

題名	我がコンピュータの旅
Title	My Computer Odyssey
著者名	スタンレー・メイザー
Author(s)	Stanley Mazor
言語 Language	日本語・英語 Japanese, English
書名	稲盛財団：京都賞と助成金
Book title	The Inamori Foundation: Kyoto Prizes & Inamori Grants
受賞回	13
受賞年度	1997
出版者	財団法人 稲盛財団
Publisher	The Inamori Foundation
発行日 Issue Date	8/30/1999
開始ページ Start page	180
終了ページ End page	205
ISBN	978-4-900663-13-1

我がコンピュータの旅

スタンレー・メイザー

始めに

このたびの受賞につきまして、まず京都賞委員会の皆様と稲盛財団の皆様にご心より御礼申し上げますとともに、私のライフワークを振り返りこうしてお話しする機会を与えられましたことにも心より感謝申し上げます。本日は私がこの栄えある席におりますのは、私が所属していたチームがインテル社で開発しましたいくつかのマイクロコンピュータが広く利用され、社会に大きな影響を与えたことを評価いただいたからです。才能に恵まれた素晴らしい方々とチームを組めましたこと、また、このような賞をいただきましたことを私自身、身に余る幸運だと思っております。稲盛財団の皆様には、こうして私の人生哲学についてお話しする機会を作っていただき、大変感謝しております。皆様がおもしろいと思われる話になればよいと願っておりますが、本日の講演は、1) コンピュータとコンピュータ設計に私が関わるようになった経緯、2) マイクロコンピュータ・チップとその発明の経緯、3) 私が学んだ一般的な教訓、の3段階に分けて進めたく存じます。

I. 私がコンピュータに関わるようになった経緯

まず、35年前に、私がコンピュータの旅を始めたきっかけをお話したいと思います。ある日、私はサンフランシスコ州立大学(photo 1)で、たまたまコンピュータ・センターの前を通りかかりました。ドアの開いていた部屋の中をのぞき込むと、ライトが100個位明滅している変わった形とした機械がありました。それがIBM1620(photo 2)で、当時は巨大なコンピュータの全盛時代でしたから、当時のコンピュータとしては小さな方でした。もし私が大きな大学に在学していれば、フルタイムのコンピュータ・オペレータがいて、私のような学生の入室は認められなかったと思いますが、幸い、私の大学は小規模でコンピュータも小型でしたから、学生は自分で操作してみることが出来ました。当時、データの入力にはたいいパンチカード(photo 3)を使っていましたが、私の大学のコンピュータにはキーボードもついていました。このIBMのコンピュータを私はかなり個人的に使わせてもらいましたから、私にとって最初の「パーソナル・コンピュータ」だったと言えるかもしれません。

コンピュータを使いこなすためには、まずプログラミングのコースをとらなければならなかったのですが、私は早速このコースをとりました。私が最初にプログラムしたのは、三目並べのゲームで、私はコンピュータを相手によく遊びました。また、統計学のプロジェクト用にもプログラムを書きました。コンピュータが最終結果を出してくれるのを待ちながら、コンピュータ室に泊まり込んだことも幾晩ありました。私た

MY COMPUTER ODYSSEY

Stanley Mazor

Introduction

Thank you to the Kyoto Prize Committee and the Inamori Foundation for this award and for giving me the opportunity to give this talk which reviews my life's work. I am here today because I was part of a team that developed a number of microcomputers at Intel which have many uses and which have had a broad impact on society. I am fortunate to have worked with a very talented team and I am humbled to receive this recognition. I thank the Inamori Foundation for this chance to share my thoughts and philosophy and I hope that it might be of some interest to you. I have organized this talk into three sections:

- I. How I got into computers and computer design
- II. The microcomputer chip and how we invented it
- III. Some general lessons I've learned

I. How I got into computers

I would like to share with you how I began my computer odyssey thirty-five years ago. One day at San Francisco State University [photo 1] I peeked through the open door of the computer center and saw an interesting-looking machine that had about one hundred lights blinking on and off. This was an IBM 1620 [photo 2] computer and it was small compared with some of the giant computers of that era. Had I gone to a larger school they would have had a full-time computer operator and I would not have been allowed into the computer room, but with the small computer at our school a student could run it himself. Most of the data was entered using a punched card [photo 3], but it also had a keyboard, and I like to say that it was the first IBM Personal Computer.

To use the computer I would first need to take a programming class, which is what I did. My first program was a tic-tac-toe game and I played against the computer. I also wrote programs for my senior statistics project, and some nights I slept in the computer room while I waited for my final results to be computed. Our professors encouraged us to use the computer for serious calculations, but did not want us to waste valuable computer time playing games. A few years later when I would see people playing games on computers at work, I would remember what my professors had said about wasting computer time. However, twelve years later as an Intel applications engineer, I helped several companies develop their video game products, and computer games turned out to be a billion dollar industry which our professors had not imagined [photo 4].

ちの大学の教授は、まじめな計算のためにコンピュータを使うことを勧められましたが、私たち学生が貴重なコンピュータの時間をゲームに費やすことは望まれませんでした。数年たって、仕事中にコンピュータでゲームをしている人を見ると、私はいつも教授が言われていた「貴重なコンピュータの時間を浪費している」という言葉を思い出しました。しかしながら、それから12年後には、私自身インテルのアプリケーション・エンジニアとして数社でビデオゲームの制作を支援するはめになりました。(photo 4)その後、コンピュータ・ゲームが10億ドル市場になるとは、教授は夢にも思われていなかったことと思います。

私がシリコンバレーの企業に入社した経緯

大学を出て最初についた職業はマウント・ビューのフェアチャイルド・セミコンダクタ社でのプログラマでした。(photo 5)マウント・ビューはサンフランシスコの南60キロにあり、現在シリコンバレーと呼ばれている地域の中心にあります。そこにはトランジスタを製造する半導体企業が約40社ありましたが、フェアチャイルド社(photo 6)はその元祖ともいべき存在でした。当時、トランジスタ・ラジオが一般に普及していましたから、私はトランジスタが真空管よりも小型で、バッテリーの消費が少なくすみ、あまり熱くならないということを知っていました。私は、技術者や物理学者がトランジスタの設計をするためのプログラムを書きました。私のプログラムは、温度が変わるとトランジスタの挙動がどのように変化するか、また、製造段階がどのように製品に影響を与えるかといったことを予測するものでした。1965年までに、私たちは10個のトランジスタをボタンより小さな1枚のシリコン・チップ上(photo 7)に載せることが出来るようになりました。これらのトランジスタをチップ上の配線とともに形成することで、「集積回路」チップが出来ました。私は、この集積回路のトップメーカーに就職したわけです。

