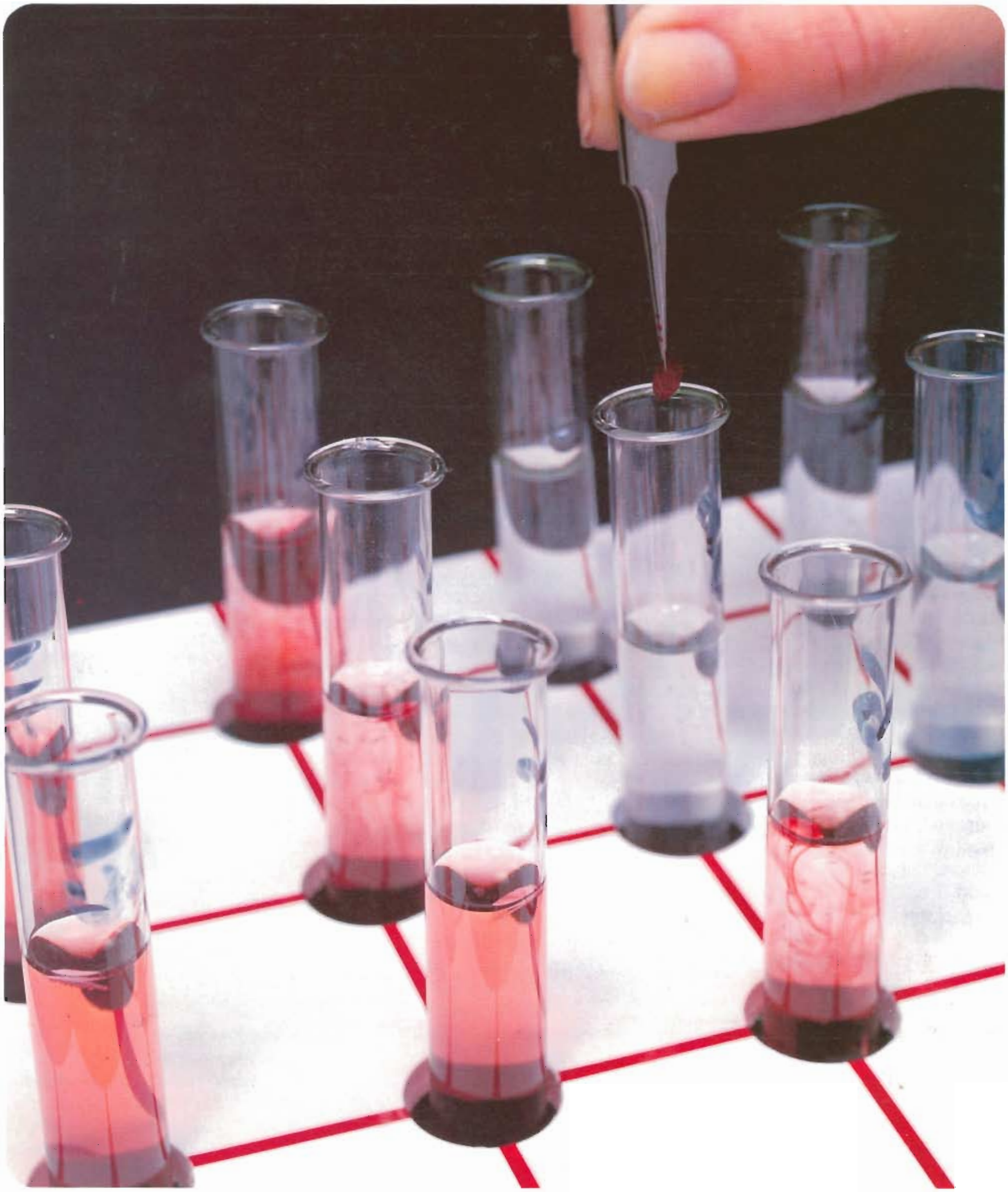
Keyboard

1978/2 A Publication of Hewlett-Packard Calculator Products Division



Keyboard

Overview

John Nairn's column, "Crossroads," will not appear in this issue, but he will be back with a continuation of the discussion on random number generation next issue.

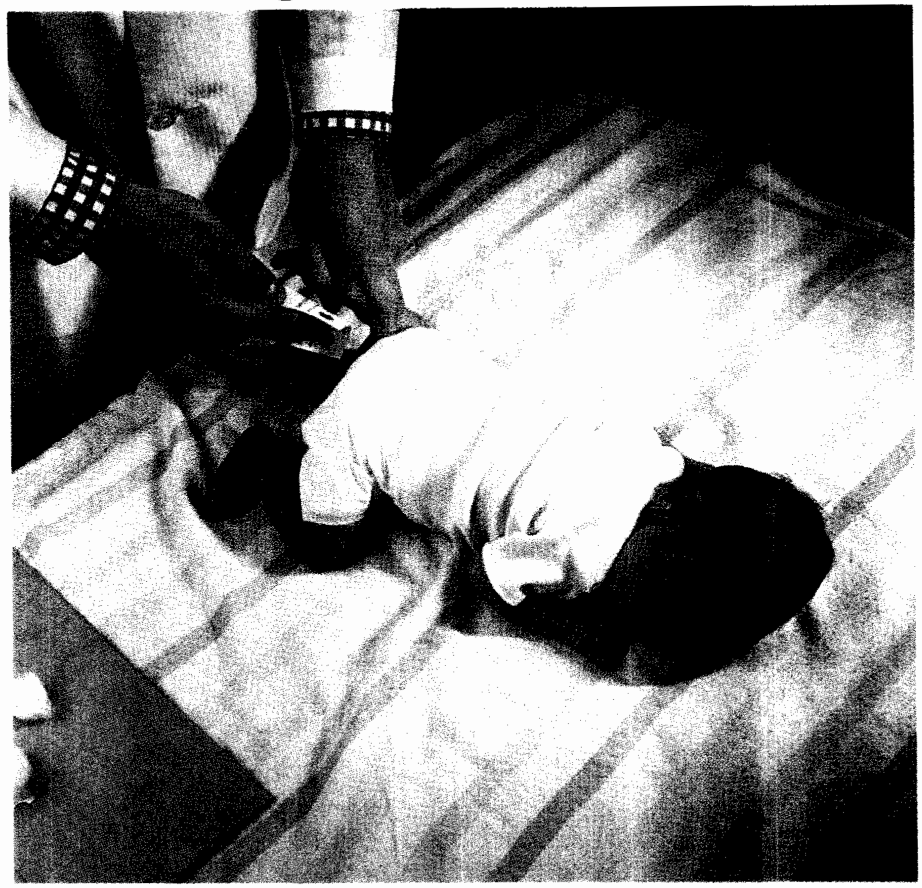
For the third consecutive year, KEYBOARD has won an award for graphic and journalistic excellence. We are pleased to gain such recognition, but it's your opinion that really counts. Many of you received a survey questionnaire last summer asking for your views on KEYBOARD as it is now and what you would like changed to make the magazine more interesting and useful. The response was very gratifying, and we thank all of you who took the time to return the questionnaire. Even more, we appreciate the written comments many of you included. You told us that KEYBOARD is very well received as it is now, that 70% of you prefer it over other similar company-sponsored magazines, and that all sections are widely read with high scores for general interest, quality and information.

Of special interest to us were the comments offering suggestions for improvement. We have already begun implementation of some of these suggestions and plan on implementation of others as quickly as we can. If you have comments on our progress, or if you did not receive a questionnaire and wish to offer suggestions, please write. Your comments are invaluable in guiding our efforts.

Features

Departments

Screening Newborns
For Congenital Hypothyroidism



by Ivan Francis

Cretin, n. Deformed idiot of a kind found especially in alpine valleys. (The Concise Oxford Dictionary)

This word borrowed from the French language originally described infants with congenital defects characteristic of the iodine-deficient alpine regions of Europe. Endemic cretinism has since been found in other alpine regions throughout the world.

Cretinism is caused by hypothyroidism, a severe deficiency in thyroid hormones at birth. Iodine is required for the production of these hormones. Cretinism caused by an iodine-deficient diet has been largely eliminated by the use of iodized salt in the diet, but there is still a spontaneous incidence of about 1 in 6000 births. This incidence is not related to an iodine deficient diet and cannot be prevented. Furthermore, infants with this condition often are not diagnosed until after irreversible physical and mental damage has occurred. The victims are completely dependent and often spend their lives in institutions.

However, hypothyroidism can be treated and the tragedy of cretinism avoided. All infants born in Victoria, Australia, are now tested for hypothyroidism by the Mont Park Pathology Centre. A 9825A-based system is used to handle the large amounts of data collected.

