SOFTWARE POLICIES FOR THE DEVELOPING WORLD

An International Development Research Centre Report

Russel M. Wills

Wescom Ltd.

September 1983

TABLE OF CONTENTS

TABLE OF CONTENTS
(continued)

PREFACE

This briefing on software policy for developing countries first describes prevalent types and levels of computer software in North America, the structure and evolution of the North American software industry, and world software markets.

It then analyzes two radically differing sets of software policy measures in "developing countries" -- that of Singapore, associated with export emphasis and extraordinary tax/fiscal concessions to the multinationals, and that of Brazil, involving a protectionist stance and reservations of market segments.

Finally, software and microelectronics policies for developing countries at differing stages of economic development are prescribed.

The two countries examined in the second part of the briefing -- Brazil and Singapore -- are at comparatively sophisticated stages of development, and it is apparent that sophisticated software/microelectronics policy measures are feasible only in countries possessing some degree of industrial development.

The author would like to thank Anthony Tillet and Christopher Smart (IDRC) for facilitating this work; Peter Booth and Steven

Globerman (Wescom Ltd.) for their policy expertise; and Juan Rada (IMI-Geneva) upon whose policy knowledge much of this work is based.

Special thanks are also due to the software firms of Canada and the U.S. who generously shared information and time.

1.0 COMPUTER SOFTWARE

Software, the instructions which enable computers to perform, may be thought of as comprising applications software, systems software and software tools which facilitate applications.

Systems software defines a computers' metabolism, i.e. how data enter and leave and how it moves about inside. Applications software, on the other hand, is always used to instruct a computer in performing a specific task -- e.g. assisting a classroom teacher in Botswanna, providing diagnostic assistance for eye diseases in Nepal, or measuring factors which will affect the size of crops in Nigeria.[1]

Such applications software may be thought of as comprising cross-industry products and industry-specific products, the latter involving chores which are unique to a specific group of businesses such as on-line banking.

1.1 Applications Tools

Applications tools are themselves packets of software which enable a programmer (a person who writes software) to easily write specific applications.

Using applications tools, a programmer with comparatively low level skills is able to describe at a computer terminal the general properties of the software program she wishes and then receive the partially written program.

In most classifications of computer software, tools for facilitating applications writing are usually considered as a part of systems software, but the actual productivity of programmers is increasing so slowly in North America and the programming needs are now so acute that many firms have emerged which specialize in these programming productivity tools.

The main U.S. and Canadian software firms are shown in Exhibits 1-1 and 1-2.

Typical of firms specializing in productivity tools is Quasar, the Canadian firm. Their program QUIZ, which can reduce the total time to produce a computer program for a specific application from several days to several hours, retails for $8,000. Such firms which specialize in productivity aids are showing major profit gains. Quasar 1980 sales of almost $6 million nearly doubled in 1981 to $11 million, and this firm has shown equivalent profit increases for 1982.

EXHIBIT 1-1
Leading U.S. Software Firms*

Leading Companies* Customized Software	1980 Sales From Custom Software (in millions)	Their Business
Computer Sciences	$337	A software pioneer with volatile history. Sells service primarily to NASA, the Navy and other government agencies. In February a court dismissed charges company had been over charging.
Planning Research	77	Provides computer services and other consulting in engineering and construction, energy management and urban planning. Earnings have been erratic.
CACI	35	Three-quarters of its sales come from computer analysis for the government of weapons systems, military logistics support and the nation's economy.
American Management Systems	30	A custom software and time-sharing company. An ambitious leap into standard software has dragged it into the red.
Logicon	27	Provides engineering and software for missile guidance and control and other strategic aeronautical systems, which accounts for two-thirds of its revenues.

* Excludes companies that derive a major part of their revenues from hardware sales, such as IBM and Burroughs.

Leading Companies* Packaged Software	1980 Sales From Packaged Software (in millions)	Their Business
Management Science America	$ 41	Sells software for personnel and financial management. Went public in April and in June acquired Peachtree Software, which provides financial management programs.
Wyly	27	Sells companies time-sharing and packaged software for financial and manufacturing management. Reported severe losses following unsuccessful effort to form data communications business.
Informatics	26	Evenly spread among packaged software and processing services. Seems to have recovered after string of losses in mid-70's.
Pansophic Systems	24	Sells data base management systems - software that runs on IBM computers and facilitates retrieval of data.
Cullinet	21	Offers data retrieval systems running on IBM computers. Is now branching out into applications programs for banking and manufacturing.

* Excludes companies that derive a major part of their revenues from hardware sales, such as IBM and Burroughs.

EXHIBIT 1-2

Canada's Leading Software and EDP Consulting Companies (2)

Company Name	% of Total 1980	% of Total 1979	EDP Revenues(1) 1980	EDP Revenues(1) 1979	% Growth '79-'80
1. Systemhouse Ltd.	32.7	36.0	19.3	14.3	35.0
2. DMR & Assoc. Ltd.	20.5	17.4	12.1	6.9	75.4
3. Systems Technology Services Ltd.	11.5	10.8	6.8	4.3	58.1
4. ACT Computer Services Ltd.	11.4	12.4	6.7	4.9	36.7
5. Quasar Systems Ltd.	9.6	9.3	5.7	3.7	54.1
6. Computech Consulting Can. Ltd.	4.2	5.0	2.5	2.0	25.0
7. ISS Information Systems Serv. Ltd.	4.2	3.6	2.5	1.4	78.6
8. Cincom Systems of Canada Ltd.	3.7	4.5	2.2	1.8	22.2
9. Art Benjamin & Assoc. Ltd. (3)	2.2	1.0	1.3	0.4	225.0
TOTAL			59.1	39.7	48.9

Notes:

(1) Figures are in $ Canadian millions.
(2) Includes those firms who derive the bulk of their revenues from EDP-related consulting and/or the sale/licensing of custom and/or "packaged" software.
(3) Now defunct.

1.2 Operating Systems

Operating systems, or systems software programs, were first in commercial use in the early 1960's. They were usually written by a series of persons and thus have undergone continuous maintenance and modification since that time. In addition to prescribing how data is moved in and out of computers, telecommunications lines and storage devices, operating systems also prescribe how work is scheduled and how access to computers serving several users at the same time is regulated.

A large amount of programmer time (approximately 80%) is given over to maintenance of systems software, and this maintenance time is usually considered unproductive. However, one of the main thrusts of the new branch of applications programming is to precisely streamline this maintenance.

Since systems software has had a convoluted evolution, it is a mark of professionalism for a computer salesman to be able to make the most inane modification in the design of a computer operating system resemble an excellent new feature. Also, new innovative techniques for the constructing of systems software has been almost impossible to implement, since such a staggering amount of money has already been invested in software to maintain the old system. Lewis Branscomb estimates, in fact, that more than $20 billion has been invested in the writing of systems software in the past three decades.[2]

Although the differences between big (so-called mainframe) computers, mini-computers and the new micro or personal computers are eroding, the personal computers (or perscoms, as the Japanese call them) have much simpler (and standardized) operating systems. An operating system for personal computers with single users called CP/M has now become the industry standard for most personal computers diffusing throughout North America, and given this standardization, companies have emerged producing software which enable almost all personal computers to communicate and message with each other over the phone lines, TV cable or virtually any communication linkage now in place in North America.

This is no trivial economic asset for business, and since more than a million personal computers were sold in Canada and the U.S. in 1981 (a 70% increase over 1980), mainly to small and medium sized businesses, within several years, any business which does not possess a communicating personal computer which can also serve word processing functions will be about as efficient as present businesses without telephones.

1.3 Writing Skill

Despite what has been said about the use of applications tools to streamline software writing and to make it more automatic, such writing remains a labour-intensive, haphazard process whose development time is almost never reliably predictable. Often estimates of total costs and development times are wrong by several magnitudes, and if one adds more people to a software writing project, it often increases development time rather than reducing it. Also, the skill requirements for writing software programs of any significant sophistication are quite strenuous, necessitating a detailed set of software and hardware skills which are acquired only through extensive training. For example, to incorporate chips into process equipment (mainly a software task) requires people who have knowledge of mechanical and electronics portions of the machinery, who have knowledge about the sensing interface between the mechanical and electronics portions of the product, and who can write software. In fact, the experience of the west thus far with labour displacements due to microelectronics developments has shown us that industrial retraining of workers displaced by this second wave of automation is difficult.[3] On the other hand, much software writing (namely business software) requires comparatively trivial skills which may be mastered by anyone with proficiency in high school mathematics, and this type of software writing will increasingly be done in the developing world.

In any case, the actual writing of a software program may be thought of as comprising the specification of the application, the design of the program, the actual writing of the software code, the testing of the program, and its modification and maintenance. These are all involved, detailed tasks, and several new operating systems have been devised, such as UNIX, to facilitate the writing of all of these steps on a single workstation. The main idea of these new comprehensive operating systems is to improve programmer productivity in both writing and maintaining computer software, and also to simplify the requirements for the design of a program so that its maintenance will be more free of errors. This is important because any error which is discovered in the design stage of a program is usually several orders of magnitude less expensive to delete than an error which is discovered only at the production stage.

Software writing involves a blend of creative insight and logical tedium and is being written by persons working at home on personal computers (or perscoms, as the Japanese call them) and in multi-national corporations.

Software is simultaneously a product and a service, a situation so strange that North American governments simply do not know how to tax it yet. The Canadian government, for example, requires that costs of software development be capitalized and considers software as a taxable asset. Also, when software is imported into Canada on magnetic tapes, Canadian customs consider the customs duties on the software to be equivalent to the costs of the magnetic tapes. However, North American governments are learning very rapidly how to tax software, since this industry is quickly becoming larger than the computer hardware which it directs.

1.4 Industry Evolution

The infant software industry, in fact, dates only from 1969, when IBM first unbundled and separately sold software as a distinct product from computers. By 1981, software sales in the U.S. comprised $13.1 billion, approximately half the size of computer sales.[4] The International Data Corporation of Massachusetts estimates that by mid-decade the software industry will total sales of more than $33.8 billion, which is approximately 60% of the size of the computer and peripherals industry itself. However, as we shall see, the figures are actually much larger since many computer users are now doing their own programming.

Software creation and distribution is almost impossible to understand in neo-classical economic terms, which evolved to deal with the production, reproduction and distribution of material commodities and not with information. But since software is essentially embedded information, it is difficult to attach a value to it. A computer program of software instructions may be distributed on a cassette, a magnetic tape or it may be shot over a packet-switched satellite or piggybacked onto an FM radio signal for distribution to home or business users. But the software's value is not equivalent to the value of the tape or cassette in which it is embedded. The value of the software is equivalent to what it can accomplish.

Thus, before the beginning of the seventies, virtually all computer manufacturers freely dispensed software along with computers. In the late sixties, however, computers cost virtually millions of dollars and were utilized almost entirely by large businesses and the government, with a few installations at universities. Today, perscom manufacturers ranging from Sinclair, Commodore and Apple Computing make products retailing for a few hundred dollars, and the costs of writing software has astronomically risen compared both to the relative price of computers themselves and in absolute terms. In fact, it is now generally conceded that if one incoporates a microprocessor into a new product line that 70%-80% of the total costs are accounted for by software. This situation, of course, is a real contrast to the early days of computing in which the ratio of software to hardware costs was approximately one to four. Today there has occurred a virtual reversal of this proportion. Illustrative of this reversal is a recently published list of U.S. Army contracts for automating administrative support in its military bases throughout the world. In previous years, bidders for such contracts would have exclusively comprised hardware suppliers; however, the predominant bidders in this project were Computer Sciences Corporation and Electronic Data Systems, i.e. software producers which are using hardware manufacturers as subcontractors.[5] Also, by 1980, software costs comprised more than half of the research and development expenditures of manufacturers of semi-conductors and may rise to 80% by 1985.[6]

By mid-1982 most computer manufacturers in the world have been spending more than 55% of their budgets for research and development on software, and given a widespread lack of competent programmers and a massive number of new computer applications, a definite trend is occurring in the software industry toward the production of software packets which are prewritten and sold to a wide variety of customers. This trend is also furthered by the growth in the past couple of years in the number of perscoms for small businesses, education and industrial training. Since many of the end users of these perscoms do not know how to write software, an entirely new business has come into existence of writing packaged software for perscoms, which is sold in blister packs (like those for audio cassettes) in computer retail stores. These software packets may easily transform a perscom into a word processor, an electronic mail terminal, a learning tool through which young children may acquire the habits of procedural thinking, or a workstation for executives.

1.5 Custom, Packaged and Embedded Software

One may think of software as being comprised of three types:

1. Custom software, which is written from scratch and is custom-designed to fulfill a unique need of the end user of the software.

2. Pre-written software packages which may be adapted to the users' needs.

3. Firmware, which is software embedded into the circuitry of Read Only Memory Chips (ROM's) and incorporated into products.

Although at the beginning of 1983 most of the software in the world is of the first type, the industrial trend toward the production of pre-written software packets is accelerating for simple economic reasons. It may cost any firm several hundred thousand dollars to implement a solution via custom software. But if a package is available for the same purpose, it may cost only a few thousand dollars.

Toward the end of 1981 the Canadian Intergovernmental Task Force on Transborder Data Flows rendered the following picture of the Canadian software industry (Exhibit 1-3).

EXHIBIT 1-3

Canadian Software Expenditures

Expenditures	($ million) 1975	1980	1985	1990
Software Development	440	830	1290	1460
Annual Maintenance	480	1070	1690	2180
Rented Packages	15	40	80	120
Service Imports	90	330	950	1620
Total Software Expenditures	1025	2270	4010	5380

* "Interdepartmental Task Force on Transborder Data Flow, Report on Software-Related Issues," (draft) 1982.

Unfortunately such figures exclude all costs for embedded software incorporated into products such as Videotex terminals, word processors, private automatic branch exchanges and virtually all other products one might think of which incorporate software hardwired onto ROM's. Thus far it has been economic orthodoxy to only include in such tallies costs for applications and custom software plus the software costs of time-sharing computer firms (those which sell computer services to other firms). These figures also exclude persons writing their own software for perscoms and all educational software, but in fact the diffusion of cheap perscoms to professionals and small businesses has meant that there is a vastly increased demand for perscom software packages which can perform virtually any type of business function such as financial planning and accounting. Small businesses simply can not afford to hire software writers and must use pre-written software packages.

