# Demand for and applications of extra large EDP systems in the EEC Countries and the United Kingdom in the seventies

## Vol. 2 - Supply of EDP systems

soris - torino

The survey was conducted for the "Commission des Communautés Européennes (Direction Générale des Affaires Industrielles)".

The survey was conducted by a research group from SORIS, consisting of: Andrea Barabino, Serena Giraldi, Donata Leonesi, Guido Musso, Iacopo Muzio, Piero Taverna

Collaborated have:

Enrico Albani, Giustino Gasbarri, Alfredo Montica, Massimo Merlino, from **Praxis Calcolo**

Antonio di Leva, Graziella Pent, Maria Teresa Reinieri, from the University of Torino

Rinaldo Sanna from the University of Genova

The interviews with users and experts have been conducted by:

| | |
|---|---|
| **SORIS** | in Italy and the United Kingdom |
| **SOBEMAP of Brussels** | in Belgium |
| **SEMA of Paris** | in France |
| **Deutsche Revisions-und Treuhand A.G. Treuarbeit of Frankfurt** | in Germany |
| **Rijkskantoormachine-centrale of 'S Gravenhage** | in Holland |

June 1970 / n. 6835

SORIS s.p.a.
Economic Studies and Market Research
11, via Santa Teresa Turin tel. 53 98 65/66

# Demand for and applications of extra large EDP systems in the EEC Countries and the United Kingdom in the seventies

## Vol. 2 - Supply of EDP systems

# CONTENTS

pag.

## CHAPTER I - Hardware and Software Supply

1. Computer Manufacturers ..................... 1
2. The Computer Market ........................ 2
3. The Software Supply ........................ 11
4. Concluding remarks ......................... 16

## CHAPTER II - Hardware

1. Introduction ............................... 19
2. Systems' logic history ..................... 23
3. The computer's components - Development and state of art ............................... 39
   - 3.1. Amplifiers ......................... 39
   - 3.2. Memory ............................. 49
   - 3.3. Input-output devices ............... 63
4. Expert forecasts on hardware ............... 68
5. Conclusions ................................ 74

## CHAPTER III - Software

1. Introduction ............................... 85
2. Operational systems ........................ 86
   - 2.1. Operating from the Console ......... 87
   - 2.2. Batch processing ................... 88
   - 2.3. Remote job entry ................... 89
   - 2.4. Multiprogramming ................... 90
   - 2.5. Interactive systems ................ 91

page

3. Programming languages ..................... 96
   3.1. Languages for scientific problems ..... 98
   3.2. Languages for commercial problems ..... 102
   3.3. Languages for processing symbols ...... 104
   3.4. General languages ..................... 106
   3.5. Special languages ..................... 109

4. Translating programmes: assemblers and compilers .............................. 112
   4.1. Assemblers ............................ 112
   4.2. Compilers ............................. 113

5. Experts' forecasts for software ........... 116

6. Conclusion ................................ 119

In this volume have been esamined:

- the problems of hardware and software supply and the computer market;
- the trend of hardware's technical characteristics, with special emphasis on the current situation and possible future developments;
- the trend of software with special emphasis on operational systems, languages and compilers, and possible future developments;
- the forecasts made by the interviewees about hardware and software.

The analysis of the supply and the market is based on the results emerging from direct inquiries and on studies of the existing literature.

The technical analysis of hardware and software has been made by specialists working with the SORIS research group and with the aid of specialized literature. The experts' forecasts have been analyzed by exploding the interviews.

# CAP. I

# Hardware and Software Supply

## 1. Computer Manufacturers

The industry's principal manufacturers are listed in the following table:

THE PRINCIPAL COMPUTERS MANUFACTURERS IN THE US AND WESTERN EUROPE

| | TOTAL SALES 1968 (millions of dollars) | EDP SALES 1968 (millions of dollars) | MAIN FIELD OF ACTIVITY |
|---|---|---|---|
| BURROUGHS | 650 | 190 | Office equipment |
| CONTROLDATA | 483 | 483 | Computers |
| GENERAL ELECTRIC | 8,381 | 420 | Electromechanics - Electronics |
| HONEYWELL | 1,281 | - | Scientific instruments |
| I.B.M. | 6,888 | 6,475 | Computers and office equipment |
| R.C.A. | 3,106 | 225 | Telecommunications |
| UNIVAC | 1,562 | 500 | Office equipment |
| TOTAL US MANUFACTURERS | 22,351 | 8,293 | |
| I.C.L. | 220 | 220 | Computers |
| C.I.I. | 56 | 56 | Computers |
| SIEMENS | 2,907,2 | 77 | Electronics |
| PHILIPS | 2,400 | - | Electronics |
| AEG-TELEFUNKEN | 1,374.4 | - | Electromechanics - Electronics |
| TOTAL EUROPEAN MANUFACTURERS | 6,957.6 | 353 | |
| TOTALE | 29,308.6 | 8,646 | |

SOURCE: EXPANSION - AUGUST 1969.

## 2. The Computer Market

On the computer market of the European countries and the United Kingdom both US and European firms are present: thus, the European market is much more lively and competitive than the US one The US manufacturers establish themselves not only through the creation of sales outlets and efficient sales networks, but also through the creation of production plants and stipulation of licence contracts with European industries.

During the last few years, the French and the English governments have tried to stimulate the national manufacturers. In France, CII was charged with designing two computer series. In the United Kingdom ICL has the task to carry out the English industrial policy in this industry and receives support in selling EDP equipment in the Commonwealth countries. The present market shares of the various manufacturers in the EEC countries, the United Kingdom and the United States are shown in the following table:

MARKET SHARES PER MANUFACTURER (1969)

| | BELGIUM | HOLLAND | FRANCE | GERMANY | ITALY | UK | USA |
|---|---|---|---|---|---|---|---|
| BULL/GE | 17.2 | 10.4 | 14.5 | 4.6 | 20.4 | 1.5 | 2.0 |
| CDC | 0.4 | 4.9 | 3.1 | 2.7 | 1.5 | 0.9 | 5.3 |
| CII | 2.0 | 0.6 | 4.0 | 0.3 | - | - | - |
| HONEYWELL | 1.2 | 2.0 | 1.7 | 2.4 | 2.2 | 5.0 | 4.1 |
| IBM | 64.9 | 54.8 | 62.6 | 63.3 | 66.3 | 40.3 | 71.0 |
| ICL | 2.6 | 3.4 | 1.7 | 0.7 | 0.2 | 42.0 | - |
| PHILIPS | 0.8 | 8.7 | - | - | - | - | - |
| SIEMENS | 2.9 | 2.2 | 1.0 | 13.2 | 0.8 | - | - |
| UNIVAC | 3.8 | 7.4 | 4.5 | 7.1 | 7.5 | 3.8 | 7.0 |
| OTHERS | 4.2 | 5.6 | 6.9 | 5.7 | 1.1 | 6.5 | 10.6 |
| **TOTAL** | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |

The importance of the American presence in this market is evident in almost all the European countries. The principal American manufacturers have in all EEC countries a share above 80%, with a maximum of 98% in Italy. Only in the United Kingdom, due to the presence of ICL, the market shares of the US manufacturers reach a total of only 60%. On the other hand, European manufacturers are just about absent on the US market.
The American superiority is particularly evident in that part of the market which is of principal interest in this survey, i.e. that of large and extra-large computers which US manufacturers control completely in Belgium, Holland, France and Italy. In Germany and the United Kingdom also ICL and Siemens are present on the large and extra-large computer market, which US manufacturers dominate, however, clearly. The US manufacturers' shares of the large and extra-large computer market are more important than those obtained for the market as a whole, because in the production of these computers the national industries of European countries cannot yet compete, whereas in other productions the national industries of some European countries have begun to offer valid competition to the American presence, this being due partly to the policies of support by governments.

The computer market has currently the structure of an oligopoly, where a single manufacturer, IBM, occupies a dominating position, followed far behind by several minor manufacturers. This structure of the market is the consequence of the vay in which the development of the computer industry has taken place.

Originally, the computer seemed to be an instrument destined to solve problems of military nature, because of the necessities of national defense. The specific technical-scientific nature, which characterized the birth of EDP in the US, is the reason why at first only research centres and the electronic industries dealt with the problems of logic and electronics, inherent in EDP, and developed basic software. Only later was commercial exploitation of the computers studied and along with it purposes which were completely different from the original ones. The possibilities of expanding the computers market were interpreted differently by the firms who wanted to operate on it.

Some electronics firms continued to occupy themselves almost exclusively with research and development of large processing systems, solliciting for this purpose government contracts. In this way, the presence of highly qualified specialists made the manufacturer's assistance in the employment of the machines superfluous.
Other, and especially IBM, directed their knowledge acquired during the participation in government-financed programs towards commercial goals, continuing their policy of exploiting the market of private investment goods.
They engaged themselves thus in the great effort of basic software development, training of qualified personnel and assistance to the users. The objective circumstance which permitted IBM to pursue this policy was the commercially predominating position which it occupied on the keypunch machine market.

IBM aimed basically at the substition, in a market which it already controlled, of a system centered around keypunch machines by another one based on the computer, making the client accept the transition as a natural consequence, and not a forced one, of technological progress and channelling into a new direction the already previously established realationship with the client.
And, in fact, IBM created the two computers which were most suited for the transformation of a UR centre into an electronic one (650 and 1401) and used two strategies (1): leasing and assistance.

## A. Leasing

Leasing, already widely used for keypunch machines, has shown to be a highly effective means of giving the computer market a precise qualitative and quantitative structure.
From the user's point of view, leasing has several immediately visible advantages:

- possibility to acquire and experiment with a service (whose importance is often ignored) without having to burden himself with a prohibitive investment;
- possibility to unload on the lessor ordinary and extraordinary maintenance of the installations;

---

(1) In 1956, a verdict of the Anti-Trust Commission ordered IBM to include sales of its own products as an alternative to leasing The verdict was, however, inefficient because it left IBM the possibility to direct the clients towards leasing through maneuvering prices, tariffs and cost of maintenance and technical assistance.

- possibility to cancel the contract on short notice, in case the installation functions badly or at the moment it has to be substituted because obsolete.

From the manufacturers view point leasing of computers requires the capacity to meet a great financial engagement(1). In reality, the manufacturers' risk is considerably greater and experiences made by the industry show how easily it can happen that able and commercially fortunate manufacturers find themselves in the situation of a deep financial crisis, arising from their success in leasing, which has often fatal consequences.

For the big computer manufacturing firm which has its financial resources well balanced with the development of the installations, leasing offers many possibilities of penetrating and controlling the market.

In the first place, being proprietor of a great share of installed equipment, the leader firm can control the process of the installations' obsolescence by directing the rythm of technological progress and exploiting to a maximum the investments already made.

Secondly, the apparently "disinterested" position which the lessor takes gives him cultural prestige with the users. Thus, his function of technical advisor is emphasized as the users become aware of EDP's meaning. Leasing becomes thus

---

(1) Usually a firm starts to make profit only during the fourth year of the installation of leased equipment.

a natural vehicle for the establihsment of a close relationship which can, in turn, be exploited profitably for promotion, that is, when convincing the clients that they should extend their installations progressively.
From what has been said it becomes evident to what extent leasing has contributed to giving the computer supply an almost monopolistic structure, all to the advantage of the most experienced and smart manufacturer (IBM).
The small manufacturers, who have a financial and organizational handicap, are unable to pursue an autonomous policy.
Once they have accepted the "rules of the game", excepting exceptional circumstances, they are limited to a marginal role on the market which cannot be changed without unbearable efforts.

## B. Educational aspects of the before and after-sales assistance

The EDP monopoly consists, above all, of control of the exploitation of the technological knowledge, first through identification and definition of the problems, then through the choice of the optimal solution.
It is typical that IBM has constantly attributed great importance to an educational policy toward the clients, beginning with the pre-sales phase until true and proper assistance during utilization. The results of this policy have been so effective in terms of customers' faithfulness that paradoxically IBM has managed on more than one occasion to sell its machines before the definite projects for their production were completed. This phenomenon is explained by the fact that in the EDP field the classic principles of industrial marketing are reversed: it is not so much a

question of interpreting the needs for supply as to provoke and direct them. For this purpose IBM has directed its efforts mainly towards persons instead towards firms, establishing intimate and continuous contacts.
Its training courses for EDP personnel have increased over the time in three aspects:

1. qualifications of participants (chiefs of EDP centres, programmers, analysts, business consultants, managers),
2. number of participants (there are many courses in many places),
3. arguments dealt with (seminars about advanced techniques to which specialists in the various matters come from all over the world).

Influence on firms is thus exerted from the inside as a consequence of the abovementioned policy. As the computer begins to become an integral part of the users' organism, new "centres of competence" are being formed which will change the traditional power structures on the business level, while on an inter-business level the osmosis of problems posed and solved will intensify according to a common ideological matrix.

In this situation, entry of competitors into the market can take place only under the condition that they communicate with the users in the same manner used by the industry's leading firm.

* * *

According to widespread opinion IBM even favoured technological innovations made by its competitors through a policy of high prices, practised in the sixties, which became, moreover, necessary because of the high cost of the services included in the equipment's price.

In reality, giving its competitors the opportunity to cope with the high cost of research & development of new models, IBM knew that in the end all the services offered by them would be inferior in quality and quantity and that, at the same time, the competitors would be weakened because of the great R & D effort.

In other words, IBM, having understood that the supply's centre of gravity moved from hardware to other "immaterial" elements of the system (software, system analysis, industry competence), was able in the sixties to profit from the equipment's innovative process to a larger degree than its own contribution would have justified.

Also with regard to applications the market was influenced by the software supply, developed especially for those applications derived more directly from the substitution of key-punch machines by computers.

These applications require, among other things, less sophisticated hardware. In this way, also the delay in the introduction of computers into Public Administration can be justified, because standardization of the applications is not so easy here as in business.

In Europe, the US model of the electronics firms did not find the necessary prerequisites for development because of different environmental conditions, despite the fact that the model was followed which in the US produced such formidable economic results.

European firms active in scientific research suffered from a lack of government support; as a consequence development of European computer firms could take place only by following the typical model of the firms manufacturing office equipment.

Much like what happened in the US to IBM, the existence of a net of commercial relationships, a sales network and market experience determined the success on the market of those firms who had already manufactured keypunch machines.

But the factors which slowed down the expansion of the European firms were the limited size of the national market on which they operated chiefly, the limited financial resources destined for computers by the various firms, a too large number of manufacturers present on the market and the always agressive competition by US firms.

To change this situation, the firms resorted sometimes to a merger into a single firm of the various national ones, guaranteeing them thus a predominant position on their own market (this happened in the United Kingdom) and avoiding being absorbed by US firms, as happened in France and Italy where General Electric absorbed Bull and the EDP branch of Olivetti.

## 3. The Software Supply

During the last decade the software supply in the US has increased at a tremendous rate, as illustrated by the following table. In it we have listed separately the expenditures of manufacturers and users, as well as the sales of independent software firms.

ESTIMATE OF SOFTWARE EXPENDITURES IN THE US (millions of dollars)

| YEARS | MANUFACTURERS (expenditures) | USERS (expenditures) | INDEPENDENT FIRMS (Sales) | TOTAL |
|---|---|---|---|---|
| 1960 | 200 | 350 | Less than 3 | 600 |
| 1961 | 330 | 670 | | 1,000 |
| 1962 | 500 | 1,100 | | 1,600 |
| 1963 | 620 | 1,300 | 5 | 1,925 |
| 1964 | 800 | 1,600 | 20 | 2,420 |
| 1965 | 1,000 | 2,100 | 50 | 3,150 |
| 1966 | 1,100 | 2,600 | 100 | 3,800 |
| 1967 | 1,150 | 3,800 | 180 | 5,130 |
| 1968 | 1,200 | 4,500 | 250 | 5,950 |
| 1969 | 1,300 | 5,450 | 400 | 7,150 |

SOURCE: EDP INDUSTRY REPORT - MAY 15, 1968.

Very little of the amount spent by computer users, estimated at $ 4,500 million, or just over 75% of the $ 5,450 million expenditures in 1968 - is available to the independent suppliers or to computer manufacturers through separate pricing. Major expenses are primarily for normal programming in the course of daily computer operation, or for software which is so specialized or proprietary that it must be handled by the computer user.

Software expenditures by computer manufacturers normally range between 15% and 25% of the value of shipments for companies that have a generally broad range of machines, Nevertheless, the total cost for software is going to be borne by the computer user - whether it is for separately priced software, or it is included in the basic charges for computing equipment.

Over a given period of time a U.S. user pays more money to his own employees to <u>support</u> a computer, than he pays to the manufacturer to <u>provide</u> that computer even though the manufacturer supplied software is covered by the latter payment.

As was confirmed by the interviews conducted with the users for this survey, the software expenses in the European countries account for a smaller part of the total expenses. The discrepancy between Europe and the US in this respect is due to a large degree to the fact that basic software is produced essentially by the US manufacturing companies. It is beyond doubt that further development of EDP depends in all the world from the reply which software supply- in the most varied forms - will be able to give to demand. Computer manufacturers tend to have a cyclical need for systems programmers and analysts. Their need peaks during the development and installation of a new generation of computers. Since completely new lines of computers are expected to occur less frequently than in the past, the highest staff requirements will be felt at longer intervals. The requirements for staff imply a large build-up of personnel who must be paid for, whether shifted to applications programming or other activities. Either the company must retain staff expertness and continuity as part of maintaining competitive posture or, if the cost appears too high, release competent and expensively trained personnel.

The latter could pose a potential threat to manufacturers and strengthen the independent programming firms, the likely beneficiaries of the migration of talents.

As programming's proportion of revenue from EDP investment continues to increase, some computer manufacturers could have the best combination of resources to meet the total needs of an information processing services industry, and consequently to change the market strategy of providing free software.

In fact, there is some evidence that marketing techniques will shift in the future: the small manufacturers will shy away more than large firms from supplying free programs not closely related to hardware sales.

Users are beginning to understand the significance of the large investments necessary in programming. As the programs become more powerful and the users become more sophisticated, they are willing to pay for a guaranteed and maintained programming package.

To the manufacturers, not serving this market could mean eventually remarkable losses and less communication with the end user.

The adoption of separate pricing for software - already effective in the US and soon also in Europe - can have considerable effects and deserves, therefore, a more profound analysis.