Causes of Hypothyroidism

The major hormones produced by the thyroid gland are thyroxine (4 iodine atoms per molecule) and tri-iodothyronine (3 iodine atoms per molecule). Thyroid hormone secretion is stimulated by the action of the pituitary hormone thyrotropin (thyroid stimulating hormone, or TSH). The secretion of TSH by the pituitary is in turn controlled by a process called "feed-back inhibition," where increasing levels of thyroid hormones from the thyroid gland act on the pituitary, shutting off production of TSH. In this way the circulating levels of thyroid hormones are kept constant.

The release of TSH from the pituitary also requires the presence of a hormone produced by the hypothalamus called thyrotropin releasing hormone (TRH). Thus, lack of hypothalamic function also can cause a deficiency of thyroid hormones due to the subsequent absence of TSH. There are therefore three major causes of hypothyroidism:

a. Absent or functionally deficient thyroid gland, termed primary hypothyroidism.
b. Deficiency of TSH from the pituitary, termed secondary hypothyroidism.
c. Deficiency of TRH from the hypothalamus, termed secondary or tertiary hypothyroidism.

In neonates (newborns), 90% of congenital hypothyroidism is due to a primary disorder of the thyroid gland, characterized by low blood levels of thyroxine and high blood levels of TSH due to lack of feed-back inhibition. The remaining 10% is due either to pituitary or hypothalamic deficiency, and in these cases both thyroxine and TSH levels are low.

Treatment for hypothyroidism consists of thyroxine taken orally and, if begun before the age of one month, will allow normal development of the child.

Screening Programme Development

The first screening programme for neonatal hypothyroidism was established in April, 1974, by the Quebec Screening Network for Metabolic Diseases. Their approach was based on a radioimmunoassay (RIA) for thyroxine. Since then, other laboratories in North America, Europe and Australia have used this model for initiating their own screening programmes.

Some laboratories use a different approach and base their screening on an RIA for TSH. This method will not detect hypothyroidism due to pituitary or hypothalamic deficiency, but has an advantage over thyroxine assay in that fewer false positive cases need to be followed up. In neonates, low thyroxine levels are found in a number of conditions unrelated to hypothyroidism, and all require further testing to determine which condition exists.

The samples used by most laboratories consist of blood collected by heel prick from infants three to five days old. The blood is soaked onto filter paper, dried, and in this form sent through the mail for testing. This type of sample was introduced by Professor Guthrie in 1963 for the phenylketonuria (PKU) screening programme. Thus, laboratories with existing facilities for PKU testing require only the extra equipment and expertise to introduce hypothyroid screening.

In Victoria, after an initial trial period we began complete coverage of newborns in May, 1977. This involves testing 60 000 infants annually. For a trial period of three months, 12 000 infants were tested for both TSH and thyroxine. After this trial we were satisfied that a thyroxine assay alone is adequate as an initial screening test.

All samples having a low thyroxine are retested for both thyroxine and TSH. Using this procedure, infants with primary hypothyroidism are readily detected and referred immediately to a treatment centre, while second samples are requested from infants with suspected secondary hypothroidism.

Assay Principle

The establishment of screening programmes for neonatal hypothyroidism was dependent on the development of assay methods sensitive enough to measure the minute quantities of hormone present in the blood. For example, in the ⅛th inch disc of dried blood we use in the assays for thyroxine and TSH, there is about 2×10^{-10} grams of thyroxine and only 4×10^{-14} grams of TSH in an average sample. In addition, they must be measured in the presence of other hormones of similar molecular structure. RIA, first used in 1964 to measure insulin in blood, has provided the means. The principle of this type of assay is as follows.

Use is made of the reaction between an antigen and its corresponding antibody. The substance to be assayed (ligand) is injected into a suitable animal in a form that is immunogenic (capable of eliciting an immune response). Small molecules, e.g. thyroxine, must be coupled to a protein molecule before they become immunogenic. The animal is subsequently bled, and the antibodies thus obtained are tested and, if suitable, are used in the assay. An important feature of the antibodies is their ability to recognize and bind the ligand to the exclusion of like molecules that also may be present in the assay system. This determines the assay specificity.

Trace amounts of a radioactively labeled form of the ligand (tracer) are added to the assay system; the tracer behaves in exactly the same way as the unlabeled ligand. A mixture of the ligand (L), antibodies (Ab) and tracer (T) is allowed to react. Both the ligand and tracer bind to the antibodies with equal avidity.

$$L + T + 2Ab \rightleftharpoons L - Ab + T - Ab$$

The amount of tracer and antibodies is kept constant for every tube in the assay, this amount being adjusted so that 50% of the tracer is bound to the antibodies in the absence of ligand; this is the maximum binding. When ligand is present in the system, it takes up some of the antibody binding sites, thus less tracer is bound to the antibody.

Once the reaction has gone to completion, the bound fraction (antibody with bound ligand and tracer) is separated from the unbound ligand and tracer. The radioactivity of the bound fraction is measured and is inversely proportional to the amount of ligand present, i.e., when ligand is absent, tracer binding (radioactivity) is at a maximum; whereas the greater amount of ligand present, the smaller is the amount of tracer bound to antibody.

Data Handling

We assay samples in batches of 120 along with standards containing known amounts of thyroxine (or TSH) and controls (identical samples included in every batch as a check on consistency). Once processed, the batch is loaded into an automatic gamma counter. (In our system, the radioisotope used is iodine-125, a gamma emitter with 60-day half-life.) Data is output in the form of counts per minute (CPM) via teletype onto punch tape.

Our data handling system consists of a Hewlett-Packard 9825A Desktop Computer with 9863A Tape Reader and 9871A Printer. The software was developed in our laboratory to meet the specific requirements of the RIA screening assay.

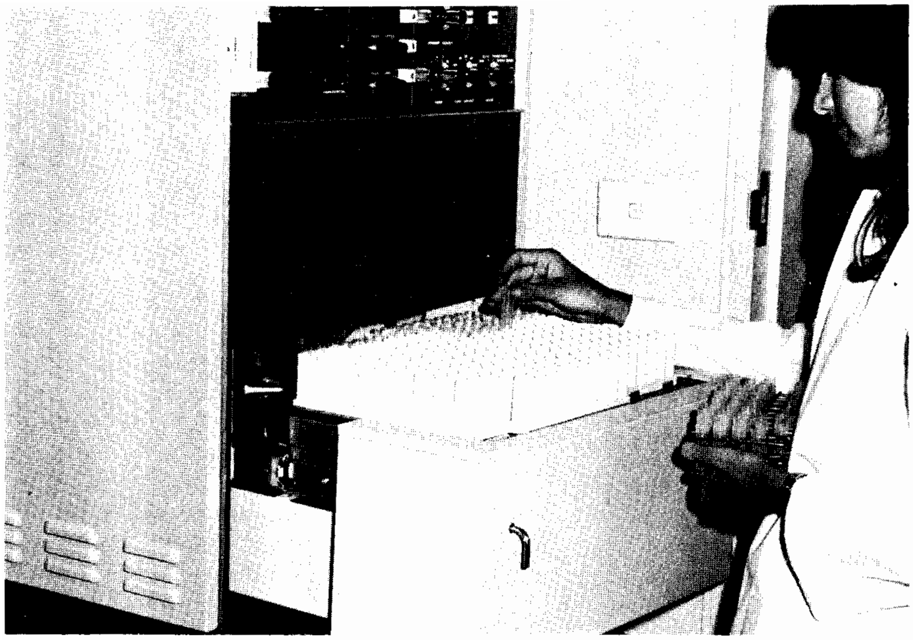
The CPM are converted to the form:

$$\frac{\text{CPM (bound fraction)} \times 100}{\text{total CPM added}}$$

In this form (B/T%) the performance of the antiserum from batch to batch can be compared. The data from the standard tubes is fitted by means of least squares to a cubic equation:

$$a + b/Y + cY + dY^2 = X$$

where $Y = B/(T-B)$ and X = amount of thyroxine or TSH present.



Kerryn Bignell loads RIA tubes from the centrifuge into the automatic gamma counter.

The computer programme will reject the standard data falling too far from the line of best fit and recompute the equation until a satisfactory fit is obtained. The plotting functions of the 9871A Printer are used to plot a standard curve, B/T% vs. concentration of thyroxine or TSH. The standard data used to compute the curve are also plotted for comparison. The printout then gives the correlation coefficients for the standards both before and after rejection of unsuitable points along with a list of rejected points.

The results of the 120 unknowns are then printed out along with an identifying number. The 9825A is programmed to give these results both in the order in which they were assayed and in order of magnitude from lowest to highest. This latter arrangement facilitates the selection of samples with the lowest (in the case of thyroxine) or highest (in the case of TSH) values. To assess the day-to-day performance of the assay, a frequency distribution curve of the 120 infants' results is also printed out with its mean, standard deviation and coefficient of variation. Any deviation from established values for these parameters warns us of a possible fault in the assay system.

