COMPUTER INTEGRATED MONITORING

VOL 2

RANJANA LALITH VASANTHA MEEGODA

Doctor of Philosophy

THE UNIVERSITY OF ASTON IN BIRMINGHAM

May 1987

List of figures, tables, charts, plates and procedures

Chapter 7

Data transfer communications for integration

7.1 Introduction

Chapter 5 described all the operations which have to be done prior to monitoring while chapter 6 explained about the actual monitoring aspect assuming that all the pre created data and information is already available on the shop floor. These are two distinct systems which run on completely different computers. These computers are totally incompatible and therefore the users will have to get the output data from one system and feed them in to the other manually. This obviously is not a practical solution and this chapter looks at some available methods of data transfer such as Manufacturing Automation Protocol and discusses a much cheaper alternative technique.

7.2 Integration

Integration should be the ultimate objective of any computer or manual system because running 2 or 3 separate

systems result in poor utilization of funds, time, labour and equipment. The word integration is in fact a keyword today in the engineering industry with many journals running special columns and many authors producing document after document. They all say that integration should be achieved and also mention the aspects which should be integrated but fail to explain how.

Hartland-Swann[153] says that a strategic development plan should exist in integration without simply investing in another slab of high technology. She also says[154] that integration is fine if not carried out at the expense of simplicity and resilience. Most of the commercial protocols lack these while the users are not supplied all the information about their internal architecture so as not to give their trade secrets away. It is therefore worthwhile to look in to these to see whether they actually help the buyer to a considerable extent or not.

7.3 Communication

Communication between computers is the key to integration in most systems. This is rather difficult because most

computers are incompatible with other makes. Most people's idea of a solution is to match up the different levels of computers which exist in a network. Brooks[155] says that it is possible to carefully match up a micro and a local network in the same way that a seamstress matches a needle and thread, but it would equally be possible that the micro would blunder like many of us would, in the embroidery world in selecting a needle too small to be threaded or a needle too large that although it might work with a thread, it produces a very unsatisfactory result. Looney[156] says that every vendor stores data on his system in a unique way meaning that it can not be passed directly to a different vendor's system. He goes on to say that the solution is to write a translator to transform data from one format to the other which requires that the vendors concerned provide comprehensive information on the way they store data. Obviously the vendors would not want to provide such information which might possibly find it's way to their rivals or if not produce new competitors. This leaves many companies who want automation to impose limitations on their work so as to make them compatible with the existing computers. Avery[157] is such a company where the latest weighing machines are manufactured to be compatible with Apricot computers.

Having made the above statements, one would argue that communication between different computers have been available for a considerable time, the typical examples being the teletext services like Ceefax, Oracle or 4-Tel and Perstel, Buletin boards or Micronet.

Both types of services provide two way communication between the user and the host enabling the user to identify himself and allowing to select the required information by keying the appropriate commands or page numbers on his computer regardless of the data handling capability of it.

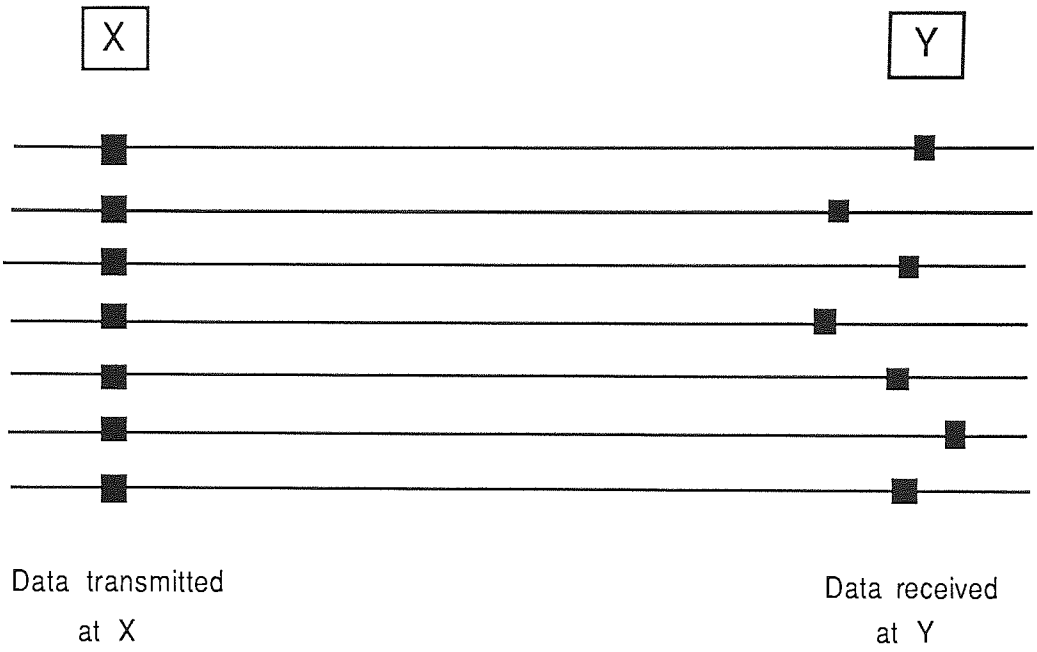
7.3.1 Principle of teletext services

Teletext is the name of the character set which is provided with the SAA5050 video chip that is made by Mullard for the use in television sets. It has the capability of creating a television picture and has a 4.3k ROM containing 96 alphanumeric characters, 64 graphics characters and 32 control codes. This SAA5050 chip can be fitted in a computer and the teletext is received in the same manner as the normal television signals. Therefore a television license and an antenna are needed to use the facility. The software

which is provided with the teletext receiver[159] allows the user to select any page at a time by using the computer's keyboard. These are displayed on the monitor. The teletext adaptor to a computer therefore is a special television set without a screen.

7.3.2 Principle of remote database access

The BBC computer uses eight individual wires (collectively known as a data bus) to move information around 8 bits at a time. This is known as parallel transfer. This data bus has to be connected physically (unlike in the case of receiving teletext) to the VAX computer with which the communication has to be performed. Unfortunately individual bits of data travel at different speeds over long distances and this communication will result (figure 7.1) in a data skew. This is the reason why short data cables are used with parallel printers or disk drives.



Data skew

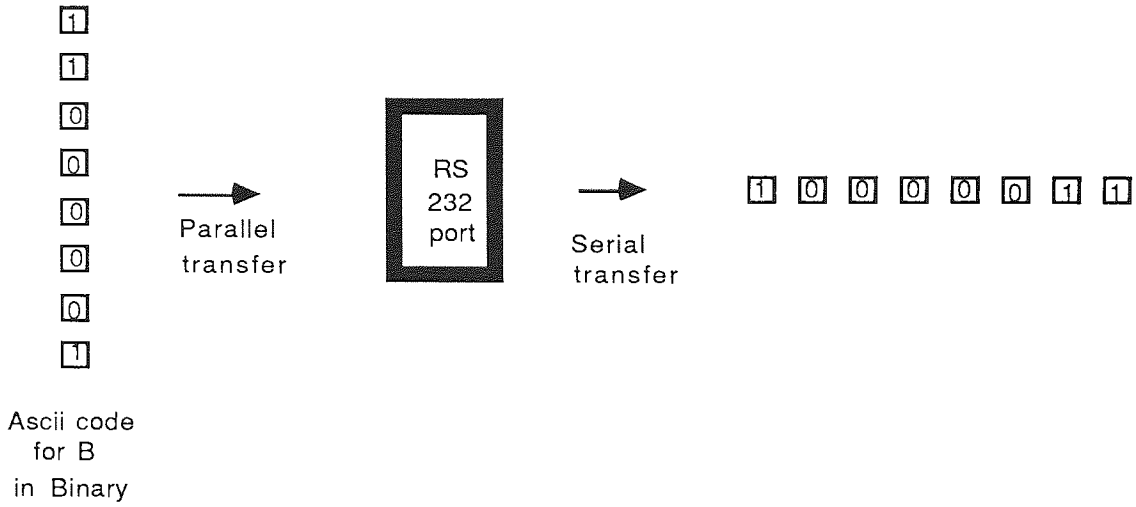
Figure 7.1

Even if this skewing effect is ignored, it will be hardly practical to have 8 wires running from one computer all the way to another. The other problem is that the VAX computer or the server is a 32 bit computer. It will be impossible to connect the 8 lines from the BBC computer or the requester to a 32 bit server or vice versa. Therefore an intermediate standard has to be used in order to transfer the data. The standard RS-232 protocol (Appendix 12) comes in at this point. The RS-232 transfers the parallel data in to a serial format (figure 7.2) and it is used in a wide range of applications today.

However, it is still not possible to use the telephone cables to transmit the data although in serial format. The reason is that the telephone cables can not transmit digital pulses (figure 7.2). Hence a device called a MODEM[158] has to be used. MODEM is the abbreviation for MOdulator/DEModulator which performs the function of modulating the digital pulses to analogue format (figure 7.3) so as to suit the transmission via telephone cables. The MODEM can perform the inverse function too, and the one at the other end demodulates the signal back to the digital format. RS-232 at the receiving end then transforms the data in to a parallel format (8 bit, 16 bit, 32 bit or any configuration) to suit the receiving computer.

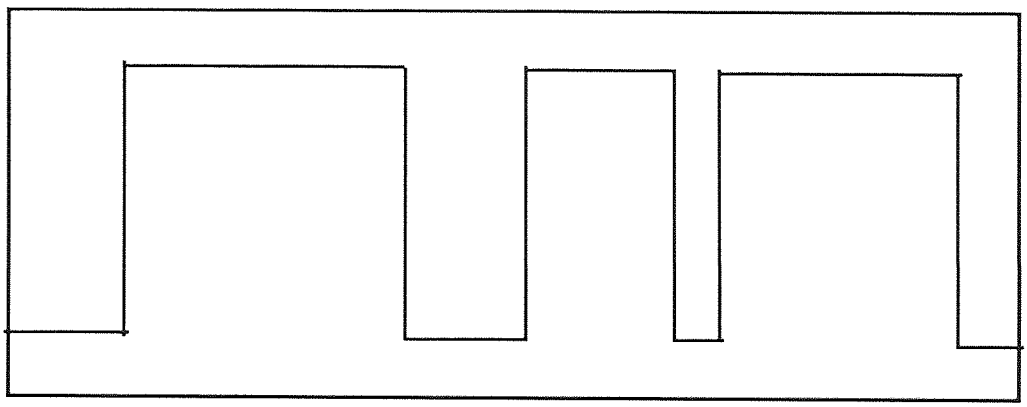
7.4 Terminal emulation

Both above systems provide a facility for the user to access the data bases in a remote system. The data can not only be seen on the monitor but also can be saved in the user's selected filing system. These methods can sometimes allow the users to write to and delete files in the host system as long as they have the sufficient privileges. In fact they allow the users to use their computers as 'dumb' terminals to see or save information on a host.