In spite of all that has been said about the growth of the software industry in North America, it is really just an infant industry which is now only a decade old. According to one marketing firm,[7] of the more than 4,000 computer software firms in the U.S. in 1980, only 147 had sales in excess of $10 million, and of these ten, seven were computer manufacturers.

Although it is anticipated that pre-written software packages will be the largest software growth product in the future, in

1981 packages comprised only $2.9 billion in U.S. sales. Even so, this portion of the software market is now growing at more than 36% per annum, and as a consequence many firms which specialize in customized programs written at large costs from scratch are going bankrupt because their former customers have switched to pre-written software packages which cost several orders of magnitude less than the custom software.

But also with this incindiary growth of software packets, there is now a superfluous production in North America of many varieties of the same product, e.g. by mid-1982 there were more than 1,200 pre-written packaged programs for accounting alone, and as a consequence the packaged software industry is now undergoing waves of consolidation in which the smaller innovative firms are being snatched up by the large hardware manufacturers or by large software and time sharing houses. Although in the first six months of 1981 there were 57 U.S. acquisitions of packaged software firms, in the first six months of 1982 there were more than 100 such consolidations. (Both of these sets of figures must be compared to 1980, during which there were only 42 consolidations for the entire year.)

Many software firms are also purchasing rights to distribute and sell software, similarly to book publishing. Many persons believe that the software industry may eventually resemble book publishing in North America, with a few large houses controlling specialized markets.

1.6 Occupational Needs

In a previous work[8] the author examined world software needs. Although the world ability to create new software is increasing by only around 20% yearly, the demand for new applications and the increased maintenance requirements are cumulatively growing at more than triple this rate.

In terms of specific software occupational categories, requirements involve:

1. Sensing designers -- people who can write software and also know about sensing devices. (This work necessitates the understanding of problems which occur at the interface between the microelectronics and mechanical parts of a product or manufacturing process.)

2. Software design engineers for computer-aided design and processing -- requiring persons who are knowledgable about both software and hardware. (The much heralded CAD/CAM is yet mostly design but not manufacturing because there is simply not software available which enables a computer to interconnect with design modules and drive a plant's machinery.)

3. Educational/industrial retraining software designers

to create the software for computer-aided learning/videodisc technologies, to produce actual educational and training content and to produce the (later discussed) expert system programs which can codify and replicate interpretative thought of highly specialized experts in technical fields.

4. Applications engineers who incorporate microelectronics into products.

5. Business software writers who produce software packets for business and governmental needs.

6. Process software engineers who produce chips.

In addition, given the above bottlenecks, there are major world needs in virtually all areas for applications tools, software which facilitates and streamlines the actual writing of applications themselves.

1.7 Telesoftware

Since software is merely information, it may be transmitted over the phone lines, packet-switched satellite channels, the coaxial cable TV links, radio, or any combination of these. With "telesoftware", computer software is "downloaded" into computers for home or business use. A new national corporation was formed in late 1982 in the U.S. to transmit software on FM radio to owners of personal computers. INC telecommunications is planning to deliver software digitally by satellite to 220 public radio stations which will then insert it into their normal broadcast signal for transmission to home users. This is essentially the first mass distribution channel for independent home computer programmers, and the fledgling telesoftware industry in North America until recently has involved merely a handful of firms such as Compuserve and The Source.

1.8 Software Languages

Software writing is such an arduous and lengthy task because a computer must be instructed in the most exasperating detail exactly what to do. Computers are also quite stupid, and the only language which they understand is the digital language of 1's and o's.[9]

The first and most primitive computer language devised was called machine (or digital) language and involved writing computer instructions directly in this code of 1's and o's. But the main problem with writing in machine language was that the only persons who could do it were persons who had an extensive knowledge of electronic circuitry. This language was soon replaced by one called assembly language, which more resembled English. (Programmers working in assembly could utilize algebraic notations.

These notations were then translated into machine language the computer could understand by an additional program called a compiler.) However, assembly language was tedious to use; it was rather analogous to instructing someone how to brush their teeth by saying, "Add a bith of toothpaste to the top of the toothbrush. Move the brush briskly up and down over the surface of the teeth, and so forth." It would be more efficient to simply say, "Brush your teeth," and toward the beginning of the sixties, so-called higher level languages evolved in which a single line of software code could replace dozens of commands in assembly languages. Some of these higher level languages were called APL, Pascal and Fortran. But even to use the higher level languages, a programmer had to be familiar with abbreviations and symbols for hundreds of computer instructions and also with the rules for their use and combination. Today however, non-procedural languages are being rapidly developed which eliminate the necessity for such knowledge. Non-procedural languages, examined in the section on artificial intelligence, enable people who know nothing about computers to program and communicate with them, e.g. to search through corporate data bases. (Although any language besides machine language is commonly referred to as "higher level language", the higher the level of a computer language, the more it resembles natural languages spoken by humans.)

Non-procedural languages are still typed into keyboards, but possibly the most startling future development in microelectronics will involve advances in speech recognition which will enable rapid oral communication with computers.

Already there are thousands of computer languages in use, including languages for children to learn logical thinking such as Logo, business languages such as Cobol, languages for educators to write courses in, specialized languages for banking, architecture and engineering, and even limited oral speech languages which enable one to verbally program a perscom by 60 words or less.

Since computers have differing internal structures in the ways they are wired up, they also have differing and incompatible machine languages, and consequently one higher level language for a specific application on one brand of computer will not work on other brands. This situation is widely recognized as a major design flaw in the evolution of computers.

A computer, of course, understands only its "native machine language". The first higher level languages were supposed to be machine independent, with each computer manufacturer supplying a translator which would translate a higher level language into the computer's native machine language. However, since computers eventually became vehicles to sell software, this machine independence simply did not occur.

A second major design fault in early programming involved the fact that since computer time was extremely expensive then, the best programs were those which were most compactly written. But programs written in a gnarled manner could only be modified, maintained or unravelled by the original person who wrote them because any slight change in one part of the program would propagage major changes throughout the rest of it. In fact, it was usually only the original writer who could even understand what an original program was attempting to do.

Since many of the programming tasks of the early sixties, such as those in the aerospace industry, involved huge interconnected software programs, this manner of writing compact, gnarled software was eliminated, and stages of any program were modularized so that parts of a program corresponded to clearly specified functions and ended with well defined and specified relations between these parts.

A third design fault involved the fact that most software written in the first decade of the computer industry had virtually no documentation concerning what it was attempting to do and how it accomplished its tasks. Most computer programmers worked in solitude, and the way that one wrote a program would be entirely different from the way another solved the same problem. Thus, for close to a decade, with most software in the world, there was either no documentation or misleading, incomplete and incomprehensible documentation which did not do much good.

1.9 Database Management

One of the most important applications of business software is the management of very large databases. In business the exact relationships between the data elements in any computerized file represent the specific business process which is being modelled and managed, but because the relations between individual data elements are often changed, much of business programming is comprised of the management of data.

Database management is so central now that software for this purpose is the main product of many North American firms. Cullinet, for example, which was the first North American firm to sell only software products, makes a database management package for IBM computers. But since the main reason for buying a database management system is to more quickly and easily write new applications, many manufacturers are now incorporating these as embedded software, e.g. VOS/3 of Hitachi and System/38 of IBM. Probably within a few years most software for the management of databases will be sold along with hardware as simply another operating system.

1.10 The Productivity of Programming

What slim gains have occurred in the productivity of software writing are rather trivial compared to those gains in hardware capabilities, but although we normally make direct comparisons between the productivity changes in hardware and software, such direct comparisons are inappropriate because hardware and software possess totally different properties.[10]

First of all, the "logical structure" of software is much more complex than the structure of hardware. Thus, in a typically sized business computer, the software may be involved in the correct logical ordering and placing of some 10^{12} bits of information. This figure is several orders of magnitude greater than the total number of logical circuits in the central processing unit of a large business computer.

Secondly, computer hardware is replaced periodically with improved technology. The actual obsolescence period of many items incorporating chips is quite short, ranging from six months to two years. Software, on the other hand, has simply accumulated as it was written, continuously modified and rewritten (usually in totally undocumented form). Also, many of the most significant business applications for large mainframe computers were written in manners which are very difficult to maintain.

It has often been estimated that replacing this body of poorly written software would cost North American business more than $50 billion, and it is thus easy to see why most programming time is still occupied with maintaining this chaotic software quilt. Most programmers, in fact, spend more than 80% of their total programming time maintaining and modifying existing software rather than creating new software.

Thirdly, the effects of software's logic are intimately tied in, we have seen, with any user's work patterns. Most of the early business software pertained to office tasks which had already been systematized, such as scheduling, inventory and accounts receivable. Thus, virtually all of the easy business software applications have already been written. The writing of new software involves modelling of office tasks and procedures which did not previously exist. Accordingly, "Much of the work associated with designing the software is in reality systematizing and designing the task that the user requires. This must be done before the software can be designed."[11]

Finally, as noted, software has become considerably more expensive than components. It is thought that the cost of providing a single line of software is around $10-$12, which is more expensive than a mass-produced chip.

The actual change in programming productivity over the past two decades, then, is insignificant, and computer programming today remains basically an art rather than science.

It has been estimated that the productivity growth in writing systems software over the past decade has been only around 10% annually. This productivity is mainly accounted for by interactive computer tools which aid one in designing systems software. However, most systems programming was first done in machine or assembly language.

> "This was fine as long as computer architectures based on the discrete electronic components of 15 years ago evolved in orderly fashion. Much of the earlier programming could continue to be used. Today, however, with integrated circuits and microprocessors collapsing the boundaries of a system onto a chip, new system structures and architectures pose a dilemma. One can either take advantage of the new technology and throw out the old software, or constrain the technology by making it run the old code. The answer to this Hobson's choice is provided by optimizing compiler technology. If the system programming is written in a high level language, it can be compiled to machine instructions that are only slightly less efficient than ones coded manually. Now, when new electronics technology dictates a change in architecture, it is only necessary to change the piece of the compiler that generates code for the target machine and recompile."[12]

One new productivity trend is illustrated by VISICALC of the Arlington, Mass. firm, Software Arts Inc. This software package is a graphic representation of a worksheet which can be quickly modified for specific tasks. One may ask "what if" questions interactively with VISICALC by changing a single figure on the worksheet, and all other figures are automatically changed.

1.11 Software Portability

Recently, venture capital has been flowing into software for the management of data bases, CAD/CAM, business graphics, and for any business applications for perscoms. A unique approach to software portability is that of Softech, Microsystems of Massachussets, which was founded in 1979 when it received the exclusive licence for the UCSD-P system (a portable version of a computer language called Pascal) which was invented at the University of California. There are several approaches to making software portable, i.e. making software run on differing makes of microcomputers. One involves:

> "... carefully designing it to reduce the amount that will have to be rewritten each time the program is moved to a new machine. Another approach is to emphasize the language used to write the program, assuming that each machine can then be made to translate these programs to the appropriate machine codes. Unfortunately ... any sizable applications program will use resources of the operating system, particularly input and output, in ways that may be unique to the hardware for which it was written. Thus, portions of the program may need to be rewritten if it is to be moved to a different system."[13]

1.12 Personal Computer Software

Probably a main reason for the increased growth of the North American software industry is the rapid spreading of personal computers to users who cannot write their own software. The number of computers in North America, in fact, is increasing much more rapidly than the number of competent software writers.

Until around mid-1981, perscoms were manufactured almost entirely by electronic firms such as the Tandy Corporation and firms which manufactured only perscoms, such as Apple. Subsequently, however, large corporations began to order perscoms in batches for their managers and engineers, and larger manufacturers entered the market. By mid-1981, the Xerox Corporation became the first major manufacturer of office equipment to enter the perscom market, soon followed by IBM.

The personal computer market is anticipated to grow by 75% during 1983. Annual sales growth of 300% is also predicted during 1983 for business microcomputers.[14]

During 1982 the Canadian microcomputer and peripheral market increased in value 75% to $200 million. More than half of all personal computers sold in Canada are used for word processing, and these perscoms may cost a small proportion of the cost of computers which only do word processing. According to Wescom estimates, in January 1983, there were 58,000 microcomputers in use in Canada -- including perscoms plus other types of micros.

Advanced Resources Development, the Massachusetts consulting firm, has predicted that the demand for business software packages will enable 16 bit microcomputers to capture a major market share of desk-top office computers during the next half decade, assuming that this type of processor will comprise almost half the volume of North American shipments by 1986. Since CP/M became a standard operating system for perscoms, the product life of 8 bit-based perscoms has been artificially extended by the massive number of applications packages which were developed for them -- comprising more than 5,000 software packages for business financial planning, accounting, word processing, and so forth.

The next generation of perscoms will have a vastly increased processing power, and will enable the usage of applications software packages which actually are "user-friendly" (almost no software presently is), and for graphics capabilities. Advanced Resources Development reports that sales of perscoms will exceed $2 billion by 1986, with more than 7 million shipped between 1982 and 1986.[15]

Canadian perscom entries include Hyperion, which is IBM compatible (i.e. will run IBM software), and the Rainbow 100 of Digital Equipment of Canada Ltd. of Kanata, Ontario. The Rainbow 100 has an 8 bit and 16 bit processor chip. It thus spans both generations of perscom software.

In the first half of 1983, increased competition is expected also in perscoms retailing between $700 and $1,000 U.S. (There are desk top computers presently on the market priced in the range of

$100 to $400, but these computers can do very little beyond games and are intended for home entertainment.) Even so, the President of the Tandy Corporation predicts that, "A whole new market segment seems to be evolving in the $300 to $1,000 range."[16]

It is also rumoured that IBM is about to unveil a new machine which will be priced in this range. It will comprise the computer and a keyboard which plugs into a TV. This cheap IBM perscom will not be marketed through the IBM product centres, but it is anticipated that it will be marketed through large retail organizations such as Sears Roebuck and Co.

The personal computer market is being driven by several factors, including growth of computer hobbyists and the decline of computer service organizations.

Time-sharing business computer services enables firms to use large, high-cost, external main-frame computers. These services are utilized by firms which can't afford their own in-house computer or, in those cases where in-house computing can be very extensive. However, with the cheap perscoms invading small businesses, this advantage is rapidly deteriorating.