As a matter of fact, the economic structure of separating the pricing of software from hardware seems to have within it a quality which is self-defeating both for the computer manufacturers and for the user.
The latter - which would have rented a system equipment for 100/month under the old pricing scheme - is faced with the choice of saving 3/month and buying his software elsewhere, or paying 115/month for hardware-software supply.
Everything else being equal, an outside software vendor can meet manufacturers's software price by offering a package to the prospect for 18/month. It is apparently a good bargain for software houses.
One must recall that the per-unit cost of software is peculiarly sensitive to volume, because reproduction cost is very low and design cost is very high. Therefore, the software producer with the highest volume will have the lowest per-unit costs. This effect is much more pronounced than in conventional mass-production equipment manufacturing where the reproduction cost dominated the design cost.
As the small manufacturer loses software customers to independent software houses, his per-unit costs go up, he is forced to raise prices, and he drives away more customers.
Finally, the separation of software pricing could give rise to an incentive structure which encourages the small manufacturer to abdicate his present role as a supplier of software, system integration and associated services.
This could have two consequences:
(a) hardware price competition vs. IBM, and
(b) a user tendency to buy IBM-compatible equipment -,
that would be particularly effective in the small

system market, where hardware margins are higher and volume differences between small and large manufacturers are greater.

The net effect, then, would be to strengthen IBM's position in the small-system market.

Some further considerations can be made about the impact of separate software pricing on what involves the help provided to a user in designing and programming his particular application, and in making sense of the software he gets. This support - previously offered "free" by the computer manufacturer - will be most reasonaly provided by an independent contractor or by the suppliers of system and application software.

The major significance of separate software pricing is that is promises to drive a wedge into the marketing relationship which now exists between the user and the computer manufacturer. The wedge is labeled "prime contractor" and the role is most probably occupied by the software houses.

## 4. Concluding remarks

The product offered by the computer manufacturers is not a tradictional product, but an information system consisting of hardware, software and organizational analysis. Consequently, the firms who want to operate on this market must assume the role of a "systems industry" and not that of a traditional manufacturing firm.

The ability to form and manage such a complex process brings into existence firms with completely new characteristics where all managerial techniques which were used separately until now, have to be integrated.

IBM has known how to carry out on a worldwide basis this transformation and it has thus managed to leave behind its competitors, creating a real managerial gap.

When trying to single out the possible lines of development of the EDP supply, the roles of the firms have to be divided depending on whether they want to assume roles which integrate with or substitute IBM's role.

By integrative industries we mean those enterprises which have directed or are directing themselves towards fields in which IBM is less interested. They are firms manufacturing highly specialized subsystems (auxiliary apparatus, integrated circuits, peripheral elements of the computer, software houses, etc.) or equipment for special purposes.

In both cases they integrate with the large manufacturer Development of these firms will depend greatly from the degree of specialization which they can achieve.

By substituting industries we mean those firms who try to

compete with IBM in "big computer systems", despite the fact that history of the computer manufacturing firms has shown that clients feel a "credibility gap" between IBM and its competitors.
The latter have always tried to imitate and often to anticipate the IBM model. IBM, for its part, has never renovated its production line, unless forced to do so by competition. But despite delays, it has always been able to destroy the initial advantage of the competition and to reaffirm its predominance. This is due to the close relationship established between user and manufacturer, a result of its "system oriented policy".

As a matter of fact, hardware sales are only the culmination of the service offered the user which begins with the determination of his specific needs, continues with the study of applications and establishes itself firmly with assistance, maintenance and renovation of the installations, and expansion and integration of the system in order to meet new requirements. In this way the service offered assures future sales of other products. In fact, one product can be substituted with another of a different brand, but it is difficult to replace a computer which makes part of a system of products and services offered by the same firm, and the probability that it will be substituted with another from the same firm is very high. However, a slowdown in technological development can arise from the leasing policy and the predominance of the market by one manufacturer. Mitigation of the present monopolistic character of the market would be desirable. But this result cannot be obtained easily, if the other manufacturers are not able to close the credibility gap with IBM by putting themselves on the same "system" supply level (by system we mean a coordinated and finalized entity of products and services), instead of dealing only in products, even if they are technologically advanced.

Finally, concerning diversification of the hardware and software supply, it is feasible that also IBM will orient itself toward diversification of its production, concentrating its efforts on the more profitable aspects of EDP, such as software or systems.

In this way, it could force also the substituting hardware manufacturers into the role of producers of subsystems.

# CAP. II

# Hardware

## 1. Introduction

This section is essentially based on a number of survey papers (1) (2) (3) (4) (5), to which reference is principally made. Other reference sources are quoted in the text. It will be apparent that our main information comes from US literature. In view of the world-wide circulation of ideas and people, sharing of experience, partaking in patents which are proper to the computer field we do not think that this gives objectionable biasing to our survey.

Concerning the development of hardware, account was taken also of the interviews with the experts, although little information has emerged from these interviews which a complete assessment of the future developments.

---

(1) S. Rosen, "Electronic Computers: A Historical Survey". Computing Surveys 1, 1, 1969

(2) N. Nisenoff, "Hardware for Information Processing Systems", Proceed. IEEE 54, 12, p. 1820 (Dec. 1966)

(3) C.J. Walter et al., "Setting characteristics for fourth generation systems computer", Computer Design Aug. 1968 p. 45

(4) G.M. e L.D. Amdahl, "Fourth - Generation Hardware", Datamation Jan. 1967 p. 25

(5) E.C. Joseph, "Trends in Computer Development"

The extremely high rate of expansion in size, the outstanding feature of computer industry from the point of view of economics, is also apparent from that of engineering. Measuring computer size by the number of logic gates an increment of order of $10^2$ is found to occur in about 20 years, from $1.8 \cdot 10^4$ logic elements of ENIAC (1946) to $\simeq 0.5 \cdot 10^6$ active, $\simeq 1{,}0 \cdot 10^6$ passive components of high-speed logic of todays typical medium/large computers.

This wide dynamic has its roots in the very essence of computer logic. While in other fields of electronics speed and performance are fundamentally ruled by the one-dimensional factor of attainable frequency treshold, logic circuitry inherently lends itself to expansion in the second dimension of parallel mode of operation, a bargain always being possible between increase in speed and increase in number of parallel paths. In fact, the "parallel vs. serial" controversy dominated the field since the early beginnings, the "serial" trend being supported by tradition (i.e. by the scientific and technical achievements in the fields of communication, TV, Radar), the "parallel" one by the intrinsic prerequisites of the art, involving both design and production choices. As a picturesque landmark in this controversial panorama of evolution we may recall the attempt to resort to purely serial mode in the "magnetic drum - computer" logic (Harvard University Mark III, 1950; IBM 650, 1953).

Nowadays, the problem is still present as a major one, e.g.

in connection with the wish to accelerate information transfer to and from external memories, disc/drum random access memories being presumably but an intermediate stage of evolution towards multi-path (static) devices. It also dominates the philosophy of high-level machine logic, bearing in one direction to the increase of parellisation through multiprogramming, multiprocessing, memory overlapping procedures. In the other direction to serial procedures, as essentially are those of time-sharing central processors among several independent peripherals via multiplexing units.

The conditioning impact of the parellisation trend on production techniques, with subsequent backlash on research trends, deserves great attention. In fact, batch (i.e.,parallel) production capability has been so far the dominant factor to decide survival of species in the numberless genus of basic devices that have been, and continuously are investigated, developed and tested by R & D divisions of companies, University laboratories etc. As of to-day, reliable LSI is the aimed-at target to satisfy this basic requirement. A highly evolved offspring, if compared to its 20-years older ancestor, the single hard-tube flip-flop.

To back the great effort required by keeping the pace set by this quick evolution, industry must master the various factors affecting its dynamics.

- Control over scientific, technologic, engineering development of components,
- Capacity of large scale production of these components within known tolerance limits,
- Ability to locate applications opening profitable mass markets,

- Power to support mass applications by adequate advisory, marketing, servicing organisations.

Joint control over the whole of these factors is a required minimum to support development, production and marketing of complex, composite, huge instruments like large digital computers.

The firm who first succeeded in seizing this control also seized the world market leadership which it still holds. A fact hardly attributable to mere chance, as are hardly those of the history of many unsuccessful attempts both in the USA and Europe. Here e.g. Italy's efforts both by private and public enterprises (Olivetti, C.S.C.E.), were backed by very deep scientific knowledge and sound engineering practice. Their failure is explained by reasons that are not difficult to trace back to a lack of control over one or more of the factors listed above.

Merging of many medium and small independent companies into ICT, which was sponsored by the UK Government with the main aim of reaching the stimated minimum of financial dimension required by a single producer (annual budget of Lst. 100 million), has probabily achieved the target also of securing this control by an European group. Of course, a joint policy by the whole of Europe's industry could give a still higher chance of challenging the present American supremacy (1).

---

(1) Cfr. M. Rose, Computers, Managers and Society, chap. 9.

## 2. Systems' logic history

Control of the first two of the factors listed above (p. 21) was on hand by most electronic research centres at the end of World War II as a consequence of the great progress which had taken place since the "penthode" year (1936) in the fields of radio communication, radar and servomechanisms.

It was then possible since 1943 to design an all-electronic arithmetic machine making large scale use of the simple toy invented by Eccles and Jordan back in 1919. As a matter of principle, many doubts existed on the chances of the working of a system relying on the joint operation of many thousands of components of widespread characteristics. But the question was put aside and the answer awaited from pratice. Thus was originated ENIAC, the first all-electronic machine performing arithmetic computing. A task it accomplished at Aberdeen, USA, from 1946 up to 1955.

Little or no consideration was given at that time to the other two factors whose overwhelming importance would be shown only by the subsequent experience. Actually, the first two externally programmable computers, EDSAC and EDVAC, were conceived in USA and respectively in the U.K. inside university circles. These projects were based on scientific requirements and regardless of industrial/economic considerations, military applications being the sole "impure" scope taken into account.

These early projects brought with them a number of consequences. The first was to stimulate the theoretical study of the basic logic operations and to start those on what would become the "software". Among main topics met by these investigations are those of optimisation of addition by parallel adders (involving the problems of carryover), of truncating and rounding off results, of protection against mistakes by means of redundant codes. The "floating point" arithmetic to exploit the full capacity of registers is one of the results of these studies, a "software" problem as initially approached by program subroutines. Another result was to individuate the problems of machine structure: e.g. that splitting the main memory into two sections, one for storing data, the other for programs, would double computing speed at the expense of little complications of hardware.

Basic theoretical problems as those of minimizing combinatory and sequential circuits were also studied in this period. One of the topics involved is that related to conversion from the asynchronous mode of operation, typical of relay and delay line circuits, to sinchronous mode. A problem that is revived today by effect of the need of merging subsystems working at different clock rates, and therefore implies that of buffering.

Perhaps the most important of these outcomes is to have evidenced that a quality step was required both by active (amplifier) and passive (memory) elements. Not the hard-tube flip-flop nor the delay-line circulating memory proved to be compact and reliable enough to enable large logic networks to operate efficiently. Efforts to climb this quality step were addressed along several lines, yielding the first practical results by the electrostatic RC tube memory system (Williams and Kilburn, Manchester Univ., U.K., 1949) and the magnetic amplifier (UNIVAC, USA, 1955). Substantial improvement of system performance was allowed by these devices which, however, would be in turn soon superseded. Among the explorations accomplished in this period one of the most promising was that of the transistor, quickly developing from the early point-contact device. The association of fast transistors to magnetic core memories (PHILCO, USA, 1954. MIT, USA, 1950. RCA, USA, 1953), marked the transition to the second generation of computers and opened the road to E.D.P. as an independent, mass-market oriented activity.

Once the first experimental computers started operating,the perspective uses of the new tool became apparent and attractive to the degree to arouse a great potential market demand. Thus a number of projects were launched by University laboratories and manufacturers, such as Univac, Rand, RCA, Raytheon and many others in the USA, Ferranti, English Electric, Bull, Philips and others in Europe.

UNIVAC I (1951, USA) is the first commercially built large computer, a very advanced design at that time including an original tape system, a buffered system that could read forwards and backwards at speeds comparable to some recent ones. Around this machine and by participation of Remington Rand Corp was formed the first commercial enterprise specialized in EDP, the UNIVAC.

Although it was initially years ahead of competition from a technical standpoint, this firm soon met many difficulties. These are mostly attributed to their failure to adopt hardware leasing policies.

The main memory of UNIVAC I was delay line, mercury. Magnetic core memory was first introduced by this firm into UNIVAC II, delivered in 1957/58, with two years delay com -

pared with competitors.

IBM entered the field of automated computation by way of electromechanical equipment designed to complement its line of punch card machines. To these well proved, very diffused devices was first added auxliliary relay logic, then partly tube logic.

Model 604 computer, an electronic calculating punch including over 1400 vacuum tubes, had been delivered since 1948 to customers in the business field. This enabled IBM to sell thousands of machines, thus acquiring a very solid position in the field of data processing. With these machines emphasis was displaced from the scientific computing field to that of - accounting and business applications which are still nowadays the most important ones. In this manner the third of factors listed on page 21 came into operation with two main results, a) to allow IBM self-financing of further expansion, and b) to strongly support the "100 percent reliability" attitude, inherent to accounting applications, that so favourably conditioned the subsequent develpments of EDP.

On the other hand, this circumstance caused a too cautious attitude of the firm towards large scale computers. While the USA National Bureau of Standards was negotiating its contract to obtain the Univac I for the Bureau of Census, IBM still contended that magnetic tape was untested, unreliable and risky as support of permanent digital information. Conversion to new ideas begun in 1950, and in 1953 IBM delivered its first 701, a large scale scientific computer using a 2048-Word Williams tube memory backed up by magnetic drum and magnetic tape storage. By that time the firm announced model 702, a character-oriented computer with $10^4$ characters of cathode-ray tube memory. The first 702 was delivered early in 1955, but soon it became clear that the machine was inadequate in a number of important respects. Above all, the electrostatic memory did not have the reliability required by bussines data processing applications.

The firm quickly reacted to this situation and the 702 was withdrawn from the market, an effort that caused a major strain in its financial resources. The successor to 702, model 705, delivered since 1957, made use of a much more reliable magnetic core memory. Another important improvement was introduced, the Tape Record Coordinator, a tape controller containing 1024 characters of magnetic core storage plus associated logic circuitry. The addition of several Tape Record Coordinators to a 705, although very expensive, made it a very powerful data processor. By 1959, the year that marks the start of the second and transistorized computer generation, the 705 was firmly established as standard in large scale data processing.

Meanwhile, in the scientific computer field the electrostatic storage of the IBM 701 was very unreliable compared with the mercury delay line storage then in use. When magnetic core storage became available, a 701 M computer was planned (later delivered as Model 704).

The 704 provided a three-index register, built-in floating point instructions, and a minimum of 4096 words of magnetic core storage with a 12-microseconds cycle time. First delivered in 1956, it was quite outstanding for its time and achieved a near monopoly for IBM in the large scale scientific computer field.

The only competition was provided by the Remington Rand 1103 series, collectively known as Univac Scientific computers, which were considered by many of their users to be superior to the IBM 700 series. However, there were relatively few installations of the 1103, apparently due to the lack of control over the fourth of factors listed on page 21. In fact, deliveries of 1103 computers were late and technical support poor.

Further improvements were introduced by IBM to its 700 series in model 709 having a new input-output system that permitted reading from tape or cards, writing to tape or printers, and computation to proceed simultaneously. This was made possible by time sharing the core memory between the central computer and as many as six data channels. Variations of this approach of internal buffering have become a standard feature of most computers, even small ones, in recent years.

Similar technical achievements, but with lesser financial success, were attained by other US companies. Among the large computers produced in this period are Datamatic/Honeywell 1000; RCA Bizmac, Ramo-Woolridge 400 and others. In Europe, the first computer to use core memory was the Bull-Gamma 60.

In synthesis, at the end of this period, i.e. about 15 years since the first projects were launched and about 10 years after the first significant applications, EDP had grown up to an important industrial activity. Among the main technical results achieved in this period are:

- A number of safe and tested components and devices, such as high-speed memories, tape, drum and disc devices as auxiliary and mass memory, input/output devices including card/tape readers and printers. Unfortunately there remained a weak, but essential element in this chain of basic components, the thermionic-tube switching amplifier with its inherent inefficiency as closed switch, erratic functioning in open circuit condition, large amount of wasted power for heaters.

However, the market response had been extremely encouraging, evidencing a large potential demand of computing systems from small to very large ones. Except perhaps by application to automatic process and machine tool control, in all fields of application the high cost of EDP was accepted in view of the unique services it offered.

In 1954 Philco (USA) was ready to mass deliver surface-barrier high speed transistors. Not much later alloy-diffused junction transistors made their appearance. These components allowed to realize the much expected compact, "cool", efficient bistable device that would match the impedence characteristics of diode gates and ferrite core memories. Better than a "second generation", this was a "species" that had attained all conditions of survival and multiplication.

The first commercially built transistorized computers were medium speed, business-oriented systems such as National Cash/GE model NCR 304 and RCA model 501. In 1958 IBM launched the 7080, a transistorized extension of model 705, and model 7070 replacing and superseding it. In 1959 Honeywell announced its model 800, a high performance system in the medium price range. The 800 had a very interesting hardware-assisted multiprogramming system with eight sets of sequencing and control registers time-sharing the control and arithmetic circuitry. Burroughs came out a bit later than the others with a very advanced computer, B 5000, which was the first commercial computer system to make use of a tree structure in logic store organisation, along the lines of the Rice University computer. This machine was also probably unique in being programmed almost exclusively in high level languages (variants of Algol, Cobol and Fortran). The success of this machine and of the others mentioned above depends largely on the efforts their manufactures devoted to compilers, COBOL by RCA 501, FACT by Honeywell 800, ALGOL by B 5000.

Transistors also allowed the construction of comparatively small, economic computers having at least equal performance to that of previous large-size tube computers. Among these are IBM 1400 (1963), RCA 301, CDC 160, Honeywell 400, Burroughs 200, GE 200, NCR 300.

These computers proved that some models could be sold by the thousands and that there was a huge market for small computers.

In the sector of very large computers IBM officially introduced in 1960 the 7090, a fast machine having 2.18 μ secs memory cycle time. This computer was designed as successor to the hard-tube model 709 (cycle time 10 μs ) and could use its software. The first delivered system proved to be of very poor performance, depending on unreliable operation of the main memory. The trouble was soon located and corrected by use of forced air cooling of the memory. The 7090 then became an extremely reliable computer, and hundreds of 7090 systems were delivered although the typical price was over US $ 300,000.