As stated earlier, cases of primary hypothyroidism are readily detected by use of a TSH assay on infants with low thyroxin. However, between 1% and 2% of samples with low thyroxine and without a raised TSH must be retested to exclude secondary hypothyroidism. The string variable ROM is used to enable us to store the data on infants requiring a repeat sample to be collected and produce a programmed letter requesting the repeat sample. When obtained, the results of the assays on the repeat sample are entered via the keyboard of the 9825A, which appends them to the previous results on the infant and produces a second letter informing the doctor of the latest results.

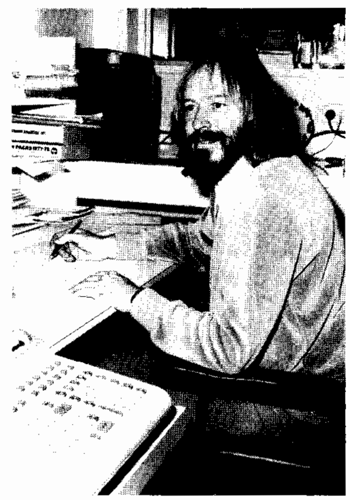
In this way, we are able to keep track of up to 100 repeat specimen requests per month without the need for secretarial assistance.

The data handling ability of the 9825A, along with its compact dimensions, makes it ideal for our busy, crowded laboratory. Information on 1000 infants who have received repeat tests can be stored on one cartridge, thus allowing rapid information retrieval. Furthermore, the data storage capacity of the 9825A system is readily increased by means of a flexible disk drive or external tape memory, so that if, in future, additional screening tests are added to our programme the system can expand to meet requirements.

Results of Screening Programme

To the end of 1977 we have tested 40 000 infants, of which 11 have been confirmed as having primary hypothyroidism. This is an incidence of greater than 1 in 4000, which is higher than that reported from other laboratories (e.g. 1 in 6000 from Quebec), though as yet the numbers are too small to come to any conclusions as to the significance of the higher incidence.

The economic benefits from screening programmes such as this come through removal of the need for provision of a probable lifetime of institutional care. However, how does one measure the benefits gained from the prevention of the tragedy of physical and mental retardation? END



About the Author

Ivan Francis completed a Diploma of Applied Chemistry at the Royal Melbourne Institute of Technology in 1960 and a Bachelor of Science at Melbourne University in 1963, majoring in biochemistry. After two years of post-graduate work at Monash University, he joined the Mental Health Authority of Victoria as a biochemist and is now in charge of the State Screening Laboratory at Mont Park.

We thank the Mental Health Authority of Victoria for permission to publish details of this work.

Ivan Francis
Mont Park Pathology Centre
Melbourne
Victoria, 3085, Australia

A Method of Business Forecasting

by Iwan Bratt and David Lindquist

Business forecasting is based on three assumptions:

1. Economic magnitudes — levels of production, income, prices, wages, interest rates, and consumer expenditures, for instance — are bound together in a system having a great deal of stability over time.
2. Future changes in these magnitudes will result from causes presently operating, and they can be deduced from observed current symptoms.
3. The properties, or nature, of these causes or symptoms and probable future consequences can be discovered by studying past experience.

A long range plan (LRP) is a business forecast five or more years into the future. It is used to make business decisions early to efficiently adapt to predicted future changes. Many different types of LRP techniques have been developed. Some of them are used, some are misused; some are good, some not.

Most LRP techniques developed in the 1960's made the assumption that outcomes followed a simple, continuous pattern, but at a faster rate of change than in earlier years. It was sufficient to adapt the properties of the enterprise to anticipated future effects of environment. Long range planning could be a rather rigid process and plans could be valid for a number of years.

At the beginning of the 1970's the overall environment in which business must survive showed a more complicated pattern — one of increasing oscillation and complexity, combined with such new, disruptive occurrences as the energy crisis. This, together with the still accelerating rate of change, makes long range planning more important than ever and places a heavy demand upon improvement of the techniques used. It is now essential to work with the properties of systems, not just their effects as in the 1960's, in order to obtain a satisfactory quality of forecasts covering larger, integrated aggregates of systems.

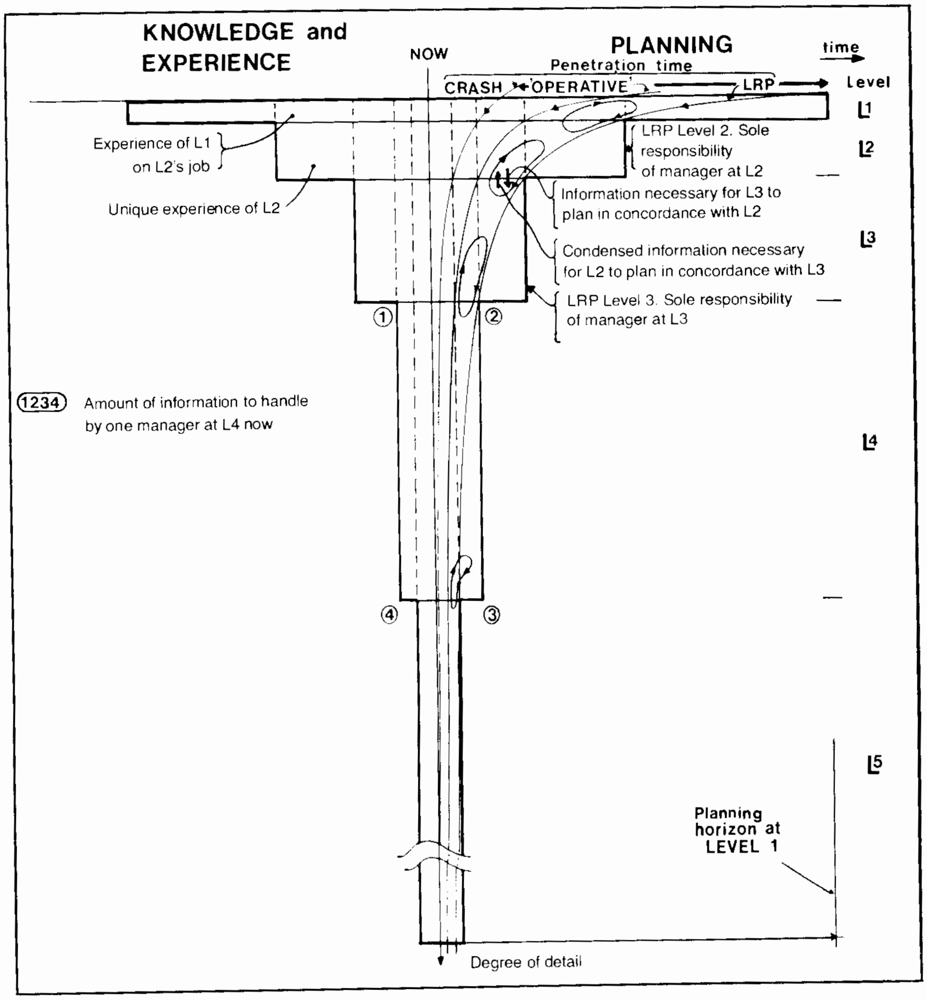


Figure 1

It is, however, not enough to have a good idea of future environmental properties. A company now must be flexible enough to adapt to a number of unforeseeable changes in the environment, and buffers, or contingencies, must be built into the company's long range plan. The forecasting technique must be designed to support management in this adaptation process by providing early warnings when unexpected events disrupt the pattern on which the plan is based. The technique must operate so rapidly and be so flexible it can be used to trigger immediate corrections to the direction of the company when unexpected events happen.

Developing an LRP starts deep within the company's organization. The planning process must be well integrated into the lower levels of management. To assimilate all the gathered bits of data into a cohesive, comprehensive format and then to forecast and simulate alternate futures for timely evaluation requires a tool adequate to the task — a computing system.