Many analysts have noted that unfortunately the makers of perscoms are building software that cannot communicate with products of other manufacturers. Thus, much money is being put into software that is not portable from one system to another.[17] However, several firms are working on emulator perscoms -- those that will run the competition's software, and many perscoms are IBM-compatible.

The main reasons for the rapid diffusion of perscoms are costs, utility and the increased availability of business software. Examples of the latter include the Visi Corporation in the U.S., which produces the highly successful Visi series of packages such as VISICALC and VISITREND. Others include Peachtree, which has developed software for IBM's perscoms, and companies such as Microsoft and J.D. Edwards which specialize in accounting and financial packages.

Xerox, IBM and Apple have been concentrating on user-friendly machines which are a blend of executive workstations and perscoms. These machines operate almost entirely by pushing a pointer on the screen to a spot presenting a pictorial representation of an office function (such as filing), and hitting one of several commands. But such machines require extremely sophisticated and elaborate software.

One of the reasons that the population of computers in North America is increasing faster than the number of software writers is that every new mainframe computer installed necessitates 100 to 500 programs in a business context. Such a "software multiplier effect", as it is named, is manifested in the fact that the average beginning wages for software programmers in North America is now around $20,000 annually, a figure much higher than other professions.

1.13 North American Market Trends

For any firms attempting to enter the North American software market in established fields, it is enormously expensive given:

1. The extremely sophisticated product differentiation which the software has undergone. In the early days of computing, with the practice of time-sharing, firms would sell nothing but time on their own computer. But after several time-sharing firms offered such services at low prices, other firms began selling time plus highly specialized software. When competitive firms began offering this same specialized software, the former firms started offering interactive capabilities in additional to specialized software and time sharing.

2. There is a real trend toward vertical markets in which the firm selling software attempts to satisfy all of the software needs for a single industry. For example, the medical software firm -- Shared Medical

Systems - can essentially meet all computing needs in the U.S. health industry through software packages.

3. Finally, there is a related trend today toward offering not merely individual software services, but software packages which may serve a diversity of business needs. For example, today there are single software packages which may compute accounts payable, general ledger and payroll as part of a single package, and also may enable the user to adapt this package to the specific structure of their firm's needs.

However, writing packaged software is very expensive, and companies which do not have considerable financial resources cannot commit funds to any speculative software development packages. Until very recently, financing in Canada has been very limited for both custom and packaged software due to the failure of financial institutions to view software as a "real" manufactured product. In the U.S. the semi-conductor industry financed itself by becoming big very rapidly. This industry has never been able to pay for its next generation of development, and innovative firms have consistently had to acquiesce takeovers (Exhibit 4) to pay for capital costs. The same thing will eventually happen with software firms, with several multi-nationals dominating North American production.

EXHIBIT 1-4

Acquisitions of U.S. Semiconductor Firms*

U.S. Firm	Est. 1979 I.C. Sales (U.S. $ Millions)	Affiliation
Texas Instruments	680	Division of TI Inc.
Motorola	425	Division of Motorola Inc.
Intel	400	
National Semiconductor	320	
Fairchild	305	Acquired by Schlumberger
Signetics	250	Acquired by N.V. Philips
Advanced Micro Devices	160	Owned by Siemens (20%)
Mostek	155	Acquired by United Technologies
RCA	145	Division of RCA
Intersil	140	Acquired by General Electric
Harris	100	Division of Harris Corporation
American Microsystems	95	Acquired by R. Bosch GmbH (25%)
Rockwell Semiconductors	85	Division of Rockwell International
General Instruments	80	Division of G.I. Inc.
Synertek	50	Acquired by Honeywell
Analog Devices	40	Acquired by Standard Oil
Monolithic Memories	35	Acquired by Northern Telecom
Siliconix	30	Acquired by Lucas Industries
Solid State Scientific	22	Aquired by Adolf Schindling
Zilog	15	Acquired by Exxon

* Source: Juan Rada, "The Impact of Microelectronics and Information Technology: Case Studies in Latin America (Part III)", UNESCO, December 1981, p.11.

With respect to taxation, Canadian software and hardware firms operate at a tax disadvantage since they are subjected to additional taxation on imports of foreign equipment and components. Taxation rules in Canada necessitate that software development costs be capitalized and that software be treated as a taxable asset.

Government financial and fiscal programs to aid software industries have been insignificant and have mainly been limited to assimilating software to research and development benefits, which themselves in Canada are insignificant compared to R&D support measures in several countries.

1.14 The World Software Market

A prediction of the world market for software was provided in mid-1982 by the International Data Corporation.[18] Predictions for sales involving applications software, utilities software and systems software are presented in Exhibits 1-5 through 1-7. Utilities software is understood as comprised of "software performing program design, supporting conversion, sort/slash, mail functions and accounting, plus all data management software (including dictionaries, modelling tools and packages for word processing)." This market prediction also noted the following trends:

EXHIBIT 1-5

Worldwide Revenues (in millions of dollars) of Stand-Alone Packaged Software Suppliers, 1980-1986*

Stand-Alone Packaged Software	1980	%	1981	%	1982	%
By Software Type						
Hardware Manufacturers:						
Systems	$ 895	29	$1,155	31	$1,515	33
Utility	630	28	805	29	1,040	32
Application	195	21	235	21	285	21
Sub-Total	1,720	28	2,195	29	2,840	31
Independents:						
Systems	145	35	195	36	265	36
Utility	420	44	605	45	880	44
Application	490	43	700	44	1,010	42
Sub-Total	1,055	42	1,500	44	2,155	42
By Hardware Type						
Hardware Manufacturers:						
Mainframe	1,315	27	1,670	29	2,150	31
Minicomputer	125	27	160	28	205	29
SBC**	235	28	300	30	390	33
Desktop	45	44	65	46	95	47
Subtotal	1,720	28	2,195	29	2,840	31
Independents:						
Mainframe	770	38	1,060	39	1,470	37
Mini/SBC	225	49	335	51	505	49
Desktop	60	75	105	71	180	64
Subtotal	1,055	42	1,500	44	2,155	42
Total Stand-Alone Packaged Software	2,775	33	3,695	35	4,995	36

* Source: IDC
** Small Business Computers

EXHIBIT 1-5

Worldwide Revenues (in millions of dollars) of Stand-Alone Packaged Software Suppliers, 1980-1986 (continued)

Stand-Alone Packaged Software	1983	%	1984	%	1985	%	1986
By Software Type							
Hardware Manufacturers:							
Systems	$2,010	33	$2,670	31	$3,505	29	$4,530
Utility	1,375	34	1,845	36	2,515	37	3,435
Application	345	22	420	21	510	20	610
Sub-Total	3,730	32	4,935	32	6,530	31	8,575
Independents:							
Systems	360	32	475	28	610	23	750
Utility	1,265	40	1,765	35	2,390	32	3,160
Application	1,435	40	2,010	37	2,755	34	3,690
Sub-Total	3,060	39	4,250	35	5,755	32	7,600
By Hardware Type							
Hardware Manufacturers:							
Mainframe	2,805	31	3,680	30	4,800	29	6,190
Minicomputer	265	30	345	32	455	32	600
SBC	520	35	700	36	955	36	1,300
Desktop	140	50	210	52	320	52	485
Sub-Total	3,730	32	4,935	32	6,530	31	8,575
Independents:							
Mainframe	2,015	34	2,690	30	3,485	26	4,375
Mini/SBC	750	48	1,110	44	1,600	41	2,250
Desktop	295	53	450	49	670	46	975
Sub-Total	3,060	39	4,250	35	5,755	32	7,600
Total Stand-Alone Packaged Software	6,790	35	9,185	34	12,285	32	16,175

EXHIBIT 1-6

Independent System Software Revenue Size (in millions of dollars) and Growth, 1980-1986*

System Software	1980	%	1981	%	1982	%
Operating Systems	$ 17	65	$ 28	64	$ 46	59
Compilers/ Assemblers	17	75	30	67	50	58
Performance Measurement	22	23	27	22	33	21
Communications	31	23	38	26	48	29
System Resource Management	58	24	72	22	88	21
TOTAL	$145	35	$195	36	$268	36

* Source: IDC

EXHIBIT 1-6

Independent System Software Revenue Size (in millions of dollars) and Growth, 1980-1986 (continued)

System Software	1983	%	1984	%	1985	%	1986
Operating Systems	$ 73	49	$109	40	$153	29	$198
Compilers/ Assemblers	79	46	115	37	158	26	199
Performance Measurements	40	18	47	17	55	16	64
Communications	62	27	79	27	100	26	126
System Resource Management	106	18	125	15	144	13	163
TOTAL	$360	32	$475	28	$610	23	$750

EXHIBIT 1-7

Independent Utility Software Size (in millions of dollars) and Growth, 1980-1986*

Utility Software	1980	%	1981	%	1982	%
Program Support and Security	$ 39	28	$ 50	26	$ 63	24
Program Design	50	30	65	32	86	35
Conversion Aids	5	20	6	17	7	29
Sort/Merge	18	22	22	23	27	26
Job Accounting	17	29	22	23	27	22
Data Management	205	46	300	47	440	41
Report Generators	48	56	75	64	115	61
Graphics	38	71	65	77	115	65
TOTAL	$420	44	$605	45	$880	44

*Source: IDC

EXHIBIT 1-7

Independent Utility Software Size (in millions of dollars) and Growth, 1980-1986 (continued)

Utility Software	1983	%	1984	%	1985	%	1986
Program Support and Security	$ 78	24	$ 98	23	$ 120	22	$ 146
Program Design	116	35	156	37	213	42	302
Conversion Aids	9	33	12	42	17	59	27
Sort/Merge	34	27	43	26	54	26	68
Job Accounting	33	24	41	24	51	22	62
Data Management	620	37	850	35	1,150	32	1,520
Report Generators	185	49	275	35	370	28	475
Graphics	190	53	290	43	415	35	560
TOTAL	$1,265	40	$1,765	35	$2,390	32	$3,160

1. The growth of system software will be accelerated by developments in language compilers, more sophisticated operating systems and in communications software.

2. There is a trend toward operating systems for personal computers such as Unix of Bell Laboratories and Digital Research Incorporated's CP/M, which are portable.

3. It is also reported that the UCSD-P system of Softech Microsystems has more than 30,000 users in the U.S. in mid-1982. Programmers using this system may move their applications from one perscom to another without modifying the software, because the machine translates automatically any language in which the program is written into the machine unique code.

Both the U.S. and Canada are importing programmers from countries such as Hong Kong, India and South Africa with large numbers of English-speaking engineers and programmers. However, by the end of this decade there will be a growth in export of both software engineers and software to the developing countries. Firms which survive takeovers will be supplying a world market with a considerable amount of software maintenance and consulting. For close to five years the Japanese have been using teams of software writers in places such as Korea and Hong Kong, but writing specifications for software projects from a distance over telecommunication lines is almost impossible.

1.15 Computer Telecommunications Systems

Software is presently the main impediment to the further blending of communication systems and computers in North America. Many firms have created computer communication networks for visual images, voice and data -- varying from so-called local area business cable networks to packet-switched satellite networks. However, all of these computer communication networks operate according to different and usually incompatible standards and protocols. Information is often stored in differing formats and is retrievable only by differing methods of access. All of these are software problems.

Sales opportunities for world software markets -- from microcomputer and office automation software to artificial intelligence software -- are presented in Exhibit 1-8.

Of particular relevance to new world market opportunities is the invention of algorithms to allow computer processing of information in the form of Chinese ideograms.

EXHIBIT 1-8

Software Opportunities

The world software industry has dozens of markets, many worth billions of dollars a year, such as the systems software market for mainframes dominated by IBM. Following are 11 of the most promising markets:*

1. **Industry-Specific Software and Turnkey Systems**

   Vertical markets provide one of the simplest and most effective businesses for a software designer: Just write a generalized program that can be used by many companies in the same industry and market it directly to computer users in the field, or package it with hardware and sell it as a "turnkey" system. Every business is becoming computerized, and each has a very different way of doing business that software must address. INPUT, the Palo Alto research house, estimates the market for turnkey systems will grow at an annual rate of 36% to $25 billion in 1985 from $5.5 billion in 1980.

2. **Microcomputer Software**

   This sleeping giant accounted for only about $437.5 million sales in 1980 but it has already taken off and could approach $25 billion by 1990, says Greenwich, Conn.-based International Resource Development Inc., a research firm. The market includes business applications and systems software as well as other uses in the home and elsewhere.

3. **Standard Software Products for Mainframes**

   The market for programs for existing and new mainframe users is growing by about 35% a year, while modernized versions of old programs as well as periodic updates that incorporate changes in laws and regulations will also find an expanding market. Says one observer, "Many products on the market today are 10 years old in concept and five years old in implementation."

* Venture, January 1982, and R. Wills, <u>Softech</u>, op. cit.

EXHIBIT 1-8

Software Opportunities (continued)

4. **Professional Services and Consulting**

   Although often scuttled as a labor-intensive industry, the professional software services market grew at a healthy 25% in 1980 to $3.5 billion in revenues, says INPUT, and it maintains an average pretax profit margin on a par with most software products companies (about 10%). For the individual freelance programmer, hired on a project basis, earnings can double the salary of an EDP staff employee.

5. **Data-Base Management System**

   This type of systems software organizes data in line with the special needs of business, makes it easier to gain access to and update files. As cheaper hardware encourages the trend toward decentralization of data storage and manipulation, more and more analysis of that data is required by more and more people. Opportunities exist in mainframes and larger minicomputers.

6. **Office Automation**

   This sweeping market encompasses software to perform a variety of office chores -- word processing, electronic mail, teleconferencing -- while integrating them with a company's network.

7. **Application Generators**

   As the shortage of qualified program designers persists, the demand will increase for tools that let users write applications programs themselves. For instance, a standard general ledger application generator would be formatted to a point, then provide the ability for a non-EDP bookkeeper to customize the program to the specifications of a particular company. The thundering growth of microcomputers is likely to spur the demand for applications generators in that range first.

EXHIBIT 1-8

Software Opportunities (continued)

8. **Speech Recognition**

   Although the technology has yet to be refined for the broadest potential use, software that activates a computer by the sound of the human voice should turn into a stunning market in the 1980's. Speech recognition applications will be used for just about any function a human now performs: inquiries into data bases, bill-paying via electronic fund transfers, even typing letters from dictates messages. The industry is still in embryonic stages, but SRI International, the Menlo Park, Calif., research firm, has projected the market for such products will grow to between $151 million and $192 million by 1985 and to $1 billion by 1990.