A lower price version of somewhat reduced performance, at a considerable lower price, (models 7040 and 7044) has been very popular since 1962/63.At about the same time Univac delivered its model 1107 which used 128 registers of magnetic. film storage as an addressable control memory along with conventional magnetic core and magnetic drum memory.

The compactness of the new transistor devices also encouraged super-computer projects such as LARC (Univac) and Stretch (IBM), both presented in 1956 but based on quite different approaches.

LARC was based on the use of components of proven reliability, such as surface-barrier transistors, Stretch on the very much faster drift transistor then becoming available. It may be noted here that the previous IBM policy had been reversed from a conservative to an almost adventurous attitude.

This depends, of course on the fact that in the latter case the risk lies mainly with public expenditure, the prestige of the firm being solid enough to withstand possible failures. Perhaps a fifth factor should be added to those listed on page 3 on the basis of this experience: the capacity of accepting risks (financial or prestige) which is limited to the sector of advanced projects, providing a solid background of sizeable standard market production exists. Although the two advanced components employed by project Stretch, drift transistors and fast (2 $\mu$ s ) memory, eventually proved very successful , this project (and LARC project) must be evaluated as a failure from the point of view of profit, but both were very successful in providing a major stimulus to the computer industry from 1956 to 1959.

The project Stretch e.g. constituted the basis for the development of the commercial model 7090, the most successful large scale computer any company has marketed.

In the same sector of super-computers Control Data Corporation had begun (1961) working on the CDC 6600, a very large system of higly advanced design. A very powerful central processor using multiple arithmetic and logic units is connected to ten peripheral processors which are themselves small computers.

An executive control allows the peripheral processors to direct, monitor and time share the central unit the speed of the central unit is over 3 million operations per second. A still improved version is the CDC 7600 which makes use of an Extended Core Storage, a large magnetic core peripheral memory designed for block transfer to and from the main memory at the rate of 10 million 60-bit words per second. The first programs run on the 7600 in 1968 indicated a speed of $25.10^6$ instructions per second.

It must be emphasized again that all these advanced projects were supported by public expenditure through government organisations such as the U.S. Atomic Energy Commission, US Air Force, Department of Defense.

In the field of supercomputers the project ILLIAC IV by the Univ. of Illinois deserves special mention, especially because its basic concept may eventually consistute the design basis of the future fourth of fifth generation of cumputers, once the open problems of multi-gate logic blocks are economically solved.

ILLIAC IV is based on superparellisation by use of a multiplicity (256) of independant processing units, each of which has its own thin film memory of 2048 64-bit words and its own high speed adders for full 64-bit floating-point operation.

These projects of very large, very fast computers started during the second generation period and are stretched over a period of time that crosses the boundary of the third.

To summarize by the late 50's and following a decade of impulsive, controversial development the "second generation" computer design was stabilized around a number of typical elements. These are:

1. High-speed switching unit, including numerical register and logic control/routing circuits, working in parallel mode in binary or binary-derived codes, based on transistors as active logic components, diodes as logic gates. The constructive trend was to assembly transistorized circuits on plug-in modular boards, with printed wire technique connections.

2. High-speed ("main") memories, consisting of arrays of ferrite cores, coincidence-driven by transistors, sometimes via magnetic switches, for storing instructions, intermediate results etc.
   Modular construction of memory arrays and stacks, inherent to the coincidence-matrix technique, pointed the way to "regular" structures.

3. Low-speed ("secondary") memories for storing permanent information (data/program files) onto magnetic tape in servo-driven, serial access transport systems. Intermediate memory on magnetic tape or magnetic drum, transistor-buffered for high-speed release of serially retrieved

information.

4. Input-Output devices including direct-to-tape access keyboards, card/tape readers/punchers, tele-typers, electromechanical printers, magnetic tape recorders/readers, devices for transferring information from one kind of support medium to others.

The years 1958 to 1964 are those of stabilization and improvement. In fact, the consolidation of the second generation of computers and the advent of the third took place without fundamentally new devices being introduced, the general technology steadily relying upon solid state devices as active circuit element and on magnetic devices as memory element. For example, the magnetic-disc random access memory, perhaps the most impressive characteristic element of the third generation computer as seen from the users' end, is a natural development of magnetic tape and magnetic drum, whose fundamental principle it shares. Integrated circuits, the "chips" constituting a major distinguishing characteristic from the production point of view, are a natural landing of automated solid-state circuits production techniques. The family concept is a spontaneous outgrow of industry change-over from the production of special task, custom-tailored large computers to mass-produced, mass marketed large systems.

The distinction between the second generation and the third is not very definite. The major technical development characterizing the third generation are monolithic integrated circuits, although many of the computers of this generation still rely on discrete components. This is true for the IBM 360 series up to model 76 and for the very advanced CDC 7600 now being delivered.

In fact, some of the manufacturers claim that it is the performance of the computer that characterizes its belonging to the third generation. Among these performance criteria is that of standardisation of such computer characteristics as instruction codes, character codes, units of information, and modes of arithmetic. For example in the IBM 360 series this standardisation, leading at least theoretically to compatibility of programmes among computers of very different sizes and speeds, lead to the "family" concept. It is implemented mainly by the technique of micro-programming in read-only memories.

Another character of the series is the ability to support a large variety of input/output as well as peripheral storage devices by means of a "standard I/O interface". Multiplexing channels are also provided so that 360 computers can operate in a single process as well as in multiprocessor operation. The 360 line has had a very great impact on the computer industry, many of its features having been accepted as standard also by other manufacturers. RCA Spectra 70 series, e.g., is almost completely compatible with the IBM 360. These RCA models use monolithic integrated circuits.

Another important characteristic of the computers of the third generation is the generalisation of time sharing features, deriving from some basic concepts first introduced in very large computers which are briefly reviewed as follows.

The British project ATLAS (started in 1959 at the Manchester University. Ferranti Ltd.) uses a single-level storage system to solve the related problems of overlay and hierarchic storage organisation and the allocation of main memory in a multiprogramming environment. The system can serve simultaneously large numbers of users by a peculiar time-sharing concept.

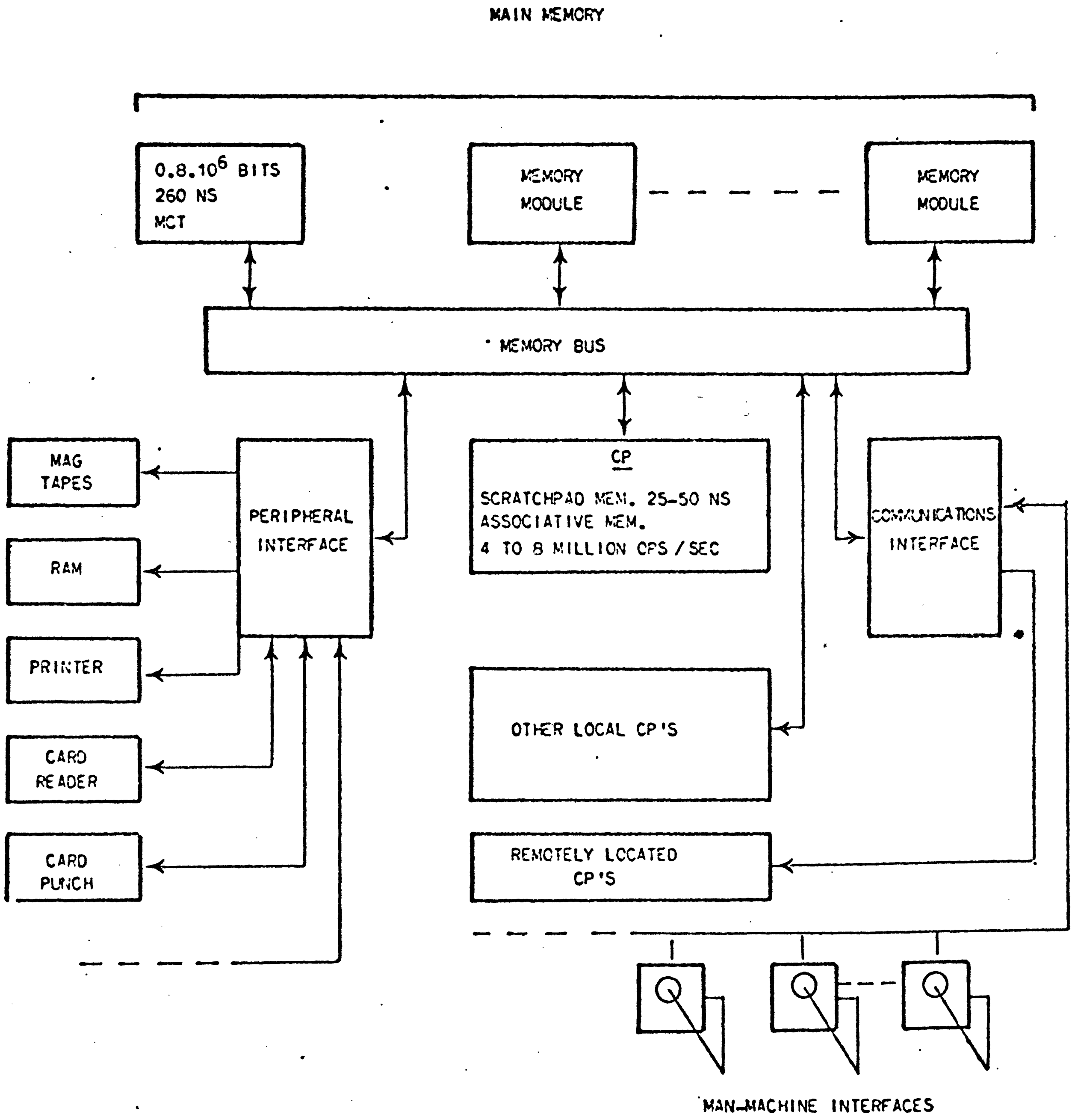
Main memory actually contains a limited number of logical pages, while the rest of the program is in a fast auxiliary storage. The same logical page may be in and out of the main memory a number of times during the execution of a program. The computer contains address-translation hardware as well ad hardwarized interrupt and memory protect system. Reallocation of the main memory at very high rates is then the main concept of this computer. In England the technical fall-out from project ATLAS was exploited in designing the ICT 1900 series, delivered since 1964. These computers are a fully modular, compatible family, less advanced than the IBM 360 series as it still uses conventional transistor techniques, but very reliable and relying upon advanced, well developed software derived from the Atlas experience. The 1900 series has met considerable success.

An adaptation and extension of the Atlas paging scheme has been adopted by GE 645, another supercomputer making extensive use of the memory-time-sharing concept. IBM also adopted the time-sharing concept in its model 360/67 (1965), but the results were disappointing.

However, the time-sharing concept has since gained much ground even in the field of medium/small systems, mainly implemented by software systems on conventional computers. But there have been a number of special models and special hardware features designed especially for time-sharing applications. One of these is the Spectra 70 model 46 containing address-translation hardware. Others are the Control Data 3300, the Digital Equipment Corporation model PDP 10 and the Scientific Data System (now Xerox) Sigma 7. The use of such computers by way of on-line remote consoles is becoming popular and will be a major consideration in future developments.

The general structure of the advanced computer of the third generation (late 60's) resulting from the developments described in the foregoing paragraphs is shown in the following scheme (fig. II.1).

FIG. 11.1 TYPICAL ORGANIZATION OF 3rd GENERATION COMPUTER



## 3. The computer's components - Development and state of art

The outline given in the following paragraphis 3.1, 3.2, is aimed at individuating, among the main lines of the technical evolution of computer hardware, those whose projection is more certain in the immediate future (1975), near (1980) and far-off (1985).

Of elementary evidence is the fact that system developments are strictly conditioned by those of basic devices. Therefore, tracing back and ahead the roads of evolution of the latter has been chosen here as an approach to the question of what future computers will be like. Some conclusions drawn from this angle are given in the final part of this section. However, the entangled interconnections between system organisation and subsystem embodiments may be understood only by merging results from concurrent approaches such as requirements by users, software problems and system compatibility which are covered by other sections of this report, the final part of wich (p. 68 ) may help to find a synthesis.

### 3.1. Amplifiers

The high-speed "drift" transistor was first disclosed in Europe (1). Experimental types were soon available in the

---

(1) H. Krömer, "Der Driftttransistor", Naturwissenschaften 40, 578, December 1953.

U.S.A., where the technique of solid-phase impurity diffusion in semiconductors was well known. After the first mass deliveries by PHILCO in 1954, alloy-diffusion transistors gained immediate popularity. Typical characteristics of low power switching transistors of this class (RCA 2N247, 1956) are: grounded-base cut-off frequency 136 MHz, grounded-emitter power gain 22 dB at 11 MHz/1mA, 40 Ω base resistance. Central processing units could thus be run at 10 MHz clock rate. In Europe, Philips/Mullard followed shortly (Germanium alloy-diffused type ASZ21, $f_T$ 300MHz, 1956). The accurate control of impurity distribution obtained by the diffusion technique and the use of alloy-junction techniques made it possible to mass-produce without too large rejects these components, soon to be incorporated into standard, machine-assembled modular boards.

Attempts in other directions had been previously made without reaching the production-line stage. Among these are all-magnetic active devices such as "Parametron" (1) which in 1958 allowed the Japanese to build a comparatively slow all-magnetic computer working at 30KHz clock rate, 2,2 MHz excitation frequency, "Transfluxor" (2), a device combining in a sense the functions of magnetic amplifier and magnetic memory,"Twistor", etc. All these descend in a sense from such devices as e.g. the magnetic shift register disclosed early in 1950 (3), nowadays referred to as "multiaperture core

---

(1) Denki, October 1955, quoted by RCA Review in 1959.

(2) Phil. Rev. 106, p. 384, April 1957.

(3) Jour. Appl. Phys. 21, p. 49, Jan. 1950.

devices". Higher speed and the ease of the fabrication of transistors by batch processes are the probable causes for obsolescence of these magnetic devices, the only one to survive for a few years being the magnetic load-sharing switch (1). This actually is a "passive" device, used for steering large magnetic core memory arrays requiring very large (several hundred mA) write-in current.

Another class of power amplifying devices, the "photo-optronic" bistable (2), has never grown out of the character of a curiosity, except for some application in visual display panels. Photoresistance would, however, become the basis for Vidicon TV pick-up tubes and, later, of "light pen" active terminal visual display devices. The "tunnel" (Esaki) diode-effect was well known in the early 60's,but its possibilities as a logic switching device for very fast operation were not fully realized (3).

Since the first large-scale deliveries in 1954 fast switching semiconductor devices developed very quickly both in the U.S.A. and in Europe with a steep increase in operating speed, current carrying capability, allowable inverse voltages. Characteristics of some representative types are listed in Table II.1(p. 43).

Important steps in this evolution are the development of

---

(1) RCA Review <u>13</u>, page 183, June 1952.

(2) Proceed. IRE <u>43</u>, p. 1911, 1955.

(3) Association of tunnel diodes to transistor logic for obtaining high-speed carry/borrow networks in parallel arithmetic units was announced at a meeting of the Inst. of Electr. Engineers of Japan in 1961.

the Silicon planar-epitaxial structure and that of the Metal-oxyde-Silicon insulated gate transistor, (MOS) both announced in the early 60's. On this basis could develop the "integrated gate" technique. Integrated solid state devices were first made available in the USA by Texas, Motorola, Fairchild, using Metal-Alloy diffusion (MAD) and thin-film electroplating techniques. The MOS principle allowed further orders of magnitude of reduction in dimensions to be concretely aimed at since 1963 (1), with the objective of producing large-scale integration (LSI) devices having packing densities of 2000 per square inch of planar surface.

After a short time the main characteristics of one active pair connected as NOR/NAND logic element in such integrated blocks were, Fan-in = 5, Fan-out = 5, pair delay time (overall, loaded) 765 ns, maximum repetition rate 500 KHz, power dissipation 5 mW, transition voltage 7.5 V at 15 V power-supply voltage.

At this time (1967) fabrication techniques were being improved to the point where reasonable yields of 100 percent perfect arrays of 100 or more gates appeared feasible. On this basis RCA could announce the project of a general purpose computer, built for the U.S. Air Force and meant as a prototype to illustrate the effective use of LSI arrays (2). This technology, however, as attracting as it can be from the points of view of "regular structures", "modularisation",

---

(1) Proceed. IEEE 51, p. 1190, Sept. 1963.

(2) S.Y. Levy et al., "System Utilisation of LSI, IEE Transact. EC - 16, p. 562 (1967).

Table II.1 - Sample of high-speed transistors

| TYPE | U - USA<br>E - EUROPE | SPEED MHz | PEAK CURRENT mA | MAX. SUPPLY VOLTAGE | NOTES |
|---|---|---|---|---|---|
| BSY 39 | E | 350 | 200 | 15 | Planar-Epitaxial, Silicon . Switching |
| 2N2410 | U/E | 200 | 800 | 30 | Planar-Epitaxial, Silicon . Memory . Driver |
| BSX 60 | E | 475 | 1000 | 30 | Planar-Epitaxial, Silioon . Memory . Driver |
| 2N 797 | U | 600 | 150 | 7 | Diffused-base . Mesa, Germanium Switching |
| 2N 2219 | U | 250 | 800 | 30 | Planar, Epitaxial, Silicon . Switching |
| 2N 3725 | U | 100 | 500 | 50 | Planar, Epitaxial, Silicon . Memory . Driver |
| TIS 73 | U | $t_d + t_r + t_F$<br>34 ns | 50 | 30 | FET, planar, Epitaxial, Switching |

"micro-miniaturisation", must still face very difficult problems both of inter-stage wiring and input-output connection. In fact, the problems of optimal circuit configuration under costraint of avoiding dangerous mutual influence between nearby connecting paths are far from being solved, even from a mere theoretical standpoint. Still more difficult is the problem of minimizing interconnection pins. These problems (and those, similar, related to large memory arrays) are strictly tied to that of the search for an optimum overall system organisation. Once more we face a complicated pattern of interaction between physical principles of devices, production techniques and system logic.

Quoting e.g. RCA's S.Y. Levy , design should aim at partitioning a computer into arrays which contain a maximum number of gates with a limitation of number of pins, the gate-to-pin ratio, irrespective of the limits of the absolute number of gates per function,constituting a measure of effective utilisation of LSI technology. One of the new characteristic structures arising from this position could be e.g. that of the Variale Instruction Computer (1). Other solutions are envisaged from a variety of theoretical approaches all aimed at an inherently efficient utilization of "Regular Arrays" (2).

This "logic to process" coherent LSI organization may give birth to a 5th generation by the middle 70's, backed by the impulsive and controversial design developments which may be expected to follow the incorpation of LSI into traditional structures in the now coming fourth generation.

