TECHNICAL DEVELOPMENTS AND THE EDP AUDITOR - ANOTHER ROUND OF TRAUMA?

Karl Reed
Lecturer, Department of Computing Royal Melbourne Institute of Technology

The Author

Karl Reed has an Associate Diploma in Communications Engineering from RMIT and has completed the MSc Prelim program in Computer Science. He is a Member of the ACS, the ACM and the I of E (Aust.) and a Graduate of the IREE (Aust.). After twelve years in industry, Mr Reed spent 2 years at Monash University as a Tutor, and is now in the Department of Computing at RMIT. He is also the Consulting Editor to Australasian Computer World, and National Chairman of the ACS Software Industry Committee.

Summary

Technological developments in electronics and computing have created new avenues for systems and hardware designed to raise productivity and increase performance of hardware and software. These will create additional problems for the EDP Auditor, and could even be considered to be another round of trauma.

The electronic office, small business systems and specialised hardware are to stay.

This paper addresses some of their Audit implications.
1. **INTRODUCTION**

1.1 **The Future has Arrived**

The rapid development of semi-conductor technology in the last ten years has created a need to completely re-think conventional approaches to many aspects of our daily lives. Computers and related devices have moved from their airconditioned "central sites" into homes, factories, petrol pumps and even domestic cassette recorders.

Predictions of very fast, very small computers made in the early 1970's have come to pass, and the future has become very rapidly the present with the advent of the so-called 16 bit microprocessors, the Motorola M68000, the Zilog Z8000 and the INTEL 8086.

We have already passed the era of a computer with the power of a UNIVAC 1108 in a cabinet no bigger than a domestic refrigerator, and are about to enter the domain of the 1000 Megabyte disc-drive.

At the other end of the scale, we can purchase a 64 kilo machine with 2 Megabytes of "floppy discs", a terminal and a letter quality printer for around $A9000.

1.2 **The Microelectronics Revolution**

The basic driving force behind developments in this area is the advances made in microelectronics.*

(See Noyce et al 1977 for an excellent presentation on the subject).

---

The rate of change in "chip density", the number of components per semi-conductor chip, has been little more than phenomenal.

A British Engineer, G.W.A. Dummer predicted in 1952 that "with the advent of the transistor and the work in semi-conductors generally, it seems now possible to envisage electronic equipment in a solid with no connecting wires. The block may consist of layers of insulating, conducting, rectifying and amplifying materials, the electrical functions being connected directly by cutting out areas of the various layers".

This prophesy, rediscovered by Wesley A. Clark in 1980, (Clark 1980, p333) lead to the first monolithic circuits in 1958.

An indication of the current state of the art is given by a paper by Terajima (1980) to IEE Congress 80 in Melbourne 1980 who said that 1 Kilobit chips were available in commercial quantities in 1971, and 64 Kilobit chips were released in 1980, as shown in Fig. 1.

![Figure 1 Developments in MOS RAM Chips (Terajima, 1980)](image-url)

He also predicted a 4Mbit chip by 1990 (ibid p129). This would be achieved by increasing chip-sizes from 5 mm square to 10 mm square, rather than by reducing the size of gate geometry.

...2...
Terajima also set the limits on disc capacity and predicted single spindle capacities of 1000 Megabytes in the immediate future. In fact, devices in this category have already been announced. However, he rightly believed that the main benefit to the customer would be in substantially reduced costs for lower capacity discs.

Recording densities of 75000 bits per inch, and 1700 tracks per inch are achievable with thin film recording heads. However, practical figures are 25000 bits per inch and 1250 tracks per inch. (See figure 2), according to his paper.

![Figure 2 Developments in Disc File Recording Densities. (Terajima 1980)](image)

These yield 4 Giga bytes on a standard disc.

Developments of this kind depend largely upon semiconductor developments which have lowered the cost of sophisticated electronics, permitting very high density recording techniques.

* Giga, \(10^9\)
The most important effect of the semi-conductor developments is to permit very complex functions to be realised using a few low-cost components. A major consequence of this is to raise, by many orders of magnitude the amount of computing power available for a given cost.

Hewlett-Packard, for example, aim to produce a 30% improvement in the relative cost of its products each year and they are by no means the most aggressive company in this regard. This has been achieved by a compounded rate of increase in the performance available per unit cost over a 30 year period.