9. **Educational Software**

   With the cost of microcomputers and network configurations coming down, a major market is developing in software that could virtually replace teachers. A few companies already have such products on the market. But if educational software maintains the same proportion of current sales for all microsoftware (10% of $140 million in 1980, says INPUT), it could blossom into a $250 million market by 1986.

10. **Computer Graphics**

    These packages, which produce graphic portrayals of alphanumeric data onto terminals, paper or transparencies, are designed to help two users -- business people in need of visual aids such as bar charts and graphs, and engineers and designers needing computer-aided design and computer-aided manufacturing (CAD/CAM) tools. Gnostic Concepts forecasts the relatively small 1981 market of $250 million for business graphics terminal equipment and software services will grow at an average annual rate of better than 50% over the next few years. The current CAD/CAM market of $740 million, typically sold now with software prices bundled with the hardware, should increase on the average of more than 35% per year.

EXHIBIT 1-8

**Software Opportunities
(continued)**

11. **Artificial Intelligence**

Tohru Moto-Oka, head of Japan's fifth generation computer project, predicts that within the decade AI software will dominate programming, and AI products ranging from export systems to robotics sensors and applications generators will grow to $50 billion by 1995.

Until fairly recently, for example, it was virtually impossible to do word processing in Chinese script at a reasonable rate, but a new breakthrough known as Sinotronic CS-4000 has recently made possible word processing in ideographic languages such as Chinese, Japanese and Korean.

This breakthrough also obviates the western prejudice that eastern languages are not amenable to text processing over telecommunications lines or to computer input and output.

CS-4000 was invented by a physicist, Lo Shui-Yin,[18] who discovered that virtually all Chinese characters could be thought of as comprising five basic strokes -- a horizontal line, a dot, a vertical line, a left-sloping line and a right-sloping line.

These strokes are first digitized, represented by 1, 2, 3, 4 and 5 and then processed by a computer.

(There have previously been other methods of inputting Chinese characters into computers, but these are very slow compared to CS-4000.)

Dr. Lo's system is equipped with a normally-sized keyboard whose keys are arranged according to frequency of use; to increase typing speed, Dr. Lo has designed 31 of the most frequently used stroke combinations (characters plus radicals) on the keyboard:

> "By using composite keys, the average key strokes for a Chinese character is reduced to 3.7. This number of strokes is fewer than that required for typing an English word, which needs about five keys to make up a word."[19]

This word processor received the name CS-4000 because it can store 4000 frequently used Chinese characters, comprising 99.7% of the Chinese vocabulary.

This stroke method is also applicable to library information retrieval, indexing, telecommunications and telex, in addition to word processing.

The CS-4000 system, which presently costs around $15,000, has already been tested for transmitting mail between Nanjing and

Beijing, and is being manufactured for general dissemination in China by Beiging's China Electronic System Engineering Co. It is also being used by several companies in Hong Kong to type letters and contracts and for document storage.

As a result of Syntronic CS-4000 and related breakthroughs, Chinese personal computers are also beginning to come into the market. One Taiway manufacturer, Multi-Tech Industrial Corp., for example, has just announced their Chinese script "Microprofessor II", retailing for less than $600 U.S. This perscom can also process Roman characters.

1.16 The Software Bottleneck

Efforts to break through the software bottleneck have included:

1. The use of so-called program or applications generators through which a businessman states his programming needs in plain English to a computer, which (through the applications package) possesses the ability to translate such instructions into a program.

2. New all-purpose programming languages such as ADA, a language being developed by the U.S. Department of Defense to replace COBOL, the most popular business language, and the 25-year old FORTRAN, the scientific language.

3. New parallel processing computers. This requires a brief explanation.

Most of today's computers can process instructions only sequentially, i.e. one after the other. The new parallel processing computers would be able to process instructions simultaneously. As reported in The Economist:

> "Researchers at Manchester University ... are working on a system comprising a ring of linked processors and memories to do parallel processing. Specialists at the University of Kent, who are developing software for such systems, point out that such a machine could not only use more powerful languages but could also make programs easier to write -- because the instructions would not have to be in strict sequence."[20]

1.17 Artificial Intelligence Software

Of all the differing types of software discussed in this work, probably the most significant involves developments in <u>artificial intelligence</u>. This new branch of computer science tries to program computers to respond as if they were intelligent.

> "AI systems involve a new type of programming which processes ideas and facts instead of merely numbers. Many AI programs utilize normal `if-then' rules (but is applied to concepts), e.g. `if an animal has tusks, then it is not a cat,' while others construct huge networks of empirical facts which inform a computer how individual pieces of information are linked together. With such a network, a computer may `know' that only mastedons and elephants have tusks, but that mastedons are extinct. To solve problems, AI software attempts to manipulate this body of empirical facts in the same way that humans reason. After sorting through a body of knowledge, the AI programs have facilities to decide on their own sequence of steps rather than following exact `if-then' rules."[21]

One of the most important applications of AI research involves <u>expert systems programs</u> -- software which allows a computer to perform the role of a smart assistant in areas of technological expertise. Expert systems programs, a result of military and medical research done mainly at the University of Edinburgh, MIT, Carnegie-Mellon and Stanford University -- can apply decision rules of human technical experts to seek solutions to specific practical problems. Examples of expert systems software include programs which can competently interpret an electrocardiogram, a program which allows organic chemists to ascertain the molecular structure of compounds more rapidly, a program for diagnosing meningitis and other bacterial infections, and a program which has successfully aided in the exploration of ore deposits.

Although there is world wide interest in artificial intelligence, the most concerted efforts in this area have been made by the Japanese, who over the next three to four years will be spending more than $50 million to build a "fifth generation" computer, which will be designed on AI principles.

New industrial applications of artificial intelligence software are just emerging from the laboratories and will have major impacts during this decade in the areas of industrial automation, applications generators, robotics, and many other fields. But how do these expert systems work?

Computers are quite dumb and of course have no empirical knowledge of the world. Thus, in each expert system program the computer must be given a knowledge base of empirical facts which are relevant to the specific problem which is to be solved.

It is a lot easier to provide a wide knowledge base about an extremely detailed but limited technical problem than it is about a general problem. Essentially, AI researchers in the field of expert systems conceived various theoretical ways to represent technical know-how and to translate these representations into specific decision rules which could be input to a computer.

The way of acquiring a knowledge base for an expert system program is to simply sit down with an expert with a great deal of practical knowledge in a specialized area and try to codify his practical knowledge and informal intuitions into rules.

Expert systems programs are "transparent" -- meaning that if requested, they can present the reasoning at which they arrived at a specific conclusion in a form which human beings comprehend. This property renders expert systems an extremely useful tool for designing and programming computers which are too large and complicated for human beings to directly interact with.

To construct an expert systems program, one first of all needs an expert, whose knowledge may be codified into rules which may be understood by computers. Experts are often incredulous that their highly specialized knowledge, which they often feel is based on intuitions, can be so easily codified into rules. The job of transferring the empirical knowledge of an expert into rules is called knowledge engineering. A knowledge engineer must first of all understand an expert's field of specialty and translate his or her formal and informal knowledge into if-then rules compatible with digital computers.

Expert systems have now been widely used in designing large-scale integrated chips themselves by both Texas Instruments and Xerox. Lynn Conway, Director of the VLSI project at Xerox Parc, has noted that, "Integrated circuit design is conceived as an invention or an art form in the past. We want to mechanize and propagate that knowledge."[22]

With the rapidly increasing memory sophistication of perscoms, by the end of the decade there will be considerable uses of expert systems by small businesses in areas such as financial advice. Exhibit 1-9 lists a number of North American expert systems programs.

We as yet know very little about future applications of expert systems programs, however if the cost of building a system is less than that of hiring a consultant, the potential is enormous. Since it is thought that "intelligent" computers will displace millions of North American workers in both industry and office work, the term AI unemployment is already being used.

EXHIBIT 1-9

Some North American Expert Systems Programs

Prospector	Developed by SRI International in Menlo Park, California, Prospector can locate uranium deposits with considerable accuracy.
R-1	Developed by John McDermott of Carnegie-Mellon University and commercialized by the Digital Equipment Corporation, R-1 aids in the proper selection of components for customized computer systems.
Dendral	Developed at Stanford, Dendral allows chemists to rapidly determine the molecular structure of organic compounds and has been commercially used by Dow Chemical and Marck & Lederle.
Mycin	Also developed at Stanford, Mycin is a type of helper in administering antibiotic drugs. The use of Mycin in diagnosis has been as good as those of doctors.
Oncocin	Developed at the Medical School of Stanford University, Oncocin is an intelligent helper to aid doctors to treat cancer patients. This expert system is a guide to administering complicated drugs.

1.18 Software for Education and Industrial Training

Until the past couple of years, computer learning in North America was a dismal failure. Early computer learning systems were difficult to use, and there was almost no software for differing subjects. The software that did exist was extremely uncreative -- limited predominantly to filling in blanks, multiple choices, etc. Most importantly, computer learning technology was extremely expensive. But the costs of the technology and the software is decreasing with the quick diffusion of inexpensive perscoms throughout North America and the mass of educational software which is accompanying this diffusion. By mid-1982, for example, there were more than 10,000 perscoms in Canadian schools, representing a total hardware investment of more than $20 million. Most of these purchases were made during the last year. One recent U.S. survey also found computer learning in over 50% of U.S. educational institutions, predominantly through perscoms. In the U.S., Apple Computer has sold more than 50,000 microcomputers to schools, with more than half of these sales occurring by the end of 1981. Apple is also seeking to consolidate its school position through offering to donate an Apple perscom to every school in the U.S. in return for tax write-offs on such donations.

Within the computer learning market, the perscom has become basically a vehicle for content, and computer learning software now comprises a major part of the educational computing market. Major decisions to purchase perscoms are now affected by the availability of good software for a specific machine, and suppliers of content face severe problems which include the ease with which teachers may copy software discs and the incompatibilities which sever this market.

The number of personal computers now in use in North American schools is unknown, but it is thought that approximately one out of four public schools has at least one perscom. In a single state, Minnesota, there are now more than 3,500 perscoms being utilized, and both the French and the Japanese are planning to put tens of thousands of them into their school systems. The driving force in this field is the software which facilitates applications. This software is comprised of the actual content of the educational or training courses (which is called courseware) and the new computer learning languages themselves, such as Logo and Smalltalk.

Logo, co-invented at MIT by Seymour Papert, is a powerful computer language which teaches children procedural thinking. Utilizing Logo, the child controls a screen animal called a "turtle". Each Logo instruction tells the turtle to perform some specific action, e.g. to go forward 50 steps or to turn left 90 degrees, etc. Utilizing Logo, young children learn in a very natural way how to communicate through a computer. In the words of Papert:

> "Learning to communicate with a computer may change the way other learning takes place. The computer can be a mathematics-speaking and an alphabetic-speaking entity. We are learning how to make computers with which children love to communicate. When this communication occurs, children learn mathematics as a living language. Moreover, mathematical communication and alphabetic communication are therefore both transformed from the alien and therefore difficult tasks they are for most children into easy ones."[23]

Logo is now widely available on perscoms, and one U.S. school utilizes Logo for the entire curriculum.

Just as Papert and his colleagues were inventing Logo at MIT, the Xerox Palo Alto Research Centre was in the process of developing Smalltalk. Smalltalk is the software portion of:

> "An educational personal computing vision, the Dynabook -- a hand-held educational computer with high resolution display, input and output devices which can support both picture and voice and linkages to outside information sources. The research at Xerox has focused on the ability of someone to creatively use the Dynabook, necessitating a comprehension of the nature of and the interactions between language, communication and knowledge."[24]

The Smalltalk project attempted to structure the Dynabook in such a way that a child could modify it by himself to build a new application, i.e. to give to children a sophisticated graphics programming tool and see what they would do with it.

1.19 Artificial Intelligence in Computer Tutorials

Of particular interest are software packets which incorporate principals of artificial intelligence into computer tutorials. Such tutorials act as if they understand their subject matter. Similar to the previously discussed expert systems programs, AI-based tutorials contain both rules describing how a subject is to be taught and a large knowledge base.

> "As the student learns, the program builds a model of the students' understanding and compares this with the entire knowledge network to determine what should be taught next. When a student has a misconception or makes an error, the program can diagnose the missing or incorrect information by means of its model of that student ... this capability to handle student errors in a meaningful way is one of the great advantages 'intelligent' computer-aided instruction, and the major failing of the original approaches to the use of computer programs for teaching. The power of such diagnostic capabilities has been well demonstrated by the 'BUGGY' program developed by Xerox. BUGGY consists of hundreds of rules which specify different kinds of errors ('bugs') students can make in doing simple arithmetic problems. When a student makes a consistent mistake, the program compares the pattern of errors against those which would occur due to the different 'bugs' it knows about. If it finds a match, it explains to the student what error is being made."[25]

Such intelligent tutorials may question a student in a natural language with either the student or the tutorial responding. Such intelligent tutorials can, in some sense, understand what a student is saying, even with spelling mistakes, and in some cases can even comprehend what a student is trying to do.

A main problem, however, with artificial intelligence-based software packets in education is that they often require several years to write and also can only run on extremely large mainframe computers.

Although such AI-based software will ultimately revolutionize computer learning, at present there is a dearth of even good (non-AI) educational software for virtually all topics.

The provincial government of Ontario is building a prototype educational microcomputer in 1983. This government will be spending an additional $5-$7 million to develop educational software for this perscom, developed by the CEM Corporation of Toronto. Two types of educational microcomputers are being built, one utilizing a standard 16 bit microprocessor and another utilizing a 32 bit microprocessor. Since a data storage device is being incorporated into both of these perscoms, they will be able to connect to other machines within a local area network.

1.20 The Semi-Automation of Programming in Educational Software -- Authoring Systems

Automated programming involves software which itself generates software. (We have already reviewed software which merely aids in the writing of software in the discussion of applications generators.)