Industry's present aim,whose achievement will probably take all its major effort up to the middle 70's at least, is to automatize the entire design procedure of active-device modules. In fact, once the logic circuit diagram is drawn, about 30 man-days are presently required to draft its photographic master mask at an expense that has been judged too high to be compatible with very large mass production processes, swallowing up the benefits deriving from the presently accomplished automation of batch-production procedure. Complete design/production automation should reach its goal once two major problems arising from the increasing number of components per module are solved to a much better degree

---

(1) E.H. Muller, in IEEE-EC 16, p. 596.

(2) See e.g. I. Aleksander (Univ. London), Array Network for a Parallel Adder, IEEE Transact E-C 16, p. 227 (1967).

than they presently are, i.e.: a) close control of batch production techniques in order to obtain bearable rejects, and b) testing against faults the single component gate of the LSI module.

Obviously the two problems are mutually dependent, the second one arising from fear that the first is not reliably solved. That of testing, however, is not an easy problem.

Obviously, integration itself prevents the single components to the individually reached for testing from outside. Exhaustive sequential testing, which is recurred to in small-scale integrated circuits, is far too slow by large, even if very fast, modules. Sequential sampling random testing, on the other hand, does not guarantee the required 100 per cent reliability.

A way out of this impasse of large module technology may be that of some built-in redundancy. A stimulating approach is that of J.B. Conolly; a mere duplication redundancy technique based on ternary-output state gates, the third "indifferent" state being reached by the faulty element of the redundant pair. Inter-pair connection is so arranged as to let the indifferent state of the faulty component have no effect on the input-output characteristics of the pair (1).

As stated by the authors, this method could achieve better failure-masking capability than the Von Neumann's triplication, majority-voting technique.However, no physical device having the required three-state characteristics was indi-

---

(1) J.B. Conolly et al., Failure Erasure Circuitry. IEEE Transact. EC-16, p. 82 (1967).

cated as available for this purpose as of 1967, except perhaps E. Goto's "Parametron".

Different new approaches to solve the 100 per cent reliability problem of LSI modules are constantly being investigated, and, although as of to-day no practical solution has yet been disclosed, these efforts are likely to reach the goal within not too long a time (middle 70's).

However, it must be noted that the LSI modular technique is far from being accepted as a unique perspective, especially in the very large, extra-fast special purpose computing field. Among other advanced research projects some relevant ones are: a) return to "serial" modes by employing gigaherz tunnel diode logic or coherent light (Laser) techniques. Both these approaches must face very hard problems of interconnection links: b),optically coupled photooptronic switching circuits.

Very fast (10 ns and less) light emitting Ga-As diodes are presently available. Now designed as pick-up devices for optically-stored information retrieval, they may eventually develop into fast optronic switches, apt to be constructed by integrated fabrication techniques in conjunction with light fibers as a means for optical inter-stage, inter-block coupling : c) pratical embodiment of the principle of superconductivity in switching amplifiers, as well as in memory devices: d) self powered, radioactive decay energized transistors.. These could allow designers to use an universal standard bistable device both in logic switching and memory arrays, much in the concept of Eckert-Mouthly's primordial machine developing into a microminiature, fast, reliable, powerful logic block.

None of these advanced techniques is yet known to have reached firm ground, and as of today it is hard to foresee the date on which some of then will, if ever so, stream into production projects.

In synthesis, the present (late 60's) state of the art is that solid state integrate-curcuit devices are the building block of computer active logic. Silicon chips of 10 gates are in normal production both in the U.S.A. and in Europe. Wafers of 100 gates may be mass-produced with reasonable yields. Main characteristics of currently available devices are the following.

- Westinghouse, USA, W M 500 series, 48 gates each (125 announced) composed of basic cells measuring 37 mils square.

- Fairchild, USA (SGS, Italy), 96 gates per wafer of 147 mils sq.

These devices are now being introduced into computer construction and their use may become standard practice and remain so up to the middle 70's.

Perspectives for the time after 1975 are alternative, i.e. a) expansion of LSI techniques towards densities of $10^3$ to $10^4$ gates per block, presumably embodying redundancy techniques to ensure reasonable yields in batch production and to by-pass the testing problem, or else b), automated assembly of pre-tested chips containing comparatively few (16 to 64) gates into (cubic) arrays of $10^3$ to $10^4$ gates. Speeds of the order of 100 to 20 ns, power consumption 20 to 5 mW per gate may become standard. Higher speeds will probably be achieved in special modules meant for special purposes very fast large computers (especially for scientific-military use) by association of tunnel diode logic to transistors. Devices based on different principles are still in a stage of research: the date by which some of them may reach a stage of application is not predictable.

As far as speed is concerned, it must be noted here that none of the users interviewed by SORIS has shown any interest in getting higher rates than the present in the Central Processing Units; clock rates from 1 to 10 MHz are considered quite sufficient, providing that the mass-memory system and the peripherals are improved to a point where the full potential of central units can be efficiently exploited.

As to the future of LSI, opinions seem more definite although this is also a question where users' opinion is inclined to be a "providing it works, providing it is cheaper" one. Large scale integration is not a mere constructive approach, since it has far-reaching effects on the general system philoso - phy. To designers the concepts of regularity, modularity, expandibility are too stimulating as not to be bound to have a strong influence on future developments. The road to get at molecular-size, regular logic-gate structures may eventually develop along different paths, but as of now LSI semiconductor arrays appear the only well explored ones. Improvement and stabilisation of this technique should, therefore, be considered the take-off point of the large systems of the 80's. Going back to the past to consider the speed at which the art has evolved in some 15 years, from the low-speed, grown-up diffusion transistor to the present fast, compact thin film modules, it is an easy guess that the still open problems of interconnecting pins and 100 per cent reliability will be solved in the very near future; a too cautious attitude in this field may simply turn out to be a short-sighted one. But it must also be pointed out that compatible, simpler, less labour-taking software is the paramount requirement by the users; a fact which must be borne in mind by envisioning the new structures to which an efficient use of LSI may eventually lead.

## 3.2. Memory

Well before the era of computers the physical and technological properties of magnetic materials had undergone extensive research work aimed both at establishing their characteristics in analogcue applications (low to VH Frequency transformers,pulse transformers, sound recording) and in switching-mode operation (magnetic amplifiers). Their use to constitute memory cells by computers was suggested very early, and resorting to circulating (delay line) or electrostatic (William's tube) memories in the early large computers was probably due to the fact that associated electronic circuitry was readily on hand, while driving and reading small-size, multielement core memories was not so immediate. Ferrite-core memory coincidence matrixes were thoroughly investigated in the U.S.A. at the Electronics Labs., M.I.T., and RCA among others since at least the late 40's; in Europe similar paths were followed by many Universities and industrial Laboratories.

Core memories developed at MIT were incorporated in their

Whirlwind I computer shortly after its first operation in the early 50's. Magnetic core memory was announced in 1953 by RCA to be incorporated into the BIZMAC computer, in 1954 by IBM to substitute the electrostatic memory in the 705 (delivered 1956), in 1955 by Remington-Rand for UNIVAC II (delivered in 1956 as 1103A). In England core memories were first incorporated into computers by Elliot (mod. 401), Ferranti Pegasus, EMI 2400 . . By 1956/57 the industry had settled upon the ferrite core matrix as the dominant technology for high-speed memories (See table II.2).

Table II.2- First generation machine main memory-systems

| MACHINE NAME | CLASS* | DATE OF FIRST INSTALLATION | MEMORY IMPLEMENTATION | NUMBER OF WORDS (MIN) | BITS PER WORD | AVERAGE ACCESS TIME (μs) |
|---|---|---|---|---|---|---|
| SEAC | RS | 5/50 | Mercury Lines | 512 | 48 | 192 |
| UNIVAC I | CB | 3/51 | Mercury Tanks | 1024 | 48 | 202 |
| IAS | RS | 1/52 | CRT | 1024 | 40 | 25 |
| IBM-701 | CS | 4/53 | CRT | 2048 | 36 | 12 |
| IBM-650 | CS | 11/54 | Mag. Drum | 1000 | 40 | 2400 |
| IBM-702 | CB | /55 | CRT | 10000 | 6 | 23 |
| UNIVAC 1103A | CB/S | 3/56 | Mag. Core | 4096 | 36 | 8 |
| IBM-705 | CB | 3/56 | Mag. Core | 20000 | 6 | 17 |
| LGP-30 | CS | 9/56 | Mag. Drum | 4000 | 32 | 8500 |
| DATAMATIC D-1000 | CB | 12/57 | Mag. Core | 2000 | 52 | 12 |
| IBM-709 | CS | 8/58 | Mag. Core | 4096 | 36 | 12 |
| UNIVAC 1105 | CB/S | 9/58 | Mag. Core | 8192 | 36 | 8 |

* RS = Research System
CB = Commercial Business
CS = Commercial System

As mentioned above (see p.29 ) the association of transistors to core matrixes marked the transition from first to second generation of computers and to the rapid expansion of the industry. All the second-generation computers have core memories whose main characteristics are shown in Table II.3.

Table II.3- Second generation machine main memory systems

| MACHINE NAME | CLASS | DATE OF FIRST INSTALLATION | MEMORY IMPLEMENTATION | NUMBER OF KILOWORDS (MIN) | BITS PER WORD | MEMORY CYCLE TIME (µs) |
|---|---|---|---|---|---|---|
| PHILCO 2000/210 | S | 11/58 | Core | 8 | 48 | 10 |
| IBM 1620 | S | 10/59 | Core | 20 | 4 | 20 |
| IBM 7090 | S | 11/59 | Core | 32 | 36 | 2.2 |
| RCA 501 | B | 11/59 | Core | 4 | 24 | 15 |
| UNIVAC-LARC | S | 5/60 | Core | 10 | 48 | 4 |
| IBM 7070 | B | 6/60 | Core | 5 | 50 | 6 |
| CDC 160 | S | 7/60 | Core | 4 | 12 | 6.4 |
| HONEYWELL H-800 | S/B | 12/60 | Core | 4 | 48 | 6 |
| IBM-STRETCH | S | 5/61 | Core | 16 | 64 | 2.2 |
| CDC 924 | S | 8/61 | Core | 8 | 24 | 6.4 |
| UNIVAC 490 | S | 12/61 | Core | 16 | 30 | 4.8 |
| IBM 7074 | B | 12/61 | Core | 5 | 50 | 4 |
| BURROUGHS B260/270/280 | B | 7/62 | Core | 4.8 | 6 | 10 |
| IBM 7094-1 | S | 9/62 | Core | 32 | 36 | 2 |
| CCC DDP-24 | S | 6/63 | Core | 4 | 24 | 5 |
| CDC 3600 | S | 6/63 | Core | 32 | 48 | 1.5 |
| IBM 1440 | B | 11/63 | Core | 2 | 6 | 11.1 |
| CDC 3200 | S/B | 5/64 | Core | 8 | 24 | 1.25 |
| HONEYWELL H-200 | B | 7/64 | Core | 4 | 8 | 2 |
| CDC 6600 | S | 9/64 | Core | 32 | 60 | 1 |
| NCR 315-100 | B | 11/64 | Core | 5 | 12 | 6 |
| GE 625 | S | 4/65 | Core | 32 | 36 | 1 |
| UNIVAC 1108 | S | 8/65 | Core | 32 | 36 | 0.75 |

In Europe, transistor-driven core memories were first built into a computer in 1959.

The adoption of core matrix as a standard element brought with it important consequences on the entire logic design, allowing e.g. word-organized memories to be announced since 1957 (1) having read-out time of 100 ns in operating arrays,

---

(1) Proceed. West Joint Computer Conf. p. 73, February 1957.

60 ns in experimental types. This was also an important score in favour of the "parallel" mode in the "parallel vs. serial" controversy. To the advantages of higher speed the word organized memory joins that of producing essentially less spurious signals than those of the bit-organized arrays; a feature which increases the overall system reliability.

Notwithstanding its peculiarities of speed, ease of construction,reliability and compactness the threaded core matrix is far from being a perfect answer to the requirements of "ideal" fast-memories, nor it has been considered as such. While 100 percent of the computers of the second generation were being equipped with threaded-core memories, research continued intensely, aimed at getting;

- Lower production cost by substituting the assembling from-parts production process by batch fabrication techniques,
- Improved read-out voltage to write-in current ratio, i.e. increased signal transfer efficiency,
- Reduced dimensions to reach increase by orders of magnitude of number of memory cells per block; and, last but not least,
- Intrinsic non-destructive reading (1) with consequent faster operation, simplified ancillary circuitry.

We shall first recall some attempts that did not lead to successful practical application, such as;

---

(1) As opposite to intrinsic destructive reading of core memory. By current techniques this is overcome by restoring the previous state after destructive reading,.thus at least doubling the theoretical read-out time.

- Twistor, disclosed in 1957 (1), a multi-turn magnetical transducer having essentially higher output-voltage to input-current ratio than that of the single-turn core transducer. This, unfortunately, is obtained at the expense of an intrinsically lower speed (read-out time $6x10^3$ ns per word in first experimental set-ups, compared to an average of $1x10^3$ ns of core-array competitors) and of the inherent akwardness of the fabrication process yielding the twisted, thin magnetic film plated wires of twistor cells. No practical application of twistor memory is known.
- Superconductivity, with its inherent capability of storing an electric current circulating along a closed path, was since the very begninning of the art thought of as a means to memorize information. Due to the major draw-back of requiring ever running cryogenic sources practical interest in the system has never developed, except perhaps where associative (content-addressable) memories became a fashionable topic of computer organization discussions (2). In fact, superconductive memory arrays are claimed to afford an intrinsic, hardwarized solution to a problem that is usually dealt with in terms of software, i.e. of programming, with the purpose of a), retrieving all data sharing a common "characteristic", inrespective of their actual location; b) allocating any available memory space to any incoming information, thus obtaining an highly efficient use of memory. Problems a) are typical, e.g. of associative information

---

(1) Bell System Tech. Journ. 36, p. 1319, Nov., 1957.

(2) RCA Review 1962, p. 147; 1963, p. 325.

retrieval techniques, while problems b) are typically originated in multiprocessor system design (1). A "software" solution to the problem has been first given by the ATLAS designers (2) by an indirect address technique, where "The position of an entry in the indirect address table corre - sponds to a program address, and the contents of the entry correspond to an actual block in memory". Indipendently from that, superconductive memories are often thought of as an element of an hypothetical, very large, very fast future computer on the basis of the very short normal to superconducting transition time ranging from $10^{-9}$ to $10^{-12}$ secs. (3). As far as it is known, no practical embodiment of s.c. memories into computers has been proposed as of this date.

Another advanced approach to the problem of fast memories is that of electro-optical modulation, based e.g. on the electro-optical properties of Gallium Arsenide (4). These operate from D-C to several hundreds MHz and have been used to transmit TV on infrared beams. In conjunction with laser as a source of coherent light, these modulators could give origin to a new class of extra-fast computers. It is also apparent that the electrical branch of such extra-fast transducers gives origin to difficult problems of interstage and

---

(1) M.E. Convay, in AFIPS Proceed vol. 24, p. 139 (Fall Joint Computer Conf. 1963).

(2) J. Fotheringham, "Dynamic Storage Allocation in the ATLAS Computer", Comm. ACM 4, 435, October 1961.

(3) "Superconductive Niobium Starmide - An Introduction" RCA Review 1964, p. 337.

(4) T.E. Walhs, Gallium Arsenide Electro-Optic Modulators, RCA Rev. 1966, p. 323.

interblock signal transmission, the speeds aimed at being in the microwave range: the need therefore arises of looking at the associated circuitry from the angle of the microwave transmission lines. (1)

An all-optical approach to memory design could be through Holography (2), yielding in principle extraordinary compact information storage with relatively simple read-out techniques. This approach could eventually lead to entirely new roads; what is still lacking is a high-speed light sensitive, erasable medium to record the information hologram. This is a very interesting approach also as far as extra-large, fast mass memoires are concerned.

As a curiosity in this respect a comparatively recent episody in the serial-vs.-parallel controversy might also be mentioned. This is the Sonic Film Memory, a device in which thin magnetic films and scanning strain waves are combined to realize a non-volatile, block-oriented random access memory. Strain pulses of 10 ns accomplish storing and accessing to this new type of circulating memory. In one project (3), reported main characteristics of an experimental set-up employing Fe-Ni-Co films as storage medium are: write current pulses 1A to obtain 10 ns strain pulses, sense signals 0.15

---

(1) M. Caulton et al., Measurements on the properties of Microstrip Transmission lines for Microwave Integrated Circuits, RCA Rev. 1966, p. 377.

(2) E.G. Ramberg, "The Hologram, Properties and Applications" RCA Rev. 1966, p. 467.

(3) Block-oriented Random Access Memory (BORAM), in Proceed. Nat. Symposium on the import of Batch Fabrication on Future Computers, Los Angeles, Apr. 1965.

mv; bit density of the order, of 50 per inch.

Aside from these projects which, as has been repeatedly stated, are not likely to produce such practical results as to warrant their adoption in actual construction practice - at least within a predictable future - more conservative research trends are developing along the well-explored grounds of magnetic hysteresis loop and solid-state transducer techniques. These are briefly reviewed in the following paragraphs.

- The thin magnetic film memory technology was announced in 1956 in the U.S.A. (1) having as major assets those of reduced dimensions, batch fabrication, comparatively small write-in current, and as a drawback that of a very small read-out signal. Ten years later, IBM could announce (2) a flat-film memory of 150.000 bits, driven by a high speed block compatible with integrated circuit technology. Operating models had access time of 30 ns; Read operations could be carried up to 50 MHz rate, Write up to 20 MHz. Magnetic thin film memories were incorporated into large computers of the third generation among the first by UNIVAC 9300 (1967; Plated-wire; cycle time 600 ns) and Burroughs 8500 (1967; Flat-film; m.c.t. 500 ns). In Europe, theoretical studies on thin film memories can be dated back to 1960 (3). Many small, very fast special purpose memories, used as slave memories e.g. for dynamic storage allocation, are cur

---

(1) Proceed. Eastern Join Computer Conf., p.120, Dec. 1956.

(2) D. Seitzer (IBM Lab., Zürich) "Drive System for a magnetic film memory with 20 ns Read-cycle time", Proceed. IEEE - EC 16, p. 562, 1967.

(3) Philips Res. Rep. 1960, p. 7

rently realized by thin-film techniques. To-day this type of memory is usually referred to as scratchpad; its uses vary from register storage to main-memory look-aside. A sampling of scratchpad memories is drawn in Table II.4.