私がコンピュータ設計に関わるようになった経緯

これらのチップ製造の他にも、フェアチャイルドではデジタル時計や、コンピュータのメモリ・ボード、テスト機器、その他の特別プロジェクトへのチップの応用を模索していました。リサーチ課では巨大で、革新的な「高レベル言語」コンピュータを開発するプロジェクトが進んでおり、コンピュータの仕様と開発に現役のプログラマの参加を求めています。私はプログラミングをするだけではなく、コンピュータを作るチャンスだと思い、この仕事を希望しました。コンピュータの新機種を作ること



photo 1



photo 2



photo 3

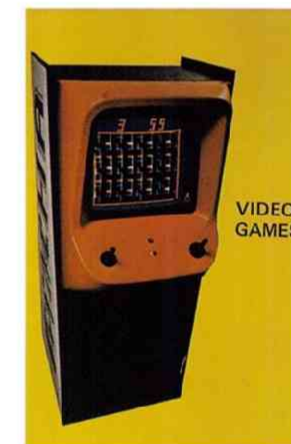


photo 4



photo 5



photo 6

は、とてもおもしろい仕事だと思ったのです。それまでの私の経験はソフト関連のものばかりでしたが、この機会にハードの設計にも挑戦しようと思いました。それで、1966年に研究所の方に配置替えになり、「シンボル」と名付けられた特別のコンピュータの設計仕様と「アーキテクチャ」を研究することになりました。私たちは「シンボル」を、ソフトウェアがほぼ不要で、使いやすく、画期的なコンピュータにしようと思いました。実際、私たちは「シンボル」がコンピュータ業界に革命を起こすと思っていました。シンボルの仕様が完成すると、その次には巨大なコンピュータの設計をしなければならませんでした。設計チームの各人(photo 8)がコンピュータの設計の各部分を担当しました。私のセクションでは10進数演算機構を扱いましたが、これは電卓で使用されている演算方法と非常に似ていて、それまでの2進数によるコンピュータとは全く異なる方法でした。この時、私は初めてコンピュータの設計をしたのですが、学校でした数学のコースのおかげでほとんどのことが分かりました。その他の、プリント基板の「デバッキング」等はこの仕事を通して学びました。

II. マイクロコンピュータ・チップの発明の経緯

私たちのマイクロコンピュータの発明について話を進める前に、「発明」という言葉にまつわる多くの誤解を解いておきたいと思います。一般に「発明」と言っても、実際に何かが完成する以前に、アイデアやコンセプトは存在しているケースが多いのです。たとえば、フランスの小説家ジュール・ヴェルヌは1850年に月に旅立つ「船」と、海中を進む「船」について空想小説を書いています。ですから、宇宙船や潜水艦を「発明した」のはジュール・ヴェルヌだと言うことも出来るかもしれません。しかし、単なる構想と、実際に宇宙や海中を進む乗り物を設計し、建造することは全く異なった次元のことだと言わなければなりません。私達の場合も、チップ上にコンピュータの機能を載せるアイデアは、すでに文献で述べられ、サイエンティフィック・アメリカン誌の特集記事にもなっていましたが、本当に重要なことは私たちが実際にそれを設計し、マイクロコンピュータを完成したことの方だと思います。

私がインテル社に入社した経緯

まず私がインテル社に入社した経緯について触れたいと思います。フェアチャイルド社での「シンボル」プロジェクトは、非常に野心的な試みでしたが、あまりにも目標が高すぎたためにコンピュータを1台組み立てた後に、プロジェクトそのものが中止に追い込まれました。1968年の後半に、もともとフェアチャイルド社の創設者であ

How I got to silicon valley

My first job out of school was as a programmer at Fairchild Semiconductor in Mt. View [photo 5], which is about 60 km south of San Francisco in what is now the heart of Silicon Valley. Fairchild was the origin of the valley's forty semiconductor companies and they made transistors [photo 6]. Transistor radios were popular, and I knew that transistors were smaller and better than vacuum tubes because they use very little battery power and do not get very hot. I wrote programs that helped engineers and physicists design transistors. My program predicted how a transistor operates at different temperatures and how certain manufacturing steps affected our transistor products. By 1965 we could get ten transistors on a single "silicon chip [photo 7]" which was smaller than a button. These transistors were connected together with on-chip plated wires to form an "integrated circuit" chip. I had joined a company which was a leading maker of integrated circuits.

How I became a computer designer

In addition to making these chips, Fairchild also investigated uses for their chips such as digital clocks, computer memory boards, test equipment and other special projects. The Research Department [photo 8] had a project to build a very large revolutionary "high level language" computer and they wanted a practicing programmer to join in the computer's specification and development. This was an opportunity to build a computer rather than just programming one, and I was excited about developing a new kind of computer. Although my experience was in software I decided to take a chance and try "hardware" design. So I transferred to the Research Lab in 1966 to work on the specification or "architecture" of this very special computer called "Symbol." We believed that this computer would need very little software, be easy to use and we thought it would revolutionize the computer industry. After the computer specification was completed, we had a giant computer design to implement. Each of us was assigned a part of detailed computer design. My section did decimal arithmetic (very similar to that used in desktop calculators and very different from conventional binary computers). This was my first experience in designing a computer, but the math courses I had taken in school prepared me for most of what I needed, the rest such as "debugging" printed circuit boards, I learned on the job.