In the field of computer learning, an authoring system comprises software which allows a teacher to write a computer-based curriculum for education or industrial training with comparatively few programming skills. The teacher specifies subject matter to be taught and how he or she wishes it to be taught, and an authoring system will generate much of the course. Thus, authoring systems allow teachers to create course material without really having to learn a program language. Such authoring systems are of critical importance in industrial training and education for these reasons:

1. Developing the content of computer training is very expensive, and it necessitates between 100 and 200 hours of development time for a single hour of actual instruction. Most of this time is put into debugging software, and authoring systems can reduce this ratio.

2. A major problem in computer learning and training is that courseware is usually not transportable from one computer to another, since different institutions have different brands of computers. (But more recently a few authoring systems now contain the facility of automatic translation to whatever computer is being utilized.)

3. Authoring systems are important because learning how to write software, even utilizing higher level languages which resembles English, radically limits the number of people who will even try to use computers.

Most of the present authoring systems, in spite of what has been said about artificial intelligence, are extremely primitive, and all of them were developed for dealing with textual tutorial materials. In these, the screen display follows a predetermined pattern, and almost none of these authoring systems can adequately deal with graphics. Also, since most existing authoring systems follow the tutorial form, they may not be used for simulations which are extremely useful in physics, chemistry and other natural sciences since experiments may be simulated on computers at a small portion of the cost of actually doing the experiment.

Although it is clear that each of the above problems will eventually be solved by knowledge-based authoring systems which are analogous to the expert systems previously discussed, so far there are no commercially sold authoring systems which are based on artificial intelligence concepts.

Having now reviewed the basic types and levels of software and world software markets, we now turn, in the second part of this work, to an examination of software and microelectronics policies in two developing countries, Brazil and Singapore.

2.0 CASE STUDIES OF SOFTWARE AND MICROELECTRONICS POLICIES

2.1 Brazil

The development policies of Brazil are generally considered to be a model for third world informatics policies. Larger than the United States, Brazil now is included in the top nine data processing markets in the world, and data processing sales exceeded $2 billion in 1981. The main manufacturers of Brazilian computers and 1980 estimates for data processing revenues are presented in Exhibits 2-1 and 2-2.

During most of the '60's there was a minimal amount of regulation of importation of computers and data processing products into Brazil, and it was only in the early '70's that the government began to regulate imports with the aim of creating a national software and computer industry.

It was only in 1974, in fact, that policy planners in Brazil began to formulate their national informatics policy, and within seven years of this date imported software has been banned from Brazil. Foreign joint ventures in Brazil had to set up Brazilian design and manufacturing facilities within five years of the initiation date of the joint venture, and the Brazilian domestic market for perscoms and minicomputers had been reserved for Brazilian manufacturers.

EXHIBIT 2-1

Authorized Brazilian Manufacturers

Company	Model	Technology	Description
COBRA - Computadores e Sistemas	COBRA-300	COBRA	Up to 48KB cpu Small single-user micro Based on floppy disks
	COBRA-400	SYCOR (US)	Up to 64KB cup minicomputer with a model oriented to data entry and MUMPS
	COBRA-330	COBRA	Up to 512KB cpu with future expansion to 1 MB
EDISA - Electronica Digital S.A.	ED-300	FUJITSU (Japan)	Up to 64KB cpu Similar to IBM System 3
LABO Electronics	LABO 8034	NIXDORF (West Germany)	Up to 256KB cpu Similar to Nixdorf 8870-1
SID - Sistemas de Informacao Distribuida S.A.	SID-5000	LOGABAX (France)	Up to 64KB cpu Similar to DEC PDP 11/34
SISCO - Sistemas e Computadores S.A.	SCC-5000	SISCO	Up to 64KB cpu Similar to DEC PDP-8
	MB-800	SISCO	Up to 256KB cpu Similar to DG NOVA 3

* Source: Datamation, May 1981, pp. 192-7.

EXHIBIT 2-2

Major Brazilian DP Suppliers

1980 Ranking by Revenue	Company	Estimate DP Revenues (in $million) 1979	1980	Capital Registered in Brazil ($million)	No. of Employees
1	IBM	350	330	130	5
2	COBRA*	70	104	9	2
3	BURROUGHS	100**	100**	2**	2**
4	SID*	19	30	5	.8
5	LABO*	5	22	2	.4
6	EDISA*	7	13	3	.3
7	SISCO*	.6	10	3	.3
8	GLOBUS*	.8	10	.8	1
9	ELEBRA INFORMATICA*	.5	9	2	.1
10	SCOPUS*	5	9	.8	.4
11	HONEYWELL-BULL	7	8	3**	.1**
12	POLYMAX*	.4	7	.4	.2
13	MICROLAB*	2	6	1	.3
14	COENCISA*	5	5	2	.2
15	PROLOGICA*	2	5	.05	.2

* Data on these companies furnished by DIGIBRAS, a federal-owned company that provides technical and financial support to national hardware manufacturers.

** Company-furnished data.

All of these policy goals, we shall see, were accomplished with basically no research and development or fiscal/tax incentives and exclusively involved national financing by Brazilian firms and the government.

Foreign firms manufacturing informatics products in Brazil were soon legally required to export a specific number of products for each one sold on the Brazilian market, and although there was considerable balking on the part of the multi-nationals concerning the decision of reserving the market for minicomputers and perscoms, the decision was soon accepted.

The first Brazilian computer policy agency which formulated and enacted these measures, CAPRE, was made a part of Brazil's National Security Council in 1979 and renamed The Special Secretariat for Informatics (SEI). All importation of computer equipment must now be approved by SEI, and there is presently an approximately two year waiting period for approval.

2.2 The Brazilian Computer Market

The Brazilian mainframe computer market is dominated by IBM and Burroughs. Given the two year delay approval by SEI, the growth of mainframe computers over the past three years has been on average around 15% per annum. Minicomputer growth, however, has averaged more than 200% per annum and by 1981 several Brazilian manufacturers began to produce perscoms. However, there are few manufacturers in Brazil of computerized industrial process control equipment, and the main inhibiting factor here is Portuguese software for industrial process control.

In spite of all of these efforts, Brazil still does not have a viable semi-conductor industry, but the present development strategy appears to focus on creating a national market for computer terminals which in turn will generate demands for components on the part of Brazilian manufacturers, since this market is protected by tariffs. However, as Juan Rada has remarked:

> "The cost of such a strategy rules out competition in the international market and means that a policy of technical obsolesence closely related to the function of the equipment has to be adopted rather than a policy of keeping pace with the speed of innovation. In addition, it applies to slow diffusion of computertechnology due to the higher price/performance of the equipment."[26]

Brazilian policy measures to enhance national software development, subsequently examined in detail, includes:

1. Commercialization and promotion of nationally developed software.

2. Development of courses and training on software in cooperation with universities.

3. Financial assistance and encouragement of governmental enterprises to invest in new software companies.

In the area of telecommunications equipment, Brazil extensively uses government procurement. The Ministry of Communication has an extensive research and development centre working on optical communications, switching and digital transmission. More than 60% of all national informatics industries depend on sales which are directly or indirectly linked to the Ministry of Communications, and as a consequence of the procurement policy, imported telecommunication equipment and components decreased from $245 million in 1975 to $54 million in 1978.[27]

2.3 Trans-Border Data Flows

In 1982, Brazil conducted several emprical studies[28] focusing on policies towards trans-border data and information flows across Brazil's boundaries.

Trans-border data flows are extremely important in connection with computer activities, since they often result in a transfer outside of the country of information resources such as software, computer data bases, hardware and computer-related employment in general.

In June 1982 MNE subisidiaries in Brazil comprised 27 of 29 computer communication linkages outside the country, and the anticipated growth rate of these is three per month.

Brazil has four main objectives in dealing with information resources and trans-border data flows in particular:

1. to enhance Brazil's political/cultural environment;
2. to maximize information resources located in Brazil, whether these are produced locally or imported;
3. to maintain control over computer telecommunications technology; and
4. to broaden the public access to information.

In implementing these objectives, Brazil has conceived policies governing telecommunications, telematics, informatics and trans-border data flows. In each of these areas, appropriate regulatory frameworks and policies have been instigated, and will be outlined in the following discussion.

The Brazilians make a distinction between two types of trans-border data flows: intercorporate flows -- those in which computer-based information and data is supportive of related activites; and commercial flows -- those in which data and information is itself the object of trade.

In the area of telematics, a national telecommunications holding company, Telebras, was set up in 1972 to digitize Brazil's public networks and to develop a national Brazilian technological capability. By 1980, Telebras had established the Sicram service, which is a computerized storage and retrieval system for messaging now operating through terminals which are linked to Embratel, Brazil's national telecommuncations company, and also linked with Brazil's national telex network. It is also anticipated that Telebras will establish a packet-switched public network during 1983.

Brazilian telecommunications policy, which is focused on reduction of technology imports and on incentives for local production via Telebras' procurement, has positively affected the development of Brazilian technology. Industrial strategy in Brazil encourages the purchasing of equity in foreign informatics multinationals by Brazilian firms and also encourages the growth of new informatics industries. There is now considerable evidence that Brazil's telecommunications policy has actually stimulated the domestic telecommunications industry, e.g. in 1978 90% of Telebras' acquisitions came from foreign affiliates, but by 1980 this figure had been reduced to 41%.[29]

The Brazilian informatics industry had its genesis in 1972 in the creation of CAPRE (The Coordinating Commission for Data Processing Activities). Given the job of creating an industrial policy for informatics, CAPRE had considerable powers to both restrict and regulate the importation, we have seen, of foreign data processing equipment, and also to protect the national Brazilian market from the unregulated importation of modems, minicomputers, perscoms and video terminals.

By 1979 SEI, the progeny of CAPRE, implemented a new strategy to upgrade the capability of national industries to produce more complex informatics technologies. As part of this policy:

> "Foreign affiliates are encouraged to exercise a comparative advantage and to produce advance state-of-the-art computer good and services for local consumption and export; they are also encouraged to improve local research and development facilities. Once a product can be manufactured by national capital, the respective market segments are protected to give the industry an opportunity to develop, while foreign affiliates are encouraged to shift toward more sophisticated products (instead of upgrading products in the same segment). The degree and type of protection is a function of the technological state of the products involved. As soon as international competitiveness is achieved, it is envisaged that protective barriers can be lowered, although measures may be considered necessary to ensure the continued improvement of local technologies and the permanence of national ownership of the country's informatics industries."[30]

Although they were initially infuriated by these policies, few multi-nationals left Brazil and most continued to prosper in the Brazilian market. Simultaneously, Brazil's share of this domestic informatics market increased to 36% by January 1982, with 71 private corporations (and one public corporation) producing more than 100 informatics products, predominantly for export. Locally produced data equipment, which comprised 30% of the total market in 1979, increased to 53% by 1981, and imports decreased from 29% in 1979 to less than 8% in 1981.

SEI's policy efforts are now focused on microelectronics hardware and software, and real time process control systems -- through the utilization of fiscal encouragement to set up specific types of manufacturing facilities, research and development centres, and increased use of government procurement and fiscal policies.

Brazil's first regulations on trans-border data flows in 1978 were a result of requests from MNE's to set up transnational computer communications facilities. At that time CAPRE ruled that these linkages would be approved for specific purposes only and for specific time periods. No restrictions were made on the content of actual messages, however. In the words of CAPRE, such a regulation was based on the principle that information and software are "economic resources, subject to trade and crucial socio-economic development". Brazilian policy in this area basically comprises a scrutiny of each potential computer telecommunications linkage and a determination of "to what extent individual applications for links fit with overall objectives."

Specific criteria which are applied involve whether the data flow is of a commercial or intercorporate nature, and what uses are being made of the information transferred in intercorporate computer communication systems, such as person-to-person communication, data processing, data base access, etc. The specific criteria for approval or disapproval of each link are summarized in Exhibit 2-3.

In the time since Brazil first instigated its policy on transborder data flows in 1978, there have been 32 applications for such linkages and seven refusals. Of the rejected seven, five involved access to data processing services located outside of Brazil, with the predominant reason for rejection involving the fact that the firms involved had made no effort to locate information resources inside Brazil. Approved and refused transborder linkages, by type of user, are summarized in Exhibit 2-4.

Any data processing done outside of Brazil over computer communication linkages is absolutely forbidden.

Although it is too early to thoroughly evaluate Brazil's transborder data flow policies, thus far there has been a definite increase in the numbers of both data bases, computers and software programmers in Brazil, and a general strengthening of Brazilian informatics industries has occurred without any significant damage to the amount of foreign investment.

EXHIBIT 2-3

Matrix of Brazilian Transborder Data Flow Policies*

On-Line Use of Transborder Data Flows	Category of On-Line Transborder Data Flows	
	Corporate	Commercial
Data Communications	Person-to-person communications are not restricted	Brazilian PTT only; co-operation agreements possible
Data Base Access	Copy of data base in Brazil, whenever reasonable	Encouraged, but in co-operation with Brazilian institutions, preferably with copy of data base in Brazil. If no local copy, services are provided by the PTT, although co-operation agreements are possible.
Data Processing (including use of software)	Not favoured	Not allowed abroad except in exceptional circumstances

* SEI, op. cit., p. 33.

Source: Special Secretariat of Informatics, 1982.

EXHIBIT 2-4

**Transnational Computer-Communications Systems
Approved and Refused Applications By Type of User
March 1982**

Type of User	Applications			
	Total	Approved	Refused	Pending
Government	2	2	-	-
Data Services	9	2	5	2
Closed User Groups	1	1	-	-
Transnational Corporations:				
Brazilian	4	4	0	-
Other	16	14	2	-
TOTAL	32	23	7	2

Source: Special Secretariat of Informatics, 1982.

2.4 Brazilian Software Policies

Realizing the importance of computer software compared to hardware, the Special Commission on Software has made the following specific policy recommendations.

Since the software and computer industries in Brazil are generally small to medium sized, they have great difficulty in obtaining development financing of software products. The SEI is cooperating with Brazilian funding agencies to make sure that there is sufficient financing available for both development and marketing of Brazilian-produced software, including funding for all phases of development, from research through start-up production. This funding also includes support of international marketing during the product's first year.

Realizing also that it is almost impossible to "establish realistic barriers to the physical importation of software produced in other countries"[31], Brazilian funding agencies have established a system in which software financing is provided for firms utilizing Brazilian technology and which insures that no exclusive use restrictions are being placed on the development of software. The Brazilians are also directing financing toward companies whose principal activities are software production.