Memo de [illegible]

Table II.4- Scratchpad memories as implemented in specific machines

| COMPANY | MACHINE | DATE OF FIRST INSTALLATION | NUMBER OF WORDS | IMPLEMENTATION | MEMORY CYCLE µs TIME |
|---|---|---|---|---|---|
| HONEYWELL | H-800 | 12/60 | 256 | Core | 6.0 |
| UNIVAC | 1107 | 9/62 | 128 | Flat Film | 0.125 |
| BURROUGHS | D-825 | ? | 64 | Flat Film | 0.333 |
| RCA | 70/55 | 7/66 | 128 | Core | 0.300 |
| SDS | Σ 7 | 10/66 | 64 to 512 | Integrated Circuit | 0.150 |
| BURROUGHS | D-8500 | 1/67 | 64<br>44 | Flat Film<br>Flat Film | 0.100<br>0.100 |

The following table presents a summary of main memory performance as of 1967.

Table II.5- Summary of third generation system main memory

| MACHINE NAME | DATE OF FIRST INSTALLATION | MEMORY IMPLEMENTATION | NUMBER OF KILOWORDS (MIN) | WIDTH (bits per access) | MEMORY CYCLE TIME (µs) |
|---|---|---|---|---|---|
| IBM 360/20 | 1/66 | Core | 4 | 8 | 7.2 |
| IBM 360/30 | 5/65 | Core | 8 | 8 | 1.5 |
| IBM 360/50 | 9/65 | Core | 16 | 32 | 2.0 |
| IBM 360/90 | 2/67 | Core | 64 | 64 | 0.75 |
| HONEYWELL H-8200 | 12/67 | Core | 16 | 64 | 0.75 |
| UNIVAC 9200 | 6/67 | Plated Wire | 8 | 8 | 1.2 |
| UNIVAC 9300 | 9/67 | Plated Wire | 8 | 8 | 0.6 |
| BURROUGHS 2500 | 1/67 | Core | 10 | 8 | 2.0 |
| BURROUGHS 8500 | 1/67 | Flat Film | 4 | 204 | 0.5 |
| SDS Σ 7 | 10/66 | Core | 4 | 32 | 1.2 |
| RCA SPECTRA 70/25 | 12/65 | Core | 4 | 32 | 1.5 |
| RCA SPECTRA 70/55 | 7/66 | Core | 16 | 32 | 0.84 |

Thin film memories have a distinguishing technical embodiment in the highly-parallelized ILLIAC IV design (delivery 1970), including 256 processing elements each of which has 2048-64 bits-words and its own high speed adders. The results of this large-scale application of the thin film memory technique will give an answer as to the future of its application. Perspectives as of to-day are that it will maintain in the next 10-15 years period its characteristic of special purpose component, especially in the logic sections of very large, very fast computers.

Meanwhile, other realistic approaches are being tried, aimed to combine the well tested qualities of core memories with the advantages of batch fabrication techniques. One is that of RCA's "laminated core" memory, disclosed in 1962 (1). Main characteristics of an operating model (1966) are (2), square modules measuring 0.945 x 0.945 sq.in., obtained sandwiching two sheets less than 6 mils thick,containig 64x64 memory cell.Read current 400 mA, write current 60 mA; out-put 25 to 45 mv.The wafers contain diodes and bussing for word selection. A conventional stack consisting of 1024 words of 64 bits is formed from 2 planes each containing 512 words to form a 32 kilo-bits module. Overall dimensions of stack are 14x4,5 inches, with 0,5 in spacing between adjacent plates. Speed is comparable with that of conventional threaded core arrays (500 ns or less overall cycle time) with better input-output

---

(1) RCA Review 1962, p. 539; 1963 p. 705. Proceed Fall Joint Comp. Conf. p. 184, 1962.

(2) I. Abeyta et. Al., "Monolitic Ferrite Memories", RCA Rev. 1966, p. 77.

characteristics. What is essential in this device is that it is obtained by a monolitic laminated ferrite sheet, embedding conductors, by a batch process apt to large-scale yields; hence, an important step along the road of "regular" uniform structures, and coherent to that of LSI active device packing. As of 1968, RCA's design objective was to obtain a system memory module with a capacity of 65 K words each 150 to 250 bits long, i.e. a total capacity of $10^7$ bits, with an overall cycle time of 1.5 to 3.5 $\mu$ secs. These large capacity blocks, designed around low-drive ferrite, would have 100 mA writing current, sense signals ± 2 mV, back voltage 40 mV/ /bit. Ranging in the field of multiples of mega-bits, these blocks may well contitute an answer for the urgent requirement of large, fast random access moderate-speed mass memories as the improved, "static" substitute for the present days magnetic discs or drums (1). However, not all problems are solved as of now, the economic contruction of these $10^7$-bit modules requiring automated interconnection techniques to be developed. There are difficulties in packaging (requiring hermetic seal). It must be considered highly probable that these remaining difficulties will be solved on a production basis within a short period, which means that computers incorporating such huge-density memory modules (as a counterpart of LSI active device modules) may be reasonably expected to be built and marketed in the very near future; all depending on the greater or smaller success which competitor devices (i.e., all-solid-state, see later) may reach in this same period. It should be noted here that a predeces -

---

(1) R. Schahbender, "Laminated Ferrite Memories", RCA Review 1968, p. 180.

sor device to laminated-ferrite was the "Microaperture Ferrite Memory" (1) designed much on the same principle from the physical and lay-out point of view, but based on an entirely different (and inherently slower) process of fabrication.

Another present promising approach to large scale, high speed storage is the MOS memory. This responds to some of the still un-answered requirements of an "ideal" memory, namely, non-destructive reading, use of same select logic for both Read and Write operations. Experimental pads of chips incorporating 64 bits per chip were available in 1967, having the very low standby power consumption of 10 mW per bit, 50 ns cell switching time. Again, as in the whole of LSI, the most serious problem to be solved in the design of a high-speed, high-capacity MOS memory is that of chip interconnection; no doubt that a solution may be reached in the very near future, thus leaving the MOS memory as a potential competitor to laminated core technique. What is peculiar to the MOS principle, is that a very low consumption of the order of 4 W operating power to a 40 K bit memory should be attained, this is a very important characteristic which, jointly with that of non-destructive reading, could bring to a prominent position this device, providing the permanence of logic state in case of power failure can be ensured. At present, MOS memories are judged still too expensive for mass applications.

Read-only memories constitute another important part of medium-large computers of advanced design. Conceived as built-

---

(1) Proceed. IRE 45, p. 325, March 1957.

in hardware substitutes for software in controlling a computer via sequences of micro-instructions, they have been in recent years widely employed as a means for "emulating" other systems which are to be replaced. The mode of implementing read-only memories spans the art, from capacitor and resistor arrays and magnetic core ropes and snakes to selectively deposited magnetic film arrays. An original approach to R.O.M. is that of LITASTOR (1) relying on light transmission over optical fibers in order to obtain a light output at choosen crossing points. The system, however, is essentially a planar (cylindrical) one, not lending itself to compact cubic stack realisations, the main advantages being very high internal speed, low cost.

Secondary memory is still relying on mechanically driven magnetic devices, such as tape, drum and disc. Development in this field has been a slow one compared with that of the associated circuitry, as no new solution has yet become practical in the problem of storing huge quantities of data in random access systems.

Discs constitute an intermediate answer to the problem, combining large storage capability ($\simeq$ 3 M bits per disc) to comparatively short access time ($\simeq$ 2 m secs) obtained by a quasi-parallel access mode through many (256) magnetic heads and tracks. It must be noted that replacement of tape memory by discs is one of the few definite requirements by many of the users, cost being the only adverse factor referred to. It is hard to believe that an essentially clumsy device, re

---

(1) F. Filippazzi, "A New Approach to Permanent Memory", Proc. IEEE-EC 16, p. 370 (1967).

quiring over-refined mechanical fabrication techniques as the magnetic disc may ever reach the low-cost level. The foreseable future tends towards substitution of the disc by a) static magnetic memories, low-cost mass-produced by some of the new techniques referred to above (p. 58 ); b) recourse to entirely different digital recording techniques. This has, for a long time, been a field subject of intense research and some of the results already reached may lead to useful solutions in a near future. The problem overlaps that of permanent recording of mass data in a small space, that of recording TV information, etc. touching a large field of interest: a further reason entitling to expect practical solutions to be attained in the very next years (within 1975). Among the investigation paths followed are:

- Electrostatic recording on dielectric-type or electret-type media. Both writing and reading are sought for via direct electric means (high-voltage electrodes; pick-up electrodes) or indirect, photo-electric transducers. High-density packing is theoretically afforded by the support medium; nevertheless, reading of, and especially obtaining electrical output signal from, high-density electrostatic latent images is still an open problem. Scanning by cathode ray beams (which could yield non-destructive reading, e.g. of electric permanent recording) requires high-vacuum enclosures and has the inherent disadvantage of serial access. Electro-optical recording and read back transducers may reach the goal through improvement of electrophotographic (Electrofax, etc.)processes. This procedure may be the first to give useful results by realizing electro-mechanical scanning systems much like those on which magnetic recording is based, with the advantage of great (of orders of magnitude) increase of bits stored per unit surface and reduction of driven speed.

- Frozen-latent image recording by photoconductive thermoplastic recording systems (RCA, 1964). Parallel recording over the whole surface may be attained; reading is serial by schlieren optics.
- Optical recording on diazo-compounds, etc. Reversible photocromic processes are known, enabling very high information packing densities to be achieved (National Cash Register Co. 1960). Much work is done on this field also in Europe (Prof. Eggert, Zürich). All leading manufacturers of photographic material and many in the electronics and allied fields are carrying on research work in this field which is very interesting also from purely theoretical points of view.
- Optical recording by laser beams to etch thin metallic films (announced in Europe by Bosch Co.), or to trigger beyond the Curie point Mg-Bi crystals (Honeywell, USA). In experimental models developed by this Compagny occupation is 1 micron per bit, with inter-bit spacing of 5 microns. Reading is by electromechanical-optical scanning.

## 3.3. Input-output devices

By the time the first members of the second generation of EDP systems were installed most of the input-output techniques which would dominate the life of that generation had been introduced; although not so impressive as those of the central unit substantial improvements had taken place - the early electromechanical teletypers were joined by their fast counterpart, high-speed line printers having 48 to 64 printable characters, 100 to 160 print positions, speed 170 to 1100 lines per minute. A sample of these printers is shown on Table II.6

Table II.6- Sample of second generation line printers

| Company | Printer Model Number | Printable Characters | Print Positions | Speed (lines/min) |
|---|---|---|---|---|
| Anelex | 4-1000 | 48/64 | 120/160 | 667/1000 |
| Burroughs | B-231 | 64 | 120 | 170 |
| IBM | 720 | 49 | 120 | 500 |
| | 1403-3 | 48 | 132 | 1100 |
| Honeywell | 822-4 | 56 | 160 | 900 |
| UNIVAC | 7912 | 51 | 100/130 | 600 |

Among these samples, typical in terms of printable characters per second is the Honeywell Model 822-4 printer (56 printable characters); 160 characters per line at 900 lines per minute giving 2.4. kilocharacters per second, an almost unbelievable speed for an electromechanical device. The corresponding transfer rate from magnetic tape to core memory was 89 kilocharacters per second (Honeywell drive 804-2), giving a fairly well balanced ratio of 37 to 1 between the transfer rate of data to central unit to that at which output data are tabulated.

As both magnetic drive and line printer speeds had reached the top values compatible with mechanical construction techniques, two ways were open to match peripheral equipment to the ever increasing speed of central units: a) sharing a single processor among several peripherals, and b) design output devices along different principles. Although ideas were ripe,practical solutions had to wait for the "third generation" computers to be adopted. The "hardware" solution was that of introducing disc memory as an intermediate storage device, non-electromechanical (electronic) visual displays and printers as output devices; the "software" solution is that of machine organization in a time sharing mode. Ancillary hardware, partly borrowed from telecommunication techniques, was of course required for multiplexing/buffering at the interface. Once again, this reacted on the general organization of computer claiming an executive control to supervise the time - shared operations.

It may be noted here that the amount of tabulated output produced by extra-fast printing devices is impractical to handle, flooding the E.D.P. centre. The need to clear out the output channels for next work is the sole justifying such a high speed. This recently suggested to record the output on some compact medium (e.g. micro-film) for handy storage and subsequent retrieval at the users' will.

High output rates are, on the contrary, required by TV digital display devices where the question is rather that of compressing the required information per frame.

Among the improvements reached during recent years, we list:

- High-speed electrostatic recording on zinc-oxyde, resin-layers coated paper, by means of thin window recording tubes (1). TV-like recording speeds are obtained on permanent records; writing rate may reach 104 characters per second (19 inches per second) with 200 lines per inch definition.

---

(1) Jour. SMPTE Vol. 69, 1, Jan. 1960.

- Special tubes for electro-setting, allowing photographic recording of selected letters. Having serial random access, these tubes are an electrical substitute for electromechanical teleprinters, working noiselessy at very high speed. Recording is done by photographic means.
- Solid-state display devices. Bistable properties of sandwiched electroluminescent photoresistive layers (1) are used to produce rectangular flat display matrixes; input may be electrical or optical. These devices are still in the laboratory stage.
- Display storage tube, a device now become popular for storing bi-dimensional information. Writing is serial, at a speed of $300 \cdot 10^3$ inches per second. Quick erase, high signal-to-noise, high brightness ratio are characteristic of these tubes.
- Light pen display devices. These are actually means for man-to-machine intercommunication used for conversational mode operation especially in the field of automated design. These devices and its predecessor, the Rand tablet, constitute a new approach to the problem of man-to-machine interface, an art becoming of even greater importance in connection with operation of many active terminals.
- Hybrid digital-analogue visual display systems. These aim at compression of digital data required to form a fundamentally digital on line output display. One embodiment is Phaseplot (2), in which graphical data to be displayed are

---

(1) Proceed. IRE 48, p. 1380, Aug. 1960.

(2) Trans. IEEE-EC 16, 1967, p. 203.

composed from primitive segments each of which corresponds to one computer command which specifies starting and stopping point, final slope and "curvature". That of compression of information to minimize occupation of computer output channels by graphic display is one of the current topics, and developments are to be expected in connection with real-time operation.

Conversational mode, implying direct access from terminals, is another current topic. The problem is obviously connected with that of "direct" languages . From the point of view of devices, a simple primitive conversational interface consists of a teletyper in conjuction with a punched card reader to charge programs. A wide range of sophisticated interfaces have been built and tested, including the "light pen" CR tube and the Rand tablet. In the "in" direction one of the problems to be solved is that of automatic pattern recognition which includes printed and handwritten character reading. So far, only the first of these problems has been partially solwed by use of special typewriter characters.

In the "out" direction, non-mechanical recording systems - including means for obtaining "hard copies" from visual display - still require much improvement in performance as well as decrease in cost. It is a widespread opinion that the whole field of the input-output devices, including its "software" aspects, deserve much improvement. Rather than big qualitative jumps regular, general improvement of existing devices is expected within comparatively short term (within 1975?).

## 4. Expert forecasts on hardware

Experts' forecasts for hardware in 1980 are sumarized in table II.6.

It will be noted that there are some discrepancies among individual forecasts, depending presumably on the different professional background of the experts. If we extract from the table the common characteristics and leave out those who strongly disagree, the following pattern is obtained.

a. Central Unit

Main memory will be hierarchically organized including large, extra fast, control storage and very large "extended" work memory ($> 10^6$ bytes). Dimensions larger than or comparable to IBM 360/195 are generally thought not advisable. Fast non-core storage, associative storage may become built-in. Modularisation is possible.

b. Peripherals

Will include large drum/disc storage, large permanent (optical?) storage for data banks.

c. Input-output units, terminals

Greatly improved input / output devices will be in use, including single-or multi-font character readers, direct data input, visual display hard copy, local disc storage.

d. System organization operating mode

The "solar" concept may prevail. Conversational mode, time-sharing, multi-processing will be common. Teleprocessing will expand depending on cost of data transmission. Real time operation will be used in process control and other applications functionally requiring it.

These conclusions fairly agree with those drawn from the users' interviews. No fundamental contradiction was found with technical development trends except that here extra-large, extra-fast computers appear not much favoured, at least as for as near future is concerned.