るノイス博士とムーア博士が(photo 9)、グローブ博士をともなってフェアチャイルドを去り、新たに集積回路のメモリ・チップを製造する企業、インテル社(photo 10)を創設しました。それまでのコンピュータのメモリは「コア」と呼ばれるドーナツ型の小さな金属片を使用していました。当時メモリ・チップを製造する企業はなく、メモリ・チップそのものが新しい概念だったのです。私はテッド・ホフ（エドワード・ホフ）が率いるアプリケーション・リサーチ課に所属するように勧められ、この申し出を受けることにしました。私たちの仕事は、デモンストレーション用のコンピュータを作り、メモリ・チップ(photo 11)が従来からあるコアに取って代われることを実証することでした。同時に、インテルが開発していたシリコン技術の新しい応用を開発することも、私達の仕事でした。ちなみに、このシリコン技術の応用として生まれたのがマイクロコンピュータ・チップでした。これらの仕事は、私がフェアチャイルド社でしていた仕事と似ていましたが、大きな違いがありました。以前は、世界一大きなコンピュータを開発しようとしていたのですが、今度は世界一小さなコンピュータの開発を目指していた点です(photo 12)。

MCS-4とビジコンの電卓

私の同僚、マサトシ・シマ（嶋正利）は電卓メーカーのビジコン社からインテルに派遣されてきました。当時ビジコンは新しい電卓用チップの製造をインテルに依頼していたのです。ホフは電卓用チップの革新的な設計を提案しました。つまり、汎用コンピュータを作り、それに電卓機能のプログラムを追加するというのです。ホフの構想どおりにすると、電卓のハードウェア設計の問題はプログラミングの問題に置き換わります。また、電卓に使用されるチップのほとんどがコンピュータ・プログラムを保持するためのメモリ・チップになりますが、メモリ・チップはインテルが最も得意な分野です。私はホフに協力してこの(MCS-4) (photo 13)コンピュータの仕様を開発し、また、このコンピュータではソフトウェアによって電卓の演算が出来ることを実証するプログラムを開発しました。

嶋(photo 14-1) (photo 14-2)が私に電卓の必要条件を説明してくれました。私はMCS-4に電卓機能を持たせるためのプログラムのサンプルを書きました。たとえば、電卓のキーを押すとそれに反応したり、結果を印刷したり表示する機能です。以前にIBMの1620や「シンボル」を扱った経験が、電卓の10進数演算機能の開発に非常に役立ちました。もちろんこのコンピュータはまだ構想段階にありましたから、全ては「机上で」進められました。私は想像上のコンピュータのプログラミングをしていたわけです。

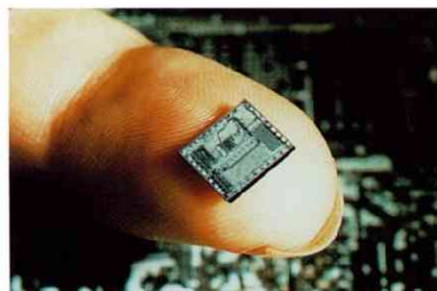


photo 7



photo 8



photo 9



photo 10

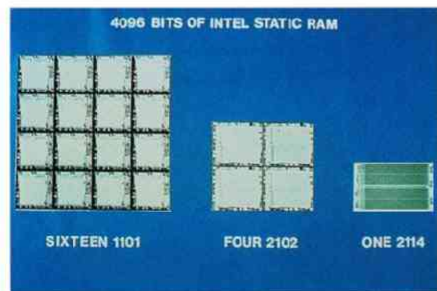


photo 11



photo 12

嶋と私は「机上の」コンピュータについて英語で話し合いました。2年後に、嶋が完成した電卓プログラムは、フェデリコ・ファジンが設計した本物のチップ上で、実際に機能したのです。フェアチャイルド社でのプロジェクトとは対照的に、このプロジェクトはビジネスとしても成功を収めました。私の以前の経験も、プログラミング、コンピュータ設計、そしてチームワークといった面で遺憾なく発揮されました。ホフ、ファジン、そして私はMCS-4がビジコン電卓以外のアプリケーションにも使用出来ることを知っていましたから、インテルの経営陣を説得し、このコンピュータをインテルの標準製品として販売する権利を取り戻しました。こうして、最初のマイクロコンピュータ・チップが1971年11月に発表されたのです(photo 15)。

困難な問題

ここでコンピュータ・チップの設計における主な障害と、それをどのように克服したかについてお話したいと思います。1970年代に「パワフルな」コンピュータを作れなかった理由は、「大きな」チップが作れなかったからです。パワフルなコンピュータには何千ものトランジスタが必要で、そのためには大きなチップがいります。ところがチップの素材に欠陥がたくさんあるため、チップを大きく出来ないのです。マイクロコンピュータのチップのことは馴染みのある方が多くはないと思いますから、ここで、婦人服の製造にたとえてみたいと思います。今、ここに婦人服の生地(photo 16)があるとします。ところがこの生地にはたくさんの欠陥、つまり穴があいているとします。穴はあちこちにランダムにあいているのですが、図(photo 16)の6個の穴のような状態だと思ってください。この生地から穴のあいていない婦人服を作ることは不可能です。図(photo 17)では穴は4個ですが、これでは婦人服は1着も出来ません。

ここで小さな女の子のための服を同じ生地から作ることとします。小さな女の子ですから図(photo 18)のように生地も少なくすみます。同じ生地から4着分とれますが、そのうち2着分は穴があいているので使えません。作れる服は2着ということです。さて、今度は人形の服を作ることにします。この場合、同じ生地から16着分がとれ、図(photo 19)にあるように、そのうち14着は穴のあいていない服になります。皆さんだったらどの服を作るメーカーになりますか。生地には穴があいているので、婦人服は作れませんが、女の子の服か人形の服なら可能です。

この服の話と同様に、半導体のウェーハ素材(photo 20)、シリコン結晶には目に見えない欠陥がランダムにあります。服地を裁断するように、チップもウェーハ素材から切り出しますが、大きく切り出そうとすれば、どうしても欠陥のある場所が入って

II. The microcomputer chip invention

Before I discuss our microcomputer invention I would like to suggest that most people have a misunderstanding about the conception of invention and its implementation. In many cases the idea or concept already exists before someone figures out how to actually implement it. As an example, Jules Verne, the French fiction writer, wrote in 1850 about a ship which could travel to the moon and another ship which could go beneath the sea. You might therefore call Jules Verne the inventor of the space ship and the submarine. However, the mere conception of such vehicles is far different than designing and building *real* vehicles which can travel under water or in space. In our case, the idea of a computer on a chip had been mentioned in the literature and even appeared on the cover of *Scientific American* magazine. It was the design and implementation of our microcomputer chip that was important.