1.3 EDP System Developments

Our real concern with technology is the effect upon society, and for practical reasons, the society that we will consider will be the electronic data processing community with specific emphasis on the EDP Auditor.

We will argue in this paper that there are three developments occurring which will alter the role of the EDP Auditor. The first of these to be considered is the Electronic Office, evidence that computer technology is finally being applied to real information processing.

Secondly, we briefly consider the use of low-cost microprocessor based systems in small businesses.

Finally, we examine special-purpose computer hardware made possible by developments in electronics with special emphasis on highly secure systems.

It will become apparent from this paper that the new technology may actually increase a system's accessibility, especially at the hardware level, and that many of the examples of fraudulent interference quoted depend upon this.

This must be considered a fourth factor.

1.4 The Auditors Problem

We will, in the process, see that there will be a substantial increase in the EDP Auditor's need for knowledge of, and access to computer science expertise. The paper will suggest that some of the plug compatible and "back-end" systems are potentially "trojan horses", bringing problems as well as offering performance gains and solutions.

However, we should briefly outline the hardware technology
The developments mentioned will force the EDP Auditor to change his approach. They will provide new opportunities for fraud, sabotage, espionage, incompetence and accident, and as a result demand considerable insecurity in the practice of his craft.

In fact, the problems could prove to be little short of traumatic.

2. THE ELECTRONIC OFFICE

2.1 Systems to Eliminate Paper Transaction

It can be argued that computers have so far contributed very little to office automation.

The majority of computer applications, no matter how sophisticated, concentrate on automating some relatively large, discrete function in the hope of a cost-benefit through labour and other savings. Classic applications such as share-registries, stock-control, pay-roll, personnel files etc. exist as separate entities and seldom have common access interfaces.

They are an ancillary to office work and do little to automate basic office routine. (Ellis and Nutt 1980 p27).

"Basic office routine" consists of a series of information exchanges, both formal and informal. The formal aspects of information exchange are not difficult to identify; and they consist primarily of such things as incoming letters, orders and telephone requests together with inter-office memo's, reports etc. In other words, we have what is basically a movement of paper, and we can easily eliminate it.

The EDP Auditor, may also may be concerned with the effectiveness of a system, and should be aware that a very considerable amount of information is exchanged verbally and informally. Two recent papers (Driscoll 1979 and Ellis and Nutt 1980 pp53-54) address this topic, and the comments, while obvious, are to be recommended.

Ellis and Nutt regard this problem as unsolved in the sense that "the trend toward automation works against the goal of maintaining a social structure" (ibid p54). So it is possible that the electronic office concept might not work.

We are more concerned with the automation of paper-flow.
It has been estimated that a saving of two hours per day would yield an annual cost benefit of $62 billion per annum in the U.S., or around $2 billion in Australia. (Driscoll 1979 p106).

An office automation system could consist of two components, both of which might be largely independent.

a) Electronic Mail and Automatic Archive

We presume that all transactions in the Automated Office are "paperless", being prepared with a word processing system. Every office worker will have his own terminal most likely a video terminal, and any item created by any means can be "filed" automatically in the archive, provided that all terminals are connected to the archive.

However, we are concerned also with office communications, most of which are now "written".

Normally, "written" communications are communicated in an office via a mailing system.

We now have the possibility of creating a memo on the word processor, and having it "mailed" electronically to its destination, as well as "filed" in the archive.

![Diagram](image)

Fig. 3 Automatic Message Flow.

b) User Interfaces for Forms and Information Handling

The word-processor itself is part of the user interface, that mechanism by which a human communicates with the system. However, we also require interfaces for information handling that are far more "user friendly" than is common. To do this,
we will call on graphics and associated
technologies which will rapidly become cost-
effective because of the advances in semi-
conductors described earlier.

User interfaces may require fairly substantial
amounts of local intelligence. An example is the
"office talk-zero" system described in Ellis and
Nutt (1980, pp29-33) and implemented by Xerox's
office Research Group at Palo Alto.

Fig. 4 shows the "starting screen" or "desk-top"
associated with that system. We will return to
it later.

Figure 4 Office Desk Top (Ellis & Nutt, 1980)
2.2 A Model Electronic Office System

Each user (ibid P30) has an out-basket and an in-basket.