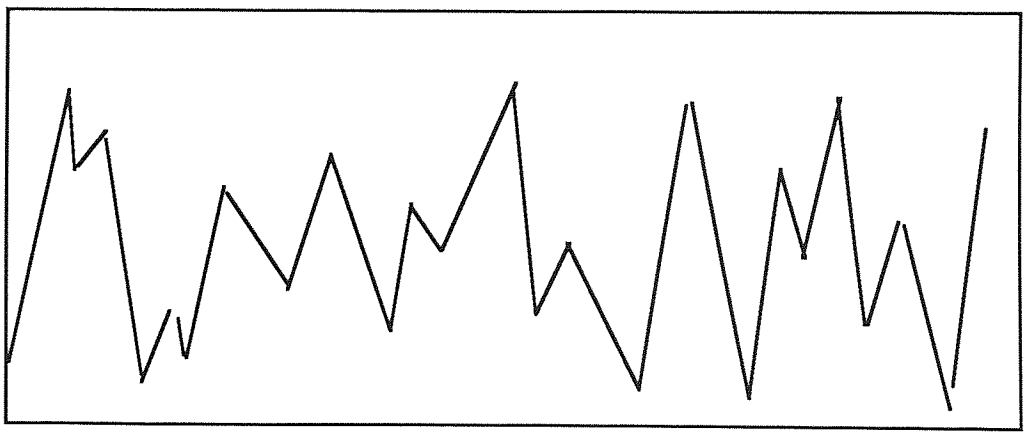
Parallel to serial transfer

Figure 7.2



Digital signal

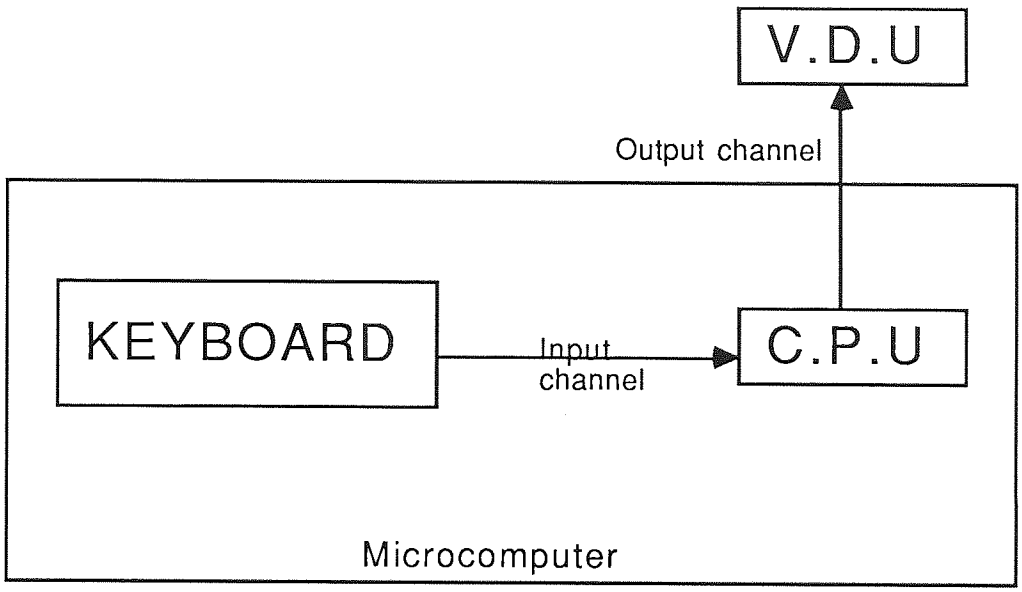
Figure 7.3



Modulated signal

Figure 7.4

This principle of terminal emulation of a microcomputer is quite simple. Under normal circumstances, the input and output channels of a micro computer are the keyboard and the visual display unit (V.D.U) respectively (figure 7.5).

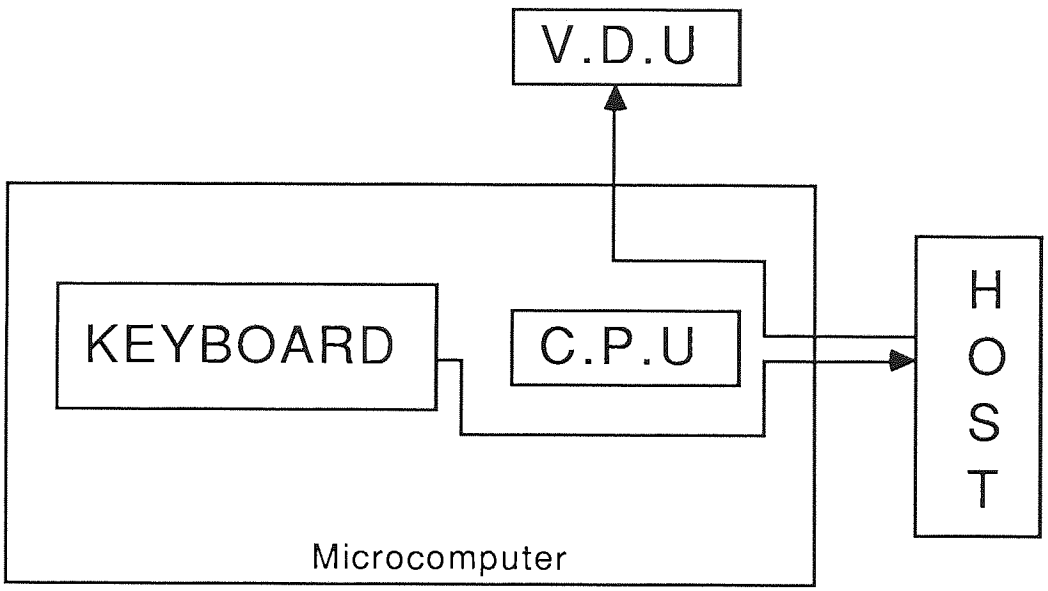


Input output channels of a microcomputer

Figure 7.5

The usage of the computer as a dumb terminal can easily be achieved by using a piece of cheap software in a ROM chip called a 'Terminal emulator'. This simply opens the serial communication port of the micro and directs all the keyboard input towards the port's output buffer while directing the input stream from the outside world towards the V.D.U. The Central Processing Unit (C.P.U)

is not involved (figure 7.6) in this operation in any way. The task which is carried out here by the micro is very trivial, and as a result a trained operator has to be present all the time in order to answer the computer's prompts so as to 'help' it to complete it's task.



Redirection of streams

Figure 7.6

The biggest disadvantage of these systems is that they provide their services via application software which re-direct input and output streams to various buffers. This means that the user who

wants to use the acquired data from the data bases will have to read them back in to his own program[160] in a format that is suitable to the micro. This involves a lot of skill and also it is a very inefficient way of running a system. This causes a problem specially in the shop floor applications where communication with the management computers have to be maintained without interrupting the application program. Unfortunately many people who discuss about data communications do not even seem to understand this problem. They assume that terminal emulation and file transfer are the ultimate solutions and even do surveys[161] on systems which provide terminal emulation software. They do not seem to understand why something as expensive as a computer has to be used for the same task that a terminal itself can perform. The research which has been done in this area so far seems to be almost non existent although many people simply mention that micros and mainframes must be integrated[162,163] without mentioning how such integration can be performed. In contrast, a method to access the data bases and to transfer of data to the user program without the involvement of any personnel is much more efficient, but such a feat has not been achieved so far.

7.5 Manufacturing Automation Protocol

There are may communications standards (Appendix 13) available today and various vendors use these which makes it impossible for different computers to communicate. The General Motors of U.S.A. have started research on a new protocol called Manufacturing Automation Protocol (MAP) which is based on the ISO model (as explained in chapter 1) to replace all the different standards. This is the most acclaimed protocol today and therefore it must be studied to understand whether it can help to eliminate the communications problems or not.

All the details about MAP (Appendix 14), it's seven layers and RS-511 protocol[164] seem to be something to look forward to. What makes MAP more appealing is that it is a non proprietary system which will make it less biased towards any vendor's system. One could assume that it would have a great future because both vendors and users support it. However, most of it is still in the early stages and it is not known when a complete system will be available. The fact that MAP is still in it's development stages is clearly depicted from the specifications of the layers as discussed

in appendix 14. Some of the claimed facilities seem to be vague while some others overlap with similar facilities or services in the different parts of the model which is developed so far.

Despite it's incompleteness MAP certainly seems to be having a lot of support which was proved at the very first demonstration in July 1984 at the National Computer Conference in Las Vegas[165] where six participants were linked over the first version, MAP 1.0. The succeeding version, MAP 2.0 is now installed at the General Motors experimental factory in Michigan. MAP 2.1 was shown at the Autofact[166] show in November 1985 at Detroit where General Motors and Boeing together explained how MAP and TOP (Technical and Office Protocol) can change the future of the industry. They said that MAP and TOP will provide the essential building blocks to complete the automation structure and that the two can be integrated together quite easily because TOP is identical to MAP in the layers 2,3,4 and 5.

General Motors expect to announce the final specification (MAP 3.0) soon and they claim that all the versions are compatible with each other. The support for MAP has grown even more since

the first demonstration, and seven countries[167] have set up MAP secretariats (table 7.1) where more than 500 companies in these countries now support MAP. The support for MAP in the UK was clearly visible[168] at the CIMAP (Computer Integrated Manufacturing Automation Protocol) event which was held at the National Exhibition Centre in December 1986.

Country	Organization
Canada	Canadian standards Organization
Japan	IRDFA
U.S.A	Society of Manufacturing Engineers
France	CIGNEF
Germany	VDT
Holland	CIAD
Italy	ANIPALA

MAP secretariats

Table 7.1

7.5.1 MAP products

MAP is going to be expensive until it becomes readily available and as accepted as RS-232. Motorola[169] recently announced their 68824 chip which costs £100. It contains the bottom two layers (Physical and Data link). This chip is a MAP token ring controller and it is also compatible with Intel's Backplate standard. Motorola also expect to launch the 372 double card with all 7 layers for £2500 and a starter pack which contains equipment for a 4 node 4 layer MAP network for £10,000.

LDR systems of Aldershot is another company which has manufactured a product which they claim to have a number of building blocks which provide services and protocols according to the ISO model.[170] This product is known as ISONET and it is in fact a communications software package where the software is written in either C or Pascal which will have no bearing on the user applications. The company also claim that they will implement MAP for any one who is interested.

7.5.2 Disadvantages of MAP

The biggest disadvantage of MAP is that it is still in the development stage which means that the permanent versions of MAP such as MAP 3.0 will undoubtedly need new software[171] although GM claim otherwise. One of the reasons for this is that the sublayer MMS on the top layer of MAP 2.1 is being replaced by MMFS in the version 3.0[172] because of the former's ambiguity. Even the 14 isolated islands of automation at CIMAP[173] made some potential buyers rather cautious because it was the very thing which MAP was supposed to eliminate. Proving this point even further, Dominico[174] says that the model is good conceptual framework but allows different implementations at each level which makes it quite common for two vendors' products to be fully OSI compatible but not able to communicate!

The layered structuring process of MAP is no different to any other communications protocol which have been developed in the past and it can be analogous to any of those.[175] All the layers of the Manufacturing Automation Protocol and can be compared with the conventional RS-232 protocol as follows.[176]

Layer 1 - 25 pin D type connector (including the specification of which the signals appear on each pin.)

Layer 2 - The voltage levels.

Layer 3 - Specification of the character encoding in the asynchronous format with parameters like receiving and transmitting speeds, number of data bits, number of stop bits and delay between characters.

Layer 4 - Specification of the meanings of any control lines (hanshaking) which are used.

Layer 5 - Local variables such as local echo and the effect of any line editing codes (line breaks or page breaks, implied or expressed carriage return and/or line feed)

Layer 6 - null

Layer 7 - The character set of the visual display unit.