How I got to Intel

But first let me mention how I got to Intel. The project goals for the Fairchild Symbol computer were too ambitious, and after building one of these computers the project was abandoned. About the same time in late 1968, the original founders of Fairchild, Dr. Noyce and Dr. Moore [photo 9], along with Dr. Grove left Fairchild to start Intel [photo 10], a new company to build integrated circuit memory chips [photo 11]. Conventional computer memories were built using little metal donuts called “cores.” No company was making memory chips and the idea of memory chips was new. I was chosen to join Intel in Ted Hoff’s Applications Research Department. Our job was to build demonstrations to show how Intel memory chips could be used instead of core memories. We also tried to find new uses for the silicon chip technology being developed by our company — which turned out to be microcomputer chips. This was similar to the kind of work I had done at Fairchild, except that rather than working on one of the world’s largest computers I would be working on the world’s smallest computer [photo 12].

MCS-4 and Busicom calculator [photo 13]

My office-mate at Intel was Masatoshi Shima from the Busicom calculator company, which had contracted Intel to build chips for a new calculator [photo 14-1] [photo 14-2]. Ted Hoff proposed a radical solution for their calculator chips—to build a general purpose computer programmed to do the calculator functions. This would turn the calculator design into a programming problem, for which I had a lot of



photo 13

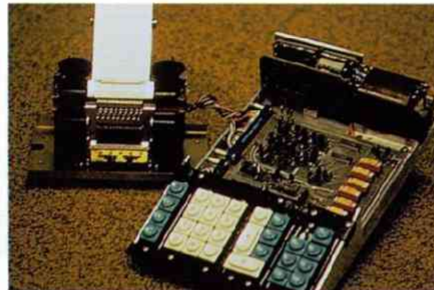


photo 14-1



photo 14-2



photo 15

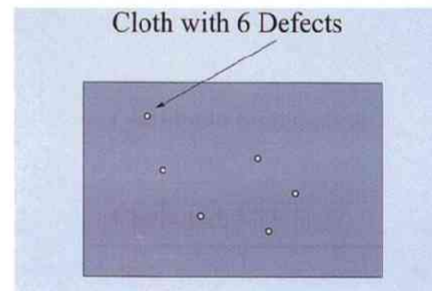


photo 16



photo 17



photo 18

experience, and would mean that most of the chips used in the calculator would be memory chips to store the computer program. Recall that memory chips were Intel's specialty. I worked with Ted on the specification of this computer, and I also wrote programs to demonstrate that calculator functions could be programmed in software.

Shima described the calculator's requirements to me. I wrote sample MCS-4 programs to do these calculator functions—such as responding to keys being pressed and printing and displaying results. My prior experience with the IBM 1620 and the Symbol computer design prepared me for the calculator's decimal arithmetic functions. However, since our microcomputer did not exist yet we did everything on paper. I was programming an imaginary computer. Shima and I had an interesting time communicating in English about a “paper” computer. Two years later Shima's calculator program was working on Faggin's real chips. In contrast to the failed Fairchild project, this project met commercial success and I had utilized my prior experience in programming, computer design, and teamwork. Ted, Federico, and I knew that the MCS-4 could be used in other applications beyond Busicom's calculator and we persuaded management to get back the right to sell this computer as a standard Intel product. These first microcomputer chips were advertised in November 1971 [photo 15].

The challenge

I'd like to explain our major obstacle in computer chip design and how we overcame it. We could not make *powerful* computers in the early 1970s because we could not produce *big* chips. A powerful computer needs thousands of transistors and requires a big chip. You can't make chips big because they would have too many defects in them: Since you may not be familiar with microcomputer chips I will try to illustrate the problem as if we were in the women's dress business. Consider a piece of cloth which has some holes or defects in the material [photo 16], and that any piece of cloth would contain defects in differing locations, but similar to the six random holes shown in Figure 1. Suppose we desired to cut out a large women's dress from the material; it might *not* be possible to get a defect free dress. In Figure 2 the dress contains four defects [photo 17], and this cutting operation yielded no good products.

However, suppose we were making smaller dresses for little girls, which used much less material, as shown in Figure 3 [photo 18]. There are four patterns on the same piece of material, notice that two of them are okay and two of them are defective. Now consider producing tiny dresses for little dolls instead. In this case sixteen patterns fit on the same piece of material and our yield rises to fourteen good

します(photo 21)。チップの設計者はチップのサイズを決め、数学モデルを使って歩留まりを予測します。図(photo 22)は欠陥のある素材を使った3種類のチップ設計を示しています。同じ円形の切り口のシリコン結晶を使用した場合、予測される歩留まりは、チップのサイズによって非常に違ってきます。大きなチップにすると良品の数は0、一番小さなチップにすれば65になります。チップは大きければ大きいほど多くのトランジスタを搭載することが出来、パワフルなコンピュータを作ることが可能になります。小さなチップは搭載出来るトランジスタの数が少なくなり、良いコンピュータは出来ません。婦人服のメーカーとは違って、実際の半導体メーカーでは誰も極小サイズのチップには関心を持ちません。設計者にとって難しいことは、そこそこの歩留まりで出来るだけパワフルなコンピュータを作るために、チップのサイズを決定することなのです。もし、非常に性能の良いコンピュータを作ろうとするなら、チップが大きくなりすぎ、歩留まりが下がります。ところが生産性ばかりに気をとられると、小さなチップになり、役に立たない代物になります。

大きなチップを作れない訳をお分かりいただけたと思いますが、この問題を私たちがどのように克服したか、別のたとえを使って説明しましょう。皆さんが6人乗りの乗用車(photo 23)の設計をしているとします。コストは車1台につき5万ドルするとします。さて、皆さんの上司が1000ドルで売れる車を作れと命令したら、皆さんはどうしますか。設計を単純化し、車を小型化し、パワーも減少させることになるでしょう。それでも、どうしたらコストを1/50にまで下げることが出来るでしょうか。そうするには、スピードが1/20しか出ない二人乗りのゴルフカート(photo 24)にするしかありません。皆さんはゴルフカートをご存じですから、そんなことは当たり前だと思われるかもしれませんが、もし、ゴルフカートがまだ開発されていなかったら、一体何をどのくらい削減すれば良いのか、皆さんはきっと迷われるでしょう。