A memo is created with a destination list, some file reference, and, when moved to the users out-basket is automatically moved to the in-basket of those on the destination list. It is also automatically filed, with appropriate* security, in the automatic archive.

Filing, by the way, should include "file by originator by contents by date" as well as according to what could be called specific files; those named on the document.

"Filing by contents" allows questions such as "I wrote a letter on the use of files in office systems - where the hell is it"? to be answered, and requires conventional information retrievable technology. (See for example Salton 1968 Chapter I, and De Jong 1980 pp469-474). This is necessary if I wish to retrieve the letter having forgotten that it was specifically filed under "EDP AUDIT".

*We go beyond office-talk here
Fig. 5 Information Flow for Electronic Mail

TO: The Boss
FROM: A Slave
SUBJECT: I Need a Raise

File: - Personal correspondence

Dear Boss,

Signed
A. Slave

Fig. 6 Sample MEMO
Table 111  Hardware Concepts Stimulated by Microelectronics

<table>
<thead>
<tr>
<th>Concept</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal Distributed Computer Architecture</td>
<td>Internal design which lowers costs and allows use of VLSI and Custom Gate Arrays. Also raises computer power.</td>
</tr>
<tr>
<td>Colour Graphics</td>
<td>Both low and high definition at various costs.</td>
</tr>
<tr>
<td>&quot;Split&quot; Screens</td>
<td>Terminals allowing different files to be displayed simultaneously.</td>
</tr>
<tr>
<td>Local Networks</td>
<td>High-speed communications networks used within a single building or site.</td>
</tr>
<tr>
<td>&quot;Back-end&quot; Systems</td>
<td>Specially designed computers for handling Data Base activity, for example.</td>
</tr>
<tr>
<td>Data Entry</td>
<td>Techniques for coding data to ensure secrecy.</td>
</tr>
</tbody>
</table>
The "in-basket" functions can be quite elaborate. It is possible to have a prioritized queue so that letters may appear in the in-basket in order of the senders' view of their urgency.

It would be possible to provide an "alarm-on-arrival" of a message that you might be waiting for, or "alarm-on-arrival" of an urgent message. All kinds of possibilities present themselves, however, we do require a general warning that there is "un-sighted" information in the in-basket, and would like to be able to view and print summaries of its contents.

Lastly, I would mention the possibility of advising the sender when a message has been sighted.

These functions are summarised in Table IV.

Table IV Electronic Office Functions

<table>
<thead>
<tr>
<th>Component</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Out-basket</td>
<td>Store incomplete memos log of outgoing messages</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>In-basket</td>
<td>Alarm-on-arrival</td>
</tr>
<tr>
<td></td>
<td>Advise &quot;sighting&quot; to sender. Summarise memo</td>
</tr>
<tr>
<td></td>
<td>Maintain Priority List</td>
</tr>
<tr>
<td>Archive</td>
<td>File by sender by date</td>
</tr>
<tr>
<td></td>
<td>by receiver</td>
</tr>
<tr>
<td></td>
<td>File by specified topic</td>
</tr>
<tr>
<td></td>
<td>File by content</td>
</tr>
</tbody>
</table>
We now have a moderately sophisticated system.

Imagine that we apply our office automation system so that:

a). ALL communications coming into the office are entered on it before distribution;

- and -

b) There are no written communications within the office - all print-out is on paper which biodegrades in 24 hours!

We have set the scene for the EDP Auditor, or rather, some of his problems.

3. AUDIT PROBLEMS IN AN AUTOMATED OFFICE

3.1 Fraud Espionage and Sabotage

A system such as the one described offers interesting possibilities for Fraud, Espionage and Sabotage, possibilities which have not been addressed by the literature to any great extent.

We note in passing that we have a moderately complex software system - which someone certainly wrote and someone probably maintains.

3.2 A Case Study

Fred is a salesman with a computer company which has a commission-based renumeration package. Sales are credited to the person who makes the first contact, and commissions are very high. A significant percentage of contacts are via telephone enquiries. Naturally, the switch board has terminals, and enquiries are "routed" via the office system. Fred also maintains a prospect file which (an option on the system) contains "refer again" items that remind him when action is required.

Fred notices a slight but rather important reduction in his telephone contact rate. He is assured that all incoming queries are being correctly forwarded, so he ignores the problem and concentrates on his personal contacts.