TABLE 11.6

## EXPERTS' FORECASTS FOR 1980

| INTERVIEW N° | N° OF SYSTEMS OF HIGHER PERFORMANCE THAN 360/75 LIKELY TO BE INSTALLED | HARDWARE | | | SYSTEM ORGANISATION, OPERATING MODE |
|---|---|---|---|---|---|
| | | CENTRAL UNIT | PERIPHERALS | INPUT/OUTPUT, TERMINALS | |
| $E_1$ | 60 IN GERMANY | Large control storage, LSI, ≃ 30 ns cycle.<br>Main storage, > $10^6$ bytes.<br>Buffer to disc/drum within unit. | Large drum/disc storage, parallel scanning of tracks 1÷2 ms access. | <u>Input</u>, optical reading single fond.<br><u>Output</u>, non- mechanical printers Graphic | <u>Solar</u><br>Free conversational mode. |
| $E_2$ | | Size 360/195, CDC 7600 advisable | Large permanent storage for data banks. | <u>Input</u>, direct from digital data; direct from graphic data.<br><u>Output</u>, visual, voice | <u>Atomic</u><br>Time-sharing Real-Time when functionally required. Conversational mode |
| $E_3$ | 30÷40<br>IN GERMANY | Hierarchic memory organisation.<br>"Extended" work memory.<br>Size 360/195 not advisable | Low-cost intermediate memory, large $(10^9-10^2)$ characters | <u>Input</u>, direct from data.<br>No from voice.<br><u>Output</u>, mechanical, visual, hard copy. | Flexible systems, general purpose.<br>Time sharing.<br>Little teleprocessing.<br>Free conversational mode |
| $E_4$ | 600÷1.000<br>IN WESTERN EUROPE | Hierarchy of memories | Large disc memory.<br>Card file obsolete. | <u>Input</u>, Card, optical reading special (OCRA) fond graphic.<br>Telereading (subject to lower priced devices).<br><u>Output</u>, Electromechanical visual/hard copy | Complex logic, multiprocessing.<br>Teleprocessing (depending on cost). No free conversational mode |

Follow: TABLE II.6 - **EXPERTS' FORECASTS FOR 1980**

| INTERVIEW N° | N° OF SYSTEMS OF HIGHER PERFORMANCE THAN 360/75 LIKELY TO BE INSTALLED | HARDWARE | | | SYSTEM ORGANISATION, OPERATING MODE |
|---|---|---|---|---|---|
| | | CENTRAL UNIT | PERIPHERALS | INPUT/OUTPUT, TERMINALS | |
| $E_5$ | 600-1.000<br>IN WESTERN EUROPE | Solid-state memory arrays. Magnetic wire : 10 to 100-fold increase of present capacity | Large mass memory. Large-scale use of data banks | Input: multifond optical readers. Microfilm readers. Direct from graphic data. No voice input. Card/band obsolescing. Output: visual display/hard copy. Disc storage, local, of info from central optical storage. Fast non electro-mechanical printers | |
| $E_6$ | 70-100<br>IN GERMANY | Hierarchy of memories Large solid state, main memory | Large mass memory, optical. | Input: optical standard fond reading. Graphic from photographs and test points. Output: multiple, visual/hard copy. Electrostatic printers (reduced cost) | Decreasing need of multiprocessing and multiple access. Conversational mode. Time sharing (scientific applications). Little teleprocessing. Solar/atomic |

Follow: TABLE II.6

## EXPERTS' FORECASTS FOR 1980

| INTERVIEW N° | N° OF SYSTEMS OF HIGHER PERFORMANCE THAN 360/75 LIKELY TO BE INSTALLED | HARDWARE | | | SYSTEM ORGANISATION, OPERATING MODE |
|---|---|---|---|---|---|
| | | CENTRAL UNIT | PERIPHERALS | INPUT/OUTPUT, TERMINALS | |
| $E_7$ | 100 to 1000<br>IN WESTERN EUROPE | | Optical mass memory | Input: multiple, direct to CPU.' Optical print reading.' Standard fond for documents, technical papers.'<br>Output: visual display, electromechanical printers.' | Mode: Conversational, Real time in process control |
| $E_8$ | 500÷750<br>IN WESTERN EUROPE | Huge internal memory | Optical storage | Input: optical readers, cards, graphic.' No acoustic.'<br>Outout: electrochemical visual display/hard copy.' Some output direct via video cable.Graphic. Some acoustic.' | Large-scale systems<br>solar |
| $E_9$ | 300<br>IN WESTERN EUROPE | Bigger computers.'<br>Modularized.' | | Input: Direct from test points.' Less punched cards.' Multifond optical claracter reading.' No acoustic<br>Output: Mechanical printers, visual display, acoustic.' | Time-sharing<br>Solar .' Software difficulties for computer systems.' |

Follow: TABLE 11.6 --- <u>EXPERTS' FORECASTS FOR 1980</u>

| INTERVIEW N° | N° OF SYSTEMS OF HIGHER PERFORMANCE THAN 360/75 LIKELY TO BE INSTALLED | HARDWARE | | | SYSTEM ORGANISATION, OPERATING MODE |
|---|---|---|---|---|---|
| | | CENTRAL UNIT | PERIPHERALS | INPUT/OUTPUT, TERMINALS | |
| $E_{10}$ | 500<br>IN WESTERN EUROPE | Fast non-core storage.'<br>Associative storage.' | Magnetic mass storage.'<br>No optical storage | <u>Input</u>: Direct from test points.' Optical character readers.' Punched cards.' Multiple, direct.'<br><u>Output</u>: mechanical printers, acoustic, graphic.' | <u>Solar</u> |
| $E_{11}$ | | | | | <u>Atomic.'</u> Integrated systems.' |
| $E_{12}$ | | | | | |
| $E_{13}$ | | | | | |
| $E_{14}$ | 70 in U.K.<br>OR MORE | | | Remote terminals, optical character readers | <u>Solar</u> |

## 5. Conclusions [illegible]

As has been stated in the introduction the aim of the foregoing outline is to single out some criteria which might be helpful in determining the paths of the probable evolution of the very intricate pattern of hardware developments we can expect in the future. This is not simple because it is peculiar to the computer field that the evolution of devices is not direct. In fact, new hardware solutions cause new software problems, and these in turn require new hardware. Both hardware and software problems have many possible solutions at every stage creating thus an almost innumerable complex of trend choices. To illustrate this point we refer, as an example, to one of the many currently available "trend forecasts" (1). Over a hundred factors - technical possibilities, trends and constraints - are listed in E. C. Joseph's article. Once these factors have been properly defined and quantified branching them into some rational decision tree would mean an arduous computing task.

A way out of this impasse is to chose at random a few out of the innumerable facets of the universe we are considering. Among these are the general factors affecting the industry's development as a whole, as we have mentioned on page 3 . To these a fifth factor can be added: the beneficial effect of very advanced research projects. The cases of Atlas, Whirlwind I, Project Sage, ILLIAC are demonstrative enough in this last respect. As to the other four factors, we think

---

(1) E. C. Joseph, quot.

that their significance has been sufficiently proven by the history of success and failure in the US computer industry. Not to speak, of course, of the European experience of the years before and after 1958.

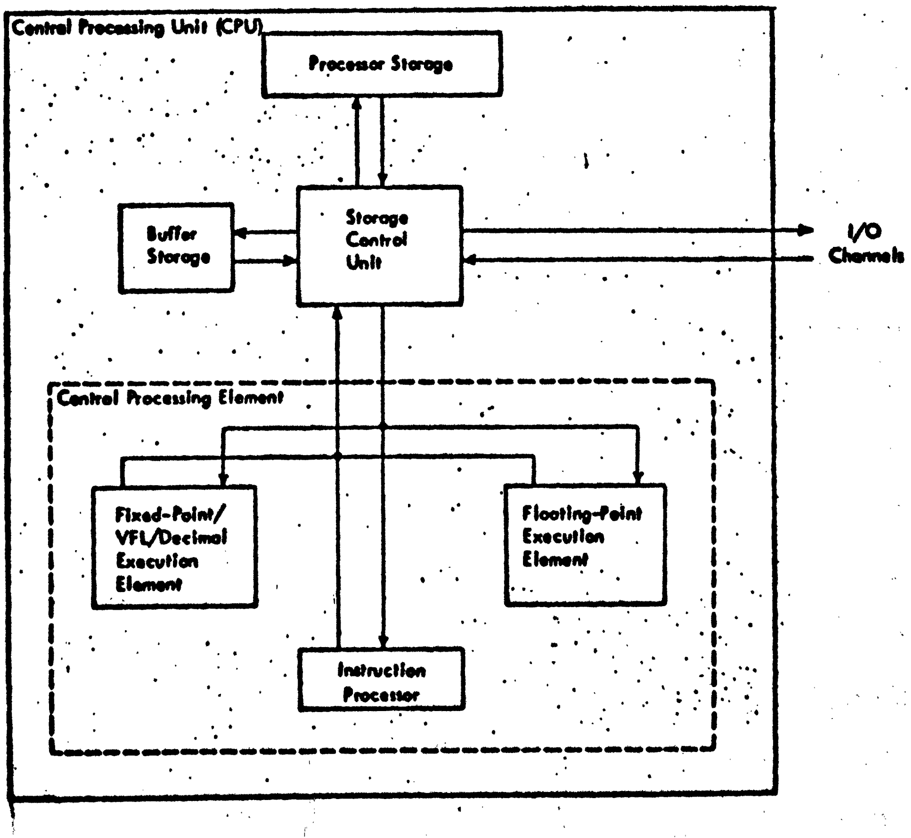
Going back to consider the computer hardware evolution, we see that a major trend is emerging. Among the variety of electronic devices which have been invented in some 20 years survive mainly those leading to: a) parallelization of logic, b) "modularity, regularity", c) comparative ease of fabrication by large-yield, "parallel" (batch) production techniques. Thus, for instance, the early serial adders were soon superseded by parallel-mode half adder/shift registers, dooming to quick obsolescence the serially working magnetic drum computers. Serial-access memories such as electrostatic and delay-line devices were soon replaced by core arrays, the oriented, serial interrogation mode of the latter being quickly succeeded by word interrogation (and word-oriented logic). In the field of intermediate memories, magnetic drums gave way to discs, again on the same ground of increased modularity, higher, "regularity".
Teletypes were quickly substituted by line printers , the latter being in turn replaced by some sort of cross-coincidence, parallel or quasi-parallel mode operating output device, such as solid-state display matrixes may eventually become. The electronic computer development state diagram passes through alternate transitions, from present state A to next state B through a transition marked by increased speed of the previous mode, immediately followed - as a tendency at least - by transition from state B to state C where the previous overall speed is obtained by increased parallelisa

tion; then the cycle repeats itself. This tendency is shared by general machine philosophy as well, as has been meaningfully shown e.g. by the emphasis presently given to multi - processor, memory overlapping, true realtime, procedures.

A similar state diagram rules production techniques. Transistor circuits were first assembled by soldering to anchorage posts on insulating boards, with wired connections; a typically "serial" procedure. Then holes arranged in regular patterns were made on boards by a single punch operation. Further, printed circuits were realized, on which the wiring is produced by a parallel, batch process, and components are soldered by "serial" operations. High-speed, automatized "serial" positioning of components followed shortly combined with "parallel" one-stroke dip soldering. Transition to integrated circuit techniques was a further step in parallelization of production procedure.

As to fast memory arrays, hand-threading evolved quickly into automation; the word-oriented mode of operation was the backlash over logic organization, marking again a step towards parallelization. Present trends of production by batch techniques are only a step in this well established-linear evolutive path.

It seems therefore that little crystalgazing is required to forecast that LSI (and related memory techniques) will gain ground in the near future. The fact that some of the microminiaturized blocks of the computers of the 3rd generation have not yet reached the stage of small scale integration constitute only a minor deviation from the general trend: as has been noted earlier, automated custom built design of inter-stage connections may very soon reverse the trend even in this respect. In fact, LSI fast memory modules are presently built-in into special-purpose computer units, as those for space vehicles, and as an added facility in some of most advanced commercial system. An outstanding example is that of the very large, extra fast IBM 360/195, whose huge core memory (maximum of 4,194,304 bytes, interleave factor 16), is backed by an extra-fast "cache" LSI monolithic memory of 32,768 bytes. Swapping from main to cache memory allows the effective overall memory cycle time to be 54 nsecs, while that of main memory alone is 756 nsecs. (see scketch below).



What remains to judge is whether the result of this tendency will be that of the huge extra-fast computer, centralizing the work coming from many peripherals or satellites or that of interconnecting into an integrated system many independent, small to medium size very compact computers. The choice depends on a number constraints with little or no bearing on the technical trend we have outlined.

Some general conclusions can now be drawn about the main directions along which substantially improved devices can be expected and predicted to become incorporated into the computer of the 80's.

a. The whole hierarchy of high speed memories (main, auxiliary, bulk) will increase in capacity by orders of magnitude; implementation will be by batch production techniques, yielding multi-bit modular "regular" arrays; magnetic (film, laminated core, wire) or solid-state.

b. External, medium speed intermediate memories may also improve by substitution of multi-track magnetic drum/disc by static devices.

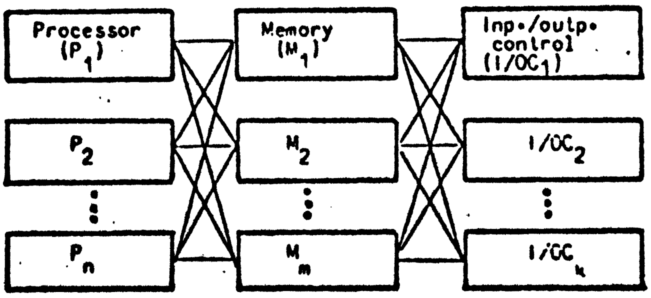
c. Very fast, comparatively small read-only memories will be increasingly used, especially for microprogramming and in connection with transfer of software functions to hardware. Probable implementation is by MOS techniques.

d. Fast access mass memories will acquire huge dimensions ( $>10^{10}$ characters). Probable implementation is by electro-optical (Hologram?) techniques. It must be noted that increased capacity, reduced access time in the whole range of mass memories is one agreed upon requirement by most users in the whole EDP field.

e. The second commonly shared requirement is increased performance, reduced cost, increased reliability of input-output devices. All advanced projects that can reach this goal will meet very favourable acceptance by users who usually complain the still existing disproportion between the very high performance of central units and that ot the terminals. This also appears, then, to be a sector where quickly developing technical projects and market demand share a common, steep upward trend.

f. Widespread use of LSI techniques will depend, of course, on production facilities. Providing the 100 percent reliability requirement is met (by strictly controlled production techniques, by redundancy techniques, etc.) modular, regular, expandable units based on LSI blocks may become a standard feature of the computer of the 80's. A very general - and very undefined - scheme of this modular structure could be the following (after E.C. Joseph, quoted)

MULTIPROCESSOR: MODULAR STRUCTURE



g. Much research and development work is presently going on over devices related to man-to-machine communication problems. This is the hardware counterpart of the software problem of conversational languages and modes. Both these problems are in the background of expansion of EDP in some important, recently opened fields of utilisation such as automated design, information retrieval, computer-assisted design, management information systems. In this sector no substantial novelties are anticipated, but there exists a wide field for gradual improvement and a wide diffusion of existing devices. Among these are,
- Cathode-ray tubes as display means, both of graphic and alpha-numeric information.
- Hard-copy printers (possibly all-electric) to obtain permanent records of information displayed by CRT's.
- Graphic input devices able to read standard typewritten pages,symbolic codified sketches, symbolic diagrams, etc.

h. On-line operation is required by the utilisations listed under item g. above and by many others, among which the outstanding are,
- Meteorology, seismology and other disciplines practicing real-time monitoring of natural processes.
- Control and optimisation of multiple-input, multiple-output processes such as passenger booking, dispatching of goods.
- Automatic vehicle guidance in military and commercial uses.

Hardwarized implementation of these tasks requires built-in input-output multiplexing, buffering, supervisory controls. These are common features to advanced third gener ation computers, but satisfying these requirements is made easier if optimal utilisation of central units ceases to be a major worry to designers. Modular construction of high-speed sub-units, and hence practical realisation of almost indefinitely expandable processors, may contribute to that goal.

As to the period of time by wich these developments will take place, two interrelated criteria can be pointed out. The first is time delay between the successful testing of new devices and their introduction into current production practice. In some exemplary cases this delay is 5 years at least. As to permanence, there are examples showing that it may be very long. This is the case of the magnetic drum which still survives after 20 years since its invention by A.D. Booth of the Univ. of London.

Taking into account these criteria it seems reasonable to expect that large-scale-integration modules will not reach the marketing stage before 1975. By this time, and as a consequence of technical fall-out from the advanced projects now being executed, some commercial computer composed of these modules may reach the marketing stage.

- Independently from the success of LSI-Metal-Oxide-Semiconductor modules, microminiaturisation will lead to increasing compactness of logic blocks. To match them LSI modular memory arrays will be increasingly used. High-density memory modules, produced by batch techniques, may find their way to production line within 1975. Other types of fast memory arrays, based e.g. on bipolar diodes or on MOS triodes, may become fully developed and tested during this period, to find their utilisation in the subsequent years.

- Disc-drum devices may keep their role as intermediate speed memory in the immediate future. A substantial increase in demand, partly independent from cost, may be expected for these devices. Meanwhile, an increase by a factor of 10 in packing density may take place. The period after 1975 will probably see some kind of static device growing out of the laboratory stage to become competitor to discs in the subsequent years.

- Peripherals incorporating improved input-output devices will be made available to users in the next years.

  Requirements are for higher performance, noiseless operation, higher reliability, reduction of overhaul expenses and of dead time for repairs. This requirement can be met by extended utilisation of existing devices such as CRT display, all-electric printers, data loggers and active terminal devices such as "light-pen", digitizer tablets/CRT and similar. Integration of a number of these devices in special purpose consoles will also expand to meet particular requirements, such as automated electronic circuit design, programmed instruction. A number of consoles may in turn be connected to small local computers acting as slave computer to a large central processor, a practice which will probably expand over the next 5 year period.

  More elaborate terminal devices, including "intelligent" ones such as pattern-recognizing may further develop at a laboratory stage in this period, to reach the production stage in the subsequent one (1980?).

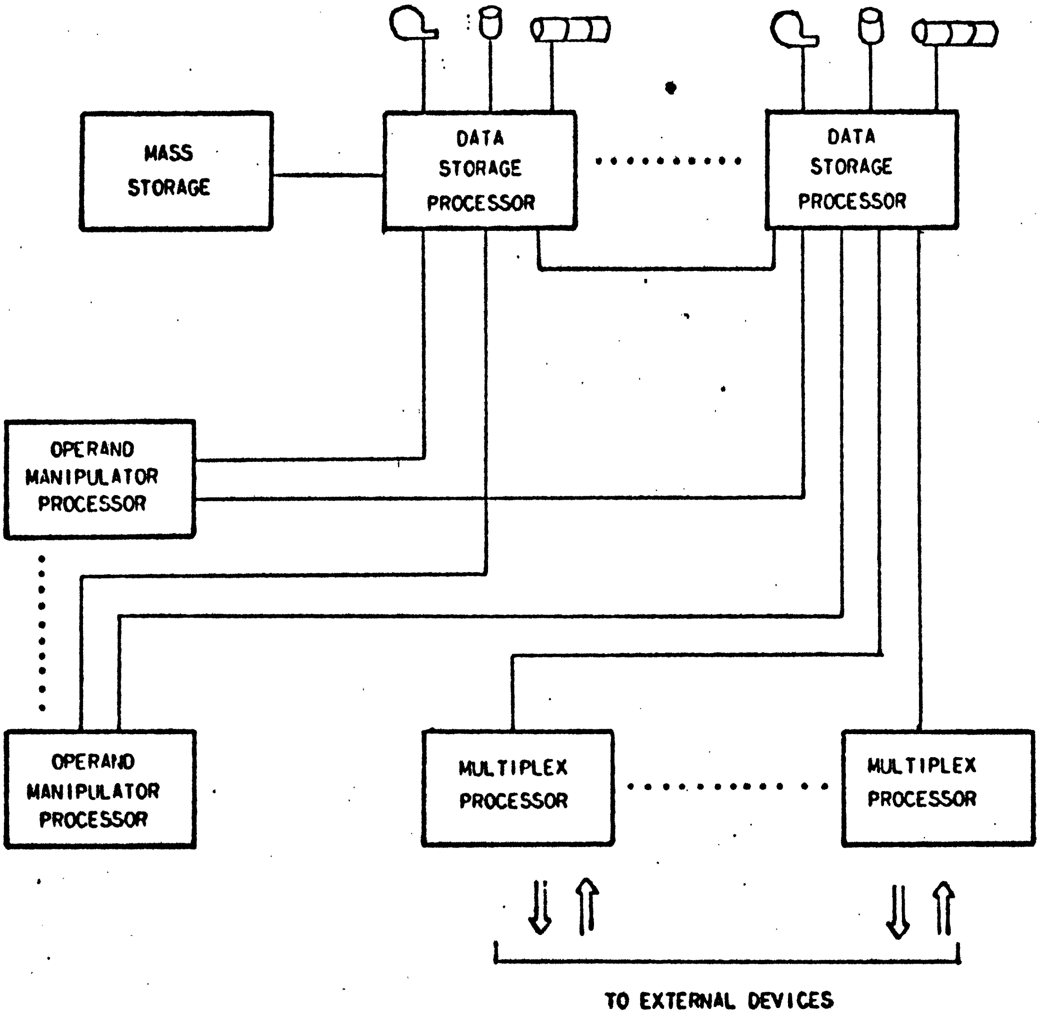
- Teleprocessing may become a common practice very soon.