確かにゴルフカートは大型乗用車より安上がりですが、機能の面でも劣っています。街の中を二人乗りで行くには役に立つかもしれませんが、ゴルフカートに乗って京都から大阪まで行こうとは思いません。チップの設計を行う人は、サイズを小さくしても必要なパワーを保てるラインを見極めなければなりません。

ビジコンの電卓用チップを開発していた時、私たちの目標は、出来るだけ小さなチップで、電卓の演算と制御機能を持つだけのパワーがあり、かつ100ドルで売れる、という3つの条件を満たすことでした。コンピュータの機能を単一チップに搭載するために、どこまでコンピュータの規模を縮小するかということが問題になりましたが、私たちは、一度に1桁の10進数を処理出来る(4ビット)能力があれば良いという結論

dresses, as shown in Figure 4 [photo 19], where only two of the dresses are defective. Now we have to ask what business would we like to be in? Because we start with defective material we *cannot* make women's dresses, but we could make girl's or doll's clothes as summarized in Figure 5.

Similar to this dress maker example, the semiconductor wafer materials [photo 20] have randomly placed defects in the silicon crystal structure which you cannot see. Chips are cut out of the wafers similar to the way that dresses are cut out of a piece of cloth [photo 21]. You can't make big chips because they would have defects in them. Chip designers choose the size chip they will make and predict using mathematical models the expected manufacturing yield. Figure 6 illustrates three different chip designs each with a different chip size on a wafer containing defects. The expected yields per round silicon wafer are very different. The largest chip size has zero yield, and the smallest chip design has a yield of sixty-five good chips [photo 22]. A large chip can have more transistors on it and provides a more powerful computer. A small chip has fewer transistors and will not be a very good computer. In fact, unlike the maker of dresses, the semiconductor manufacturer may find no one interested in a "doll" sized chip. Hence, the challenge for the designer is to define the right size computer chip that can be manufactured with a reasonable yield and is powerful enough to do something useful. If your goals are too ambitious the chip will be too large and will have a low yield. If you are too cautious, the chip will be too small, the yield will be good, but the chip will be so simple that no one will want to use it.

Now that you have an idea of the limitation we had in making big chips, I would like to describe by another analogy how we overcame this limitation. Consider for a moment that you are a designer of a 6 passenger car that cost \$50,000 to make [photo 23]. Your company asks you to design a new car which sells for \$1,000. You know you must simplify your car design, and make it smaller and less powerful. But how can you reduce the cost to 1/50th? The result might be 2 passenger golf cart that goes 1/20th the speed [photo 24]. But although this is an obvious solution to you now, if a golf cart did not exist already, it is not so easy to figure out what to scale down.

While it is clear that a golf cart is much less expensive than a large passenger car, it is also clear that it cannot perform the same function. A golf cart might be suitable for transporting two passengers within the downtown area of a city, but it is not suitable for getting from Kyoto to Osaka. Hence one must consider if your scaled-down design is powerful enough to meet your needs.

Our goal for the Busicom calculator chips was to define a small enough

に達しました。それに対して1970年代のミニコンピュータはほとんどが一度に4桁を処理出来、私たちのマイクロコンピュータと比較するとスピードも100倍、価格も100倍していました。当時普及していたコンピュータと比べると、私たちのマイクロコンピュータは価格が安い代わりにスピードも遅かったのです。ホフと私は、電卓やその他何百もの用途に使えるようにプログラミングすることが可能な小さなチップのスペックを書きました。過去25年の間に、チップのサイズが大きくなり、コンピュータのスピードも上がりました。私たちの最初のマイクロコンピュータは約2,000個のトランジスタをチップ上に搭載していましたが、今日のチップは何百万個ものトランジスタを搭載しています。今日のコンピュータは、低価格を維持しながらも非常に速度が上がり、しかも複雑な設計になっています。

マイクロコンピュータの波及効果

たとえ発明品が完成し、新製品が出回っても、誰も使用しないのでは無意味と言えます。インテルで、私は嶋がしていた電卓機能のプログラムを手伝いましたが、そのうちに新しい技術や製品が広く採用されるためには、教育が必要だと痛感するようになりました。私は過去20年間、業界における講師として、世界各地で新しい技術をいかに応用するか説明して参りました(photo 25)。また、技術者やプログラマに古い習慣やツールを捨て、新製品を使うように勧めてきました。エンジニアやプログラマは、大学を卒業してからも学び続けなければなりません。そのためには良い講師が必要です。私は時間の大半を新しい講師の教育と実地訓練に費やしております。教育プログラムに関して私が特に力を注いでいることがあります。それは、難しい概念を身近な例を挙げて説明することで、本日お話しした婦人服製造の話も、如何にして欠陥のある素材から大きなチップを切り取るかという問題を分かりやすく説明するために私が考案したものです。

このように教育に携わって来ましたが、それでも全ての企業が新しい技術の採用に成功している訳ではありません。ここで例を二つ挙げて、新技術が企業や個人にどのような影響を及ぼすかについてお話ししたいと思います。1975年に私はインテルで最初のヨーロッパ在駐のアプリケーション・エンジニアになりました。そこでインテルの顧客が私達のマイクロコンピュータを応用して新製品を開発する支援をしたのですが、その時ドイツで私が訪ねた2社についてお話ししたいと思います。1社は機械工が作った機械部品を使ってキャッシュ・レジスタを組み立てる企業でした。もう1社では、電子部品を使ってコンピュータを組み立てていました。インテルの新しいマイ