7.5.2 Alternatives to MAP

Texas Instruments UNILINK[177] is the only direct rival to MAP at present and it's principle is almost similar to that of MAP. Unilink uses common hardware and common software held in easily exchangeable ROM cartridges which in fact are personality interface modules. Texas Instruments claim that a wide variety of equipment such as PLCs, Micros and operator interfaces can be linked to hosts using UNILINK through gateways called TIway-1.[178] These gateways can even be used to connect to MAP networks via MAP gateways.

7.5.4 Verdict on MAP

With all the publicity and the growing support one can forget to weigh the pros and cons of MAP and see whether it stands out in front as claimed. The exitement about the seven layers and the interface boards can also obscure the most vital ingredient in the system, the software. The MAP software is bound to contain enough information about all the standard systems in order to translate every detail to a universal (i.e., MAP) format first, and then to

re-translate it to suit the second system. This software will act like a Personality Interface Module (PIM). In effect everything narrows down to software ultimately. The next question is 'How standard will this software be ?', although undoubtedly it will be of the highest standard. One can wonder whether MAP will contain enough software to translate to and from all systems or will it be only those systems which the manufacturers think as 'standard'. This is quite an alarming prospect.

Looking at the details of MAP carefully, it appears that the remote memory access facility is the only revolutionary feature and this realization effectively shadows the lustre of the whole system to quite an extent. The other protocols like File transfer and the Virtual terminal are quite standard and can be used to communicate between any two systems although it may sound novel to the amateur.

Installing MAP will most certainly make things easier and eliminate the problem of writing expert software, but the purchasers will be paying a major part of their money towards redundant features like all the standardization software out of which

only one type will be used. Therefore a small or a medium sized manufacturing company would not gain anything by spending a lot of money in implementing MAP while in the point of view of a large company, the question remains whether it is better to install MAP or to employ an expert to use an alternative method for communication which will use tailor made methods to suit each individual's needs. A suitable cheaper alternative to MAP is certainly needed to carry out the data transfer as far as this research is concerned.

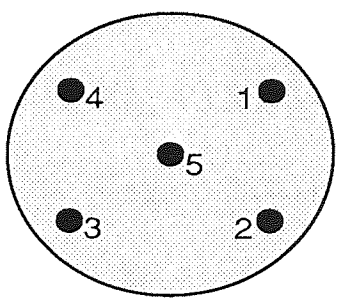
7.6 Hypothesis for data transfer

Reading from the host's data bases is a role which must essentially be performed and the idea of transferring these to the micro is simply unthinkable because of the limitations of the micro such as it's response time, memory size, speed of data transfer and the different formats in which the data is kept. We have therefore already established that the micro has to be used as an intelligent terminal so as to carry out the data transfer communication while running it's own program without any help from the operator. Both these tasks can not be performed simultaneously and therefore the hypothesis which can be used to achieve this would be to use the

micro as a terminal at certain times to send and receive data to and from the host while using it in it's own capacity as a computer at other times. The data that is captured while in the terminal mode must be processed in the computer mode. The switching between the two modes has to be done automatically using appropriate software which has to be used within the main program or programs which attend to monitoring.

7.7 Interfacing

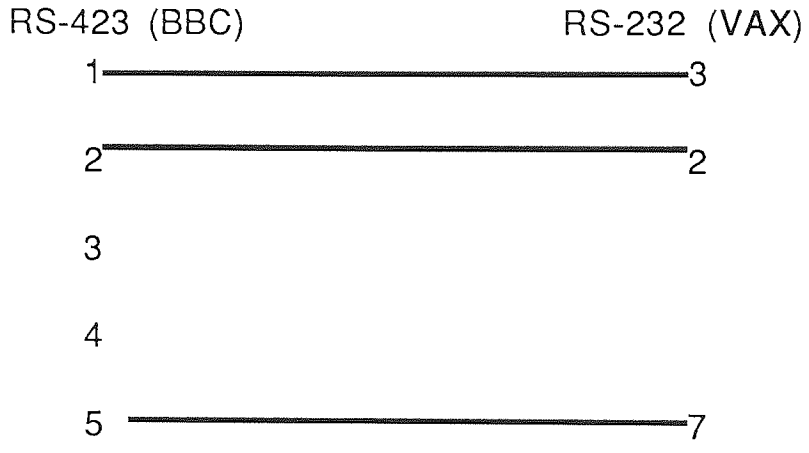
The VAX computer uses standard 25 pin D type RS-232 ports for the users to interact with the system. The BBC computer has a 5 pin RS-423 serial port (figure 7.7) and it can be connected to the VAX computer as shown in figure 7.8.



1 - Data in
2 - Data out
3 - CTS
4 - RTS
5 - SG

RS-423 serial port

Figure 7.7

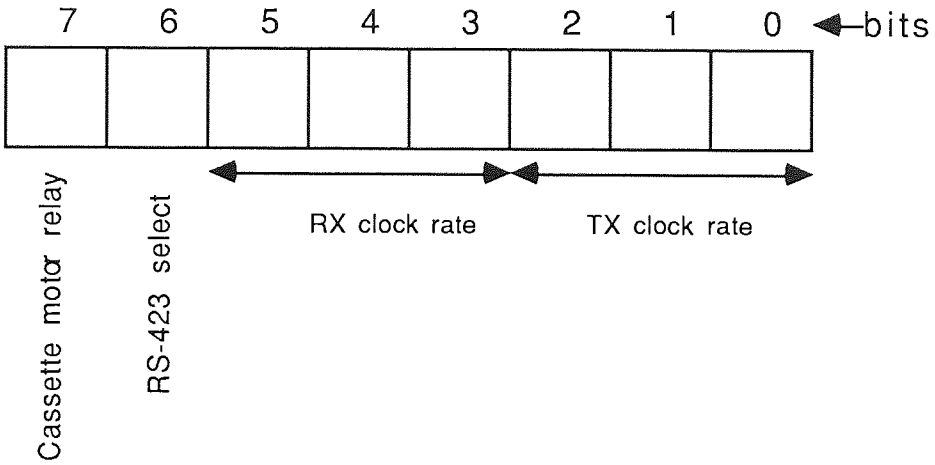


BBC to VAX connection

Figure 7.8

The transmission speed is an important parameter to set up and a reasonable speed is 1200 baud. Some other parameters which can be set up according to the user's requirements (if not satisfied with the default settings) are parity, duplicity and the delay between characters which can be set up quite easily by software. The system manager must set these on the VAX computer using software (SET TERMINAL TXA1/SPEED=1200/PERMANENT) while the ULA (Uncommitted Logic Array) device which is known as a serial processor[179] can be used to define these parameters on the BBC. This device is mapped in to the memory of the BBC computer at the memory location FE10 (in hexadecimal) and the status register (figure 7.9) can be used to access this. An easier way to initialise the

parameters is to use the operating system commands directly. The commands FX8,4 and FX7,4 set the transmission and receiving speed to 1200baud respectively and these commands can be used within any language by preceding them with an asterisk.



Status register

Figure 7.9

The next step is to log in to the VAX computer. Once logged in via the terminal mode of the BBC, the VAX computer considers the BBC computer as any other user. This has to be carried out manually during the development stage by using the BBC computer as a terminal and therefore some stream directing software has to be used. The 'RETURN' key must then be pressed continuously to log in until the host responds.[180] (If the connection

is established via a switching system like the X25 Packet Assembler Disassembler or Gandalf then the 'BREAK' key and the 'RETURN' key have to be pressed continuously. The 'BREAK' key on the BBC is not sufficient for this purpose because it is simply a soft reset key and does not send a pulse which is long enough for the switching system to recognize as a signal to log in. This problem can easily be eliminated by coupling a mechanical switch to the connection cable.)

7.8 Data transfer

7.8.1 Data receipt

Pressing the 'BREAK' key after logging in, would terminate the BBC from running the terminal emulation program. In other words it would now act in the usual way as a computer in it's own right. The important fact that should be remembered here is that the connection which is established will not be lost even after the BBC computer has finished acting as a terminal. The connection can now be used to communicate with the host when the serial port is used. The principle that must be used is the same as the Virtual terminal principle.

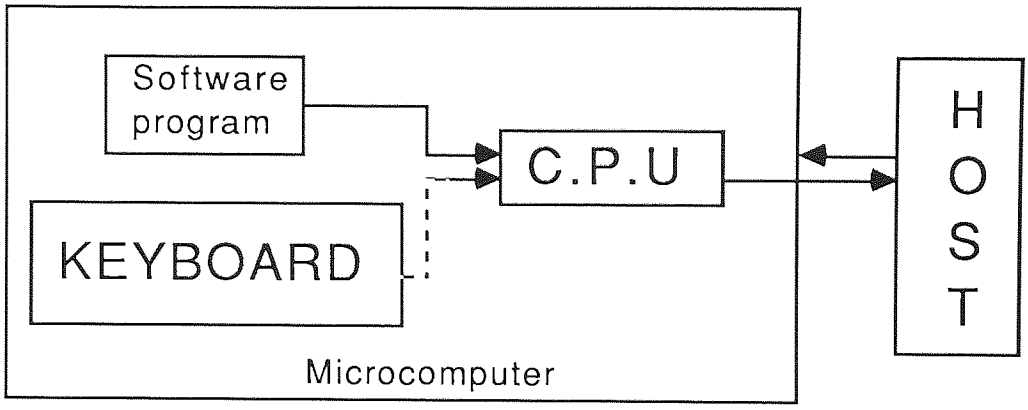
The very first task that should be done in an integration process such as this is to synchronize the time on both systems although battery backed RAM is used on the BBC system. This synchronization can be done by obtaining the time from the VAX computer and setting the time on the BBC.

The resident variable TIME in the BBC gives the number of centi seconds from the time the machine was switched on and this can be reset manually.[49] The VAX has a similar resident variable SECNDS in FORTRAN which gives the number of elapsed seconds from midnight[113] and therefore the variable TIME can be equated to SECNDS multiplied by a hundred.

A short program (named SETTIME) was written on the VAX in Fortran to read the value of the variable SECNDS, multiply it by a hundred, convert the resultant to a character string as opposed to data and then write it to a data file called TIMEFILE.DAT. It writes the current time in centi seconds in to the data file every time this program is run. The BBC computer can therefore run this program, read the data file and transfer it's contents to it's own variable TIME. This has to be done by the Virtual terminal principle as explained

above where the BBC computer is configured as the master while VAX is the slave. The VAX computer therefore assumes any command which is received from the BBC as a command from a user who is logged in to a remote terminal.[181]

The RS-423 port of the BBC is opened to run the above program in the host and the simple *FX3,7 is used to select the RS-423 as the current output channel and to direct the output stream to it[49,139]. The command that has to be sent to the host is 'RUN TIMEPROG' but this can not simply be given as a line of the program even after the communication port is opened. The reason for this is that any string from a program is treated slightly differently to that from the keyboard (figure 7.10) and in this case the C.P.U can not be bypassed. BBC's MOS operating system will not understand the command 'RUN SETTIME' and will send an error message. This will go to the RS-423 port as it is the current output device and also to the host because the RS-423 port is connected to the host. This message will not be understood by the host's VMS operating system and it will send it's own error message and the whole procedure would end up in disaster.



Software directed output

Figure 7.10

This problem was avoided by embedding the command string 'RUN SETTIME' in a PRINT statement of the Basic program which is resident in the BBC. BBC's MOS operating system does not look at the contents within this PRINT statement. This command was effectively executed in the VAX and the resulting data file was created. This data file can be read either by DCL or any other higher level language.