Integrating LRP Into the Organization

The prehistory of a policy decision is, and should be, an unstructured process occurring at various levels in the organization. Needs and opportunities are compiled with plans at all levels, and a policy in accordance with the needs and opportunities of the entire company is generated. Once such a decision is made, it causes a large number of detailed decisions to

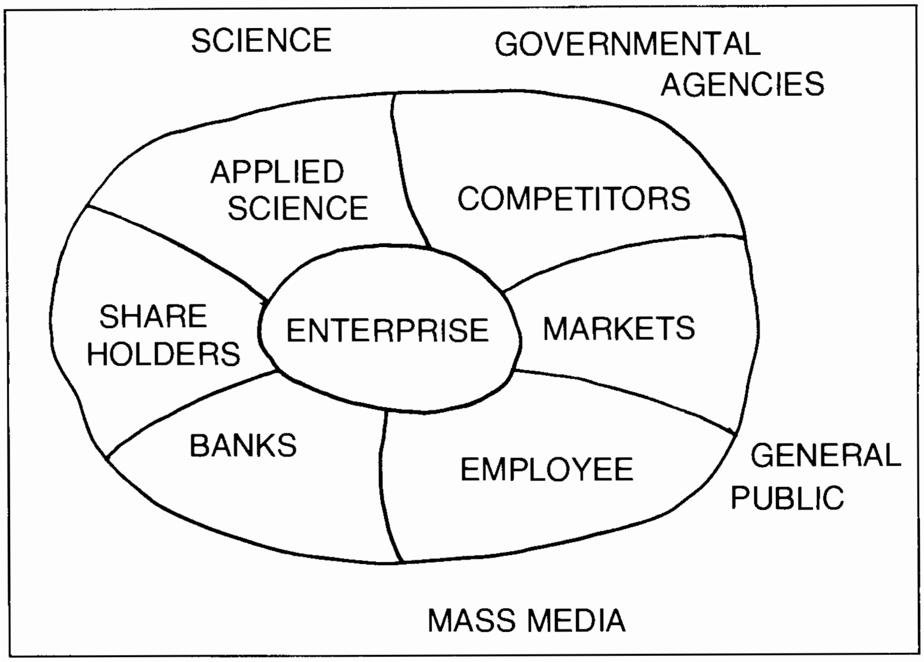


Figure 2

be made at such diverse locations as on the workshop floor and in the sales organization. The efficiency of this costly but necessary process must be optimized.

Long range planning is an integral part of this process, illustrated by Figure 1. The figure is based on the fact that the capacity of a person to handle information in a rational way is limited, even though it may vary greatly among individuals.

Investment such as developing a new product or market or building a new factory puts our planning horizon at Level 1 (L1) some decades into the future, while the planning horizon for the worker at Level 5 (L5) may be a week or day. The length of the planning horizon and the degree of detail indicate the amount of information each member of the organization should be able to successfully handle.

As an LRP decision develops through the organization along the penetration line, each level adds information to it. When it reaches the workshop floor early enough to be put into practice, all necessary information should be available. Crash programming can shorten penetration time, but this obviously will delay the penetration of other decisions, some of which may reach their destination too late or with insufficient information.

The LRP at L2 is based on concepts operative at L1. L2 has to reformulate this information and add further information in concepts relevant to L3. On the other hand, L2 receives from L3 more detailed and sometimes critical information that is available only to L3 but affects L2's planning.

We find that planning — operative or long range — covers a band along the right side of the time-detail diagram in Figure 1. The quality and speed of information flow within this band has great impact on the successful adaptation to future events. As long as LRP at each level is based on realistic information, expensive crash programs can be avoided and profitability can be improved.

Forecasting

All planning deals with the future. It is essential at each level to predict the development of the environment of the enterprise in concepts relevant to operations.

In Figure 2, three types of environments are depicted. Group A has environments affecting but hardly affected by the enterprise. Such systems as world market, governmental agencies, mass media, science/technology, and the general public are in this group.

Environments in Group B(1) interact with the enterprise; applied sciences, competitors, markets, employees, banks and shareholders.

Group B(2) comprises the enterprise's internal environment, such as management style and ability, facilities, philosophies, etc.

We think it is necessary to use at least two methods of forecasting in parallel, which are as different as possible in order for them to act as cross-checks for each other:

- The judgement of an experienced manager, which takes into account factors not readily quantifiable, and
- A computerized forecasting technique which treats facts and assumptions according to a totally defined logic.

Most forecasts are based on some combination of judgement and historical data. The subject we address in this article is the second of the two methods.

Our knowledge of the structure of interactions between Group A systems is very limited, but the basic assumptions are:

1. All systems may interact, and
2. The properties of the systems involved continue changing as before.

We can make a reasonably good prediction of the future behavior of the Group A systems as long as these assumptions are valid. Entirely new properties may appear, but we have means to alert us when there is reason to believe that substantial deviations from earlier patterns may be expected — a sort of early warning system.

Suppose we want to forecast industrial production (IP) in a country (C) for the next few years. It is reasonable to assume that the volume produced is a result of interaction between a number of systems affecting raw material supply, cost, quality, marketing, etc., in a number of countries directly or indirectly involved (see Figure 3).

It is also reasonable to expect that industrial production (IP) from the industrial complex system in the country (C) interacts with an unknown number of unknown systems in a multiple feedback loop system. This is likely to generate oscillating deviations in a number of loops superimposed upon a general long-term trend. Assumptions about unknown mechanisms causing these interactions should be avoided, since guesses may affect the validity of the forecast. We can, however, make assumptions on the mathematical framework within which an optimizing system could determine values of (clusters of) property parameters from historical data.

We can for example, assume that the algorithm for IP is the sum of:

1. A general trend, either linear or exponential.
2. Possibly a crisis of the 1975/76 type.
3. A seasonal cycle.
4. A number of cycles having varying amplitudes and durations.

We want to determine how each of these components have contributed to IP in the past, since we assume that each will continue to develop as before. This can be formulated in a mathematical model.

(t = time)

$$IP = P_1 + P_2 \bullet t + \frac{P_3}{1 + (P_5(t - P_4))^4} +$$

$$(P_6 + P_7 \bullet t) \bullet \sin(\pi(1.5 + 4 \bullet t)) +$$

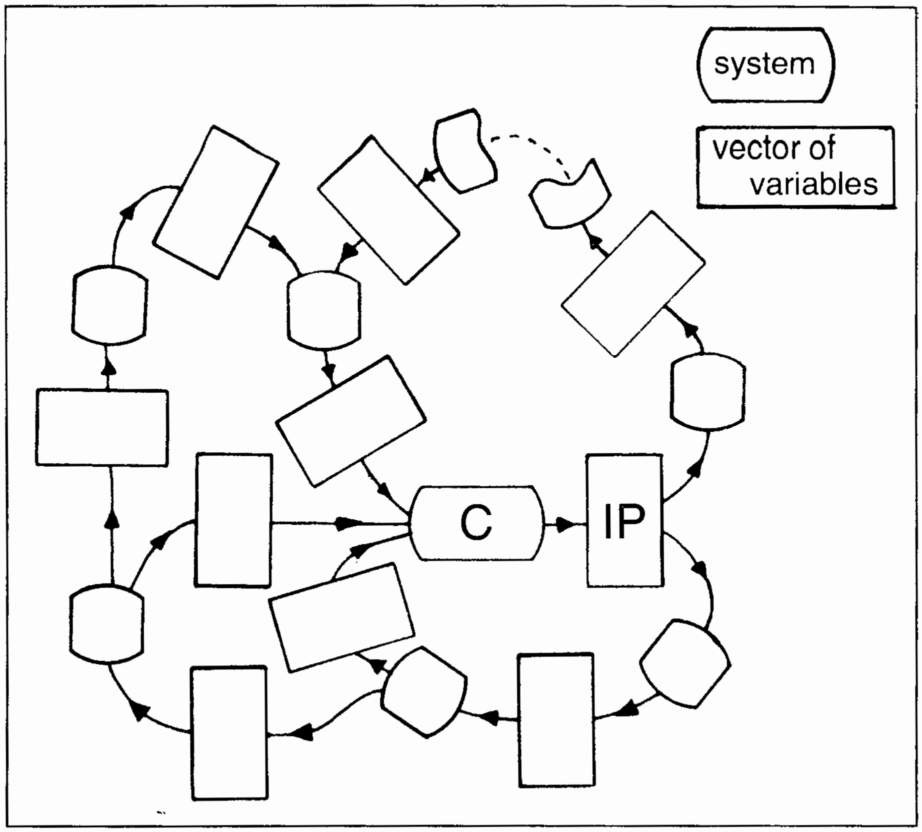


Figure 3

$$\sum_{i=0}^{n} \left[(P_{8+5 \bullet i} + P_{9+5 \bullet i}) \bullet \sin(P_{10+5 \bullet i} + P_{11+5 \bullet i} \bullet t + P_{12+5 \bullet i} \bullet t^2) \right]$$

We now do a computer search for values of all P:s, so that for time passed (T = period of available historical data)

$$S = \sum_{j=-T}^{now} (IP_{calculated} - IP_{measured\ j})^2$$

is minimized. The parameters P represent clusters of property values and changes in the properties of systems in the different feedback loops, and for the present we can do no better than assume the various P:s — representing the development of properties — keep their values in accordance with our basic assumption; i.e. that properties of the systems involved will go on changing in the same manner as before.

We measure the accuracy by which the model describes past events by

$$\text{Sigma } \sigma = \sqrt{\frac{S}{n - m}}$$

where n is the number of observations and m is the number of parameters utilized.

Experience from a few hundred forecasts produced during the last ten years shows that:

1. If sigma is less than 1%, the forecast generally is reasonably correct for the next few years.
2. When adding a new value to a time series, sigma should decrease. If several new values give greater sigma values than the smallest value previously found, this is a signal that actual events have ceased to follow the pattern defined by the model. This means that something new and unexpected has come into the picture and should be regarded as an early warning that a new term has to be added to the model.