In the area of international tenders, the Brazilians have drafted regulations which govern the importation of computer equipment and peripherals which are purchased with international financing. If computer equipment and systems are available from national producers they must be purchased inside Brazil, thus decreasing technological dependence and excessive imports.

In terms of software investment priorities, Brazil (and, as well shall see, Singapore) has given priority to software investments in which the country is least developed, i.e. basic business software.

The Brazilians are also actively attempting to increase software exports, which have already included software for banking terminals in Africa. Also in the area of internal fiscal incentives, the special commission recommended that the Ministry of Finance make available for up to ten years fiscal incentives according to which software users would be allowed a 200% deduction on any direct expenditures involved in the purchase of Brazilian developed software.

One of the most basic and important policies of SEI involves the registration of software. Noting that it is in the interest of Brazil to specify those types of software whose importation should be encouraged and to establish official prices for software to bolster the practice of separating software sales from those of computer equipment, the special commission on software has recommended that:

> "The SEI create a mechanism for registering all software products available in Brazil; that the SEI issue regulations instructing all government agencies and organizations operating either directly or indirectly under the federal administration to purchase only registered software products; that the SEI issue regulations which establish a method of making the granting of import licences on computer equipment dependent upon registration of the accompanying software; and that the SEI urge the Ministry of Finance to issue a regulation stating that the purchase price of software products may be included under operating expenses and will be tax deductible only in those cases where the software in question has been duly registered".[32]

Furthermore, with respect to the separation of software and hardware purchases, the Brazilians have realized that first of all it is in their interest to create a market for software products which is not dependent on foreign computer manufacturers, and also that it is absolutely necessary to establish some degree of competition between computer and software producers. The Commission on Software and Computer Services has thus recommended that the prices for hardware and computer software be clearly defined, and that in the case of computer imports, inclusion of the cost of the software is prohibited on the same import licence with equipment, and all software must be covered by separate contracts.

With respect to specific controls on basic software, the Ministry of National Revenue now requires that contracts with purchase of "basic software" are examined for the end of registration with INPI. The registration of software contracts with the INPI is the most predominant means that Brazil uses to control purchases of foreign software, since each of these purchases always entails payment in foreign currency.

Finally, in the area of software training, the Brazilians have created new full degree programs in their universities, since there are no programs which are currently available which allow satisfactory levels of software training in Brazil, especially in the areas of basic software and firmware. The Brazilians are also encouraging teaching and research of software aimed especially at industrial automation processes.

2.5 Singapore's Software Strategy

Singapore is a small country at the tip of the Malay Peninsula with few natural resources, save her 2.5 million people. With the constraints of this limited manpower resource, Singapore has recently conceived and implemented an industrial strategy which involves the development of high value-added technology and knowledge-based industries -- such as software, computer service industries, automated banking and finance, engineering and medical/scientific consultancy services.

This industrial strategy is comprised of three sets of interlinked measures which will be described in this section -- development support measures directed at these new industries, export support measures and manpower training/upgrading.

One of the main thrusts of Singapore's new industrial development strategy is manpower development to train workers to meet demands of the high technology industries and "brain" services which Singapore is promoting. Here the Economic Development Board has supplemented other governmental training efforts through the operation of its <u>own special training centres for technology, which are collaborative efforts with the private sector and other governments</u>. These are: The Japan-Singapore Institute of Software Technology, the German-Singapore Institute of Production Technology, and The French-Singapore Institute of Electrotechnology.

As of March 1982, 2,298 apprentices and technicians had completed two years training programs in these institutes and had been placed in 215 companies, while 1,242 were still in the process of being trained.

In total, these training institutes have covered specialized areas such as computer-based tool and die design, computer control of production processes, robotics, microprocessor and computer applications, and software technology.

2.6 Recent History

In March 1980 a ministerial Committee on National Computerization was put into place by the Singapore government to create a strategic plan with the aim of, "establishing Singapore as a centre of computer services and software." This committee, chaired by the Minister for Trade and Industry, was given the task of creating a master plan to build up microelectronics expertise in Singapore and to create an export-oriented software industry.

Specific recommendations of this report involved:

1. increasing the number of computer professionals;
2. accelerating computerization in the civil service; and
3. promoting the computer software industry.

One recommendation also involved the establishment of a National Computer Board (NCB) which would implement this plan for the development of computer software.

In addition to promoting the computer services industry, the National Computer Board has formulated a series of innovative programs and industrial incentives to induce firms to set up software development companies in Singapore, particularly those which deal with software developed for state-of-the-art technology. Alsoto answer the shortages of software professionals, a series of programs has been instigated to increase and improve the training of data processing professionals, and these programs are collectively producing a supply of approximately 700 information systems professionals yearly.[32]

2.7 Computer Education and Training

In its preliminary 1980 report, the Committee on National Computerization had identified the absence of trained software and computer professionals as the "main barrier" to further developing Singapore into a centre of excellence for computer services. Thus, one of the Committee's main recommendations asserted that efforts should be placed into quickly training computer professionals. Since the Singaporeans estimated that the number of software writers necessary for the computer service industry would be between 5,800 and 7,800 by 1990, and that their current number of professionals in this area was around 1,200 in 1980, they estimated a need to increase the software writing pool by five to six times within eight years. On a yearly schedule, they have kept to this ratio thus far.[33]

To satisfy this demand for computer professionals, Singapore has instigated a plan involving the following elements:

Before the 1980 formation of the committee on National Computerization, Singapore's software and computer professionals came from in-house training programs of multi-national computer vendors and firms widely using computers such as national banks and insurance organizations. The government first instigated financial/export incentives to upgrade and expand the quality of these multi-national training programs and also established computer-related training institutes.

These efforts were correlated with upgradings and expansions of computer programs throughout the school system, and involved the creation of the Institute of System Science (a collaborative effort between the National University of Singapore and IBM), the Japan-Singapore Institute of Software Technology, the French-Singapore Institute of Electrotechnology, the German-Singapore Institute of Production Technology and the Centre of Computer Studies.

The Institute of Systems Science is a post-graduate institute instigated in October 1981 with IBM, which trains around 100 systems analysts yearly. It is also providing courses in training senior executives in Singapore to use computers.

The Centre of Computer Sciences is a new institute which trains around 200 programmers yearly at the polytechnic level. Begun in 1983, it is a joint collaborative effort between the British Council and the Ngee Ann Polytechnic.

The German-Singapore Institutde of Production Technology began operations in February 1982, and trains production engineering technicians and vocational instructors who specialize in computerized machining processes.

The Japan-Singapore Institute of Software Technology (JSIST), which also began operations in February 1982, provides training in computer software. Its purpose is to increase the supply of computer personnel requisite for the national computerization program as well as for the devleopment of the computer software industry in Singapore. The Japanese-Singapore Institute is also upgrading EDP managers and is providing middle and senior management training in the applications of computers. They now have an output of around 300 persons per year. This project was officially launched with the signing of the Record of Discussions in Singapore in December 1980. With this agreement the Japanese government (i.e. NEC) agreed to provide technical assistance to Singapore in the form of NEC computers and a team of six software experts for five years.

Although it is thought that Japan's expertise in alphaphotographic Videotex and facsimile transmission will enable her to be one of the first to penetrate the Chinese market, Japan is not exactly known for software expertise, and this co-operative venture with singapore must be seen as the first real effort to develop software markets in Southeast Asia.

- The French-Singapore Institute of Electrotechnology began operations in April 1983, and will specialize in training for electronic manufacturing and servicing industries. Main emphasis here has been placed in the area of robotics, microprocessors and computer applications.

All of these manpower training institutes are a creation of Singapore's Economic Development Board, which as we have seen gave high priority in 1981 to ensuring that an adequate supply of highly skilled persons would be available for the computerized industrial sector.

Figure 1 presents this entire framework for computer education/training in Singapore. It is expected that the total output of computer professionals in Singapore will soon be around 700-1,000 per year.

In implementing the new computer software policy, Singapore is acting very quickly and it has achieved most of its stated goals already. Because it is a little country, once policy has been formulated Singapore may respond very quickly without the need, as in Canada, for task forces and field trials of new technology. Quite simply, Singapore is attempting to guarantee that training of its software writers and computer professionals will be able to respond to market demands as Singapore implements its industrial strategy to transform itself into an area of software expertise for Southeast Asia.

2.8 Industrial and Export Support

Fiscal, tax and export incentives are actively used by Singapore to implement a policy of restructuring industry toward more technology-intensive activities which are higher value added. Progressive tax incentives are given to firms involved in technology upgrading, automation and software training projects.

These are the major incentives offered by the Economic Development Board to industry:

1. Export Incentives

   These incentives give a 90% tax exemption on all profits above a specified base which are a result of export sales. The incentive is granted for three or five years, with the longer period being granted to companies which have not received a Pioneer Status Incentive.

2. Pioneer Status Incentive

   The Pioneer Status allows for total tax exemptions of 40% of a corporation's income tax for five to ten years, with this exemption being granted to firms which engage in sophisticated software development.

3. <u>Investment Allowance</u>

This is an incentive for both manufacturers and technical service companies. It is considered as an alternative to the above two incentives.

In Singapore the investment allowance has mainly been used to promote automation. Under this incentive a firm is given tax exemptions on, "a specified amount of profits equal to the approved investment allowance which is a percentage (up to a maximum of 50%) of the fixed investment in plant, machinery and factory buildings actually incurred by the company on the project."[34]

Such allowances are granted as a bonus over the normal capital allowances given to a firm. Thus, any firm can at the same time claim a three year accelerated depreciation on plant and any machinery and not have any benefits of the investment allowance reduced.

4. <u>The International Consultancy Services Incentive</u>

The incentive was instigated to increase the number of software, consultancy and engineering firms located in

Singapore which export. it gives a five year 20% tax write-off on export profits which are above a specified level. Export services hwich are eligible for the incentive comprise software, design and engineering, machinery production, data processing and all technical advisory services. However, only companies which have a gross export income of at least $1 million are eligible.

5. Tax Incentives for Research and Development

In addition to the above incentives, Singapore has introduced the following R&D incentives:

a. a double deduction of 200% of any R&D expenditures, except those on buildings and equipment;

b. accelerated depreciation over three years for machinery and plants used in R&D;

c. investment allowances of a total of 50% of an R&D capital investment, with the exception of building costs; and

d. capitalization of lump sum payments for manufacturing licences for five years.

In addition, the Singapore Science Park, which is situated next to the National University of Singapore, has recently been developed to accommodate R&D organizations and activities in connection with the manufacture of high technology products. As part of this science park, the National Computer Board will operate a Software Technology Centre.

2.9 The Singapore Electronics Industries

Heavily implicated in computer and software industries, Singapore is discouraging labour-intensive factories by increasing wages. In 1981, the government-controlled pay scale increased 20%. (Having long restrained wages, Singapore ended up with a shortage of labour which was filled by 30,000 Asian "guest workers", and the present strategy of higher wages has not decreased foreign investment, which now provides nearly 73% of all investment in plants and machinery.)

All of the newly industrialized countries, and not merely Singapore, are expert now at the Japanese game of creative copying, improving and cheaply exporting technology. (An improvement on an Apple II computer may now be purchased for $300 U.S. in Taiwan.)

Typical if Singapore manufacturers are:

1. Digital Equipment, which set up a $30 million manufacturing plant for mini-computers;

2. Micro Peripherals, Singapore, a subsidiary of Micro Peripherals (U.S.) which manufactures floppy disk drives for export to world markets;

3. Apple Computer, which has just begun a $40 million project to produce microcomputers and to test encoder boards; and

4. Juki, a Japanese producer of Chinese script word processors.

Now that it has made concerted efforts to develop the European market, Japan is focusing its computer and software efforts in the Asian markets, including Singapore, since it now has a large number of computer users, and its government is actively promoting computerization. Fujitsu, in fact, first set up operations in Singapore in the early 1970s and NEC in 1977. NEC now has a regional centre to train their customers' engineers and software programmers how to use, service and maintain NEC's ACO mainframe computers.

The Japanese have been invited, we saw, by the government to help develop Singapore's EDP user community, and the recently established Japan-Singapore Institute of Software Technology is now training senior level systems analysts and programmers. The

Republic has also committed more than $100 million to computerize its 10 ministries over the next decade.

An equivalent computerization effort is occurring in the commerce, banking and financial sectors, and one international consulting firm projects that Singapore will acquire more than 1,000 minicomputers in 1983 alone and that this number will double in 1984 and 1985.

Over the past decade Singapore has really graduated from an assembler of consumer products to an integrated manufacturer of advanced consumer electronics. In mid-1979 it branched out also into manufacturing high grade components for industrial electronic equipment. The new policy of upgrading artificially suppressed wages is forcing many manufacturers to automate and also to diversify into more sophisticated electronic products.

Government policy encourages industry to specialize in two main areas: high quality components such as semi-conductors and microwave components, and industrial electronics such as avionics equipment, electronic measuring equipment and computer peripherals. Principal statistics of the Singapore electronics industry, the industry's structure and exports of major electronics equipment are shown in Exhibits 2-4 through 2-6.

EXHIBIT 2-4

Principal Statistics of the Electronics Industry*

Year	No. of Establish- ments	Output (S$ Million)	Employ- ment	Value Added Per Worker (S$)
1968	5	8	700	2,000
1969	15	67	7,500	6,600
1970	35	213	11,250	8,810
1971	49	319	15,870	8,890
1972	53	617	27,270	10,500
1973	64	1,097	39,210	10,830
1974	91	1,603	46,230	11,300
1975	95	1,458	32,030	14,800
1976	105	1,988	43,720	14,600
1977	117	2,323	46,440	15,200
1978	135	2,822	53,440	16,700
1979	168	4,093	66,840	19,100

* Source: Singapore Business, November 1981, p.51.

EXHIBIT 2-5*

Structure of the Electronics Industry

	1974	1977	1980	1990
Industrial Electronics	4%	2%	3%	20%
Consumer Electronics	41%	48%	40%	35%
Electronic Components	55%	50%	57%	45%

EXHIBIT 2-6

Singapore Exports of Major Electronic Products ($million)

Year	All Exports	Domestic Exports
1979	4,397.3	1,709.6
1978	2,944.2	1,314.8
1977	2,238.3	1,065.5
1976	1,761.6	748.7
1975	1,113.0	471.3

* Source: Singapore Business, November 1981, p.51.