However, he begins to notice that his "refer again" are occasionally missed-by him, but checks on his files show that his records are correct, so he
assumes he some how missed the "refer again" messages. (See Table V).

Later, he finds that one important prospect cannot be retrieved from his prospect file, so he calls in the EDP Auditor. The EDP Auditor has now to decide whether-or-not:-

a) There is a software or hardware error;

b) Unauthorised variations have been made to the software (or hardware);

c) Unauthorised access has occurred to Fred's files.

None of this is in itself new, except that we are dealing with a highly volatile real-time system which will be based upon sophisticated operating system concepts. It may also have special purpose hardware (Ellis and Nutt p55) and might have a multiple-cpu realisation.

For example, the system described above, the "refer-again" facility has been implement in such away that a "next event" list is constructed when an item is added to any file. Every item has or has not identifiable "refer again" (R.A.) fields, and the modules which store items create a list for the target of the R.A. which contains the date and time, the item's identification and a short message.

The "next event" list is a "binary-heap" (Knuth 1973, pp145-150) (see fig. 37), and, the auditor finds that the module concerned has been tampered with, and routes some of Fred's R.A.'s to Jean! Likewise with his telephone queries.

Jean, it turns out, is a "gun" system's programmer, turned, well, let's just say turned.

However, an EDP Auditor had been associated with the design of the system, and had almost foreseen the problem. An "R.A." log was kept which showed all items placed on the R.A. list. Jean was pretty clever and had only modified the dispatcher, leaving the audit software in the dark.
### Table V  Extended Application Electronic Office

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic Routing of Queries</td>
<td>All queries treated as &quot;memos&quot; and routed automatically</td>
</tr>
<tr>
<td>Refer Again</td>
<td>Message referred back to someone at a specified future time, or repeatedly</td>
</tr>
</tbody>
</table>

![Diagram](Image)

**Fig. 7** Next-event List for Refer Again
Fred, however, was in very serious trouble, he had been carefully chosen because he was a bit forgetful, so it was months before he actually complained about the information retrieved system in misclassifying some of his material. . . . . this, after lengthy investigation, does not prove to be sabotage.

We have been painting a hypothetical picture which is familiar to the EDP Auditor, although the complexity is rather different, and we are no longer concerned with quantifiable items, such as stock items etc.

We have not yet finished with the peculiar problems which might occur with the Automated Office system.

An interesting example relates to the "check-on receipt of message" (COROM) facility which tells the sender when the receiver actually viewed the message. This is part of a more sophisticated "check-on response to message" (CORTM) which advises the sender as the recipient carries out each activity on a list. (This is part of the Business Office Supervisory System or BOSS for Short).

<table>
<thead>
<tr>
<th>Component</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>COROM</td>
<td>Check on receipt of message - advises sender of recipients &quot;sighting&quot; of messages.</td>
</tr>
<tr>
<td>CORTM</td>
<td>Check on response to message - advise sender of action taken to carry out nominated tasks.</td>
</tr>
</tbody>
</table>
Jenny, who is an area supervisor, is having trouble with John. John will not accept direction from Jenny, so she uses COROM and CORTM to monitor his work. These systems show that John has begun to respond and is acting promptly on all directions and requests, except that the material results are not apparent at the end of a month.

Again, the software has been tampered with.

Experienced EDP Auditors will be aware of the danger of that problem, and some of the difficulties are alluded to in Stanley (1978) p1725 but there seems to be no simple answer.

We have, however, seen some of the facilities which will be available, and their potential for misuse.

4. LOCAL NETWORKS

Automated office systems will necessarily be distributed, and this raises the possibility of the extensive use of "Local Networks".

"A" Local Network" (see Kryskow and Miller 1981 for example) is one covering relatively short distances, say .1 km to 10 km operating at relatively high transmission speeds, say .1Mbits/sec to 10Mbits/sec.

They are characterised by inexpensive transmission media and the existence of a "base" communications protocol which allows messages to be sent between any two or more modes. Local Networks arose from the need to connect small numbers of CPU's together within one industrial complex or even one building.

Another characteristic is the absence of a central mode - although this is not essential.

Coaxial cable, "twisted pair" (or telephone) cable and optical fibre and all used as transmission media. Our Automated Office, by the way, may require quite high band widths, and access to more than one main CPU.

Another extremely important requirement is that it should be possible to add modes cheaper and without disturbing the rest of the network. Further, the network has no controlling intelligence, and does not of itself know which modes are present.