  No different problems than those of time-sharing by different users arise for computer hardware. Problems lie mainly in the policy followed for data transmission.

- Utilisation of read-only memories will quickly expand in connection with the needs, a) of simplifying programming, b) lessening the charge on computers by elaborate display devices. It may be expected that microprogramming of frequently recurrent programme sequences by read-only memories will be an added optional facility in the next five years. The years after the use of read-only memories may see a further expansion in connection with the expected evolution and standardisation of languages.

- The general features of system organisation for the immediate future correspond largely to the following sketch, except perhaps for gradual substitution of rotating memories by static devices (see Fig. II.2). The near (1980) and far-off (1985) future are subject to too many variables to allow realistic forecasts of system organization.

FIG. 11.2 FOURTH GENERATION COMPUTER SYSTEMS



# CHAPTER III

# Software

## 1. Introduction

Facing the problems of software the first thing which has to be pointed out is that they cover an extremely vast field, e.g. the manipulation of symbols, invention of new languages, new techniques and algorithms. Moreover, programming a digital computer is an individual job, while hardware design is a typical teamwork.

This feature of individuality has caused the today available software to be so varied and heterogeneous. It is a commonly shared opinion that this is the principal reason for the current inadequacy of the man-machine interaction.

In the following analysis we have tried to keep the survey along traditional lines, and especially:

1) Operational systems
2) Programming languages
3) Compilers and assemblers

In the assessment of software's future development account has been taken also of the interviews with the experts which do, however, not furnish elements of particular interest.

## 2. Operational systems

Under operational or monitor systems we understand those programmes which permit the automatic handling of jobs, eliminating as far as possible manual operations and reducing the idle times of the central units.
In the case of small size,slow speed,primitive computers the normal programmation did not affect very much the total working time.
The increase in operational speed caused the gradual development of operational systems, starting with first monitors and growing up to time sharing and multiprogramming systems.
At the very beginning the "monitor" was just used to control input/output and interruption operations in order to make overlap, at least partially, the data's input an output times.
Thus a fortran monitor system (FMS)is introduced in an operational way in 1958 on an IBM 709 which can be considered the forerunner of all future operational systems.
With IBSYS on 709 - 7090,which is a thesaurus of older systems and may be referred to as the monitor of monitors, substantial improvements are attained.

IBJOB in one of its sub-sets covering compiler and assembler, whereas,strictly speaking,the proper functions of monitoring are fulfilled by OIOCS and IOEX routines.
IOEX is the program which controls all input-output and interruption functions. IOCS controls the systemation and use of the buffers. Any reading of recording request makes

these programmes intervene.
Their structure is complicated and not easy to use,when applications, other than the very few ones for which the systems have been conceived, are ever tried.
In fact, the 709 is a computer with and input-output buffer placed inside the machine.

## 2.1. Operating from the Console

A first generation computer is basically a system involving the programmer as part of the operational system. He is not only responsible for running the programme, but also when debugging he runs the computer freely by pushing the control key and looking at the contents of the various registers.
Obviously this systems is insufficient, not only because expensive and not very useful means are employed, but even more,because of the machine is standing still while the operator is scheming the next step.
It becomes therefore necessary to develop new efficient operational procedures in order to save idle times between one job and the next, as well as to develop at the same time systems for the diagnostics of errors,in order to enable the programmer to find out quickly the origins of the malfunctioning of a programme without checking it step by step from the console.

For small and medium computers procedures have therefore been developed aiming at increasing efficiency in the field of processing and programmes capable to detect the cause of errors.

## 2.2. Batch processing

In bigger systems the previous technique is completely abandoned.
In this case it is possible to integrate the basic monitor with automatic programme loading routines in such a way that the computer remains always operating. The operator has to make only occasional interventions, such as loading the magnetic reels or answering simple requests from the console programme. These systems are called "batch systems". Jobs are pooled in batches and their control is handled by the system.
The overall performance increases thus at the expense of increased turnaround time. Debugging, which will be only possible off line, becomes a grave problem.

A more favourable situation, with shorter answering times, can be created by adding to the system auxiliary casual access memories. Jobs are loaded in batch discs, whose casual access can permit choosing in the batch the priority jobs (being determined, for example, on the basis of CPU time required).

Thus efficiency of the machine is increased, but the programmer has no possibility of checking or modifying his programme during processing.

## 2.3. Remote job entry

During the last few years small computers at acceptable costs have appeared on the market.

These computers, however, cannot cope with all the computing requests,nor give the help which is obtainable with monitors of bigger systems.

On the others hand, the possibility of integrating these small computers through telephone lines into bigger systems makes it necessary to increase their access speed and performance.

Above all the turnaround, already a hampering factor in local batch processing, becomes unacceptable to a long distance user. It is therefore imperative to reduce the access time in order to secure a quick service and increase the capabilities of remote computing. Some systems are simply modified in such a way that remote jobs which enter into a normal batch get a certain priority over local computing jobs.

Depending on the kind of system, priority selection and control may be performed automatically or through the operator. In more advanced systems special control routines analyze the remote job batches, so that through this first error scanning the turnaround time will be further decreased.

## 2.4. Multiprogramming

Development of batch systems used on large computers led to multiprogramming systems.
We are referring to multiprogramming, when two or more programmes, quite independent one from another, are stored in the central memory at the same time, operating alternatively. Programmes are sharing the CPU time sequentially and based on their priority level.
Local batch systems such as DOS, OS, MVT, OS-MFT, EXEC8, etc. are multiprogramming systems. They are supplying a remote processing service, still showing, however, the traditional batch systems' shortcomings, such as very scanty integration operator/machine and long waiting time for answers.

The ever more pressing requirements of long-distance users have caused research to move into a different direction, and the problems were faced from a completely new point of view, but still based on the same requirement to keep CPU employed at a maximum.
Systems of this type (interactive) emphasize to the highest degree the interaction operator/machine and fast answering times.

## 2.5. Interactive systems

In the past few years interactive systems have become synonymous with time sharing.
The basic difference between these systems and multiprogramming systems consists in the scheduling method.
In M.P. systems CPU time is distribued according to the priorities of the jobs, in T.S. systems jobs are accepted by CPU on a time-slice basis which, disregarding the jobs, determines the CPU time available and when the next control transfer can take place.

Setting-up of third generation computers equipped for T.S. starts in 1964. Dennis (1) suggests a new technique of dynamic control of the memory, where programmes are developed in sets of constant length called "pages", and only the pages necessary for the processing of the programme are stored in the memory for the assigned time-slice. But the basic concept is the recognition that a storage address means two things: a single name attributed to a certain information and further its storage location.
Before it was always thought that an address meant only the information name, and that by mentioning this name it would be possible to find the address.
Thus, there arises the idea of a series of names whose variability does not depend from the memory's actual size, but from addressing facilities of the hardware.
This is the concept of virtual memory being entirely at the disposal of each user, which is the basis of the operational T.S. system.
A dynamic control of the memory, such as we have described, required changes and improvements in hardware, both of which were made in the GE 645 and the IBM 360/67.

---

(1) J.B. Dennis - Segmentation and the design of multiprogrammed computers - I.E.E.E. 1965.

In addition to the new way of addressing the memory, these two models are the first ones in which the T.S. system consists of a double processor. This feature produces a more balanced system, where more than one control and calculating element have equal access to the machine. This dual capacity has been created in order to increase the T.S. potential.

While these T.S. systems are being developed, also systems geared toward special applications are created in the same years, e.g. JOSS by the Rand Corporation in 1964 and the very efficient, even if very simple, BASIC.

The number of systems labelled T.S. increases very rapidly in the period 1965-68, losing any coherence among themselves, so that a homogeneous picture can hardly be given.

The leading commercial T.S. are based on a particular language, such as GE BASIC - BBN'S, TELECOMP(JOSS), IBM QUIKTRAN (FORTRAN). Other systems are based on a certain class of applications, such as KEY data for commercial applications and the IBM TEXT Data Service.

One of the best systems having the clear features of a general purpose T.S. is SDS 940. Having been started on an experimental basis by the University of California at Berkeley, it is now extensively used in many installations. Despite some limitations due to the type of computer, this system gives way to reasonable hopes about the success of T.S. on a commercial basis. Also GE BASIC, conceived as a T.S. with a single language, can now be used by many languages.

The MULTICS and T.S.S. systems, linked to GE'S 645 and IBM's 360/67 respectively,have progressed very slowly.There are many reasons for this. Big programming efforts have to be made on rather new hardware (associative registers, for instance). They are aiming at an improvement in the quantity and quality of services of a certain size.
Future will tell, whether or not these goals can be reached.

In the majority of general purpose T.S. today on the market, the number of contamporaneous users is 30 or more in machines with only one CPU unit. It is obvious that services offered by the recent systems are greatly improved when compared with the first T.S. systems.

Special purpose T.S. can serve up to 400 simultaneous users which confirms McCarthy's forecasts of 1962, at least in the case of computers with limited applications.

One of the most interesting aspects of the interactive systems is that of the kind of service provided. With T.S. one can obtain, under economically advantageous conditions as compared with traditional systems, on-line simulations, programmed instruction, search for information by data banks, and all those applications obtainable by keyboard-terminals. With more sophisticated devices, of course, more sophisticated applications are possible, such as, for instance, examination of three-dimensional shapes by rotation on displays, recognition in real time of handwritten characters and of mathematical formulas or hypotheses-tests which use a display for the presentation of alternatives and the relative weights of the different factors (e.g. medical applications at the University of California).

In some special systems, such as APEX, structure, language, form of data have been created for the solution of graphic problems. In 1964 Jacks presented a system for contemporaneous use of display in the designing and studying of the various components of a car.
Until now computer networks in T.S. are limited to a few installations, such as the Q 32 of the Lincoln Laboratory and T-X2, SDC Q-32 TSS and SRI CDC 160A. In this application programmes running in one computer are able to request and make run programmes in a different computer, in such a way that incompatible computers pool their performances (a Lisp is running on Q 32, used by a programme on the T-32).

The slow progress in this field is due to the scanty experience in computer networks. Experiments of this type are by force very expensive, involving many computers and still high transmission's costs.
Only few companies are willing to study an expensive project, before its economic value has been demonstrated, even in the US, where the taxation system is quite generous with firms in this respect.

As is always the case with new ideas, also about T.S. systems there have been many disagreements. One point of discussion is the cost of using such a system.
The programmes' control cost is certainly higher when operating in T.S. and longer is also the time of using a computer for performing the job.

Some studies have been made in order to compare the efficiency of a particular interactive system with traditional batch systems. For quite a while it was only an idea that gains in performance were making losses in "efficiency" tolerable. Today, the idea is largely accepted following some proofs during the last few years which consisted of a comparison between the production of programmes and their solution with interactive systems.

The efficiency can be evaluated on the basis of measuring a certain number of more significant parameters (elapsed time, computer time, programmer's time).

For example, Sackman (1) concluded in his research that interactive systems the programming time (manpower time) decreases, whereas machine time for the solution of problems increases.

Critical comments came from several sources, who judged these studies as too hasty.

The present offers a large quantity of interesting applications and many experts are working in the hope of being able to furnish undisputable proof of the economical advantages or disadvantages of these systems.

---

(1) Sackman - Time Sharing versus batch processing in the experimental evidence. Processings of the spring joint Computer Conference, 1968.

## 3. Programming languages

The programming languages are a set of rules established for constructing through characters expressions with well defined operational significance which can be processed by the computer.
In fact, each computer understands directly during the processing of a programme only a limited number of operational instructions which have been coded in machine language, generally in the binary system.
Programming with such languages, used in early computers, was very complicated and tiresome, because of the need of formulating programmes in every detail.
On the other hand it became clear that passage from a defined problem, e.g. using a mathematical formula, to a programme in machine language ready to be processed by the computer, was nothing else than a process of manipulating symbols which could be processed by the computer itself.

Thus the first translating programmes (compilers and assemblers) were created as well as the first programming languages which differentiated gradually into high level or "problem oriented languages" and assembling or "machine oriented" languages.

The "problem oriented" languages offer the following advantages:

a) easy to learn: the notation used is nearer to the problem to be processed than to the computer's hardware;

b) coding and debugging: in addition to being problem oriented, a high level language is generally much more synthetic than an assembling language, which makes coding, debugging and the possibly necessary modifications easier;

c) compatibility: since high level languages are problem oriented, they show a high degree of compatibility among the different computers in which they have been implemented, so that only a little effort is necessary to adapt a programme to a different computer;

d) speed: the easy coding and correction of such languages reduces considerably the total time necessary for the solution of a given problem.

Programming languages can be divided into:

- languages for scientific problems,
- languages for commercial problems,
- languages for processing symbols
- general purpose languages,
- special purpose languages.

## 3.1. Languages for scientific problems

These languages are used for the solution of scientific problems through numerical techniques.
The first languages compiled for this purpose were the SHORT CODE (UNIVAC) and the Speedcoding (IBM-701), both a far cry from permitting coding of problems similar to their mathematical formulation.
The subsequent languages, such as A-2 and A-3 (Remington Rand), BACAIC (IBM-701), PRINT (IBM-705), MATHMATIC (UNIVAC 1), UNICODE (UNIVAC 1103/1105), IT (IBM-650), now only of historical interest, contributed to the development of compilation techniques and of logical and Input-Output procedures.
Among the languages developed later and still in use today, we will examine first those based on compiling systems, in which the programme is translated into machine language in a single phase and then those based on the more recent conversational systems in which translation is made statement by statement.

* Compiling languages are;

FORTRAN: the FORTRAN language (Formula Translation) has been developed since 1959 by a team of IBM researchers headed by J. Backus (1). The first version was presented in 1957 (IBM/704), and was improved later in FORTRAN II and FORTRAN IV (IBM 7090/94).

---

(1) J. W. Backus et al. - Internal translator IT, A compiler for the 650, Ann Arbor, Mich. 1957.

FORTRAN became soon a remarkable success, due also to the success of the IBM 704 for which it had been prepared. Shortly thereafter, all computer manufacturers constructed compilers able to accept more or less extensive versions of this language.

The FORTRAN language has, on the other hand, restrictions due to the fact that it has been produced for a specific computer (IBM 704) and is therefore influenced by its hardware. In subsequent versions these limitations have been partially eliminated (for instance, instructions linked to IBM 704 hardware). Many restrictions have remained, such as the incapability to process lists of symbols and the lack of recurrent procedures.

ALGOL: the language (Algorithmic Language) has been developed from 1955 on as an international strictly defined language for the description of computing procedures (algorithms) by a European team (GAMM) with the collaboration of an American committee from ACM.

After a preliminary version (ALGOL 58) a first version called ALGOL 60 followed, which took account of the basic research by Backus (1) about formal definition methods of a language's syntax.

---

(1) J.W. Backus - The Syntax and semantics of the proposed international Algebraic language of the Zurich ACM-GAMM Conference, Proceedings of international Conference on Information Processing, Unesco, Paris, 1959.

This research opened a series of studies about principles, syntax and semantics of languages. After some adjustments the final version, ALGOL 60, was published.
ALGOL-60 in the standard version shows many features which are similar to Fortran, but is more fleible and general.
The major drawbacks of ALGOL are, above all, the input-output statements which are not defined in the language (and therefore left to the compiler constructors), and the difficult implementation of some of the language's most important features such as the recurrent recalling of procedures, dynamic allocation of the memory and own variables.

Conversational systems have been developed in order to create the means for a continuous dialogue man/machine avoiding the long delays of traditional batch systems. Manufacturers tried to use these new operational systems (time sharing) by either transferring there the languages already in use in traditional systems or creating new languages especially studied for them. Given the increasing importance of conversational languages, due also to the efficiency of debugging and the flexibility in the use by practical and economical telepointer-type terminals, we will examine briefly the most important languages.

JOSS has been the first significant example of a conversational language. It was created for the on-line processing through a computer of small scientific problems.
In order to minimize the learning time, the number of instructions has been reduced which do, however, permit complete and synthetic processing of the small problems for which this language is destined.

QUIKTRAN is a transfer of FORTRAN which was created as a compiling language on conversational systems. It can thus accept programmes written in FORTRAN up to the FORTRAN II level, with some limitations. A characteristic application of this language is using it during the study of coding and correction of programmes, which are then processed using a normal compiler (which generally furnishes more efficient object codes).

BASIC : this language has been constructed with the purpose of supplying a simple instrument, useful introduction to the study of more sophisticated languages and which can process small scientific problems. This language has been quite a success and is one of the most widely used ones in this field.

CPS: is a rather limited subroutine of PL/1 (IB67) and can be employed in small size scientific and commercial jobs. In the first version it was working in a micro-programmed IBM 360/50; but there exists also a version with subroutines replacing the micro-programmes and avoiding thus the need of special hardware.

APL/360: is a subroutine of a more general language: APL was developed by Iverson (1) for scientific applications and programming of operational systems as well as for the description of hardware. APL/360 is oriented toward scientific problems and exploits a concise and synthetic notation in mathematical operations, introducing a series of factors for the algebraic and logic processing of vectors and matrices.

---

(1) K.E. Iverson - A programming language, J. Wiley & Sons, 1962.

Other languages have been created especially for the solution of mathematical problems and are therefore equipped with a series of high level standard operations (integration, operations on matrices, etc.). We want to mention among the most popular MAP:developed by M.I.T.,DIALOG used in connection with a video terminal and AMTRAN.

Other languages in the same category, such as COLASL, MADCAP and MIRFAC, the latter developed in Great Britain, make use of special equipment at the hardware and terminal levels.