The data files must first be opened in the VMS operating system using a statement which is similar to 'OPEN/READ INFILE TIMEFILE.DAT'[182,183] and this too can easily be sent via a PRINT

statement as before. The next statement can be 'READ TIME' which must also be embedded in a PRINT statement. Finally PRINT "CLOSE INFILE" can be used to close the file. All these statements are true and will work except that the variable TIME will contain the current time in centi seconds in the VAX and not in the BBC computer for the simple reason that the PRINT statement in which the command exists is simply a string of characters for the micro although it is an executable statement for the host. Therefore this procedure is not useful for the intended task of getting the required variable to the memory of the BBC.

7.8.2 Data capture

It was not discussed as to what is to be done to the RS-423 input stream, although it was mentioned that the output stream has to be directed towards the RS-423 output buffer. The principle that should be used here is the fact that any response from a host is echoed back to the remote terminal (figure 7.10) from which any commands are received. The VAX in this case assumes that the BBC micro is in fact a remote terminal and sends it's response via the RS-232 line. This will come in to the BBC's RS-423

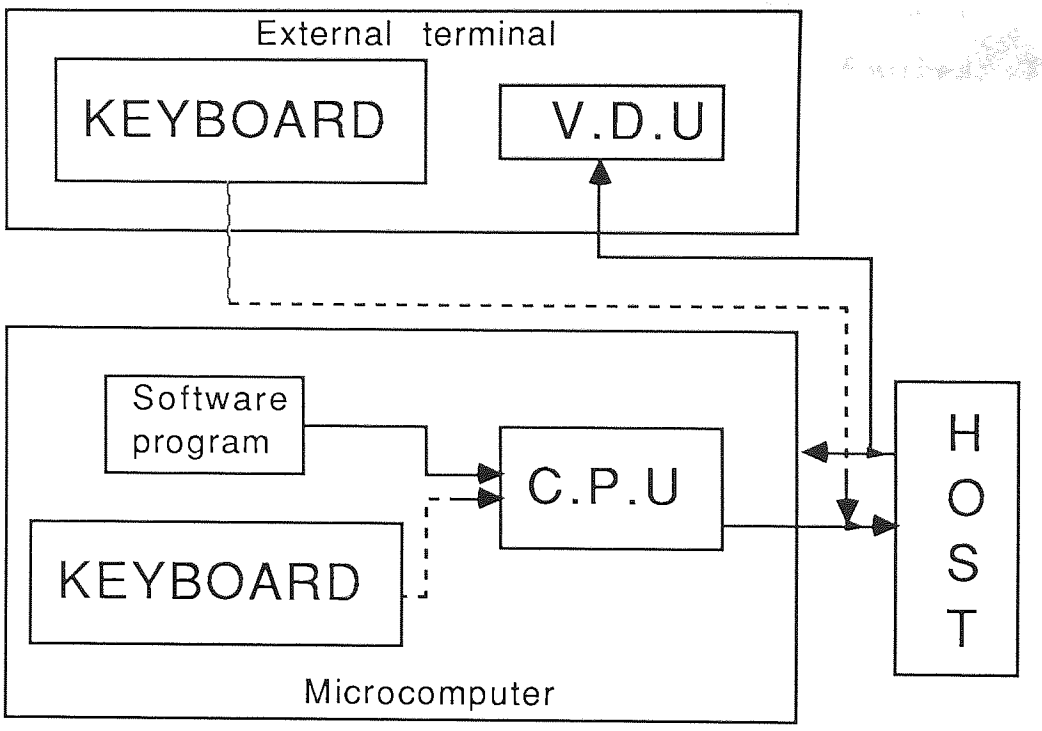
input buffer even if this buffer is not directed towards anywhere. The interactive user or the application software can either capture the contents of this buffer or direct it to any device.

The previously discussed 'READ' statement will just read the content of the file in to a variable of it's own called TIME but will not send it to the BBC because it was not commanded to do so. Alternatively a 'TYPE' statement[114] sends the contents to the BBC's RS-423 input buffer which is the host's equivalent to sending the response to the visual display unit of a remote terminal. This was tested by connecting a 'Cyfer' terminal in parallel (figure 7.11) to the BBC, and all the response from the host was echoed in the V.D.U of the terminal.

The statement in Basic to get a character from the current input channel is 'A=GET', where the character 'A' represents the ASCII code of the first character in the buffer. This statement was used to capture the contents of the buffer and the codes were converted to give the corresponding characters by A$=STR$(A). Then the complete string was evaluated by converting each ASCII code to it's corresponding character and adding them using 'B$=B$+A$'.

Unfortunately the ASCII codes of the carriage return, line feed and other characters that are not necessary were also in the buffer and were added to this string. This was eliminated easily because we need only the numerical characters and the decimal point of which the ASCII codes are between 46 to 57. The character addition statement was therefore altered to read 'IF A>45 AND A<58 THEN B$=B$+A$'. This is the reason why it is not advisable to capture characters from the buffer directly by the usage of 'A$=GET$' instead of 'A=GET'.

This principle was successfully used in transferring data from the VAX to the BBC. The process of obtaining the data was further simplified by getting the Fortran module 'SETTIME' to write the resultant to the output stream with the channel number 6 rather than writing it in to a data file which has to be 'typed' to the output stream later.



External parallel display

Figure 7.11

7.8.3 Termination characters

The process of looking in to the buffer to capture characters must be halted once the characters which form the required variable or variables are obtained. In other words, the communication with the host has to be terminated at this stage in order to transfer control to another part of the program and carry on with the shop floor tasks. The VMS prompt (the '$' sign in this case) was used as a termination character to achieve this because the VAX

computer displays it's prompt once any task is finished. The complete procedure of obtaining characters and forming the string was inserted in a 'REPEAT UNTIL' loop (procedure 7.1) while it is also possible to use an 'IF THEN' loop (procedure 7.2). The RS-423 is disabled after the termination character is received and the keyboard is once again selected as the current input stream while the V.D.U is selected as the output stream. Finally the variable B$ was evaluated using the powerful BASIC function VAL in the form TIME=VAL(B$).

It is interesting to note at this point that the personality interfacing has already been carried out by transferring data from the VAX to characters which have ASCII codes so that they are read by the BBC computer which in turn translates the character string to it's own data format. This is done easily because both systems use the ASCII standard but the conversion can be done easily even if the devices use different standards such as ASCII and EBCDIC, by adding a file which contains all characters and the equivalent ASCII and EBCDIC codes to the data base. An application software module which can be executed from the process computer can do the code transfer before each code is sent down the line. The code transfer can also be

done within the process computer after receiving the strings but it not a very good practice because of the limited memory of the process computers.

```
31000 A=0 : A$="" : B$=""
31010 REPEAT
31020 A=GET
31030 A$=STR$(A)
31040 IF A>45 AND A<58 THEN B$=B$+A$
31040 UNTIL A$="$"
31050 (rest of the program)
```

REPEAT-UNTIL loop

Procedure 7.1

```
31000 A=0 : A$="" : B$=""
31010 A=GET
32020 A$=GET$
32030 IFA$="$" THEN 31060
32040 IF A>45 AND A<58 THEN B$=B$+A$
32050 GOTO 31010
32060 (rest of the program)
```

IF-THEN loop

Procedure 7.2

The BBC computer must have the working parameters such as the estimated machine setting times, strip times, rolling targets, quantities to be produced, number of press operations per

length and scale factors before the actual shop floor monitoring or control begins. These quantities are prepared at the time the job is created and are written to files (under the job numbers) in a special format (figure 7.12) as explained in section 5.10.

```
1!
220!
1.5!
200!
1.3!
2!
4&          1.5¬        1000£       200!
3&          2¬          1500£       500!
4!
220!
33401!
1!
2108!
0!
1912!
0!
0!
0!
0!
0!
0&          8382!
23456!
*
```

Data file for the shop floor

Figure 7.12

Each direct variable is written in a single line in this file. The array variables which correspond to one particular operation are written in the same line but each is separated by a different character for the micro to differentiate between variables. A different character is defined to terminate each array variable. (The $ sign can not be used in this case because there is more than one variable.) The BBC computer had to be programmed in such a way so as to understand that a particular variable is finished when a termination character is received.

7.8.4 Transmission of data

The shop floor computer can successfully perform it's operation as soon as the above variables are obtained but it also must send the completed job details back to the host. Additionally the error messages also have to be sent to the host computer for the attention of the supervisors if the shop floor personnel ignore their messages. Furthermore, the management need to monitor the machine status constantly and this will have to be done by periodical messages which the shop floor computers can send to the host. The theory which was used here was to use the BBC computer

as an intelligent terminal and write the relevant information to files in the host which have to be opened every time a message is sent. The management too can open these files and obtain information via their terminals. Finally each and every occurrence during a job, the downtime categories, displayed error messages and operator change overs with the dates and times are also logged in a file (under the job number) which the management can list after a job is finished as explained in chapter 6.

Sending a message to the host is somewhat easier than receiving data as the data evaluation has to be done by the host and not by the shop floor computer. The command 'OPEN/APPEND OUTFIL LOG1' is used within a 'PRINT' statement to open a file called 'LOG1.DAT' to write all the occurrences during a job, where 1 is the job number and 'OUTFIL' is the logical name. This command always opens an existing file while the word 'APPEND' must be replaced with 'WRITE' to open a new file. The format to write the message 'Roll speed is too slow by 120m/hr at 1307hrs on 23.04.1995' in to the opened file in the VMS interactive mode is as follows, where the actual message has to be included within quotes.

$ WRITE OUTFIL "Roll speed is too slow by 120m/hr at 1307 hrs on 23.04.1995"

This message has to be written within a PRINT statement to send it without generating a MOS operating system error as before but the total number of quotes becomes 2 pairs since the message is already included within a pair of quotes as shown above. The PRINT statement has to go through the MOS operating system first and will be subjected to an error because it contains two pairs of quotes although the inner pair is meant only for the VMS operating system. The message therefore must be prepared first (procedure 7.3) with The ASCII code of 34 for the inner pair of quotes.

```
32010 Send$="WRITE/OUTFIL "+CHR$(34)+"Roll speed is too slow by
      120m/hr at 1307 hrs on 23.04.1995"+CHR$(34)
32020 PRINT Send$
32030 PRINT"CLOSE OUTFIL"
```

Previously prepared message

Procedure 7.3

The last statement (line 32030) is to close the opened file, and there is no need to wait for the VMS prompt as in the previous case once the file is closed. This procedure was used quite successfully.

The processes which were carried out to send the status of the job, finished job details and the suspended job details are similar and use file names 'STATUS.DAT', 'FIN.DAT' and 'SUSn.DAT' respectively where 'n' corresponds to the job number.

7.8.5 Alarm messages

Unlike the data, the alarm messages are useless if sent to files because they will then be read only when the management want to. These are required to be sent to the management terminals overriding any other task the managers may be doing at the time. The analogous task on the VAX computer is done by using the 'REPLY' function which is used in the two following formats.

```
$ REPLY /BELL/TERM=TXA1 "Over produced."
$ REPLY /BELL/USER=JONES "Setting target exceeded"
```

Either the terminal number or the VMS 'user name' has to be given so that the message is sent to the correct destination unless the parameter 'ALL' is used. The parameter 'BELL' is optional and corresponds to ASCII code 7 which sounds an audible 'beep' on the destination terminal. The actual message was included within quotes as before and therefore the whole command was prepared beforehand. One important consideration is that the sender should have the 'OPER' privilege to carry out such a high level procedure. Therefore the system manager should authorize this privilege and all other required privileges to all the line terminals. Each terminal should also request the required privileges when logging in.(180)

7.8.6 Remote database access and start characters

So far it has been discussed only about receiving data which has been prepared in a pre determined fashion but the 'operator log in' is different to this. It was discussed in section 5.12 that a general module which consists of all operator details was added to the main data base and therefore the clock number which the operator enters has to be compared with the clock numbers in the data base for identification. The initial thought would be to

download the complete data file in to the shop floor computer so that it can carry out the comparison. This is practically impossible because of the limited memory of it and an alternative solution would be to download record by record which can be destroyed until a match is found. This method does not occupy a large amount of memory but is very long and time consuming. It also is a very inefficient way of approaching the problem.