Figure 8 shows the topology of an Ethernet, and Figure 9 the method of connecting.
Fig. 8 Ethernet Configurations (Xerox 1980 Fig. 7).
Fig. 9 Ethernet Architecture and Typical Implementation. (Xerox 1980 Fig. 4.1).
Fig. 10  Ethernet Basic Message Format
(Xerox 1980 Fig. 6-1)
Fig 2 CATV topology. CATV systems are becoming popular for local data distribution because of excess "free" bandwidth where they also carry video and voice signals.

Fig 3 Star topology. Historically, most common data network topology, not decentralized.

Fig 4 Ring topology. Failure at single node can bring network down unless some form of bypass is provided.

Fig 5 Bus topology. Ability to insert spurs facilitates network reconfiguration.

Figure 11 Common Local Network Topologies (Kryskow & Miller 1981)
5. **SMALL BUSINESS SYSTEM**

Auditing small business systems will present problems for a variety of reasons. These hinge upon the fact that the capital cost of the systems and software is extremely low, in the range of $5000 to $15000 dollars. An extremely wide variety of software is available for microprocessor based systems. There are, for the Z80 based systems, a wide variety of operating systems based upon CP/M, a small scale, single user operating system. A multi-user version of this is also available, and called M/PM.

All major programming languages are available as are the commonly required utilities such as SORTS, Merges etc.

Simple data base systems can be obtained (see Whinston and Holsapple 1980 for an example) in addition to an extensive collection of very low cost applications software. Figure 12 contains an incomplete list of software available for Z80 based systems.

Substantial volumes of software are also available for the 6502 based systems (e.g. the Apple series) and for the 8080 based systems, including manufacturers operating systems and compilers.

This picture will be further complicated by the so-called 16 bit microprocessors such as the Zilog Z8000, the Motorola M6800 and INTEL 8086 (Refer Zilog 1979, Motorola 1979 and Intel 1980) which will have more substantial operating systems.

It is even possible that software will be written by the owner of the system, and that the question of program security and integrity will be irrelevant. For example, the Auditor, who may need to examine the floppy discs which hold the accounts, need not be aware that the software recognises two sets of "books", one for the auditor and the tax inspector, and the real set. These could be on separate a floppy, complete with a copy of the operating system.

The accessibility of small business systems hardware and software, and its low cost will make it difficult to justify a detailed examination.

It would, for example, be a simple matter to write an assembly language program which modified floppy disc directories so that files holding incrimination evidence might "temporarily" disappear from the directory!

The use of read-only-memory may prevent an EDP Auditor from modifying a program to "trap" all incoming transactions. Phillip Stanley's concept of sealing the system
as it now is.

Table I summarises the dominant microelectronics developments. Table II lists some "single chip" functions which are either available or are projected, while Table III lists the major hardware concepts made realizable by low cost electronics.

Table II  Major Microelectronics Developments

<table>
<thead>
<tr>
<th>Development</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very large scale integration (VLSI)</td>
<td>Techniques yielding single chips with 60,000 to 100,000 devices</td>
</tr>
<tr>
<td>Custom Gate Arrays</td>
<td>Very low cost methods of building special purpose components. Around 800 devices per chip.</td>
</tr>
</tbody>
</table>

Table III  Some "Single-chip" Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Floppy Disc Controller</td>
<td>Now</td>
<td>Complete controller for dual floppy discs</td>
</tr>
<tr>
<td>Data Comms Controller</td>
<td>Future</td>
<td>Handles either HDLC or SDLC protocols</td>
</tr>
<tr>
<td>CRT Controller</td>
<td></td>
<td>Generates signals for control of video generators</td>
</tr>
<tr>
<td>Data Encryptor</td>
<td></td>
<td>Encrypts data to NBS standard</td>
</tr>
<tr>
<td>Dot Matrix Printer Control</td>
<td></td>
<td>Controls dot matrix printers, accepts characters and produces solenoid drive signals.</td>
</tr>
</tbody>
</table>
General Ledger
Accounts Receivable
Accounts Payable
Letter Writer
Sorts
Word Processors
Data Base
Merges
Operating Systems
Games
Fortran
Basic
Cobol
PL/I
Pascal
Text Editors
Linking Loaders
Screen Formatters
File Utilities
Plotting Packages
Graphics
Debuggers
Mail List Maintenance
Payroll
Inventory

Figure 12 - A List of Microprocessor Software (incomplete)
(Stanley 1978) is useless.