This was the case in the middle of 1974 when industrial production in the U.S. ceased to follow a pattern that had been stable for about a decade. We added a "crisis" term to the model. This term, having no support in past events, lowers temporarily the forecasting power of the model until enough data is available to determine the pattern by which the crisis affects events (see Figure 4).

Comparing the period of fluctuation of components found in different forecasts we generally find periods with remarkably small differences, often less than 1%, which

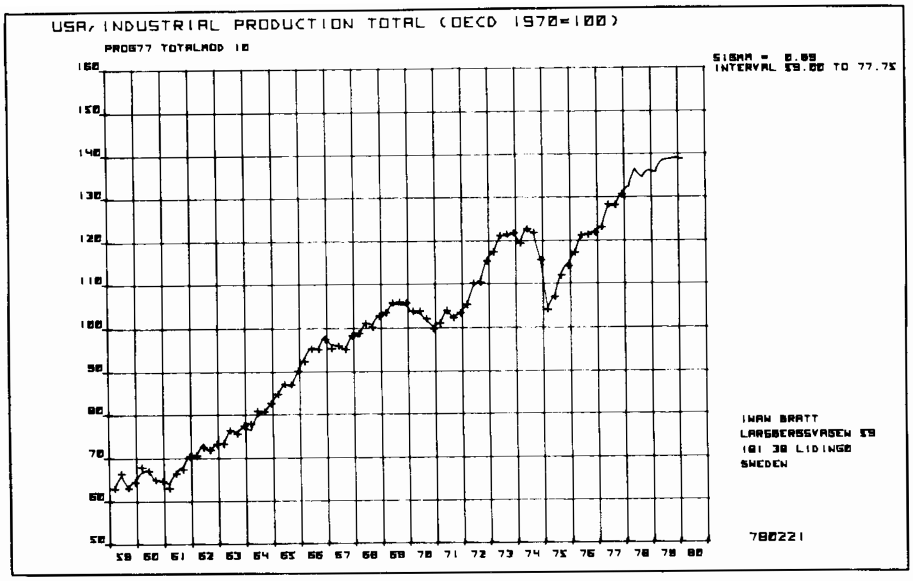


Figure 4

indicates that these components are likely to correspond with real events. This method may be developed further by deducting interaction patterns from the way these events spread. It is unreasonable to expect two sequences of events to so closely parallel each other over decades if there is no interdependence. Minor deviations should be regarded as phase shifts. Utilizing the crisis term in forecasts of IP in a number of other countries, we could follow the way the crisis traveled worldwide.

Simulation

Forecasting events in the Group B systems is more complicated. Group A systems are little affected by Group B types of environments, but Group B systems are affected by Group A environments. Since the reason for forecasting is to find new and better ways to adapt to future environments, we must formulate models describing the relevant features of Group B systems and the interaction between them. We then try various means of influencing the Group B systems subject to effects from Group A systems such that the future of the enterprise is near optimum. This we call simulation.

We model, for instance, Group B(1) systems (Figure 2) and the way they interact with the enterprise, B(2), one at a time. We then search the property parameters that reasonably fit the historical data of the B systems if such are available, or we use our best judgement. By cautiously combining models into larger systems, we arrive at a workable model large enough to test the long-range effects of various policy modifications. We also test the sensitivity of property parameters against such important effects as liquidity, solidity, and profitability in order to direct model improvement efforts toward submodels which impact these effects.

To facilitate this process, we have developed a simulation language where each type of subsystem is represented by an algorithm. We then can make a graph showing how various systems interact. The computer assembles a program corresponding to the graph and produces a list of variables and parameters in clear language to facilitate input of some hundred data. We then simulate gradually changing strategy parameters to find a near optimum strategy, keeping an eye on feasibility and risk involved. See Figure 5.

We generally use the HP 9830A for simulation, as this gives us an extra and very important facility. We can stop computations at any time and modify properties of any system or modify the model structure when a critical situation seems to be ahead of us and so try various means to avoid pitfalls very much the same way a manager would do it.

At best, forecasts of effects from Group A systems are predictions of what would happen if all systems continue changing their properties as before. The models of our Group B systems are not what they actually are now or what they will be in the future, but what we think they are or will be. So, having made a forecast for our enterprise we must be aware that it is based on what we think will be the properties of our systems.

Deviations of actual events from forecasts could then mean:

1. That one or several effects from the Group A systems deviate. By comparing actual events with simulated ones we can locate which events do not follow our predictions, and this tells us that something new and unpredicted is likely to have happened in systems operating in a specific loop — an early warning of any kind of coming crisis. We can then concentrate our efforts to find what really happens to a limited area.
2. That our models of Group B systems are wrong or not sufficiently accurate. We can study deviations between actual and simulated values of the different variables and sometimes trace the deviation back to its source and so locate the system, the simulation model of which is likely to need improvement. We think it is most dangerous to modify a good section of a model in order to compensate for inaccuracies in another section. Model improvement is a difficult task, and the difficulties grow very rapidly as the size and complexity of the model grows. We recommend partitioning the model and checking sections both on reasonability and against historical data whenever possible. Our simulation language is designed to simplify this process.

Uncertainty

Since all planning deals with a more or less distant future and no model is complete or infallible, we must calculate with uncertainty. This uncertainty has several sources, and part of it can be computed, but never all of it. Our models can never

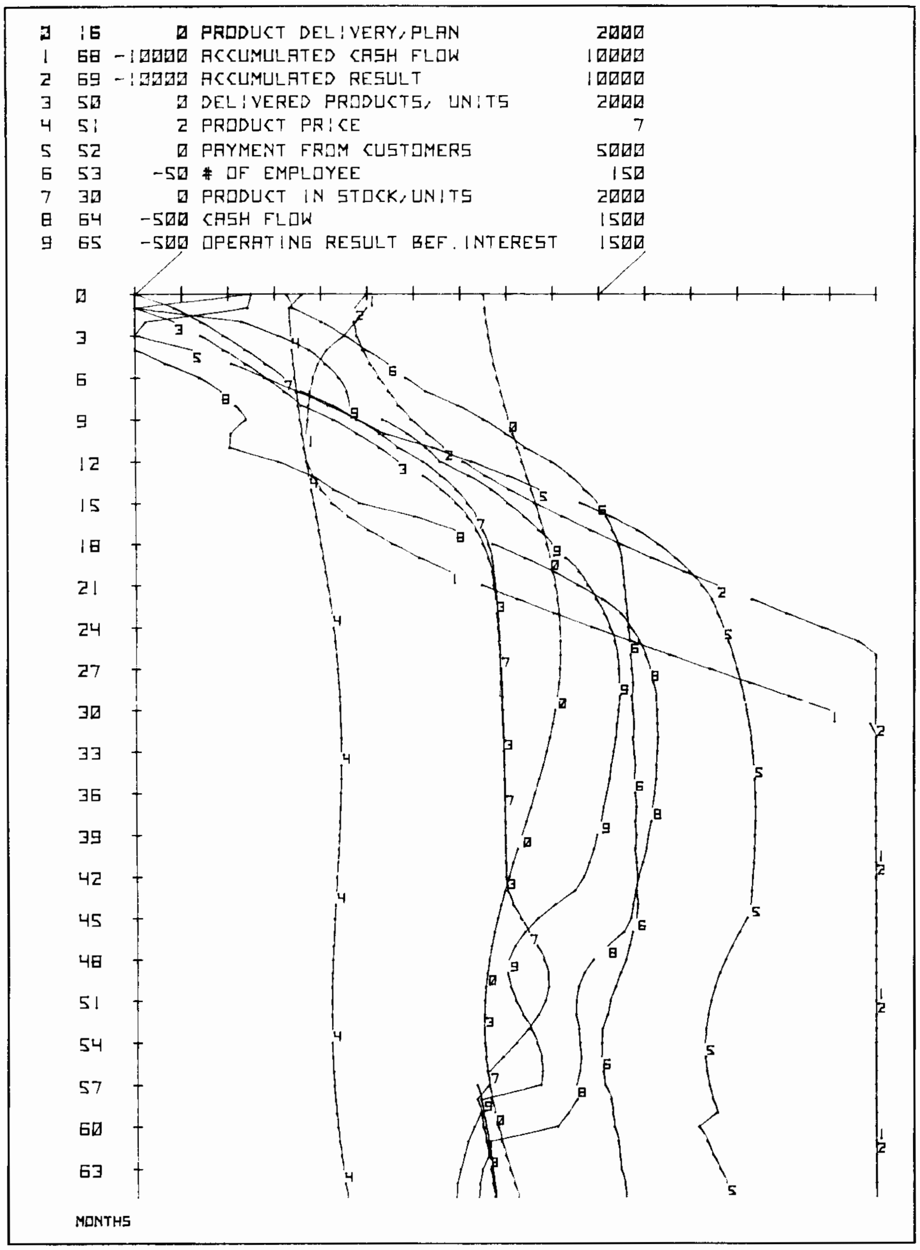


Figure 5

reproduce historical events exactly. This means that our optimal parameter values can be determined with only a limited accuracy, but we can estimate the approximate magnitude of inaccuracy of each parameter if we assume that all parameters for instance contribute equally to the overall inaccuracy. Using the Monte Carlo technique, we then can make a number of forecasts and we get a band, the center value of which is our most probable future.