photo 19

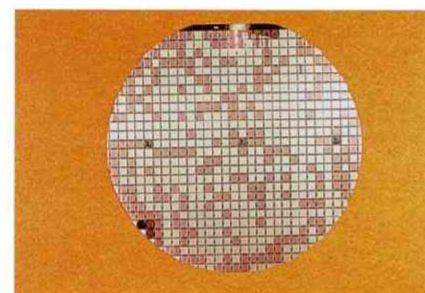


photo 21



photo 23



photo 25

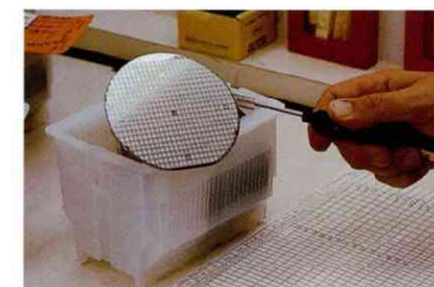


photo 20

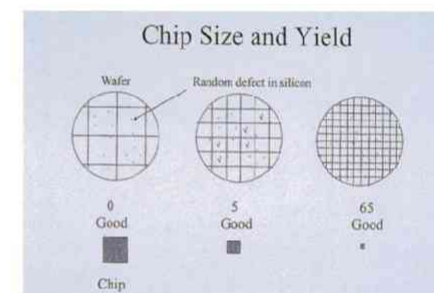


photo 22

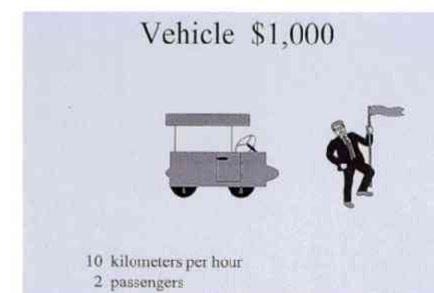


photo 24

クロコンピュータがこれら両社に影響を与えたのですが、コンピュータ企業の方はマイクロコンピュータを利用してうまく「技術革新の波」に乗り、その後の10年間、非常に大きな成功を収めました。ところが、機械部品を使った組立企業の方は、エレクトロニクスの利用に失敗し、電子キャッシュ・レジスタが出現すると、市場競争に敗れキャッシュ・レジスタ市場からの退散を余儀なくされました。

私達が発明したマイクロコンピュータには何千もの応用が考えられ、また、パソコンは今日世界中で使用されています。コンピュータは濫用されることもありますが、それでもコンピュータの使用によって多くの人々の生活が潤っているはずです。有名な発明家ホールで、私は昨年、目の不自由な教授からお礼を言われた経験があります。教授は、私の方にやって来られ、「私は目が不自由ですが、コンピュータのおかげで仕事が出来ます」とマイクロコンピュータの発明に対して感謝の言葉を述べられました。私はこの言葉に非常な感銘を覚えました。

CAD

また、コンピュータの応用方法の一つとして、設計支援があります。コンピュータを使って設計することを、コンピュータ・エイデド・デザインもしくは略してCADと言いますが、CADはエンジニアが新しいコンピュータの設計をする時にも使われます。コンピュータの設計にコンピュータを使うことをおかしいと思う方もいらっしゃるかもしれませんが。ロボットを作るためにロボットを使うようなものですから。チップの製造プロセスでは写真技術を使うので、いったん設計を変更するとチップの修正に約8週間かかります。ですから設計を変更しないですむように、初回で99.9%正確な設計が求められます。このような設計のチェックにはCADを使用する以外方法がありません。インテルでは、ホフともう一人の物理学者の下で、私は（回路シミュレーション用の）CADプログラムを書き、チップ設計者をサポートしました。後年、私はCAD企業で、CADツールの研究に専心することになります。

私が学んだこと

私の半生を振り返るに当たり、稲盛財団の方から私の思想や哲学を披露して欲しいという要望がありました。皆様に関心をお持ちだということを中心に、以下に5項目ほど、私の経験を通じて学んだこととお話しさせていただこうと思います。

長期計画は失敗する可能性が高い

computer chip that was powerful enough to do desktop calculator arithmetic and control functions and that could sell for \$100. Our problem was to figure out how to scale down a computer to fit onto a single chip. We concluded that a computer that could process a single decimal digit at a time (four bits) would do the job. By contrast most of the minicomputers in the 1970s could handle four digits at a time. Overall, they were about 100:1 times faster and 100:1 more expensive than our microcomputers. Our early microcomputers were low priced and very slow when compared to popular computers of that era. Ted and I wrote a specification for a small computer chip that could be programmed to be a desk calculator and also be programmed for hundreds of other uses. Over the past twenty-five years the chips got bigger and the computers got faster. Our first microcomputer had about 2,000 transistors on the chip, today's chips contain millions of transistors. Today's computers are still low cost and very, very fast, and their designs are very complex.

Impact of the chip

After an invention is complete and a new product is created, it is of no real significance if no one uses it. At Intel I worked with Shima to help him program the desk calculator functions. To get new technology and products adopted, it is necessary to educate people. My job for the last twenty years has been a teacher in industry, explaining how to use these new technologies [photo 25]. I try to get engineers and programmers to abandon their old practices or tools and use new products. Engineers and programmers need to continue learning after they graduate college and many good teachers are needed. I spend most of my time teaching and also training new teachers to do industrial training. One of the training challenges I enjoy is developing new examples and analogies to explain new concepts. The dress making example I used today to explain the problem we faced in making large chips from defective material is an example of the training analogies I develop.

However, even with education, not all companies can adopt new technology. Let me share a couple of examples of how technology affects companies and individuals. In 1975, I became Intel's first Application Engineer living in Europe. I helped customers develop new uses of our microcomputer products. Two interesting companies that I visited were in Germany. One was a company building cash registers using mechanical parts made by machinists. Another company was building computers using electronic parts. Intel's new microcomputers affected these two companies. The computer company was able to use microcomputers to ride the "technology wave" into the next decade very successfully. However, the mechanical company did not succeed in using electronics and failed to compete with

まず、初めに長期のプロジェクトは続けることが難しいということです。長い期間をかけていると、その間にビジネスにおける優先順位やスタッフも変わりますし、何より志気を維持することが難しいのです。フェアチャイルド社で失敗したプロジェクトは5年以上かかりました。また、「非常に実験的な試み」であったために、いくつもの新しいアイデアを試しましたが、どれも現実的ではなく、あまりにも多くのことを試したために長い期間をかけすぎたと思います。それに対して、インテルで成功したプロジェクトは18ヶ月でした。全ての長期計画が失敗するというわけではありませんが、私自身は経験からプロジェクトは短期の方が良いと思っています。