A possible solution is auditor - certified software.

5.1 Large Systems Analogues

Duplicate physical disc-packs or magnetic tapes, deliberately falsified disc-directories and special hardware will always be a problem, and are not restricted to small systems. However, successful execution of such a fraud would, in a larger installation, require a major conspiracy involving operators and systems programmers.

6. SPECIALISED HARDWARE

Fundamental audit issues arise with the use of specialised hardware. In particular, the exact internal functions may not be accessible, even for examination. In addition, there may be no mechanism for modifying the special hardware for audit monitoring purposes.
Table VII shows some currently available special hardware functions of direct interest.

Table VII - Add-On Hardware

<table>
<thead>
<tr>
<th>Special Hardware</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Ball-end Database&quot;</td>
<td>Hardware to Software which reduces the load associated with Database system.</td>
</tr>
<tr>
<td>&quot;Front-end Systems&quot;</td>
<td>Systems which act as terminal concentrators, or provide &quot;intelligence&quot; to terminals.</td>
</tr>
<tr>
<td>Associative Processors</td>
<td>Special Hardware for reducing cost of searching texts, or directories associatively.</td>
</tr>
<tr>
<td>Remote Processors</td>
<td>CPU's, systems which are physically remote, and which may not be a direct part of an audit.</td>
</tr>
<tr>
<td>&quot;Associated&quot; Processors</td>
<td>Devices attached to the CPU's memory bus for handling special instructions.</td>
</tr>
</tbody>
</table>
We should also consider user-modifiable hardware, the most common being the provision of user microcode facilities which are now relatively widely available. Table VIII also includes disc and type sub-systems as potentially user modifiable, as well as terminals.

Table VIII User Modifiable Hardware

<table>
<thead>
<tr>
<th>Type</th>
<th>Source</th>
<th>Effect</th>
<th>Motivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Microcode</td>
<td>Supplier Option</td>
<td>Allows user to use specially modifiable code</td>
<td>Cpu Performance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Special instructions for cpu savings</td>
<td></td>
</tr>
<tr>
<td>Magnetic Media</td>
<td>User Modifications</td>
<td>User may modify recording techniques or add</td>
<td>Security</td>
</tr>
<tr>
<td>CPU</td>
<td>&quot;Add-on&quot; or user modes</td>
<td>Improved performance using &quot;add-ons&quot; that are not available from suppliers</td>
<td>Improved performance</td>
</tr>
</tbody>
</table>
User-Microcode

It is not difficult to construct an extremely unpleasant scene in which an installation has undertaken extensive microcoding to achieve improved speed and reduce program volume, since hardware which enables the user to tailor his own instructions is now quite widely available, especially on large-scale minicomputers.

A plausible scenario would involve a large-scale mini-computer such as an Interdata 3200 or a Prime 750 which has been "tailored" to improve (say) online performance and database index maintenance. It is possible that this may effect the logging of transactions in the database software, with the result that an auditor may have extreme difficulty verifying the correct operation of the modified system.

For example, the software may not be readily transportable to another machine of the same type unless user microcode is available.

Modifications to "trap" transactions may prove difficult, depending upon the actual choice of function, since it can sometimes be difficult to initiate logical input/output from within a microcoded sequence. The Auditor may, if there existed a version of the software which did not use microcode, insist that the application be run using the original code.

This is, however, a futile gesture because the Auditor wishes to vet the software actually in use!

One can imagine an extreme case in which user microcode had been used to deliberately obscure the purpose of a program such as might possibly be done for reasons of national security or extreme perversity. The Auditor would then be faced with a copy of a virtually unreadable program plus the micro-code itself. For example, the microcode instructions on the PDP11/60 are 48 bits wide and have a four address format, including a "next instruction" address. There are a total of 100 registers visible to the microprogrammer, and we are promised (Dec 1979 p268) that the user Writeable Control Store (WCS), the Random Access Memory which contains user micro-code, is also available as a "local storage" for micro-coded instructions.

The PDP11 microcode by the way is relatively intelligible. Examples in which the micro-instructions control internal gates directly and have little relationship to conventional computer instructions so-called "horizontal" micro-code (See Varian 1972 for example
are common, even in "user-microcodeable" machines.