These forecasts of effects from mostly Group A variables may be introduced in our simulation model of Group B systems. The future properties of these systems are never known precisely and, besides, we have no means of measuring how well we know them. We have tried to quantify these uncertainties and how they are linked to each other by discussing them with managers concerned. The subjective uncertainties agreed upon are introduced into the simulation models and the properties varied using the Monte Carlo technique. The overall uncertainty is used to derive the risk that certain variables such as solidity and cash flow may be critical at a certain time in the future.

Should effects from a Group A system deviate from forecasts (due to bad modelling or a crisis) but our Group B simulation model (the description of how the properties of the enterprise (B2) and its immediate environment (B1) interact) has proved to be correct, the simulation system allows us to compute alternative strategies in a matter of minutes searching for a modified strategy by which we adapt to the new future of A-environment. This is one of the main objectives on modern LRP.

We feel we will be faced with increasing difficulties in finding ways to adapt to an uncertain future; improved forecasting and simulation techniques are necessary. These techniques also give information on foreseeable risk so that we can organize for replanning and building buffers in time. We think that when things tend to go wrong, more reliable predictions give us an early warning, so that we can start replanning and reallocating buffers as soon as possible and avoid paying a huge penalty because of delayed countermeasures. END

About the Authors

Iwan Bratt received a Mechanical Engineering degree in 1939 from Chalmers Technical University, Gothenburgh, Sweden. During World War II, Mr. Bratt worked with steam plants, packaging machinery design, and guided missile development. More recently he has been involved in industrial automation, systems research, mathematical forecasting and simulation, and long range planning.

David Lindquist received his Electronic Engineering degree in 1960 from the Tekniska Gymnasiet, Göteborg, and his Bachelor of Commerce degree from Handelshögskolan in Göteborg in 1966. He started his career with military electronics, but has since the mid-1960s worked with company development and long range planning for a number of companies in different lines of business in nine countries. He is presently Director at the head office of Esselte AB, a Swedish company.

David Lindquist
Esselte AB
Fack, S-100 41
Stockholm, Sweden

Iwan Bratt
Larsbergsvaegen 59
181 38 Lidingoe, Sweden

The 98224A Systems Programming ROM For The 9825A

by Steve Leibson, Hewlett-Packard Calculator Products Division

What would you call a programming language enhancement that enables the host machine to interact in a data communications environment? Historically this would be known as a Data Comm ROM such as the one available for the 9830A/B and would provide a firmware-based emulation of a communications terminal. Based in ROM, such an emulation is limited in flexibility. No problem exists if the capabilities provided are sufficient, but if greater capability is needed, a software-based emulation is more useful. The 98224A Systems Programming ROM for the 9825A offers just such an approach. The provision of additional statements to the HPL (High Performance Language) vocabulary of the General I/O, Extended I/O and Strings-AP ROMs and the 9825A mainframe enables creation of emulator programs with greater capabilities and more versatility.

As a bonus, basic tools such as keyboard operation under interrupt, end-of-line command formatting, a programmable Store statement, and the increase of continuous input rates by a factor of 20 expand the application of the Systems Programming ROM far beyond the realm of serial data communications. Machine operation by untrained personnel, programming for systems having memory configurations that are unknown, in process of change or dissimilar, and interfacing to peripherals with unusual interfacing problems are also applications that can benefit.

For users of the 9825A who require a prepared terminal emulator, the Terminal Emulator Applications Software Package, 09825-10040, configures the 9825A to act as a printing terminal at rates of up to 9600 baud.

Communications Terminal Operation

In order to understand the use of statements provided by the 98224A, the operation of a communications terminal must be understood. A block diagram is shown in Figure 1. Two tasks are to be simultaneously and continuously performed:

- Characters typed on the keyboard are to be transmitted.
- Incoming characters are to be displayed/printed.

The serial receiver and serial transmitter are implemented with a 98036A Serial Interface Card, and the 9825A keyboard and display are the terminal keyboard and display.

Those familiar with the 9825A programming language know that both of these tasks can be performed using statements in the General I/O ROM with no enhancements required. The 9825A keyboard can be read with a Read Binary (RDB) statement to select code 0 and bytes output to the serial interface with a Write Binary statement. Incoming characters may be collected into strings with another Read Binary statement and the assembled lines output to a printer using a Write (WRT) statement.

Problems are encountered when both tasks are to be concurrent, as they must in a proper terminal emulation. Using a Read Binary on the 9825A keyboard will halt program execution until a key is pressed. The same is true for the input process, since the time that a character may be received cannot be known in advance. If the program should halt while performing one task, characters may be lost in the other task and be overwritten in the interface if not obtained by the program before the following character arrives.

The solution to the problem is to take advantage of the 9825A's interrupt capability to read interfaces. Using the Transfer statement in the Extended I/O ROM, incoming characters can be placed in a buffer until the mainline program has time to print them.

There remains, though, the problem of the characters coming from the 9825A keyboard. If a read is performed on the keyboard, program execution still will halt until a key is pressed. If no key is pressed, the program will never have time to print the characters entering the transfer buffer from the 98036A, and eventually the buffer will overflow; characters will be lost. This problem is solved by operating the keyboard under interrupt.

Keyboard Interrupt Operation

None of the interrupt-oriented statements in the Extended I/O ROM may be used on the internal keyboard of the 9825A, so the 98224A ROM provides a set of statements that allows keyboard interrupt operation. The On Key statement establishes a buffer for incoming key codes resulting from key presses, creates a link to a specified program routine that will process the buffered key codes using the end-of-line interrupt subroutine jump (used by all interrupt service routines in the 9825A), and finally will sever the link normally existing between the keyboard and the operating system in the 9825A. The HPL program now has complete control of the keyboard at all times the On Key is in effect. The interrupt service routine for a terminal emulator would simply output the ASCII equivalent of the key code to the serial interface for transmission. This ASCII equivalent may be obtained from the ASC function, also in the Systems Programming ROM.

Interrrupt operation of the 9825A keyboard is not only useful for a terminal emulation program, but for any program requiring input from the keyboard. The program can be

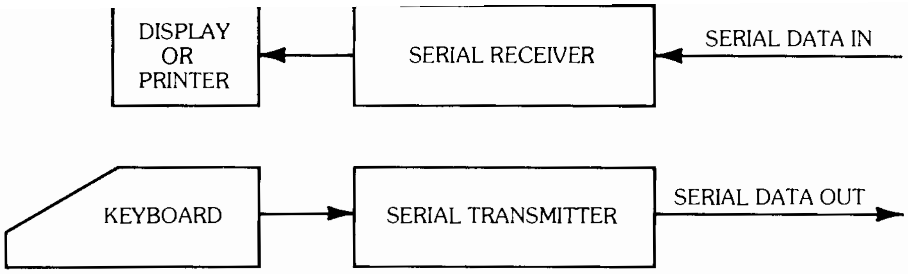


Figure 1. Communications Terminal Block Diagram

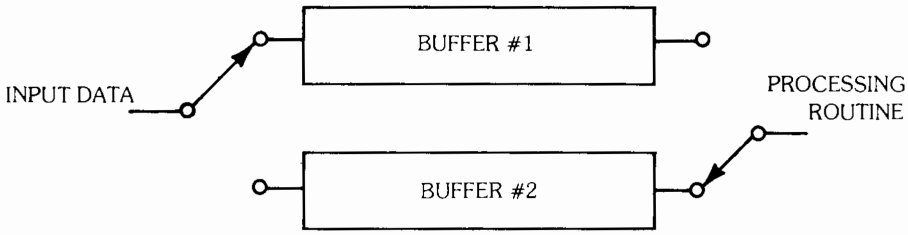


Figure 2. Double Buffering Scheme

protected from disruption by ignoring illegal key codes or issuing warning beeps or messages when illegal keys or key sequences are encountered. All keys are diverted from their normal operation except for the reset key.

Higher Speed Buffering

The usual method for buffering input data continuously is to use a double buffering scheme as shown in Figure 2. Two buffers are used, and one is filled while the other is processed. Assuming the buffer can be processed faster than it can be filled, the only factor limiting input rate is the time required to switch between buffers. This must be less than the time between the arrival of adjacent characters or some characters may be lost. The double buffering scheme on the 9825A allows a continuous input rate of about 60 char/sec. A higher speed method is necessary for the higher bit rates encountered in typical data communications environments.