3.0 SOFTWARE AND MICROELECTRONICS POLICIES FOR DEVELOPING COUNTRIES

Having now reviewed two case studies of software and microelectronics policies of considerable sophistication in two (comparatively wealthy) developing countries, we turn to a consideration of general policy measures in this area for developing countries.

The first factor to note here is that the great diversity in stages of development of third world countries eliminates the possibility of appropriate but specific policy recommendations in these areas and that sophisticated microelectronics/software policy measures are feasible only in those countries possessing a significant degree of industrial development.

Although we have noted in the discussions of educational software and applications generators that advances are occurring in the automation of software writing, widespread commercializations of the latter are still a decade or two away, and in the meantime software creation remains a labour intensive process. Thus, during the following decade there are real market opportunities in software for developing countries which still have comparatively low wages, and this portion of the report will focus on software and microelectronics policy issues of relevance to such developing countries.

Software policies are of particular importance, given current attempts by multinationals to strengthen terms of agreements for software use. This is occurring because of the difficulties in patenting software, the ease with which it may be copied, and simply its increased economic importance.

Software, we have seen, is an "information product", and this fact means that it may be more cheaply copied than the purchase cost of an original version. (Although it is possible to write software in such a way that it will self-destruct -- i.e. be erased if one attempts to copy it from one disc to another -- such a situation would create innumerable consumer complaints from users accidentally copying discs and inadvertently erasing their software. In fact, few companies have protected commercial software in this manner.)

Some software is patentable, but patent and copyright procedures always involve publically disclosing the software, and companies do not want to do this, since other companies would then have access to the copyright and patent details and could slightly modify the software just enough that the patent is not applicable.

For all of the above reasons, multinationals are attempting to strengthen terms of licences and agreements for software transferral to developing countries, and since this is one of the main ways software has traditionally entered these countries, the economic importance of multinational transfers becomes evident.

The second background trend to note is this: because of the great number of consolidations in the software industry among firms producing "low level" business software, and the fact that one, for example, might hire an Indian programmer at a tenth of the developed world salary cost to write business software, much software writing will be done off shore in the developing countries.

This will occur for at least a decade until advances in the automation of software writing render it more economical to produce in the home country (as has been the trend with hardware production in the export-free zones), after which time low programmer labour cost will no longer be an advantage.

3.1 Determining What is Needed

One of the first decisions that any developing country must make is this: amongst the range of microelectronics-based processes and products -- from integrated circuits and other components through products incorporating chips, computers, information systems and types of software -- which they actually <u>need</u>, and of these, which they will produce and which they will purchase. Specific policy instruments may (sometimes) then be devised to support these purchase vs production decisions. The Brazilians, for example, we have seen, have reserved a portion of their internal microelectronics market for specific types of Brazilian products (but they have a large internal market all speaking Portuguese) and there is little reason for any developing country to protect its internal market if it is miniscule.

It must be also noted that policies with respect to integrated circuits and components, "should be subsidiary to a general strategy in the area of informatics, which must also include ... applications in general."[35]

Secondly, developing countries, it is well documented, often have problems in both obtaining and utilizing technology because:

1. they do not have the technological institutions which could help in selecting appropriate technologies;

2. their often non-existent productive sytems cannot select and utilize the transferred technology; and

3. given the multinational domination of indigenous economies, they often have little real autonomy in selecting technology.

Many of these traditional problems are being exacerbated with such "intangible" microelectronics products such as computer software for the reasons previously described.

This question of which microelectronics hardware and software technologies to purchase and which to produce, becomes especially important given the growing protectionism in the developed world, and the need of the developing economies for more economic diversification and greater trade. The main motivational factors involved in the growing protectionism and this need for diversification involve:

1. changes in comparative advantages arising from the real increase in energy costs and man-made factors;

2. the growing protectionism amongst the industrialized countries of the world against exports of light manufacturers such as textiles and clothing;

3. the increased competition from the new low-cost countries in selected light industries; and

4. the impact of microelectronics on current and future export markets.[36]

As a response to this pressure, many of the NICs of Southeast Asia, we have seen, are now promoting information industries via low corporate taxes and extraordinary export incentives.

3.2 Licensing of Microelectronics Technologies

One means of purchase is to licence the microelectronics technologies. Traditionally many of the successful NICs of the Pacific Rim have made extensive use of licensing of foreign technology, which ranged in 1979 from 0.2% of GNP in Japan to

in such "heavily licensing" countries, most existing R&D expenditures are devoted to short term product development. But before they licence, many East Asia countries developed methodical means of assimilating foreign technology, the first being the so-called foreign enclaves such as Canton in China, in which foreigners could be observed from a distance without corrupting the entire society. This was also the role of Hong Kong over the past 20 years for the People's Republic of China.

But in connection with licensing, probably the most robust institution for the absorption of foreign technology in the developing countries of East Asia has traditionally been the trading companies and foreign firms. To take South Korea alone, between 1962 and 1976 Japanese trading companies (Sogasosha) were the source of more than 66% of 737 licensing agreements with foreign firms.[38] American multinationals also provided the source of much original technology in Korea and Japan. For example, the Japanese computer industry arose from a series of patents granted from IBM in early 1962 to make mainframe computers in Japan, and the technological base for integrated circuits of NEC, Hitachi and Fujitsu arose from patents granted in 1968 from Texas Instruments. Similarly, the electronics industry in South Korea and Taiwan were founded on licence transfers from foreign multinationals. (There is currently a similar trend in Singapore's new optics industry.) China is just now beginning to buy the technology its needs on a licensing basis, and such trends will continue in the near future.

In summary, many of the industrial gains in the developing countries of East Asia have not been R&D driven; excepting Japan, many of these countries have spent on average less than 1% of their domestic product on R&D annually between 1975 and 1981.

However, when these developing countries have searched for licensing opportunities for technology in other countries, they have been consistently selective about the precise terms of licences, what specific technology is actually licenced, what firms are allowed to licence, and have always focused on technology necessary for industrial development.[39]

Furthermore, in many countries such as South Korea, Taiwan and Singapore, the governments have long maintained lists of desired technologies for the purpose of evaluating specific terms of licensing agreements with foreign firms, and one of the main ways of intervention of such agencies in both Japan, Taiwan and South Korea is to ensure that all of the large firms in any specific industry have an equal access to specific new technologies. (The best known example of this occurred in the mid-1970's when MITI postponed an agreement between Hitachi and RCA for access to integrated circuit technology until NEC, the main competitor of Hitachi, found an equivalent licensor.) It is thus recommended that developing countries with the human resources to so do instigate methodical proceudres to search for software licensing opportunities in a range of countries, paying particular attention to precise terms of licences which may involve restrictive practices.*

3.3 Productivity Centres

Other institutes serving the purpose of formal assimilation of foreign technologies in many developing countries have involved the national productivity centres, whose purpose is to investigate foreign technology and determine how it might be modified for use in the home country. Such national productivity centres have usually been co-financed by government and business, and many productivity centres have sent engineers and researchers abroad to examine advanced technology and see how it could be utilized. For example, one of the main scientists who founded United Technolo-

* These have been enumerated in the discussion of Brazilian software policy.

gies, the Taiwan chip manufacturing facility, was lured away from Intel by this means. Both Singapore and South Korea have productivity centres, and China and South Korea both engage in such an assimilation through national science academies, which have their own laboratories. Again it is recommended that developing countries with the requisite human and fiscal resources set up productivity centres and that countries with existing centres concentrate on formal assimilation of appropriate foreign microelectronics software and hardware which may not be easily produced in that country and which is necessary for short-to-medium term economic development.

3.4 The Brain Drain

Many developing countries have a problem in the microelectronics area due to the systematic loss of scientific and engineering people to that country in which they were educated. However, many developing countries of East Asia have managed to partially stop this brain drain. How is this accomplished?

Many East Asian countries have offered significant monetary and psychological incentives for persons who have been educated in the west to return to the home country. South Korea, for example, conducts annual national searches for the best of the Korean scientists who are working abroad, and Taiwan also conducts yearly "reunification" seminars to repatriate well known Taiwan scientists. It is recommended that developing countries with the financial resources conduct annual searches for national scientists working abroad in microelectronics software areas of strategic importance.

3.5 Restrictive Practices for Microelectronics Licensing

In spite of the wide diversity of both industrial conditions and differing stages of political progress in a number of developing countries, there is evidence that specific policies for regulating the importation of technology have resulted in:

1. the elimination of some restrictive practice clauses from transfer agreements;
2. the shortening of the duration of agreements' validities;
3. a general reduction of prices paid for technology; and
4. a growth of imported technologies to such countries.[40]

There is also some evidence from a number of countries that the imposition of maximums on royalty rates for technology has reduced overall prices of technology contracts, and that various prohibitions on parent/subsidary and subsidary/subsidiary technology payments have contributed to the reduction of royalties. This factor will become especially critical with respect to software since future balance of payments in many developing countries will hinge more directly on software and on content for information utilities than on hardware and equipment.

Finally, in several developing countries, direct governmental intervention in technology markets has, in a minimal way, contributed (slightly) to the reduction of balance of payments for technology. However, the elimination of restrictive practice clauses from technology transfer agreements has more often than not been avoided by tacit agreements to circumvent regulations, and in fact regulations on technology payments may be easily circumvented by transfer pricing.

Also, case studies of technology registries in several developing countries have all concluded that:

1. persons responsible for their operations infrequently possess knowledge of either state-of-the-art technology or the economic/socio conditions in their country to engage in useful evaluation of transfer contracts;

2. although several countries (including all the signers of Andean Pact Decision 24) have forbidden transfer contracts between subsidiary and parent firms to minimize balance of payments in the guise of royalties, using sophisticated financial accounting multinationals

can quite easily make it impossible to determine who is paying whom for what;

3. since bureaucrats in either developed or developing countries do not usually know or understand the productive infrastructure of their country in any great detail, they are often incapable of determining whether or not a specific new technology may be usefully assimilated -- also, since they possess a limited knowledge of the country's indigenous technological infrastructure, they are also often incapable of judging whether a specific technology might be generated in that country rather than imported;

4. since they are often ignorant of world market prices for types of technology, they often have made major errors in setting, "fair prices to be paid for its importation."[41]

These are somewhat dismal observations, but given the recommendation that developing countries should instigate methodical procedures to search for appropriate microelectronics software and hardware, it is (obviously) important that searches be conducted by persons knowledgeable in indigenous industrial conditions and software technology.

3.6 Purchasing Power of the State

An extremely powerful instrument to define the pattern of demand for microelectronics software and hardware involves the national government as purchaser. In many a developing country the government budget is a significant portion of all investments in a specific industry, and the government is also often a major producer of essential industries. It is clear then that governments may, through their intermediary corporations, determine and set policies which favour the development of specific national scientific and engineering capabilities in the microelectronics area. Governmental firms, when purchasing goods, may also set standards which require the use of indigenous technology. However, this implies:

> "that the state firms or departments must be willing, in some cases, to incur possible additional costs, tolerate possible delays as well as venture into possible lower qualitative results, taking all these as necessary risks involved in the learning process and its consequence of the dependent character of the country. Without making virtue of necessity, it can nevertheless be said that such flexibility is not always possible and is seldom present."[42]

In fact, in many developing countries, governmental firms have shown little real concern for technological aspects of development, and many, such as Venezuela, have often acted as if technology were a fixed, exogenous productive variable, whose properties and conditions of access were determined by the foreign supplier which, in a very negative sense, they often are. In other cases, the exigencies of immediate production have pushed aside other developmental concerns.

Any concern with short term economic goals always necessarily precludes "learning-by-doing", and the disaggregation of any technology package -- which is necessary for any development strategy based on experiential learning -- will produce:

> "benefits that almost certainly will not be capitalized on by the plant owner who imports the technology in the first place, but will be external to his project. In the case of a private firm, it can easily be understood that the incentives offered by disaggregation might be minimal, since they might even benefit competitors. The situation should be different, however, in a state firm, where the external economies generated in this way should be seen in the long term from a social point of view."[43]

It is thus recommended that governments and state firms of developing countries aggressively utilize procurement for relevant hardware and software with special attention to essential business software, software for industrial processes and training software.

3.7 Control of Foreign Investment

One means of regulating microelectronics technology imports which has traditionally proved more effective in developing countries than measures such as import licences and tariffs is the direct control of foreign investment. As with import controls, investment policies have had major effects on technological advancement in developing countries, in which the average firm is so tiny that its equipment needs and capitalization are comparatively insignificant, and in which the investment structure is often highly concentrated:

> "in larger firms so that their technology choices, particularly in machinery and equipment, have a disproportionate effect on the technology ruling in a sector."[44]

Multinationals usually have a corporate policy with respect to technology and long term planning horizons. These firms also possess well-developed production and marketing know-how, which are usually not possessed by domestic firms, whose marketing know-how has often arisen only from experiences in the home country. Given these advantages, developing countries realistically must often rely on foreign investment to develop high technology sectors such as computers, microelectronics, electrical equipment, scientific instruments and so forth.

It should be obvious, however, that there are real economic reasons for upping the entrance barriers, when possible, to multinationals which are seeking markets in developing countries, and many developing countries have already created barriers to entry based on investment criteria -- including requirements for matching funds for foreign/domestic investment, and the reservation of certain sectoral investments for domestic production. Thus it is recommended that developing countries undertake strategic assessments of their levels and areas of industrial development, with special attention to the productivity gains and labour effects of microelectronics, with the aim of 1) reserving certain sectors for domestic production; and 2) simultaneously granting incentives to the multinationals (as was the case with Singapore) to develop other sectors. Here there must be a mix of protection and inducement, since purely protective regulations concerning the importation of microelectronics technology always disrupt the free flow of technology between a country and the rest of the world. Benefits to developing countries have included minimal savings in the balance of payments for high technology goods, employment increases, the protection of industries, and some increased growth of indigenous industries. But the major costs have involved price increases, technological inefficiency while the national infrastructure is being built up, and a scarcity of technological goods. Such scarcity in price increases may be tolerated only if the developing country in question is not an international trader or if import controls are selectively applied for certain classes of goods, as was the case with microcomputers and software in Brazil.

Finally, because developing countries often have very small domestic markets, especially for goods of a high technology nature, any protective market stage will grow very slowly compared with international markets. Also, the absence of real competition in technology will result in technical inefficiency.