The approach which an auditor should take is not apparent to this author!

6.2 "Plug Compatible" CPU Modifications

An illustration of the relative ease which which a CPU, particularly in the modern large-scale mini, might be tampered with can be found from an examination of the "add-ons" available from Plug Compatible Manufacturers.

Examples (see Table IX), taken mainly from advertisements in the U.S. Publication "Computer Design", details "add-ons" which:

a) Interface to the memory address* and data buses
   - and -

b) Could be readily tampered with remitting interference with information flow.

Table IX CPU Add-Ons and their Vulnerability

<table>
<thead>
<tr>
<th>Add-On Device</th>
<th>Vulnerability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Boards</td>
<td>Diversion of information substitution of program</td>
</tr>
<tr>
<td>Caches</td>
<td>As above</td>
</tr>
</tbody>
</table>

*It should be acknowledged that the use of virtual memory may complicate such interference
The EDP Auditor's only problem would be to verify that the "add-on" hardware does not perform some additional undesirable function.

6.3 Disc and Tape Subsystems

One would not normally regard disc and tape subsystems as "user modifiable", but there is no doubt that they may be. Further, as mastery of the technical problems associated with modifying these items becomes easier, as it clearly is, then such modifications will either be made to improve performance or to increase security.

6.4 "Plug-Compatible" Disc Controllers

It should also be noted that there are already a wide variety of "Plug Compatible" disc-controllers, some of which have microprocessors (see advertisement for Burroughs 6Mbyte floppy drive, Computer Design April 1980 p237, also Western Peripherals advertisement in Computer Design December 1980 and Dilog's advertisements for special interfaces evaluating DEC devices) on the market, and it is only a matter of time before associative stores which speed-up disc accesses are available.

The EDP Auditor's problem in this case is to ensure that the controller functions correctly. Such a device is capable of maintaining multiple copies of files, unknown to a user. It is also capable of allowing only a user with a special "software" key to access the copy.

Forging disc records would also be possible.

6.5 Altering Recording Methods

A security conscious installation could also alter the recording method used on its disc drives. Such an unlikely action would make it impossible or difficult to move packs from one installation to another.

6.6 Encryption

Dr. Pearcy's paper discusses this in detail, but encryption essentially consists of encoding a message so that a knowledge of the encrypting algorithm and perhaps the key are necessary before the message can be read.
The concept of a "back-end" system is one that has developed to enable special functions to be carried out by special hardware and software.

Database back-end systems are reasonably well known and have been well documented.

Maryanski's paper of 1980 presents the details of several proposed systems. One of the first (ibid) was developed at the Bell labs. "Back-end" systems are predicated on a rather simple proposition which is that specialisation is more efficient than generalisation. The support software for basic file handling, and the instruction sets of most computers are designed for general applications. Any special purpose software must contend with this.

In addition, it can be difficult to ensure that resources are allocated to specific tasks, such as a data-base package on many systems. Adding more CPU capacity, more channels and more discs may lead to better overall service rather than just better data-base service.

Further, one might implement such a system on a recent "maxi-computer" such as a PDP11/70, VAX11/780, PRIME 750 etc. machines with excellent cost/price characteristics, and connect it to a main-frame computer such as a large IBM370.

There is also the possibilities of using special hardware, as we shall see.
The "Back-end" processor accepts commands of varying complexity from the "Most", and notifies the "Most" when the task is complete.

The basic concept is shown in figure 13.
Currently, Cullinane corporation and AGS have released "back-end" systems, and a company called "Data Fusion" offers an Associative File Processor which has a separate CPU, and is available as an attachment to PDP11's. Another, Britton Lee, Inc. offers a special relational data-base system which can be interfaced to mainframes.

There are also a very wide range of sophisticated "back-end" data-base processors based upon experimental hardware. Examples are a very advanced project using the concepts of data flow computer architecture (see for example Comte, Hildi and Syre, and Curd and Watson 1980 parts 1 and 2), and parallel computers described recently by Tanaka et al (1980) and another based upon multimicroprocessors and bubble memories (Uemura et al 1980).

These systems are shown in Figures 14 and 15 respectively.

Figure 14 Tanaka's dataflow database machine
Figure 15 Vemuro's Magnetic-Bubble Data-Base Machine
One should also mention the ICL Content Addressable File Store (Maller, 1980 and Maller, 1979), a special device which reads all disc-surfaces in parallel.