This method is provided by the Buffer Read (BRED) statement. A single interrupt buffer is used for the incoming data and is periodically flushed using BRED. The interrupts are disabled for a brief period when the buffer is emptied and then the characters are again allowed to enter the buffer. The BRED statement is actually a function that returns a string of characters representing the contents of the interrupt buffer before the BRED was executed. Continuous input rates of over 1000 char/sec have been possible using BRED, and since any byte-oriented buffer containing input data may be acted upon by BRED, any application requiring the handling of continuous input data can benefit from using the BRED statement.

Beyond Terminal Emulation

Although one of the most important applications is terminal emulation, the 98224A Systems Programming ROM is used in other applications as well. Instead of emulating a communications terminal, some applications require communications with a terminal. Terminals are of two major types, teleprinters and CRTs. Both terminal types can have unique interfacing requirements, many times complicated by the lack of handshake capability over a serial link. For example, teleprinters may require several hundred milliseconds to execute a linefeed/carriage return sequence. The transmitter must wait until it is sure the sequence has been completed or characters subsequently transmitted to the teleprinter may be lost or printed in the wrong place. CRT terminals may require special character sequences to maintain certain modes such as inverse video or underlining. These sequences are generally required at the start of each line.

To aid in the interfacing of such terminals, the End Of Line or EOL statement was created. This statement redefines the normal carriage return/linefeed sequence output by the Write (WRT), LIST# and Catalog (CAT) statements of the General I/O and Flexible Disk ROMs. The replacement sequence may be up to seven characters in length. In addition, a time parameter allows slower teleprinters to complete sequence commands before further data is output. The EOL statement affects software only and may be used in conjunction with any interface for the 9825A.

The 98036A Serial Interface is a very flexible device and can be programmed to perform in many different ways. This programming is accomplished through a series of registers within the interface. Access to these registers is complex, and the method of access varies depending on interrupt operation modes. To relieve the access complexity, three statements are provided by the Systems Programming ROM to perform all transactions necessary with the programming registers and automatically handle the various access methods. Write Serial Control (WSC) and Write Serial Mode (WSM) access the control and mode registers respectively. The control register controls the Clear To Send and Data Set Ready output lines (Request To Send and Data Terminal Ready lines for a 98036A Opt. 001), transmitter and receiver enabling, and the resetting of error bits in the serial status word. The mode word controls the character length, parity and stop bits that define the serial link operation. Read Serial Status (RSS) is a function that returns the contents of the serial status register and reports on reception errors, input line status, and internal transmitter and receiver state. Using these three 98224A statements, the 98036A may be more easily interrogated and controlled according to the needs of the application.

Remote Keyboard

One of the special abilities of the 98036A Serial Interface is simultaneous bidirectional operation by operating the transmitter with programmed I/O (write and write binary) while operating the receiver under interrupt. This capability prompted the inclusion of the Remote Keyboard (RKBD) statement in the 98224A. A device such as a CRT terminal may be selected as a remote keyboard having all the capabilities of the local keyboard except reset. The function of the local keyboard is retained, but it may be disabled through a dummy on key routine as previously discussed.

In addition, the remote keyboard may be activated upon power-up of the 9825A by removing the appropriate jumpers on the 98036A. The remote keyboard facility enables control of the 9825A over long distances, since a data communications link may be interposed between the 9825A and the remote keyboard device.

Programming Aids

The final group of statements provided by the Systems Programming ROM do not relate directly to the 98036A or I/O. They are aids for programs used in systems where the system configuration may not be known.

The most powerful of these statements is STORE. STORE works the same as the store key on the 9825A, placing a specified line into the currently resident program. The line specified may be either a literal or a string variable. The line may be stored at any currently existing line or at the end of the program, creating a new line number. Only lines conforming to HPL syntax may be stored.

This feature is provided to enable programs to be stored on general ASCII devices, using LIST#. The programs may be retrieved with I/O statements line by line and stored into program memory to recreate the program. The short program at the end of this article does this. Note that error trapping is used for lines that do not meet syntaxing rules. Such lines have a "%" placed at their head. The "%" is a universal syntaxing symbol provided by the 98224A making anything following simply a character list. Such a line is not executable, but the program loading is not halted. "%" is useful for comment lines as well, since anything may follow a "%".

Although STORE was intended as a device for loading programs, it is finding use in programs that generate other programs. For instance, a program can be created that understands how to program an entire line of voltmeters. If the program asks the proper questions starting with "What voltmeter do you have?" to querying as to function, range, etc., a subprogram can be generated by creating strings that are HPL syntaxable lines, which are then stored into program memory using STORE. This control subprogram can be recorded on tape or disk for later incorporation into another program, say an automated testing program. By creating program-writing programs for all programmable instruments in a system, a large test sequence can be created by an operator untrained in the operation of the 9825A or the instrumentation.

The three remaining functions in the 98224A Systems Programming ROM are also used in writing program-generation programs. The Next Available Line (NAL) function returns the value of the last line number in the currently resident program plus one. To prevent interference with the running program, this is generally the best area to build a subprogarm. The remaining amount of free read/write memory may be determined using the Available Memory (AVM) function. This function provides an approximate measure of the remaining free memory. An exact reading cannot be made because memory usage is dynamic. Finally, the current line number may be obtained from the CLN function. Uses for CLN include program tracing via live keyboard, absolute jumps (JMP) and absolute computed GOTO's.

Who Needs It

The 98224A Systems Programming ROM has many applications in the field of data communications. However, with the unique capabilities of controlling 9825A software and internal hardware, users with applications involving nonprogrammers as operators or host machines of unknown memory capacity may have use for systems programming statements. END

```
0: % "Program Loader or PTAPE"
1: ent "Input Select Code = ?",S;ent "Record on Track# ?",T;ent "File# ?",F
2: dim A$[85];avm→A;nal→N;% "save available memory and next available line"
3: "input":red S,A$;if len(A$)<=2;gto +0
4: if A$[1,1]="*";gto "out";% "asterisk is end of listing"
5: if num(A$[1])=0;gto "input";% "ignore null lines"
6: on err "err";store A$,nal
7: if avm<250;beep;dsp "INSUFFICIENT MEMORY";stp
8: gto "input"
9: "out":A-avm→A;trk T;if F>=0;gto "rec"
10: for F=0 to 9999;% "search for null (last) file
11: fdf F;idf F,Y,C,Q;if Q;next F
12: mrk 1,A+500,Z;if Z<0;beep;prt "Not enough tape",A,"bytes needed";stp
13: "rec":rcf F,1N;prt "PROGRAM ON FILE#",F;stp
14: "err":"%"&A$→A$
15: if not (pos(A$,";")→X);gto 6;% "search for internal ;'s"
16: "%"→A$[X,X];gto -1;% "replace internal ;'s with %'s"
*7417
```

Programming Tips

Duplicating Tape Cartridges (9815A)

Submitted by F. William Schueler, Analytical Engineering, Rollway Bearing Company, Syracuse, New York 13201, U.S.A.

Here is a program to duplicate the contents of a tape cartridge either for use at another location or as a safety measure to insure against accidental loss of programs due to tape damage.

The program runs on the 9815A Opt. 001 (2008 steps) and has the following limitations:

1. Only cartridges containing file types 0, 2, 5 or 6 can be duplicated, and
2. Files containing more than 1816 program steps or 227 data registers will not be duplicated. A 2000-step empty file will be marked and this will be noted on the printout. An empty file will be marked as such and also noted on the printout.

The operation of the program is as follows: After entry "END" and "RUN" are keyed. The printout calls for "MIN.FILE#". This is the algebraically lowest file number, i.e. $:-3<-0<0<1$. This number is entered and "RUN" is keyed. The printout calls for "1st In". The cartridge to be duplicated is placed in the tape drive and "RUN" keyed. The printout calls for "2nd In". The cartridge on which the duplicate recording is to be made is placed in the tape drive and "RUN" is keyed. Continue alternating the two cartridges until duplication is complete and "END" is printed out.

Steps 116 through 121 govern the size of file in which each program is recorded. In order to allow for changes that may be made in the program, the file will be marked at least 150 steps longer than the program.