## 3.2. Languages for commercial problems

Commercial problems are characterized by the need of processing a very large quantity of data at different levels through standard procedures. Historically, the first language which was created for this purpose was the FLOW-MATIC.
This and other languages, today obselete influenced the features of a new language, COBOL, created especially for commercial and administrative applications. This new language was created by a joint committee of manufacturers and US government officials. COBOL is today the only one used and is a big success also in Europe.
A first preliminary version was followed by other more sophisticated ones.
As is known, COBOL works with sets of signs which may indicate names (of data or procedures) constants, verbs (defining functions) and operators (arithmetic, logic and relational).
The programme coding consists of four different parts (or divisions):

- identification division;
- environment division;
- data division;
- procedure division.

The main contributions of COBOL in the field of programming languages are:

a) the separation between description of data and procedures on one side and the physical elaboration units on the other, so that the same programme requires only some modifications in the environment division when changing from one computer to another;

b) development of a form for describing data and procedures, which is independent from the computer;

c) efficiency in processing the record files;

d) the effort made in making the programme assume the aspect of an easyly understandable English text.

Since data processing is a crucial point in commercial processes, many COBOL extensions have been suggested: it is a question not only of programming languages, but of special systems for processing data stored in large mass memories. One of these systems is IDS (Integrated Data Store), which predicts e.g. the possibility of interconnecting records in lists by supplying new procedures for their processing.

## 3.3. Languages for processing symbols

The first computer applications concerned exclusively processing of numerical data.
The necessity to process not numbers, but general symbols attributed fundamental importance to the concepts of lists and strings.

The fundamental operations on lists are the creation and combination of the lists as well as the extraction of the first and last elements. In the processing of strings it is essential to create the string and to seek in it characteristic sequences which can be substituted by other assigned ones. Other symbol groups are formal algebraic expressions with the operations which can be made on them.
Symbol processing languages can be divided into three groups: languages for the processing of lists, languages for the processing of strings, languages for the processing of algebraic formulas.

* Languages for the processing of lists are:

IPL-V- : IPL (Information Processing Language) was the first language for the processing of lists; it has been improved and revised up to today's IPV-L.
This language looks formally like a sequence of operational instructions (like an assembling language) which process binary lists or list groups, control the memory (e.g. "garbage collection", i.e. cleaning of the memory is performed by the programme) and allow construction of subroutines and

recurrent processes.

$L^6$: also this is a assembler-type language. Available instructions allow direct assignment of the memory which means very efficient object programmes for the processing of lists (at not so high a level).

LISP 1.5 (49-50) is the most powerful means available for recurrent processing of lists. This programme is a sequence of functions, elementary or deriving from them, which are evaluated during processing and which work on the lists or complex structures. Control of the memory and the recurrent system is automatic. The drawbacks of this language are mainly the heavy notation and the inefficient handling of numerical data.

* Languages for the processing of strings are:

COMIT: the oldest language of this type. It was used originally for the processing of linguistic texts (e.g. automatic translations). The basic operation of the language consists of extracting a sub-string from the original strings and the transformations of them following assigned sequences.

SNOBOL: was created by COMIT and is based on the same functions and characteristics, but on different structures.

PANON: was created by the Italian University of Pisa and offers wide possibilities of defining and recurrent processing of the string structures.

TRAC - GPM: the former was created in the US, the latter in Europe, for the processing of strings using macroinstruction techniques.

* Languages for the processing of formal algebraic expressions are

FORMAC: is an extension of FORTRAN whose principal features it preserves. It accepts, in addition to the usual numerical variables of FORTRAN, a new kind of variable, whose value is not a number, but an algebraic expression. Besides this type of variable FORMAC introduces a series of functions and subroutines for their processing. This language was a great success within the field of its specific applications, because it is practical, easy to learn and powerful.

MATHLAB: was the first conversational system in the field of algebraic processings. It is coded in LISP and uses LISP for the execution of its functions, even if they are very complex, such as differentials, integrals, Laplace transformations, systems solutions, etc.

ALTRAN is also an extension of FORTRAN, but more limited in applications (rational functions) than Formac. It is therefore not much used.

## 3.4. General languages

The general languages are these capable of processing in a satisfactory way two of main field in wich programming languages are used: risolution of scientific problems; processing of commercial and administrative problems; processing of lists and sequences; processing of algebraic expressions.

Great diffusion and importance enjoys PL/1, developed by IBM at first as an extension of Fortran IV, later as an independent language, combining the most important features of FORTRAN, ALGOL, COBOL and of the languages for the processing of lists.

PL/1 can process a large variety of data which are organized in matrices and structures, even if they are very complex and on which instructions can operate treating them formally as sequentials.

Among the general features of PL/1, which do not exist in the component-languages, can be mentioned:

a) modularity: PL/1 has been structured in such a way that it can be divided into simpler subroutines covering special uses;

b) multiprocessing: PL/1 is capable to process parallel to the instructions for a major procedure those of a subprocedure;

c) capability of macrodefinitions: PL/1 is capable of defining macro-instructions for the purpose of optimizing the object programme. Before compilation a macro-processing reads the programme and modifies it according to the macroinstructions which it meets. The modified programme is then compiled.

The most advanced features, such as memory control, multiprocessing, macro-possibilities, classify PL/1 as the first language able to interact efficiently with the operational system of the computer, despite the fact that the interpretations of these features and their dependence on the specific features of the computer, are not yet clear.

<u>C.P.L.</u> (Combined Programming Language) developed in Great Britain by Cambridge and London Universities is language similar to PL/1;

<u>Formula ALGOL</u>: this is language derived from Algol which is able to create and process lists and sequences of characters and to define formal variables on which algebraic processing is possible;

<u>LISP 2</u>: this language is partly an extension of Algol, whose formal aspects it has preserved,towards LISP 1.5 from which it takes some functions and internal structures. Partly it is an attempt to overcome the major drawbacks of LISP,such as the difficult coding of programmes, the slow speed of interpretation, the poor efficiency in numerical calculations;

<u>ALGOL 68:</u> essentially a generalization of ALGOL 60 at PL/1 level. It is still in the stage of elaboration and final completion.

## 3.5. Special languages

There exists a very wide range of languages created for very special uses which require generally specific knowledge of the fields to which they refer.

Many of these languages are structured like subroutines (and can have been derived from more general languages) or are based on macroinstructions,in which case they have a very rigid structure, which is, however, not a disadvantage, due to the very limited field of application. Because of the great number and the quick developments taking place in this field, we will just mention the more important applications starting with languages oriented towards specific classes of problems.

- Tool machine control: the languages are suited for programming all the operations to be carried ont during the working (with cutting tools) of a piece, in order to get a finished product; we have thus languages which permit, on the one hand, bi-and threedimensional descriptions (final surface of the piece to be worked) and supply,on the other hand, a programme written in a code which can be directly interpreted by the tool machine and carried out in the form of elementary operations (moving the tool, moving the piece, determining the revolving speed; etc.).

  The most prominent of these languages is without doubt APT (Automatically Programmed Tools), developed by M.I.T. from 1955 on.

- Civil Engineering: the languages are destined for particular applications in this field and are prepared in such a way as to require only a minimum of information on the computers for their use. Among languages of this type let us remember COGO which M.I.T. developed and which is suitable for the solution of geometrical problems presenting themselves in civil engineering (e.g. topographic surveys), STRESS for calculating articulate structures under various loading conditions, and ICES which is an integrated system for the processing of various problems, such as structural analysis, bridge calculations, road engineering, etc.

- Computer simulation: languages belonging to this class have the purpose of allowing, as far as possible, studies about the computer's hardware before its actual production, simulating on another computer its logical design and behaviour. Languages of this type are APL and Simulating Digital System and various others.

- Simulation of continuous systems: the languages have the purpose of defining the dynamic state of a system which varies continually with time; this can happen when considering either particular systems (mechanical, electrical, chemical, etc.),and in this case we have a specialized terminology or general systems, i.e. integral and differential equations determining the evolution of the system over time.
  In this way it is possible to simulate analogical and hybrid computers. The most popular languages used are CSMP developed by IBM and MIDAS.

- Discrete simulation: in discrete simulation the system's evolution is thought of as a whole of many parallel processes, each of which is related to one element of the system, whose state is upgraded at the end of a suitable waiting interval: during these intervals a queue of events forms (due to the instantaneous verification of some processes and the termination of others) which have to be taken into account during the next upgrading.
  These systems can therefore become extremely complex: and then considerable theoretical as well as practical problems present themselves, concerning the parallelism of these processes (e.g. the problems deriving from simultaneous upgrading of processes which interfere with each other).
  Among the most popular languages can be mentioned GPSS, SIMSCRIPT (derived from FORTRAN), SOL, MILITRAN and SIMULA.
  Also languages for interactive systems were developed, such as OPS by M.I.T.

- Information processing: this is a very wide field which comprises the creation and maintenance of archives, the recording and information retrieval, the question-answer system: generally speaking, these are more complex systems than real languages.

- Management of graphic terminals: the languages are used for the description and synthesis of drawings, and in general of patterns, allowing e.g. composition, rotation and translation; generally these languages use i/o optical devices.

## 4. Translating programmes: assemblers and compilers

Translating programmes (or better, "metaprogrammes") have been created for the purpose of eliminating the difficult task of codifying programmes into basic languages.
They can be divided into " machine-oriented "languages and " problem-oriented ".

### 4.1. Assemblers

Assemblers are metaprogrammes which translate programmes written in "machine-oriented" language into a sequence of instructions (objective codes) written in machine language and ready for processing.

The first assemblers were produced by computer users as routines for facilitating the writing of their programmes.

One of the first organic standardization attempts led to the SAP assembler for IBM's 704. This and other assemblers were not much more than one to one translators.

With the development of operational systems and in connection with batch processing, assembling programmes became more and more important. Since this technique foresaw the processing without any interruption of a large number of jobs, the assembler was incorporated into the operational system and supplied by the computer manufacturers.
Power and flexibility of assemblers and the connected symbolic "machine-oriented" languages have been substantially increased by adoption of macro-operational techniques.

Depending on these techniques a set of instructions can be thought of as a single instruction to a pseudo-computer which is far more sophisticated than the one at disposal.

## 4.2. Compilers

Generally, compiler is a programme which accepts as input a "problem-oriented" programme like Fortran, Algol, etc. and produces a sequence of "machine-oriented" language instructions. This whole of instructions undergoes later an assembly process during which the final translation into machine language, the allocation of the memory and the storage in it take place.

The first compilers for "problem-oriented" symbolic languages were developed in the years 1956-1958. The most important of these compilers, which is a milestone in the history of automatic programmation, was the Fortran compiler developed for the IBM 709 by the Backus team.
The efficiency of the Fortran compilers produced by IBM contributed to a larger diffusion not only of this language, but also of IBM computers.
This language is still today the most widely used algebraic language for scientific applications.
The success of Fortran forced other manufacturers to produce compilers for this language, and thus the compiler ALTAC for the computer Philco 2000, the Honeywell Algebraic Compiler and the Control Data 1604 Fortran were produced.

At the same time (1958), because of the work of the European GAMM team, a new algebraic language called Algol 58 which was completely independent of the computer, was developed. This , was the first universally accepted language, with a strictly defined syntax. This fact led to many other studies in the field of languages based on the work of Chomsky (1). Current development of compilers is related to these studies which were directed towards:

a) lexical analysis: stage of the compiling process which transforms the entry programme into a list of characters, each of which has a well defined meaning (e.g. at this stage a character used with different meaning will be translated with a different character on the list).

b) syntax analysis: the list of characters is examined for the purpose of verifying the compatibility of entry instructions and generating a structure reproducing the instruction to which it refers in such a way as to be coded directly.

---

(1) A.N. Chomsky - On certain formal properties of grammars - Information and Control, vol. 2, 1959.

The syntax analysis is followed generally by a transforming process of the resulting structure into a sequence of macro-instructions. Algorithms optimize later this sequence (e.g. eliminating from the cycles invariant calculations, distributing in a most rational way index registers, etc.) and develop the macro-instructions.

An assembly phase concludes the compling process.

The complex techniques used in compilers make their construction long and difficult. In order to speed up their construction, special languages have been created in order to produce at least partially the compilers.

These languages are called CC (Compiler Compiler) and include the development of a whole of standard techniques in compiler construction.

Research in this field was conducted by American teams which have realized FSL, VITAL, CABAL and by a European team from the Manchester University which created SPG. Another rather interesting field, from a theoretical as well as from a practical viewpoint, is the possibility of supplying high level symbolic languages and the relative compilers able to expand their capacity by the macroinstruction technique.

With this technique it is possible to extend without restriction language and compilers creating pseudo-instructions capable to introduce new semantic and syntactical forms.

## 5. Experts' forecasts for software

Experts' forecasts for software in 1980 are summarized in table III.1.

Extracting from the table the common features and leaving out those who strongly disagree, the following pattern is obtained:

"Simplified ("natural") programming languages will be used. More specialized languages are required, among them enquiry languages. Distinction between basic and application software will become sharper.

Hardwarisation of some software functions (e.g. supervisory control programs) will be achieved. Increasing sophistication and importance of applications will bring with them an increase in the role of the system analyst. "

TABLE III. 1 EXPERTS' FORECASTS FOR 1980

| INTERVIEW N. | SOFTWARE |
|---|---|
| $E_1$ | Simplified languages, increased role of system analyst |
| $E_2$ | Special-purpose Languages |
| $E_3$ | Simplified languages<br>Hardwarized software functions |
| $E_4$ | |
| $E_5$ | |
| $E_6$ | |
| $E_7$ | Simplified programming languages |
| $E_8$ | Simplified programming languages<br>Enquiry languages |
| $E_9$ | Enquiry languages, increased role of system analyst<br>Highy sophisticated software |
| $E_{10}$ | Enquiry system<br>Natural languages<br>Increased role of system analysts<br>More separation of basic from application software |
| $E_{11}$ | Hardwarized software functions<br>Simplified languages<br>Associative files |

Follows: TABLE III.1 EXPERTS' FORECASTS FOR 1980

| INTERVIEW N. | SOFTWARE |
|---|---|
| $E_{12}$ | Hardwarized software functions (indexing relative adressing, reallocations, etc.) |
| $E_{13}$ | Inter-file communication language to enable establishing data banks |
| $E_{14}$ | |

## 6. Conclusion

In the past great attention was given to techniques concerning the applicational programmes, while concerning the programmes of system control only unorganized and improvised efforts were made. On the other hand, to-day and in the foreseeable future the percentage of programmes for the system control (operational systems, utility programmes, subroutine archives) is considerably greater than the applicational programmes, although the latter are the actual purpose of employing a computer. For this reason, the major innovations which can be expected in the field of software are bound to be in the sector of supervisory and operational systems. This can take place partly also with their transfer to hardware through implementation of read-only memories, along the lines of technical evolution already mentioned in the chapter on hardware.

The predictable development of languages is related to this process.

Concerning programming languages, it must be remembered that the great proliferation which happened in this field has resulted in a very confused situation.

In order to put some order in this field studies are constantly made about the theoretical foundations of the languages in general and about the formulation of the programming languages in particular. These studies are essential also for the construction of the translating programmes of these languages.

The narrow definition of programming languages is however still an open problem, and the development of a uniform theory and a universal language is still a long term objective.

Together with the basic development of the definition of the languages, there exist two individual lines of development: in the first, the computer is seen from the users's viewpoint and research is, therefore, oriented towards production of easier and more practical systems, in the second line research is oriented towards the development of theoretical means for the study of these languages.
In the first line are included the research concerning the construction of basic languages equipped with techniques which permit the user to develop his own language and the attempts to adjust the computer for the acceptance of languages which are more and more similar to the natural ones. Moreover , more powerful techniques are being studied for the dialogue with the operational systems and for the use of new means of communications with the computer (e.g. video terminals).
From a theoretical viewpoint, in addition to the already mentioned basic research, the second line of development concerns the following fields: verification of the compiler's efficiency from the point of view of translation and debugging, the methods for the construction of languages capable to generate other languages, the construction of languages for the efficient handling of big masses of information. Great attention is also given to problems related to the new techniques of management of computers (parallelism, multiprogramming, time-sharing), transfering to hardware some of the functions still performed by software (mathematical subroutines).

The developments which will take place in the translating programmes are related to these developments in programming languages.
First of all, the skill of the translator has to be verified, i. e. the degree of equivalence which they manage to establish between the programme to be translated and the object code produced. These problems are solved for the time being in an empiric way.

One of the crucial points emerging is the necessity for a strict definition of the semantics and syntax of programming languages, so that more clearness in this field (still far away) can produce positive reactions in the field of translators, reducing at the same time the high cost of producing these programmes.
Related to the problem of cost and formalization of the languages, the writing systems is the most interesting branch of development and research in the field of translators.
The most interesting research concerns meta-assemblers (used for the construction of assemblers and compilers using macro-instruction techniques) Compiler-Compilers and extendable compilers which allow the introduction of techniques into the compilers and languages which permit their development.
In the field of interactions between translating languages and hardware, we want to mention the possible and predictable applications of microprogramming techniques in the translation process. It is not unreasonable to think that such techniques can be used successfully in the process of lexical analysis, in the construction of charts which guide the syntactical analysis, and in debugging.

In the US, the cost of programming as compared to the total data processing cost (cost of hardware, programmes, programme maintenance and upgrading of data) has gone from 5% in 1950 to 50% in 1965 and will reach 80% in 1970. The problem will become even more critical in the future.

This is the central problem which has to be coped with during the seventies. Its solution requires an increasing compatibility among the various systems, not only vertically among successive generations and within computer families of the same generation (e.g. IBM's 360 series), but also horizontally among the families produced by different manufacturers, in order to avoid further efforts of reprogramming.

Such a compatibility implies also a common man-machine interface, i.e. the possibility for the users to be able to change from one processing system to another without further training and the standardization not only of the programming languages, but also of the principal features of the operational systems of computer management (utility programmes, diagnostics, control files, input-output management).

Standardization of the features of the operational systems will permit the exchange of data among independently produced programmes and give the possibility to combine data from various origins into an integrated whole (data bank).

From this point of view, the road to follow would be that of the creation of a common machine - machine interface, i.e. creation of one single language of description of the machine's internal formats for the writing of data: such a language would have to be managed directly by the operational systems and supply the basis for an effective horizontal compatibility (computer networks).

An ideal long term solution could be the realization of a standard interface on the "extended machine" level (hardware plus operational system), obtained, for example, through the microprogramming technique.