The EDP Auditor's problems, in these cases, expand to include more conventional questions of system "auditability; to what extent was the system designed to be auditable?

This author is unable to answer the question, but it is a simple matter to identify the problems.

Problems with "Backend Systems"

a) Modifiability

Access to the programs are likely to be impossible, and modification for audit purposes a matter for the original designers.

b) Independent Verification of Disc Contents

One might wish to verify that a "backends" discs contain valid data that has not been tampered with. This may require special action since the files may be organised in such a way that they may not be accessible on some other machine. This will be the case if the back-end machine runs a special operating system.

By the way, this assumes that the disc drives themselves are not some obscure incompatible type!

Of course, the disc pack can always be read if there is a "programmable" machine with a suitable drive attached. The contents can be read directly and then analysed.

However, this is not totally satisfactory, except as a last resort.

c) Deliberate Interference

Such systems are likely to be harder to interfere with than conventional systems for exactly the reasons that they will be difficult to modify. There is a strong likelihood that the programs may be in Read Only Memory, for example, and complete documentation may not be available.
8. ASSOCIATED PROCESSORS

Another area of concern is that of "Associated" Processor designed to perform special functions. These are available on a wide variety of computers. Comments can be transmitted either via I/O channels or memory buses, and interrupts used to signify task completion.

So-called "bus" architecture machines, such as the PDP11 can have such devices attached with relative ease, giving the device access to the system memory.

The major difference between "back-end" and "associated" processors is that the former perform large tasks and the latter performs small tasks which might be instructions on other machines.

Examples include floating point and special business instructions.

"Back-end" data-base systems may be connected in the same manner as "associated" processors as may be hardware for processing or data communications.

This approach is about to become standard practice with the next generation of large-scale microprocessors.

The Intel 8086 has a number of these devices, two of which are the 8089 Input-Output processor (INTEL 1979 Chapter 3) and the 8087 Numeric Data Processor (Palmer 1980)

An example of using the 8089 is shown in Figure 16.
Figure 16 Intel 8089 Connected to an 8086 (INTEL 1980).
These devices provide a degree of modularity, allowing the user to pay for only those features that he requires, and allowing functions to be performed in hardware which would take much longer in software.

Again, these "associated processors" produce the "trojan horse" effect, providing an "authorised, unaudited listener", and are a function which may not be readily audited.

The associated processors are not limited to special functions, and it should be noted that multiple CPU systems, are now relatively common, being available almost all major computer suppliers.

Auditors need to be aware that the security of such systems depends upon hardware and software in a complex relationship, and that it is not impossible to gain control of part of the system, given sufficient knowledge. The associated CPU could then obtain access to data and programs without the knowledge of the remainder.

A detailed examination of this subject is beyond the scope of this paper.

9. CONCLUSIONS

It will be clear from this paper that EDP systems are potentially at risk from malevolent systems programmers and computer systems designers. The paper has detailed some of the possibilities arising from the introduction of a number of new concepts, all of which depend upon new technology, however, an underlying proposition is that the hardware is becoming more and more accessable, and that this accessability reduces system security.

Alternatively, the new accessability through micro-code and other techniques may make the system unauditable.

It is fairly clear that the EDP Auditor is going to be placed in an extraordinary difficult position in which he may need to be the compleat hardware and software engineer to ensure that a system is secure.

It has been suggested that the availability of "passive media" local networks such as ethernet will also reduce system security, permitting unauthorised access in a manner that may be difficult to detect.

The author is, unfortunately, able to offer much encouragement, although it would seem that there will be employment opportunities for computer scientists in audit terms.
It is encouraging to see that the computer science related areas of the EDP audit problems are being addressed by other speakers since this shows a general recognition of the problems. However, this is the right time and place to suggest that the EDP Audit fraternity might fund graduate research at colleges and universities aimed at examining some of these questions.

One should always conclude on a positive note, and the EDP Audit profession has an excellent track record when it comes to meeting complex challenges.

A new challenge has arisen, the profession is aware of it and is in a better position to join with other sections of the data processing field to solve them.
References


Pearcy, T.  "Encryption" EDPAC 1981


Zilog  "Z8000 CPU Instruction Set" Zilos Inc. 03-8020-01 1979.