The technique, Inversion was applied here to make the task more efficient and the software was written in such a way that the clock number which is keyed in by the operator is sent to the VAX computer for comparison (instead of downloading VAX's records to the BBC) while the BBC waits for this task to finish. This was done by triggering a procedure in the VAX once the operator enters his clock number on the BBC computer. Such a procedure which is named 'OPNAME' was added to the VAX computer and is executed by the command 'RUN OPNAME'. This command is sent from the BBC computer after receiving the operator input and is written it in to a file (OPNAME.DAT) as before. The procedure OPNAME reads the clock number from this file and compares it with all the numbers in the operator data file until a match is found. If a

match is found then it outputs the operator name in to it's output stream as before which is collected by the BBC computer. It outputs an error message if a match is not found which is interpreted by the BBC so that the operator is requested to re-enter his clock number.

The BASIC command 'IF A>45 AND A<58 THEN B$=B$+A$' can not be used in this case because a name consists of characters which have different ASCII codes. The name will not be correct if all the characters from the VAX computer are collected and added. Therefore a start character is needed for the BBC computer to start collecting characters and the module OPNAME was designed to send the operator name preceded by the '*' character which has an ASCII code of 42 while the termination character is '!' which has an ASCII code of 33. The BBC computer collects all characters within these start and stop characters to form the name.

7.9 Handshaking

The transfer of data as explained above is theoretically possible but the different response times of the two computers present a major problem. It was found out that the response time of

the BBC computer is much faster than that of the VAX when performing this data communication, because the VAX computer continuously failed when receiving data at 1200 baud while the BBC computer carried out it's task successfully at the same speed. The BBC computer could receive data even at higher speeds such as 9600 baud while the VAX computer failed even at low speeds like 300 baud. This is quite understandable because the VMS operating system is a time sharing operating system whose disadvantages were described in appendix 2. This problem is quite a common one and therefore handshaking must be used to prevent the loss of data. There are two types of handshaking called hardware handshaking (Appendix 12) and software flow control of which the former is not supported by the VAX computer.

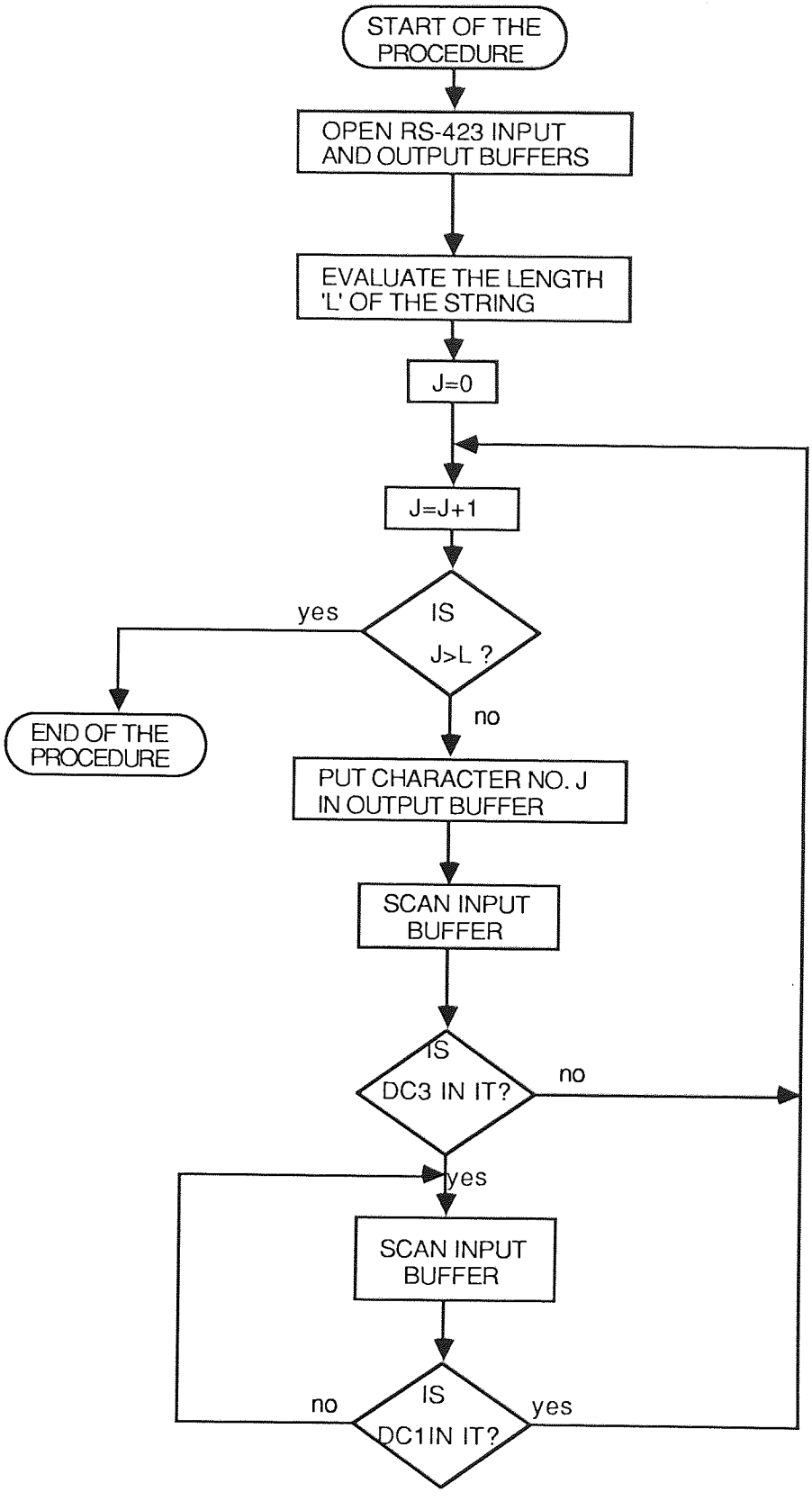
7.9.1 Software flow control

Software flow control which is also known as the Xon/Xoff protocol is simple but is not as fast as hardware handshaking. This type of handshaking can be used with only two wires (ie, data in and data out).

The basic principle is that the receiving device sends a code which is known as Xoff when it is unable to receive any more data. The transmitting device waits after receiving such a code until it receives the Xon code to continue. The Device Control (DC) codes DC1 (ASCII 17) and DC3 (ASCII 19) are used for Xon and Xoff respectively.

The VAX range of computers support this type of handshaking and send DC3 when the input buffer is $^3/_4$ full and continue to send it for each subsequent character it receives until the buffer is only $^1/_2$ full in which case it sends a DC1.(184)

Unfortunately the BBC computer does not support this type of handshaking and therefore ignores all the device control codes which are received from the VAX. This presented a major problem but was eventually eliminated by writing a routine (chart 7.1) to recognize the Xon/Xoff control. Therefore all the messages had to be sent by placing one character at a time in the output buffer of the BBC computer rather than sending a command or a data string within a 'PRINT' statement. The data communication became fast and successful after implementing this.



X-on/X-off

Chart 7.1

7.10 Usage of the 4 channel serial card

It was explained in chapter 6 that the BBC computer was used in the development stage but a racking system with a second processor and a 4 channel serial card was purchased later to be used with a separate keyboard. It was also explained that the software was modified to use this equipment.

The two DUARTS are referred to as A and B and are mapped to the memory at locations FD80 and FD90 while they consist of the following fourteen registers of which the first five are more important and have to be set up prior to communication while the other registers are not so important and that their default values can be used. Each register has an offset value (table 7.2) form a base address which is different for the two DUARTS.

(1) Mode Register 1 (MR1)

(2) Mode Register 2 (MR2)

(3) Clock Select Register (CSR)

(4) Command Register (CR)

(5) Status Register (SR)

(6) Output Port Configuration Register (OPCR)

(7) Input Port Configuration Register (IPCR)

(8) Auxiliary Control Register (ACR)

(9) Interrupt Status Register (ISR)

(10) Interrupt Mask Register (IMR)

(11) Counter Timer Upper Register (CTUR)

(12) Counter Timer Lower Register (CTLR)

(13) Rx Holding Register (RHR)

(14) Tx Holding Register (THR)

Offset	Read	Write
00	MR1 and MR2 of DUART A	MR1 and MR2 of DUART A
01	SR of DUART A	CSR of DUART A
02	reserved	CR of DUART A
03	RHR of DUART A	THR of DUART A
04	IPCR	ACR
05	ISR	IMR
06	CTUR	CTUR
07	CTLR	CTLR
08	MR1 and MR2 of DUART B	MR1 and MR2 of DUART B
09	SR of DUART B	CSR of DUART B
0A	reserved	CR of DUART B
0B	RHR of DUART B	THR of DUART B
0C	reserved	reserved
0D	input port	OPCR
0E	start conter command	set output port bits command
0F	stop counter command	reset output port bits command

Register offsets

Table 7.2

7.10.1 Mode Register 1 (MR1)

MR1 (table 7.3) is used to initialise the parameters such as parity, number of bits per character and the receiving control types. The value 00000110 which is equal to 6 (hexadecimal 6) was used to set up to have no parity, 7 bits per character and no receive control for both channels 0 and 1.

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Rx RTS Control	Rx INT Select	Error mode	Parity mode		Parity type	Bits per char	
0=no 1=yes	0=RXRDY 1=FFULL	0=char 1=block	00= with parity 01= force parity 10= no parity 11= multi-drop		0=even 1=odd	00=5 01=6 10=7 11=8	

Mode register 1

Table 7.3

7.10.2 Mode Register 2 (MR2)

MR2 (table 7.4) is used to set up local parameters, transmit control and the stop bit lengths. The value 00001111 which is 15 (0F in hexadecimal) was used to have 2 stop bits, no transmit control at normal mode.

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Channel mode		Tx RTS Control	CTS Enable Tx	Stop bit length			
00 = normal 01 = auto echo 10 = local loop 11 = remote loop		0 = no 1 = yes	0 = no 1 = yes	0=0.563 1=0.625 2=0.688 3=0.750	4=0.813 5=0.875 6=0.938 7=1.000	8=1.563 9=1.625 A=1.688 B=1.750	C=1.813 D=1.875 E=1.938 F=2.000

Mode register 2

Table 7.4

7.10.3 Clock Select Register (CSR)

This register (table 7.5) is used to set up the transmit and receive speeds independent of each other. The four least significant bits are used for transmit baud rate and the other four for the receive rate. 1200 baud was selected for both transmit and receive speeds with the value 01100110 which is 102 (hexadecimal 66).

7.10.4 Command Register (CR)

CR (table 7.6) is used to enable data in/out lines and/or to use miscellaneous commands which were not used. Therefore the value 00000101 which is equal to 5 (hexadecimal 5) was set up.