The 1816-step limitation on the length of the program to be duplicated is obviously caused by the necessity of retaining the duplicating program in memory. In order to handle as many steps as possible, I have tried to reduce the ALPHA to a minimum.

```
0000 0
0001 #REGS
0002 2
0003 EEX
0004 3
0005 STO      J
0006 1
0007 9
0008 2
0009 STO      I
0010 PRNTα
0012 M
0013 I
0014 N
0015
0016 F
0017 I
0018 L
0019 E
0020
0021 #
0022 ENDα
0023 STOP
0024 PRINT
0025 STO      B
0026 IF -
0027 SFG      1
0028 PRNTα
0030 M
0031 A
0032 X
0033
0034 F
0035 I
0036 L
0037 E
0038
0039 #
0040 ENDα
0041 STOP
0042 PRINT
0043 STO      C
0044 IF -
0045 SFG      2
0046 IF CFG   1
0047 GOTO     0136
0049 0
0050 IF SFG   2
0051 RCL      C
0052 +⇌-
0053 STO      A
0054 RCL      B
0055 +⇌-
0056 STO      F
0057 FOR      A→F
0058 PRNTα
0060 1
0061 S
0062 T
0063
0064 I
0065 N
0066 ENDα
0067 STOP
0068 GOSUB    0181
0070 IDENT
0071 ROLL↓
0072 STO      D
0073 ROLL↓
0074 STO      E
0075 1
0076 8
0077 1
0078 6
0079 IF X<Y
0080 SFG      3
0081 ROLL↓
0082 ROLL↓
0083 2
0084 IF X=Y
0085 SFG      4
0086 CLX
0087 5
0088 IF X=Y
0089 SFG      5
0090 IF CFG   4
0091 GOTO     0098
0093 RCL      E
0094 8
0095 ÷
0096 #REGS
0097 STO      G
0098 RCL      I
0099 IF SFG   4
0100 0
0101 GOSUB    0181
0103 IF CFG   5
0104 LOAD
0105 PRNTα
0107 2
0108 N
0109 D
0110
0111 I
0112 N
0113 ENDα
0114 STOP
0115 RCL      E
0116 2
0117 0
0118 9
0119 +
0120 2
0121 ROUND
0122 IF SFG   3
0123 RCL      J
0124 IF SFG   5
0125 RCL      D
0126 1
0127 GOSUB    0181
0129 MARK
0130 IF SFG   3
0131 SFG      5
0132 IF SFG   5
0133 GOTO     0165
0135 RCL      G
0136 IF SFG   4
0137 0
0138 IF CFG   4
0139 RCL      I
0140 GOSUB    0181
0142 IF SFG   4
0143 RCDATA
0144 IF CFG   4
0145 RCPRGM
0146 CFG      3
0147 CFG      4
0148 CFG      5
0149 NEXT     A
0150 IF SFG   2
0151 GOTO     0185
0153 IF CFG   1
0154 GOTO     0165
0156 RCL      A
0157 IF SFG   1
0158 0
0159 STO      A
0160 RCL      C
0161 STO      F
0162 CFG      1
0163 GOTO     0057
0165 GOSUB    0181
0167 PRNTα
0169 #
0170
0171 PRINT
0172
0173 E
0174 M
0175 P
0176 T
0177 Y
0178 ENDα
0179 GOTO     0165
0181 RCL      A
0182 IF SFG   1
0183 +⇌-
0184 RETURN
0185 PRNTα
0187 E
0188 N
0189 D
0190 ENDα
0191 END
```

READ DATA Error Recovery (9845A/S)

Submitted by Bonnie Dykes, Hewlett-Packard Calculator Products Division

Sometimes, because of extreme use of a mass storage media or adverse environmental conditions, the contents of a particular file may become "lost." That is, the desktop computer cannot successfully read a particular area on a tape cartridge, disc, floppy, etc.

This READ DATA error recovery program recovers most of a "lost" file by trapping the READ DATA error and bypassing the unreadable portion of the storage media.

The recovered data or program is recorded onto a storage medium specified by the user. The program is usable only on data files that have been SAVE'd rather than STORE'd.

```
10   ! THIS PROGRAM WILL HELP A USER RECOVER FROM A READ DATA ERROR IN A !
20   ! PROGRAM FILE WHICH WAS SAVED AS DATA !
30   COM A$(50)
40   DIM B$[160],E(10),File$[6],Mediumfrom$[10],Mediumto$[10]
50   PRINTER IS 16
60   PRINT PAGE,"READ DATA ERROR RECOVERY PROGRAM",LIN(1)
70   PRINT "NOTE:  The recovered program will be stored in a file called 'Recvr
' on the",LIN(1),TAB(8),"specified storage medium.",LIN(1)
80   PRINT TAB(8),"SOME LINES OF CODE WILL BE MISSING SO CHECK YOUR RECOVERED P
ROGRAM"
90   PRINT TAB(8),"CAREFULLY AND REPLACE ALL MISSING LINES.",LIN(1)
100  INPUT "ENTER NAME OF FILE WITH READ DATA ERROR",File$
110  INPUT "ENTER SPECIFIER FOR STORAGF MEDIUM CONTAINING FILE WITH READ DATA E
RROR",Mediumfrom$
120  INPUT "HOW MANY RECORDS DOES THE FILE WITH THE READ ERROR CONTAIN?",Recor
ds
130  INPUT "ENTER SPECIFIER FOR STORAGE MEDIUM ON WHICH TO STORE THE RECOVERED
PROGRAM",Mediumto$
140  ASSIGN #1 TO File$&":"&Mediumfrom$
150  CREATE "Recvr"&":"&Mediumto$,Records
160  ASSIGN #2 TO "Recvr"&":"&Mediumto$
170  ON ERROR GOTO 230
180  E=1
190  ON END #1 GOTO 270
200  FOR N=1 TO 9999999999
210  READ #1,N;B$
220  GOTO 260
230  E(E)=N
240  PRINT N
250  E=E+1
260  NEXT N
270  READ #1,1
280  READ #2,1
290  E=1
300  ON ERROR GOTO Nextr
310  ON END #1 GOTO Out
320  READ #1;B$
330  PRINT #2;B$
340  PRINT B$
350  GOTO 320
360 Out: PRINT "RECOVERY COMPLETE"
370  BEEP
380  PAUSE
390  END
400 Nextr:   READ #1,E(E)+1
410  E=E+1
420  GOTO 320
```

WRITE and FORMAT Incorporating Variable Length Strings (9830A/B)

Submitted by G. Fletcher, Plessey Telecommunications Ltd., Beeston, Nottingham NG 9 1LA, England

It is often desirable to produce output with embedded variable length strings but maintaining tabulation of output columns. By extending each string to the maximum length as dimensioned, the subsequent variables remain tabulated.

Lines 30 - 50 in the example below will maintain string length and hence tabulation of results.

```
10 DIM A$[32]
20 READ A$,A,B,C
30 FOR I=LEN(A$)+1 TO 32
40 A$[I,I]=" "
50 NEXT I
60 WRITE (15,80)A,A$,B,C
70 STOP
80 FORMAT F3.0,3X,2F12.0
90 DATA "BRISTOL-TAUNTON H & G",1,400,12
100 END
```

Update

9845 Clinical Laboratory Library Vol. 1, 09845-14250

Contains programs for radioimmunoassay data reduction, blood gas analysis, and clin lab quality control statistics. RIA transformation models include logit log, spline approximation, and four-parameter logistic. Quality control programs allow storage of 100 data points for 240 chemistries with basic and correlation statistics, Levey-Jennings plots, scattergrams, and Youden plots. Requires a 9845S. The 9883A Tape Reader, 2631A or 9871A Printer, 9872A Plotter and mass memories are optional.

HP-IB Programming Hints for Selected Instruments, 59300-90005

This book offers suggestions and instructions on assembling HP-IB systems using the 9825A. The following instruments are covered:

3455A	Digital Voltmeter
3437A	Systems DVM
3438A	Digital Multimeter
3495A	Scanner
5328A	Universal Counter
5345A	Electronic Counter
5340A	Microwave Counter
59309A	Digital Clock
59307A	VHF Switch
59306A	Relay Actuator
3571A	Spectrum Analyzer
59501A	D/A Power Supply Programmer
59308A	Timing Generator
3330B	Frequency Synthesizer
59304A	Numeric Display
9871A	Printer
9830A	Desktop Computer
59403A	Common Carrier Interface

98224A Systems Programming ROM

The 98224A Systems Programming ROM enhances the I/O capabilities of the 9825A Desktop Computer and is particularly useful for:

- Asynchronous terminal emulation,
- Program input from ASCII-coded devices,
- Control of 98036A Serial Interface,
- Remote keyboard operation of the 9825A,
- Keyboard redefinition,
- Annotated HPL programs,
- Efficient memory utilization,
- High-speed data logging.

The ROM occupies the same memory space used by the 98211A Matrix ROM. Therefore, the two ROMs cannot be used in the 9825A at the same time.

Keyboard
Hewlett-Packard, 3400 E. Harmony Road
Fort Collins, Colorado U.S.A. 80521

Keyboard

Publications Manager - Ed Bride
Editor - Nancy Sorensen
Graphics - Hal Andersen
Typography - Paula Dennee

For further information on HP products or applications, please contact your local Hewlett-Packard Sales and Service Office or write to Keyboard.

CHANGE OF ADDRESS: To change your address or delete your name from our mailing list please send us your old address. Send changes to Hewlett-Packard Keyboard, 3400 E. Harmony Road, Fort Collins, Colorado 80521, U.S.A.

July 1, 1978