間口を広げすぎない方が良い

私はインテルを辞めてからソフトウェア企業2社の設立を手伝いました。最初の1社はある程度成功しましたが、一つの製品にあまりにも多くのことを追い求めすぎたために、結局挫折しました。2番目の企業では、構想は似通っているものの、もっと「単純な」製品を作ることを目指しました。「単純な」ソフトは小型で高速のチップを作るのに適していたので、このCAD企業は成功しました。時には、汎用性を追わずに、このように利用範囲を絞った製品を作ることも重要だと思います。アメリカには「手にあまることには手出しするな（直訳：飲み込めないほどの食べ物をほおばるな）」という諺があります。

また、「シンボル」コンピュータのプロジェクトに関して、フェアチャイルドの経営陣の中には、目標は数多く、高く設定し、そのうちいくつかが達成出来れば、それで成功したとすれば良いと考えている人がいましたが、私の経験では、それは違うと思います。「シンボル」プロジェクトはあまりにも野心的で、しかも的を絞らなかったために失敗したのだと思います。

人生は計画どおりにはいかない

3番目の教訓として私が学んだことは、私たちには情報が全てあるわけではないので、将来のことを完璧に計画することは出来ない、ということです。フェアチャイルド社で行った「シンボル」プロジェクトは現実的ではありませんでしたが、非常に先進的なコンピュータの構想を実現しようとする試みに私は胸の高鳴りを覚えました。ですから、その後私が携わるプロジェクトが「シンボル」プロジェクトと同様におもしろいものになるとは、思っていなかったのです。実際、私がフェアチャイルド社を辞めた時、私は次の仕事にそれほど期待していませんでした。しかしながら、私のキャ

electronic cash registers as they were developed. They eventually went out of the cash register business.

Our microcomputer invention has led to thousands of specialized uses and the personal computers are being used everywhere. I feel that the computers are often abused but that many lives are enhanced because of their use. At the Inventor's Hall of Fame last year, a blind professor came up to me and thanked me for our microcomputer invention which has enabled him to be productive on his computer even though he has no sight. I was very touched by his comments and his sincerity.

Computer Aided Design

Another use of computers is to help designers design something. When a computer aids our design work, we call it Computer Aided Design or CAD. Computer Aided Design is software that helps engineers design new computers. Sometimes it seems funny to people that we used computers to help design computers. It's like using robots to assemble more robots. The chip-making process is photographic, and it takes about eight weeks from the time you change a design until you get a chip in your hands; you must get the design 99.9% correct the first time. You can only do the checking using a computer program. Working under the direction of Ted and an Intel physicist, I wrote programs (for simulating electronic circuits) to help chip designers get working chips. I would later spend many years working on CAD tools in CAD companies.

What I learned

As I review my life's work, the Inamori Foundation has asked me to share some of my thoughts and philosophy which I hope may be of some interest to you. I will highlight five ideas for you to consider.

Long projects are likely to fail

I learned that on long projects it is difficult to have continuity because of changes in business priorities, staffing, and project morale. The Fairchild Symbol computer project lasted for more than five years. Because it was "very experimental," we tried many new ideas which were not practical; we tried to do too much, it took too long and it failed. In contrast to this, the Intel microcomputer design projects took about eighteen months and were successful. Now I am not saying that all long projects will fail, but based upon my experience I prefer shorter projects.

リアの中で最も意義があり、輝かしい出来事（マイクロコンピュータの発明）は、フェアチャイルドではなくインテル在籍中に起きたのです。この経験を通して私は、人生には未知のことがたくさんあり、人生の結末などとうてい予測出来るものではないことを知りました。私たちは間違っただけで正しいことをすることもあれば、逆に正しい目的のために間違っただけをしてしまうこともあります。

もう一つ、将来を予測出来ないと思わせる出来事がありました。私は大学に入学した当初、特定の職業的な技能を得ようという気持ちは全くなかったのですが、結局、在学中にコンピュータを学び、卒業後はプログラマーとして身を立てることになりました。その後、コンピュータの設計や集積回路について学ぶことになるとは、夢にも思っていませんでした。言い換えれば、一般的な教養を身につけようと思って大学の門をくぐりながら、最後はコンピュータのキャリアを持つことになったのです。

私はハイスクールの頃、生徒会活動に従事していましたが、そこでチームワークと妥協することの重要性を学びました。マイクロコンピュータのプロジェクトでもチームの一員として働くことが求められ、学生時代の経験が非常に役に立ちました。また、生徒会活動を通して人前で話をする経験を積みましたが、これは後に講師活動をした際に非常に役立ちました。大学で学んだ数学は、コンピュータの演算ユニットの設計に役立ちました。学んでいた当時、これらのことが後年私にとってどれほど重要になるか、全く予想していなかったのです。

自分のしていることに情熱を持つこと

私の半生を振り返って見ますと、私は、それなしには生きていけないほどの情熱を何かに傾注した経験が何度かあるのです。16歳の時は、一人乗りのヘリコプターの写真を見て非常にヘリコプターに興味を覚え、本を読みあさりました。ヘリコプターの設計をする会社を見つけ、ヘリコプターの建造を夢見ました。結局は1機も建造しませんでした。それでも長い年月をヘリコプターの建造を考えて過ごしました。

初めてコンピュータのプログラミングを考え出した時は、完全にコンピュータにとりつかれ、1日のほとんどをプログラムの書き換えに費やしていたので、心身共に「燃え尽きた」状態だったと思います。私以外にも、文字どおり「コンピュータの虫に食われた」人が大勢いるのをご存知だろうと思います。

初めて本を執筆した時も、私はキーボードから離れられませんでした。私の人生を通して、コンピュータの設計からエンジニアに指導することまで、色々な仕事に熱中出来たことを私は幸せだったと思います。さらに、熱中してやった仕事で成果をあげ、

Projects which are too broad in scope are likely to fail

After I left Intel I worked at two software start-up companies. Although we had some successes, the first company's product tried to do too much and was abandoned. The second company's product was based upon similar ideas, but with a *simpler* product. This simpler software did a very *good* job of making chips smaller and faster. Sometimes it is important to do a better job on a project which has narrower scope; in America we say: "do not bite off more than you can chew."

Some of the Fairchild management on the Symbol computer believed in an engineering principle which is to set very high goals, so that if you achieve only some of the goals you might still have a good result. In my experience at Fairchild that did not work; the broad scope of the project was too ambitious and caused the project to fail.