Receive baud rate	Bits 7 to 4	Transmit baud rate	Bits 3 to 0
75	0000	75	0000
110	0001	110	0001
134.5	0010	134.5	0010
150	0011	150	0011
300	0100	300	0100
600	0101	600	0101
1,200	0110	1,200	0110
2,000	0111	2,000	0111
2,400	1000	2,400	1000
4,800	1001	4,800	1001
1,800	1010	1,800	1010
9,600	1011	9,600	1011
19,200	1100	19,200	1100

Clock select register

Table 7.5

Bit7	Bit6 Bit5 Bit4	Bit3	Bit2	Bit1	Bit0
Not used	Miscellaneous commands	Disable Tx	Enable Tx	Disable Rx	Enable Rx
must be 0	000 = no command 001 = reset MR pointer 010 = reset receiver 011 = reset transmitter 100 = reset error status 101 = reset interrupt 110 = start break 111 = stop break	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

Command register

Table 7.6

7.10.5 Status Register (SR)

The status register (table 7.7) is used to determine the type of errors on which to enable interrupts. This was set up to have only 'ready to send' and 'ready to receive' with the value 00000101 which is 5 (5 in hexadecimal).

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Received break	Framing error	Parity error	overrun error	TxEMT	TxRDY	FFULL	RxRDY
0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

Status register

Table 7.7

The disadvantage of this 4 channel serial card is that the ports are not fully buffered as in the BBC computer and that they can not be selected as input and output channels. The ports therefore have to be driven by software and this is time consuming. It was also discovered that the response time of this system was much slower than that of the VAX computer which effectively made the

above explained Xon/Xoff routine redundant. The feasibility of implementing Xon/Xoff in the other direction was then looked at but it appeared to be impossible because the ports buffer only 3 characters. Therefore even if a DC3 code is sent as soon as one or two characters are in the buffer, the host computer is bound to transmit a few more before that DC3 is received.

After long hours of testing, the device appeared to work at 110 baud without failing and this is the speed which was used throughout. This was set up by setting the value 00010001 in the CSR which is equal to 17 (11 in hexadecimal) for channel 1 to communicate with the VAX while channel 0 was not altered because the keyboard could communicate at 1200 baud. The communication with the host at 110 baud is successful but the author of this thesis strongly recommends the use of either a shadow RAM card[185,186] which can be used as a port buffer or a device which is known as a 'Sprinter'[187] which can be used as an intermediate buffer.

Chapter 8

Management Level 2

8.1 Introduction

Chapter 5 described how the management modules were prepared to create jobs to be monitored on the shop floor. The data from the completed jobs have to be processed in order to create long term reports and statistics for the management. These can be preserved and referred to as guide lines in the future expansion of the company. This chapter describes the modules which obtain the raw data from the shop floor to process and create reports so as to achieve the ultimate objective of the monitoring system.

8.2 The Production Report

The production report is the single most important document which gives all the information about a completed job. Unlike the job card, the creation of the production report has to be automatic. An application software module was therefore prepared for production reports and was named 'PRODREP' with the usual

facilities but without an option on the main menu for creation of a report. Manual creation of reports (section 8.2.2.3) was however added via another menu so that the users can carry on with the process even if the shop floor computers break down.

The first part of the production report is almost identical to the corresponding job card but the produced quantities of each length (Appendix 15) are shown in item 18. There is also a comparison between the manually entered values by the operator and the automatically calculated values by the computer. The module extracts each category of downtime from the data which has been passed from the shop floor and compiles the total downtime for the job while also showing each category of downtime individually.

It was explained in section 5.10 that the total estimated job time is calculated in order to schedule the job. This quantity is compared with the actual time for the job. The estimated and actual rolling times are also compared. The setting up and strip times are not true downtimes since they are essential for any job. Therefore the total productive time should be the total of these two and the rolling time rather than only the rolling time.

$$\text{Hence the efficiency} = \frac{\text{Total productive time}}{\text{Total job time}} \times 100\%$$

Obviously the job is estimated to be 100% efficient without any downtime (other than setting up and strip times) but the actual efficiency has to become less than this because of other categories of downtime. It is also possible for the efficiency to decrease because of increased setting up, strip or rolling times. Therefore all these estimated quantities and efficiency are compared with the actual values in the production report which will become invaluable information for the management.

One other important factor which contributes to the profitability is the total cost which is broken down to various different costs as entered in the sales pricing card. The costing is different for each customer and the total cost of each category is calculated by considering the proportion which is ordered by each customer on the job. The total material cost and the overhead cost too can either increase or decrease from the estimated values because of the changes in the total weight of material used and the total job time. Hence the actual total cost can also be different from the estimated cost and therefore is calculated by considering each

individual actual value. The total selling price is calculated considering the same variables which have been used to calculate the estimated value. For example the selling price is calculated as a percentage of the material cost sometimes and the actual selling price is calculated to be higher than the estimated value as a result of an increase of the material which is used although the unit cost of material is not changed. The profit, profit per hour and the percentage profit are also increased as a result. This means that any job is not expected to earn a loss, as per calculations. This however is not practical as the selling price may not be able to be increased once quoted. Therefore provision has been made to alter the selling price manually. It is also possible for the unit material price to be different from the estimated ones while the different items of cost also might change although not as a result of the job performance. Hence provision has been made for the users to change these quantities too; and the total cost, profits and percentages are altered automatically. These quantities of new (incomplete) records are presented to the user when he enters the module so that he can either complete them skip to carry on as in the case of other modules. A useful feature of the module is that the user can specify the items he needs in the reports when a full report is not needed.

8.2.1 Tied Jobs

One important facility of the module is the identification of tied jobs as explained in section 5.10. In the case of tied jobs, this module creates only one production report by addition of all the details, and numbers the production report with the number of the last finished job of the lot. If a further tied job is added subsequently, then the production report takes the number of that job.

The deadline and the finishing times (estimated and actual) become the latest ones since the tied jobs are considered as single jobs while the starting times are the earliest of all jobs.

8.2.2 Auxiliary Modules

8.2.2.1 Production Start Module

The starting time of a job is determined at the time of creating it but the actual starting time and date can be quite different to this. This time and date can quite easily be maintained

in the shop floor computer and can be passed on to the host along with other variables when the job is finished, but the management will not be able to see actual starting time and date until the job is finished.

Therefore a module called 'PRODST' was prepared to write the time and date in to the appropriate job card as soon as the job is started on the shop floor. This module is executed by the shop floor computer by sending the command 'RUN PRODST'.

8.2.2.2 Production report creation - automatic

It was explained above that the production report is automatically created. This is done at the end of a job and the shop floor computer sends a command to execute a module called 'PRODCREA' at the end of each job.

This module searches for a file 'FIN.DAT' in the line directory and extracts the job number from it. It then extracts the corresponding record from the job card file. Finally a record in the production report file is created while the corresponding records in

the job card file and the scheduling file are deleted so that obsolete records do not get accumulated. The production reports have been assigned the same format as the job cards and therefore two data files have been created with the names 'PRODREP1.DAT' and 'PRODREP2.DAT' with channel numbers 10 and 11 respectively.

8.2.2.3 Production report creation - manual

Only the top management are allowed to create the production reports manually and this is quite a useful facility at a time when the shop floor computers break down. It is however expected that a job card should be existent before any attempt is made to create a production report.

A stand alone application software module called 'PRODMAN' was created for the manual creation of production reports and the first item of information that it requires when executed is the corresponding job number. The user gets the chance to enter '0' and exit at this point but will have to enter all the other information which is usually collected form the shop floor if he wishes to proceed. This information includes the starting and

finishing times of the job, produced quantities and each category of downtime.

The module, although a stand alone module, is accessed within the 'PRODREP' module where a new process is created(188) when the user selects the appropriate option. If the parent process is known as 'JONES' then the child process gets the name 'JONES_1' and the former waits at this point while the latter executes the module 'PRODMAN' and hands the control back to the parent. It must be remembered that each user therefore should have sufficient VMS privileges to create a second process. One other important consideration is that the any other data file does not get automatically updated if a production report is created manually. The users must remember to update the periodical downtime, operator data, bonus and also the job history files manually.

8.3 Periodical Downtime Report

It is important for the management to see the total downtime under each category for a particular period so that they can observe the efficiency of operations for the whole company. It

was decided that a suitable period would be a week, after a number of discussions with the client's staff .

Making these reports for each week is very complicated especially to be carried out on an operating system such as VMS. A module which is named 'DOWNTIME' was prepared on the VAX computer to be executed by the shop floor computers as soon as they pass the downtime information to the VAX. This module first checks the current date and determines the last date of the current week which must essentially be a Sunday. It then scans the directory to find a file whose name consists of that date. (The format of the name is 'DOWN120886.DAT' where the last day of the week is 12.08.86). If such a file is non existent then it opens a new file and writes the downtime data whereas it adds the data in the case of an existing file.

The determination of the date, search and the file names are transparent to the user. The user may not be able to identify the appropriate file even if he lists the directory because a part of the file name would simply be a series of figures while the contents are also coded. Another application software module named

'DOWNDET' was created to access these data files to extract information. The user simply has to inform the computer whether he wants to see the current downtime file (Appendix 16) or a previous one. He has to supply the last day of the week if it a previous file that he wants to access. This module also has all the usual facilities of any other management module and was later extended to give information about the bonus for the week too. The same principle was used in extracting bonus details and the file names are similar to 'BONUS120886.DAT' as before.

8.4 Operators' working details

It was explained in section 6 that number of hours worked, the total length rolled, and the total bonus earned by each operator are calculated on the shop floor computer and are passed on to the host. It was decided that those details are also more useful periodically than against each job. Therefore a module 'OPERDAT' was created to carry out a similar search as in section 8.3 and to create data files similar to 'OPERDAT120886.DAT'. These data files (Appendix 17) are accessed in the same way as the downtime details and the application software module for this is 'WORKDET'. This

module can quite easily be extended to calculate the wages for each operator for every accounting period.

8.5 Job Monitoring

The users who are logged in to the host via their own terminals can not observe the progress of a job on the shop floor because these details are on the shop floor computers, but it was explained in section 7.8.4 that periodical details are written in to a file named 'STATUS.DAT' every five minutes or if a change to the normal routine occurs before that. This file is updated regularly on all four lines whether the management wants to see it or not. The information is extracted from the status files by an application software module called 'STATUS' and is displayed on the management terminal in an easily readable manner (Appendix 18) at any time the management wishes to monitor the progress of the shop floor work. The information which the management see will not be 'the up to the minute information' since the information is extracted from the data files, but it can be five minutes old in the worst case.

8.6 Complete History of Jobs

Section 7.8.4 explained how each occurrence, each alarm message and all other relevant information during any particular job is logged in a file which is exclusive to that particular job. A module which is named 'LOGFILE' was created so that the management can see or print these details at any time after a job has finished (Appendix 19) by specifying the job number. This is a very useful facility if one needs to investigate a job which has taken a very long time or made a huge loss.

8.7 Stock Control

It was not planned to create software for stock control functions according to the design specification (Appendix 4) but was planned to make provisions to use some software which was available. It was later discovered that the existing software on stock control was not adequate after considering the complexity of the complete project which was not foreseen at the time the design specification was made. Therefore it was decided to design a completely new set of modules for stock control although this is a

complete deviation from the planned work. These modules were also planned to be compatible with the other software to transfer data in and out within the system.

