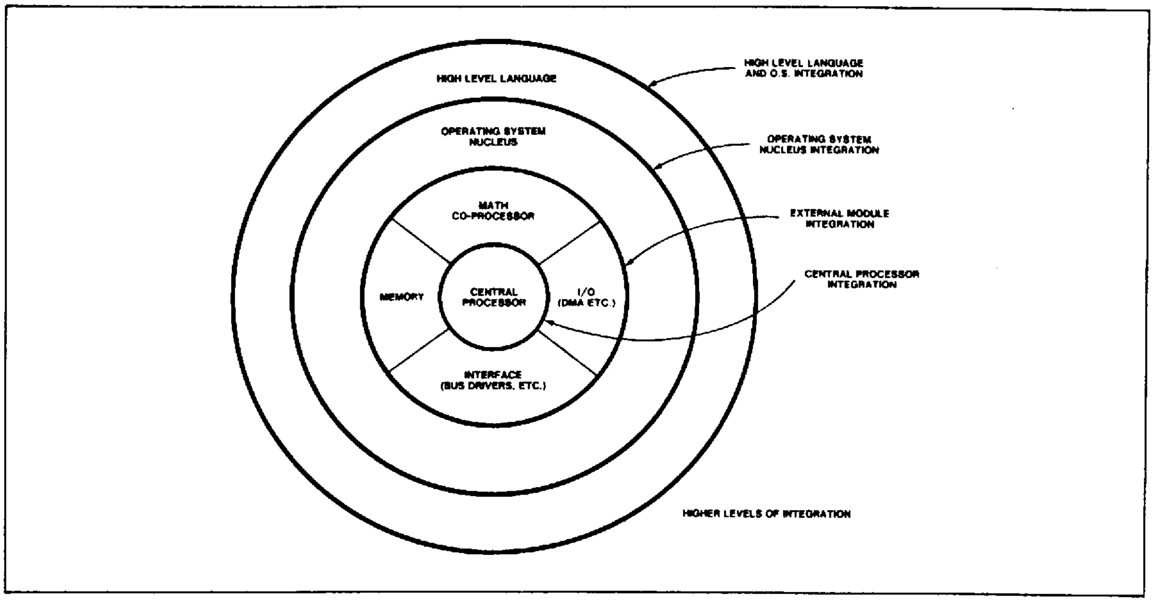


**Figure 1. Microsystem 80 Multiple Integration Levels**

system solution. What this will give you is a *minimum total cost* for your application.

Figure 2 shows classes of increasing functionality of microcomputers and, in the absence of better names, we call them microcontrollers, microcomputers, micromidis, micromaxis and micromainframes.

In order to keep your total cost minimal, the combination of the cost of the hardware and the cost of implementing the total solution must both be kept to a minimum.

At the microcontroller end, where software costs are not as significant, we integrate only those functions necessary to reduce replicated costs in higher volume. As we move toward the higher end, where large scale software programs are developed, we integrate the basics of the operating system and high level language features to lower your engineering investment.

Our goal is to deliver the optimum level of system integration that gives you a minimal total cost for the complexity level of your application.