3.8 Transfer Restrictions

In the developing world, sales of technology are often conditional on acquiring related elements necessary for the functioning of a manufacturing process, elements which might often be obtained in the country themselves. Transfers also frequently involve restrictive practices -- e.g. the forbidding of licence knowledge use after the licence has expired, price fixing, export prohibitions, restrictions on production levels, and so forth. All of these factors obviously lead to an absence of economic decision-making capability on the part of the importing country concerning the commercialization of manufacturing processes and products which are built on foreign technology. Thus, if the developing country is not in a position where the multinationals may massively retaliate against its economy, the following types of microelectronics contracts should be entirely banned:

1. those which implicity or explicitly significantly delay or forbid acquiring related technologies from sources other than the licensor;

2. contracts placing real limitations on export or production volumes;

3. contracts whose time duration is longer than the technological obsolescence period of the licenced technology -- such is especially critical for micro-electronics technologies whose products often have an obsolesence period of 1-4 years; and

4. contracts containing _de facto_ clauses which imply that the recipient must sell, to the licensor, products of the technology at lower prices than those of the international market.

In summary, the main transfer problem in developing countries, we have seen, is that they possess little independence in acquiring technologies; when technology is transferred to firms it often resembles the "high cost rental" of packages of production techniques (which is essentially what obtaining software packages involves) and little effective transfer actually occurs. The second reason for the lack of technological autonomy involves the fact that many of the manufacturing techniques are being transferred to "mixed enterprises" -- whose capital is an indigenous/foreign mix. In such cases it is usually the economic decisions of the foreign investors which are followed, especially those relating to the importation of technology.

3.9 Transborder Data Flows

Much computer software will increasingly be incorporated both into terminals and computers, and in instances in which software has been separately developed in parent firms, it will continue to be sold to third world subsidiaries via interfirm transfer pricing. It is extremely difficult for governments to attach any values to services traded on an interfirm basis, and this problem is complicated when traded items are "information-intensive", such as software. Yet given the future probable growth of world trade in software, this is a problem which must be examined by developing countries.

The imposition of restrictions on transborder data flows out of developing countries may cause more data being processed internally in these countries by domestic service bureaus or by the subsidiaries themselves. Actually, such restrictions might accelerate the computerization of subsidiaries, but as we have also seen, restrictions inevitably result in short-term technical inefficiencies. It is thus recommended that developing countries review their policies toward transborder data flows with the end of strengthening the domestic service bureau/data processing industry.

3.10 Long Term Technology Assessment

- Comparative advantages now are not the result of historical accidents concerning a country's climate or natural resources but are human created. For example, in the case of Canada, her comparative advantage is usually conceived of as comprising her vast natural resources. But increasingly these will be supplanted by the products of the new science of the substitution of materials derived from biological sources grown in the tropics.

As comparative advantages become increasingly dependent on man-made factors, natural comparative advantages are becoming less significant. It would seem obvious then that developing countries must create a real ability for short and long term technology assessment, focusing on the ways that man-made comparative advantages will directly affect their industrial strategic sectors.

Also, given the wide hiatus in most developing countries between the industrial and educational infrastructure, it is mandatory to develop science policies which will forge closer links between these two, and such policies must be conceived in terms of critical economic sectors of the country that could be developed.

3.11 Copying

We have discussed the necessities for developing countries to create an ability to both select and absorb useful microelectronics technology in both the private and public sector. This absorption has traditionally been achieved via a number of ways including imitation or what is called "reverse engineering". Normally it is thought that reverse engineering and imitation (copying) will be much more difficult for software than it is for machinery. However, as anyone knows who has visited the Sum Sui Po section of Hong Kong (where copies of North American perscoms are manufactured), software may be quite easily copied. However, it is more difficult to modify and maintain it in such a manner that it may be utilized in the local productive infrastructure, and special attention must be devoted to software maintenance. One analyst suggests that, "international joint ventures could be developed where production takes place in developing countries and service and maintenance in developed ones due to relatively lower high-skill labour costs and the fact that most of the market is in the advanced countries. This is preferable to subcontracting arrangements (already used) since the latter only covers revenues for production.[45] However, if this arrangement prevails, developing countries will miss out on the major source of software revenue, maintenance.

3.12 Diversification of Software and Hardware Sources

Developing countries are presently dependent on the developed countries for the vast majority of microelectronics hardware and software. Given this situation, the dependency on both integrated circuits, computers and software should be spread out as much as is possible. As Rada has noted:

> "In addition, national guidelines should exist to set standards for the increasing number of suppliers of systems and also specialized components. These guidelines should convey the message to suppliers that they will be authorized in the market providing they show a clear commitment and seriousness(toward the local market). Many developing countries have been invaded by vendors which do not necessarily transfer the necessary back-up to make the applications of components or systems successful. Client support should be one of the standards for operations. The size of the market obviously conditions the willingness of vendors to make a firm and long term commitment, and thus the bargaining capacity of different countries differ widely."[47]

This "necessary back-up" for microelectronics technology must be examined in more detail. With respect to software, we have seen, it comprises software maintenance. (In the product lifecycle for software, initial development costs represented only about 20% of total costs, with the rest going predominantly into maintenance.) Accordingly, the developed countries will attempt to gain software revenue directly through maintenance. However, the actual maintenance of software necessitates a close relation with users, and in spite of much rhetoric about working at a distance over a satellite terminal, it is very arduous to even prescribe software specifications for new applications over telecommunications lines. Thus, in spite of the fact that the developed world will attempt to have new software cheaply written in developing countries and maintained in the developed countries, the developing world must make every attempt to create software trade between developing countries so that they may appropriate the revenues for both software creation and maintenance.

3.13 Equity Participation and Transfer of Technology

The concerns of technology transfer made some sense when applied to non-microelectronics based technologies, whose obsolescence period was usually greater than five years. However, what is now important is not merely to have access to newtechnology, which may soon be obsolete, but to the entire technological/innovation process. One way of gaining this access is through equity capital in the developed countries' firms. Several analysts suggest that developing countries, in other words, should invest in the developed countries for the purpose of obtaining technology, just as the developed countries' firms invest in the developing world to obtain market access. But although equity participation is one of the most prevalent means of accessing technology in both the semi-conductor and software industries, obtaining such equity access in the developed countries' economies, for a number of reasons, is extremely difficult.

3.14 Summary

In summary, developing countries must apply microelectronics policies which incorporate these principles and observations:

1. Although a number of developing countries are now producing their own mini and micro computers, most third world software and hardware will often have to

be compatible with equipment of IBM. In fact, the third world possibility of manufacturing microprocessors is a real possibility only if, "the adopted criteria of obsolesence differ from the international market and the system is developed behind strong protective barriers. The technical specifications of equipment would, however, be several years in generation behind the leading producers. Under these conditions, the manufacturing alternative is only justifiable within a large potential internal market (e.g. India or Brazil) or within a regional market that can permit appropriate economies of scale."[47] Developing countries which seek to produce microelectronics software as hardware, then, must make strategic market assessments which emphasize regional configurating within the developing world.

2. A real concern of many developing countries is that the new microelectronics hardware and software be appropriate to local production conditions, and that it therefore be potentially modifiable. Machinery may be broken down, reassembled and modified -- computers and software less so. The latter, in fact, we have seen, is often now hardwired in chips and cannot be as easily understood and modified as machinery may.

3. New lines of technological dependence may appear, given the economic importance of software. Software sales in the developing world may take the form of consulting, purchased software packages, contractual licensing agreements with provisions for training in original writing, modifying and maintaining software. With the exception of second source agreements, however, licensing has often given developing countries access to obsolete technology, and there is no reason to expect that situation will change with microelectronics software and hardware. Software licensing terms must thus be scrutinized through the criteria previously enumerated -- length of licence, restrictions on export, modification, production levels, etc.

The developing world's policy response to this situation has been to attempt to build the technological capabilities of the productive infrastructure, especially engineering firms -- to identify problems which will necessitate importing foreign technology; to depackage the relevant technological alternatives; and to create favourable negotiating conditions for their application, improvement and adaptation. Instruments utilized to implement these policies must be economic and fiscal in nature and involve:

a) the use of strict preference in government procurement for national firms;

b) selective tax exemptions for the private sector, institutes and universities for imported equipment, such as computers and software, which are used in R&D;

c) export financing for software and services;

d) the granting of tax incentives and special credit lines for the development of new technologies and new applications of existing technologies;

e) a selective awarding of government contracts according to priorities in development plans; and

f) specifically with respect to software, initiatives must, when possible, be directed toward the following: Unpackaging of computer software systems and of large software packets to increase national revenue from indigenously-produced software and from software maintenance. Also, since software may be transported as an intangible or unembodied good across national boundaries, or simply shot across telecommunications lines, developing countries must create import/export policies with respect to computer software. In this context, measures devised by Brazil should be useful.

We have seen that there is a world trend for the multinationals to eliminate their low-labour production facilities in developing countries and move these facilities back to the home country where production is done via CAD/CAM. Thus, it would seem apparent that the future productivity and world competitiveness of the developing countries will ultimately hinge, as it does in the developed countries, on how rapidly and competently firms incorporate microelectronics innovations into products and manufacturing processes.

Juan Rada has enumerated a number of principles which directly pertain to third world microelectronics/software policy and amplify these areas of concern.

The first principle emphasizes this need for technology/marketing policies based on selectivity of applications which may optimize the use of a country's resources rather than merely increasing efficiency. Since future balance payments considerations in many developing countries will hinge more directly on software and information creation rather than hardware production:

> "By avoiding imports of services, the foreign exchange savings could be greater in the medium term than the costs of installing equipment, while at the same time developing local expertise and capabilities. It will also have the added advantage of organizing a system which meets the needs ... of the country. Selectivity can also gear applications to areas where it is beneficial to the country as a whole to maintain international competitiveness."[48]

The second principle involves the need to ensure diversity of suppliers for components, accompanied by specific policies on the creation of software of benefit to the country.

The third principle involves the need to monitor the, "national integration of locally assembled or partly manufactured electronic-based products," because many countries use an index of national integration combining volume, weight and value to determine how much of any specific product is domestically produced. Integrated circuits and software embedded on chips -- main contributors to the "value" of products which incorporate them -- have low "volume", value and weight ratios, and if any country is trying to upgrade efforts in these areas, such monitoring is necessary, especially since many intelligent products have been clandestinely classified under the archaic standard industrial trade classification system (SITC).

4.0 REFERENCES

1 These are all actual uses of Apple perscoms in developing countries.

2 Lewis Branscomb, Science, Volume 215, 12 February 1982.

3 See, for example, R. Wills, "Computer-Based Scheduling and Billing in the Canadian Transportation Industry", Task Force on Microelectronics and Employment Effects, Department of Labour, Ottawa, 1982.

4 Andrew Pollack, "Computers: The Action's in Software", The New York Times, Sunday, November 8, 1981.

5 Alfred R. Berkeley, "Millionaire Machine?", Datamation, August 25, 1981, Volume 27, No. 9, p.32.

6 Juan Rada, "Structure and Behaviour of the Semiconductor Industry", (draft), UNCTC, May 1982, p.62.

7 Input Inc., quoted by Pollack, op. cit.

8 Russel M. Wills, "Microprocessor-Based Media", IDRC, Ottawa, 1981.

9 In this language, any number or letter is represented by a sequence of 1's and o's.

10 Glen Bacon, Science, Vol. 215, 12 February 1928, p.775-776.

11 Ibid.

12 Lewis Branscomb, Science, Vol. 215, 12 February 1982.

13 Edith Myers, Datamation, Vol. 27, #9, August 25, 1981, p.56.

14 Wescom estimates, see P. Booth & R. Wills, National Evaluation of the Telidon Field Trials in Canada, Vol. I, Economic and Marketing Study, Dept. of Communications, Ottawa, March 1983.

15 Globe and Mail, March 25, 1983.

16 Ibid.

17 Wills, 1981, op. cit.

18 The Straits Times, Thursday, December 30, 1982, p.1.

19 Ibid.

20 The Economist, August 8, 1981, p.68.

21 R. Wills, "Softech," Ministry of Industry and Trade, Queen's Park, Toronto, 1982, p.98-99.

22 Quoted in Robin Webster and Leslie Miner, "Expert Systems," in Technology, Feb 1982.

23 S. Papert, Mind Storms (Children, Computers and Powerful Ideas), New York: Basic Book, 1980, p.6.

24 Wills, "Softech," p.137.

25 G. Kearsley and B. Hunter, "Electronic Education," HUMRRO, Washington, D.C. (draft), June 1982, p.7.

26 Juan Rada, "Structure and Behaviour of the International Semiconductor Industry," UNCTC, New York, 1982.

27 H. de Mattos, "Technology and Developing Countries," in ITV Third World Telecommunications Forum, 1979, Vol. I, p.II 2.5.

28 "Transborder Data Flows," SEI, Brazilia, 1982.

29 Ibid.

30 SEI, op. cit., pp.28-29.

31 Special Commission on Software, Brazilia, 1982.

32 Ibid.

33 Personal communication, Sushil Chaterjee, National Computer Board, Singapore.

34 Annual Report of the Economic Development Board of Singapore, 1982, p.37.

35 Rada, "Structure and Behaviour ..., p.201.

36 Kurt Hoffman and Howard Rush, "Microelectronics and Industrialization in the Third World," Development Research Digest, No. 3, Spring 1980.

37 Financial Times (London), July 2, 1979, p.5.

38 "Kagaku Gijuksu Yoran," Tokyo, 1981, pp.22-23.

39 R. Wills, "Canadian Technological Dependence," The Science Council of Canada, Ottawa, 1982.

40 R. Wills, op. cit., pp.38-52.

41 See, for example, "Criteria on Applying the Law on National Registry of Transfer of Technology and the Exploitation of Patents and Trademarks," RNTT Internal Publication, Mexico City, undated, p.11.

42 S. Bario, "Policy Instruments to Define the Pattern of Demand for Technology," IDRC, Ottawa, 1980, p.20.

43 C. Cooper and P. Maxwell, "Machinery Suppliers and the Transfer of Technology to Latin America," SPRU, 1975.

44 F. Sagasti, "Policy Instruments to Promote the Performance of Sand T Activities in the Industries Enterprises," IDRC, Ottawa, 1980, p.20.

45 Rada, loc. cit., p.212.

46 Ibid., p.205.

47 Ibid.

48 Ibid.