8.7.1 Raw Material File

The raw material file had to be designed in such a way that five basic functions are available to the user as follows.

(1) Order

(2) Receipt

(3) Issue

(4) Reject

(5) Audit

The variables for each transaction were set out as

(1) Date of transaction

(2) Reference number

(3) Supplier

(4) Unit price

(5) Quantity

(6) Balance

In addition, the header details (Appendix 20) like the product code, description and material description were decided to be included. It was also decided to include the customer name since each company prefers to maintain data under it. It must however be noted that this parameter has no significance to the records and it is included simply as an extra means of extracting them. One product is entered in only one raw material record even if it is sold to a number of different customers. Hence the uses are free to enter just one customer name in any card.

The computer prompts for the initial balance at the time of creation and if this is entered as zero then the other details of that transaction are not requested. If however the balance is greater than zero then the user gets the chance to enter all other details. The user does not have to type the appropriate word for the status of a transaction such as order, receipt etc. but is presented with a menu which consists of all five options of which he can select one.

The computer calculates the balance for each entry while not allowing the user to issue more than the existing balance. The computer uses the FIFO (First In First Out) rule for issuing and calculates the unit price of the amount issued by locating the corresponding receipt or receipts for that transaction.

8.7.1.1. Zero Value Stock

The stock sometimes gets depreciated when it gets very old until it has no value. The management can do this on the computer by issuing certain stock on a dummy basis and repurchase at a lower cost. This can be carried out until the stock value is nil. This zero value stock sometimes has to be issued for certain jobs and in such a case the computer does not apply the FIFO principle but displays all the available zero stock items and asks the user to select the one to be issued. If a higher amount than any one available zero value stock has to be issued then it has to be done in two transactions. The reason for this is to maintain the age of the stock.

8.7.1.2 Audits

The user can specify the balance in the case of audits and this balance becomes the starting point of all subsequent transactions although the age of each item of stock is maintained and the same FIFO rule is applied when issuing.

8.7.1.3 Reject stock

Rejected stock is similar to issued stock but each receipt of stock is shown in LIFO (Last In First Out) order for the user to select the stock to be rejected. Once again the age of stock is maintained. Furthermore, each reject transaction is entered in another file.

It was initially decided to allocate 16 transactions per record so that all can be fitted in to one screen but later all the arrays were increased to provide a second page which gives a total of 34 transactions. The module was also extended to be linked with sales pricing card, job card, the customer file and the gauge tolerance file. At the time of creation of any particular record, it looks for a

sales pricing card which has the same product code and extracts the product and the material descriptions. Then the user has to supply 4 characters of the customer name in the usual way and the appropriate name is extracted. If however the sales pricing card is non existent, then this module places an incomplete record in the sales pricing card file while also specifying the current record as 'incomplete' to be displayed for the user to complete later.

It is useful to see the stock level of the appropriate material at the time of creating a job by using the job card module. Therefore the job card module was extended to look at the stock level by approaching this module. The stock availability, part availability (with the amount which has to be ordered) or the non availability is therefore shown on the job card and it places an incomplete record in the raw material file in the case of a non existent raw material card. As in the case of sales pricing card, the raw material module also scans all the cards as soon as a user enters it to look for incomplete records so that the user can either complete them one by one or skip.

The top transaction of a raw material card is lost when the number of entries get accumulated and exceed 34 (which is the maximum number of transactions maintained at one time) and all the subsequent transactions scroll up to make room for the new entry. This would cause a problem if the first entry had some stock which contributed to the final balance. In such a case where 'live' data is going to be lost, the module informs this to the user with the amount of stock involved and requests him to do a dummy issue and repurchase the same amount. The user is not allowed to proceed without doing so.

Another facility is that an invisible 'marker' is placed against the first entry of a new card and a copy of this card is sent to the default printer if this entry with a marker is going to be lost by the accumulation of entries, even if the data is 'dead' (i.e., does not contribute to the final balance). A new marker is then placed at the last entry and the same procedure is carried out when that entry is reached the top subsequently.

This module has become a very long and a complicated one so as to fulfil all the requirements. The data channel of it is 14

while the data file which is named 'RAWMAT.DAT' has a record length of 260 bytes.

8.7.2 Reject File

This is a relatively shorter module and has the inverse format of the raw material file where the user has no facility to create records but has a facility to remove records. This is because the records are created automatically at the time the stock is rejected via the raw material module. The name of the data file is 'REJECT.DAT' and the data channel is 12 with a record length of 100 bytes.

8.7.3. Finished Product File

The finished product file is very much similar to the raw material file but is less complicated. The basic functions of this module are as follows.

(1) Produced

(2) Despatched

(3) Reject

(4) Audit

The balances in this case are integers (Appendix 21) rather than real variables which have been allocated for material weights in the raw material file. Another difference is that the rejected products are added on to the balance because it is the customer who rejects. The age of stock is maintained against each entry on the module while audits are also similar to the raw material file.

The name of the module is FINPROD while the data channel is 13. Each record of the data file is 200 bytes long.

8.7.4 Instantaneous Stock Value

The raw material file and the finished goods file have the stock details and the balances of all stocks and materials but the total value of these stocks is not seen at a glance.

A special module was created to calculate the individual stock values to total them which consists of two main sections, one for the raw material and the other for the finished goods.

The raw material section accesses the raw material file and scans each entry of each card. If the entry is either a receipt (in which case it is an addition to the stock) then it looks at all the issues to see whether that particular receipt forms a part of the current balance. If so, the value is calculated by considering the unit cost which is written against that particular stock. Similarly all such stock is written in to a temporary file named 'FOR015.DAT' and is sorted in the chronological order. The module then codes all stock as A,B,C and D depending on age (Appendix 22) before being presented to the user. Hence the management is able to see the total value of stock at an instant.

The second part of the module is to calculate the finished product value. It is more complicated than the raw material value calculation because a price is not available in the finished product card. The finished product card is accessed in the same way as the raw material card, first to ascertain the contributing stock from each

line. Then the sales pricing card file is searched to extract the corresponding card in which the actual selling price is available. The rest is similar to the raw material value calculation and even the coding is carried out in the same way.

Chapter 9

System configuration and security

9.1 Introduction

The previous chapter discussed about building and implementing a general purpose computer integrated monitoring and management system for which a VAX II 750 computer was used as the host. By this stage 2 large scale VAX 8650 computers and a VAX II 785 computer which were clustered together were available within the university. A MicroVAX II computer was also available at the client's premises and all the computers were discovered to be compatible with the old VAX II 750. The management software was moved to the new computers at this point to run at both locations. The software and hardware were then used with actual data by the actual users after they were trained to use the system. The configuration of the system as a multi user environment which needs total security also forms an essential part of the planned research and this chapter describes how it was done.

9.2 File handling

It is essential to understand how the files are set up in the VMS operating system before the security is introduced. The user can write software in any language and these programs are compiled by the specific compilers which can be found for any language. The user can also give commands directly or write programs in a language which is called DCL as explained before. This language is an interpretative language and is slower than the compiled languages as can be expected. It is however essential to have a knowledge of this language because the usage of it is the only means of creating software files regardless of the language which they are written in.

9.2.1 Device

The software obviously has to be saved in some device as explained in chapter 3 and such an applicable device in this case is a hard disc. The access of one or more hard discs is transparent to the user but they have to be named at the system initialization. The default name for the hard disk in the micro VAX computer which is

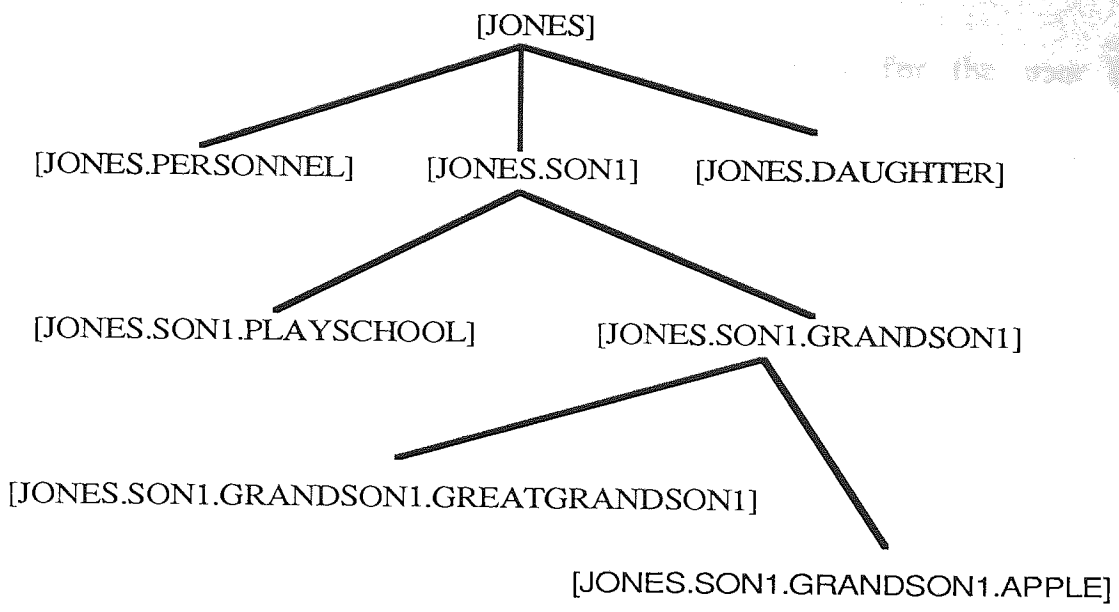
being used is '$DISKI:' while the name of the disk which is being used in the cluster is 'IDUA6:'. The colon (:) must essentially be the last character in a device name.

9.2.2 Directory

A directory or directories must first be created in a device to locate the software and a directory name can consist of up to 20 alphanumeric characters but must be enclosed within square parenthesis. A typical directory name is '[JONES]'.

9.2.2.1 Sub directories

Any directory can have any number of sub directories and even the sub directories can have their own sub directories. The names of the sub directories are placed within the same pair of square parenthesis but are separated from the main directory name with full stop characters. Two examples are '[JONES.PERSONAL]' and '[JONES.SONI.GRANDSONI]' (figure 9.1) where '[JONES]' is the root directory. The software can be placed either in the root directory or in any of the branch sub directories.



The structure of the directories

Figure 9.1

9.2.3 Files

The file names can also have alphanumeric characters and it is up to the user to name his files in a logical manner by which he can identify them.

9.2.3.1 File types

File types usually consist of three characters as explained in section 5.7 and table 5.1. These are the usual types which are

recognized by the computer. It is not compulsory for the user to adhere to this convention and he can even use file types which do not consist of three characters. The only rule to use a file type is to precede it with a full stop character so that the subsequent characters are differentiated from the file name. Alternatively the file type can be omitted and only the file name can be used.

9.2.3.2 Version numbers

The version number of a file which is a numeric value is determined by the VMS and this is preceded by a semi colon (;). The version number is determined as 1 when a user creates a file for the first time, and any subsequent version is saved with a version number which is one unit higher than the highest existing version in the current directory. The user however does not have to specify the version number if he wants to select the latest version. This is automatically done by specifying the file name and the file type although the version number has to be specified to select an earlier version. An exception to this rule is deletion where the version number has to be specified even to delete the last version.

9.2.4 Selecting a file

The file name in a directory is similar to CUSTOMER.DAT;10 which can be selected as CUSTOMER.DAT if the latest version number is 10. A file can be selected by specifying only the file name this way if it is resident in the current directory.