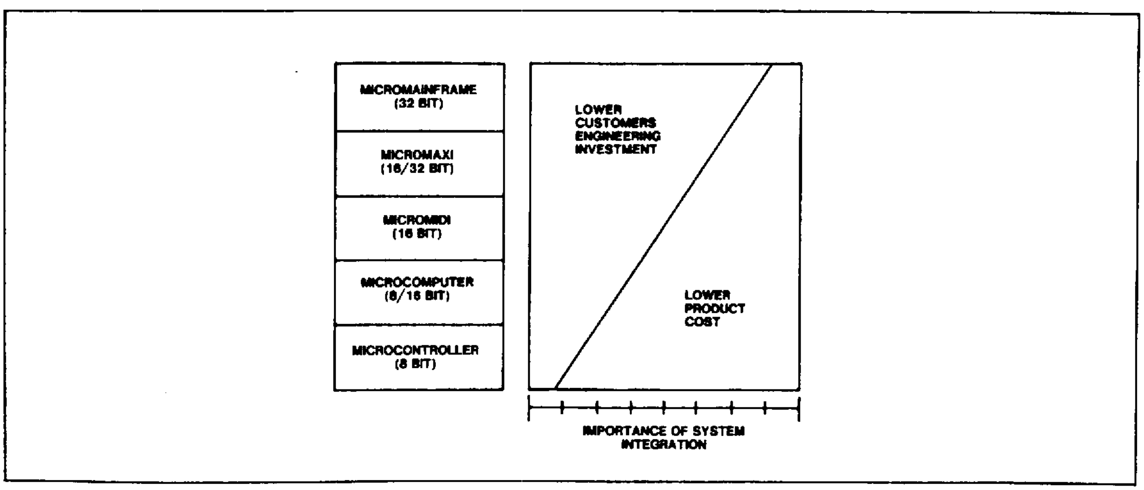


**Figure 2. The Driving Factors of Integration Costs**

Let me illustrate this by a quantitative look in terms of dollars. The implementation of a simple system of the early 70's, using components, may have cost a few hundred thousand dollars. By now, the typical cost of system implementation may be a million dollars. The complex system of the 80's, starting with a bag of components, will cost perhaps five million dollars to implement, assuming the programmers can be found to do the job! If we provide the optimum level of system integration as we have described, we can reduce your cost dramatically. (Figure 3.)

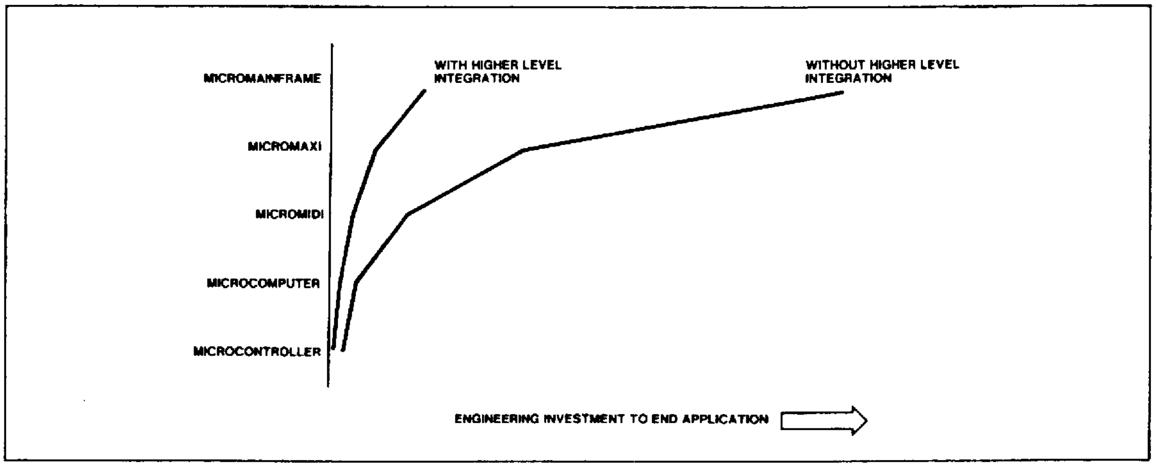


**Figure 3. Total Engineering Investment To Typical End Application, With And Without Integration**

This approach was the rationale for the design of the evolving Microsystem 80. This book covers how the iAPX 86 product line was constructed and how it will grow and be supported to implement this strategy. You will see, both in the sections that follow and in many of the individual product descriptions in the rest of this book, that much of our investment in our future product line is aimed at integrating software concepts and elements into the hardware to offer you a rational step in dealing with the "programmer catastrophe."

In summary, the iAPX 86 product line provides the foundation for meeting our objective for the 1980s: to offer you the world's highest performance VLSI systems, and to provide you the total solution at the lowest possible cost.

A. S. Grove

Andrew S. Grove
President