We cannot plan our life exactly

A third lesson I have learned is that we can't plan our future completely because we do not have all the information. Although the Fairchild Symbol computer project was not practical, it was exciting to be working on very advanced computer ideas. I did not expect that any other work would ever be as interesting as that project had been. When I left Fairchild I doubted that my next job could be as interesting. But as it turned out the most significant event and right thing in my career (the microcomputer) happened at Intel, not at Fairchild. Therefore, I conclude that we cannot predict the outcome of life's events which involve many unknowns. We do the right thing for the wrong reasons—or the wrong thing for the right reasons.

As another example of the difficulty in predicting my future, I started college with no particular profession in mind, learned about computers and entered industry as a programmer. I then learned about computer design and integrated circuits. Although I started out to get a general education I ended up with a career in computers.

My early experiences in high school student government taught me about teamwork and making compromises. These lessons helped me when working as a team member on the microcomputer. My student government public speaking experiences helped me in my teaching career. My college mathematics helped me to design computer arithmetic units. I could not have predicted that these experiences would have been so valuable to me later in my career.

人助けにもなることほど嬉しいことはありません。最近では、古い建築物に関心を持っていますので、どこかに古いお城があると聞くと何が何でも行ってみたいのです。

チャンスを逃がさないこと

私が人生の大半を過ごしたシリコンバレーでは、常に新しい企業、新しい製品を生み出すチャンスがありました。私自身は、文字どおり好機をつかんで成功したこともありますし、「シンボル」コンピュータやシリコン・コンパイラのように成功出来なかったこともあります。しかし、私は成功からも失敗からも多くのことを学びましたし、これらの経験を通じて人生を豊かにすることが出来ました。私が設立に参加した企業の中では3社が成功しています。

私のコンピュータの旅は大学時代から始まったのですが、私はそこでプログラムを学ぶチャンスを逃がしませんでした。フェアチャイルド社では、ハードウェアを設計するチャンスをつかみ、コンピュータの設計を始めました。インテルに入社したことも、インテルで最初のフィールド・エンジニアとしてヨーロッパに赴任したことも、自分にとってチャンスだと思うことは全て前向きに向かって行きました。その後、2社の設立を手伝うチャンスに恵まれました。現在は、「クライアント・サーバー」と呼ばれるコンピュータの分野で別のチャンスをつかんでいます。私がコンピュータの分野に入ったのは、偶然の要素が大きかったのですが、私は以後ずっと、この急速に発展し拡大している分野に関わり続けてきました。今でも私の好奇心は衰えることを知らず、新しい製品、そして新しい企業を生み出すチャンスを探しています。

まとめ

本日ともに京都賞の栄誉を分かち合った素晴らしいエンジニアと共にインテルで働けたことは、私の生涯で非常に幸運な出来事でした。また、国際的なチームの一員としてヨーロッパで仕事をすることも素晴らしい思い出です。本日の喜びを、この席で息子、姉妹、そして妻と分かち合えますことも、大変幸せなことだと感謝しております。

稲盛博士、財団の皆様、素晴らしい賞をありがとうございました。最後に、私の人生哲学を披露して、本日の講演を終えたいと思います。

私の哲学

人生の意義について

Passion for what you're doing

As I review my life I recognize that I have had several great passions, almost compulsions. When I was sixteen, I saw a photograph of a one-man open helicopter. I was very interested and read all of the books I could about helicopters. I found a company that had printed plans for constructing a helicopter. I had a passion for building a helicopter and although I never did get to build one, I spent a great deal of time thinking and dreaming about helicopters.

When I first discovered computer programming I was “consumed” by the computer. I spent most of my time writing and improving my computer programs. You can find many people who have “been bitten by the computer bug.”

When I was working on writing my first book, I found that I could not get away from the keyboard and I found myself working on it with a passion. I feel very fortunate that in my life I have been very enthusiastic about my work, from designing computers to teaching engineers. In addition it has been very satisfying that this work has been productive and helped people. In the last few years I have become very interested in architecture of old buildings and I have a compulsion to visit old castles.

Take chances

I realize that where I live in Silicon Valley we have an opportunity to take chances with new companies and new products. For me some of these chances have been successful and some of the projects like the Symbol computer and silicon compiler were not successful. But I learned a lot in either case and my life is richer from these experiences. Three of the start-up companies I joined were successful.

When I started my computer Odyssey in college I took a chance and learned to program. At Fairchild I took a chance at designing hardware and became a computer designer. I took a chance when I joined Intel. I took another chance when I moved to Europe to become their first field engineer. I took a chance and helped start-up two software companies. Now I am taking a chance in an area called “client/server” computing. By accident I fell into the computer field and have stayed in this expanding and growing field. I still am inquisitive and take chances in new products and with new companies.

Summary

It was gratifying to have worked with these fine engineers and fellow laureates when we were together at Intel. It was great to have been part of an international team and to work in Europe. I am lucky to be surrounded by my sons, my sister and

人は、一瞬一瞬、人生に意味を与えながら生きているのです。

ですから、意味のある瞬間で人生を埋め尽くすことです。

あなたにとって何が大切で意味があるのか、それを決めるのはあなた自身です。

私にとって意味のある瞬間とは、

子供や妻と過ごす時間

創造力あふれる研究仲間と過ごす時間

記事や本を執筆している時間

新しい教育コースを考え出している時間

そして、外国を旅行している時間です。

本日はご列席の皆様とともに、ここ京都で大変意義のある時間を過ごすことが出来ました。ありがとうございました。

my wife as I share my story with you today.

Thank you Dr. Inamori and to the Foundation for this wonderful award. Finally I was specifically asked to tell you my philosophy.

My philosophy

What is the meaning of life?

You give your life meaning—moment by moment.

Fill your life with meaningful moments.

Only you can decide what is important and meaningful to you.

My meaningful moments have been:

the time I spent with my wife and children

the time I spent with my creative co-workers

the time I spent writing articles and books

the time I spent creating new courses

the time I spent traveling in other countries

Thank you for giving me these wonderful meaningful moments, the time I have spent in Kyoto with you.

稲盛財団 1997——第13回京都賞と助成金

発 行 1999 年 8 月 30 日

制 作 財団法人稲盛財団

京都市下京区四条通室町東入函谷鉾町88番地 〒600-8009

電話 (075) 255-2688

ISBN4-900663-13-1 C0000