A file in another directory can be selected by either selecting that directory first with the command '$SET DEFAULT [JONES.SONI]' where the directory name is [JONES.SONI] or specifying the full file name. The full name consists of the device name, directory name and file name such as, $DISKI:[JONES.SONI]CUSTOMER.DAT. The full file name can be shortened to [JONES.SONI]CUSTOMER.DAT in the case of the directory being in the same device (which is the usual case).

9.3 Users

The system manager has to introduce users to the VMS system to open their accounts and has to specify a default directory which must be an existing directory in the system for every user.

All the user's work is done within this directory by default when he logs in unless another directory is selected manually.

9.3.1 User Identification Code (UIC)

The system manager has to allocate a UIC to every user at the time of opening an account. This consists of two numerical values separated by a comma and held between square brackets such as [201,1], where the first value is known as the group number and the second, the user number in the group. The UICs are located in an area called the 'RIGHTS' data base in the VMS system.

9.3.2 Identifiers

Identifiers can be used in addition to the UICs and these can be of the same format as the UICs where an alphanumeric equivalent to both the group and the user number can exist. The advantage of identifiers is that all users can be given unique identifiers even in a case where many UICs have to be the same. The identifiers are located in an area called the 'Access Control List' (ACL).

9.3.3 Classification of Users

All the users in the system are classified in to four categories as follows and any file has an identification tag which gives either the owner's UIC or identifier/s.

(1) System

(2) Owner

(3) Group

(4) World

9.3.3.1 System

This is usually the system manager who has a unique set of identifiers.

9.3.3.2 Owner

A person who has created a file is classified by the VMS system as the owner of it.

9.3.3.3 Group

The users who have the same group number in their UIC as the owner of the file are known as the group. This also holds true for the users who have the alphanumeric equivalent identifier for the group as the owner.

9.3.3.4 World

All the users in the system belong to the world category.

9.4 VMS based protection

The built in protection in the VMS allows the users to do the following operations to files.

(1) Read (R)

(2) Write (W)

(3) Execute (E)

(4) Delete (D)

The owner of a file can give any of these privileges to any category of the users as explained above. Additionally the same users should be given the same privileges to the appropriate directory in which the file is located for them to carry out the required operation. If the file is in a sub directory then the users should also have the sufficient privileges in the root directory and the branch until the destination directory is reached.

9.5 Security

VMS based protection is inadequate to carry out internal file operations while it is essential to allow the access of a group of users to a set of files. The biggest disadvantage of the VMS based protection is that it allows operations to complete files rather than to different parts of files. For example, any user who has the privilege to write to a direct access file can delete records via application software while a user who has only the read privilege can not enter such a file unless the file is created as 'read only'.

While recognizing the fact that all the users who have to use the created system must be put in to a single group, it also must

be understood that their task is only to run the application software modules which perform the other tasks like calculations, reading writing and communications. Hence the users do not have to edit or create files manually which make it unnecessary to enter in to the operating level. There is also no need to have one directory for each user and it was decided to specify a directory called '[CAM]' as the default directory for all the users.

9.6 System structure

The development of software was carried out in a directory called [MASTER] which was specified as the default directory for an account holder who was given the user name 'MASTER'. All the authorized users and the 'MASTER' account were given the group number 201 and an individual member number, while opening the [MASTER] directory for all the group members to read, write and execute. The users can access the files in the [MASTER] directory to perform their tasks from the directory '[CAM]' while the '[MASTER]' directory was expanded later to include two sub directories '[MASTER.SOURCE]' for the source files and '[MASTER.IMAGE]' for the executable images for extra security. All

the users were given the privileges to read, write and execute in the latter while no one was given any privilege to access the former.

9.6.1 Subsidiary Companies

Usage of a file name within a Fortran module opens a data file in the [CAM] directory since the default directory of all users is [CAM]. This presents a problem because a number of companies who have to have their own data files are involved in the system. One way to eliminate this problem is to have a different system for each company by duplicating the image files and another is to allow the users of different companies to have their own default directory similar to '[CAM]' while running the same application software modules in '[MASTER.IMAGE]'. The latter case is more efficient but it is difficult for the policy makers of the group to go in to all the directories and is better if all the users can use the directory '[CAM]' so that nothing is necessary to be duplicated. The solution for this is to have a different sub directory for each company's data files within the '[MASTER]' directory. The companies therefore were given a numeric value such as 1, 2, 3 etc. and their sub directories were given the names '[MASTER.BAY1]',

'[MASTER.BAY2]' and '[MASTER.BAY3]' etc. respectively. The Fortran modules specify the full file name such as '[MASTER.BAY1]CUSTOMER.DAT' in allocating a channel number to read and write data to these files. The numeric values which correspond to the company numbers are parameters and these have to be set up each time a user enters a module by recognizing which company he belongs to. Therefore a file must be maintained in the system which maintains such parameters.

9.7 User authorisation file

This module was built to maintain the user details and it consists of essential features like each user's name, the default directory, account name, UIC and the company number to which the user belongs.

The design specification shows the need for different levels of access to personnel depending on their seniority. Therefore it was decided to set up three categories of users for each company in addition to the policy makers of the group who obviously must have an additional privilege of being able to access the files of all

companies. The system was decided to be set up in such a way that the users who have the level 3 are the lowest in the hierarchy and 'reading' being the only operation which they are allowed to perform while those who have level 2 can read and write data. The category of the users who have been assigned the privilege level 1 are the only category who can delete data. These parameters were also added to the authorisation file.

All the software was extended so that each module extracts the UIC from the VMS system. It then reads the authorisation file to compare the system UIC with the UICs in the authorisation file until a match is found. The corresponding details of such a matching UIC give all the necessary information and therefore the data files can be opened in the appropriate directory. All the modules were also added the extra facility of deletion which can be accessed by keying in a number which is one higher than the highest option (which is usually the exit) from the menu. This option which is not visible for extra security takes the users to an extra menu when selected, providing the user has the sufficient privilege. If the privilege is less than 1 then the user is returned to the main menu with a polite message informing him that he has not got the

sufficient privileges. It is only via the higher level menu of the production report module that the top management can create production reports manually as explained in section 8.2.2.3. The new record creation in any other module and updating are also disallowed if the user has a privilege which is less than 2.

9.8 Unauthorised Users

One important rule to remember is that every user's details must be entered in the authorisation file for him to use the software. If a user who's details do not appear in the authorisation file attempts to use the software then he is automatically logged out and his UIC is written into another file with the time and date of the attempt. This file can be listed by the top management.

It may be assumed here that the users can first alter the authorisation file by either adding their details or giving themselves higher privileges but it is not possible because the authorisation file has been prepared in a special format which allows only two pre specified people to use it. If any other user attempts to execute it then it too carries out the above task of logging the user out and

writing his details to the unauthorised file. This is a very powerful utility where such procedures have been provided by advanced programming. Even the VMS system manager with his full set of VMS privilege can not access this.

9.9 Company File

Each subsidiary company was allocated a unique number for the purpose of directing the user outputs to the appropriate directory while being totally transparent to the users. This utility was further extended to use the company name in headings of different documents by creating a special module.

An abbreviated name was also built in since it is not possible to show the full name in each individual screen of display. A user's company number is extracted from the authorization file when he enters any module. The company's full name and the abbreviated name are extracted from the company file using this number. This feature is very useful to the policy makers because they can not forget which company's files they are using at any time because virtually every screen displays the abbreviated name while

the full name is displayed in every menu and listing. The hard copies also carry the full name as the heading which is an essential feature when summarizing the reports of the whole group.

The application software module is an independent module and the policy makers can use it to enter or delete company names. The name of the module is 'COMPANY' while the data channel is 29 and the length of each record is 45 bytes. The company file, authorisation file and the unauthorised users' file can be accessed via a menu program called 'UTILITIES' which can be selected by typing 'UTILS' from the master menu. This is available only to the pre specified policy makers of the group of companies.

9.10 'Login' Command File

The users have to type the file name (and the directory) manually when they want to run a particular module after they log in to the [CAM] directory. The appropriate command is similar to 'RUN [MASTER.IMAGE]CUSTOMER'. This is rather a long string to type and also the security is adversely affected when the users are allowed to enter in to the operating level. Therefore it was decided

to create one module in DCL which can run all the different modules. The user is shown the list of main modules in terms of a menu and is allowed run any one of them at a time by simply pressing the appropriate numeric key. The whole system therefore appears to the user as one program and is quite easy to run. This menu was named 'MASTER MENU' where the name of the DCL module is 'MENU.COM' which can be executed by typing '@MENU'. This once again has to be done in the command mode of VMS. The VMS utility of maintaining a file named 'LOGIN.COM' in the default directory eliminates this. The VMS automatically executes a file with this special name if found in the default directory as soon as a user logs in. Therefore the main command file 'MENU.COM' was placed in the [MASTER.IMAGE] directory while creating a file LOGIN.COM in the [CAM] directory to issue command '@[MASTER.IMAGE]MENU'. Therefore the users are taken to the master menu as soon as they log in and are not allowed to do any thing other than selecting and running programs from it. The log out command was also placed in this file and the users are logged out as soon as they exit from the master menu. All the software is thus protected and no user can modify any source or image file.

9.11 Additional Information

The interrupt procedure on the VMS system is carried out by pressing 'CONTROL' and 'Y' keys together at any time and the users are returned to the command level regardless of what they have been doing. This feature was disabled for individual users by setting the appropriate flags on the VMS RIGHTS data base, in order to make the system foolproof. The master menu was also divided to two and a secondary menu was created by placing the less used modules in the latter because the earlier became too long. It is assumed that modules like the customer file and gauge tolerances are not used daily once they are set up. Therefore six main modules were modified to be accessed via the secondary menu while the remaining 12 were left (Appendix 23) in the master menu. The feature to change the individual users' passwords was also added to the master menu.

Finally all the data files and text files in the system were opened to the group for reading and writing while the image and command files were opened for the group for reading and executing. The 'world' category was not given any type of an access privilege.

9.12 Super User

The need for a 'super user' was briefly explained in chapter 4 while it was already explained about the three levels of privileges (section 9.7) which were set up.

Three special UICs were allocated to be used for this 'super user' level and a module called 'BAY' was created to such personnel to select the company which they want to access. The theory which was used for this is to extract all the company names from the company file and display them in terms of a menu for the user to select. Then the corresponding number is entered in the authorisation file against the user. This number is altered in the authorisation file (only for these special users) every time a new selection is made. Therefore the data files of only the company which is selected are opened at one time since the modules extract user information from the authorisation file.

Identifiers were introduced for the whole group and currently each user has two identifiers; one being equivalent to his user name and the other to 'HADLEY'. The complete identifier for

each user take the form of '[HADLEY,MASTER]' and they prevent any user other than those who are specified from executing the module 'BAY'. This is done by a small routine in the command file and only the policy makers are allowed to enter the module while all other users are directed to another part of the program. In addition, the module 'BAY' itself checks the user parameters for extra security and prevents any user other than those who are specified to proceed any further. This program can not be executed as a stand alone program by any user.

9.13 Special Levels

The super user level and the three privilege levels are sufficient in most cases but the sales pricing card is a special module which is executed by sales representatives sometimes. Therefore they should have the privilege to create and update data although such operations should not be allowed in other modules.

If a special privilege was created for the sales personnel then all the software has to modified to accommodate it. It was therefore decided to use an alternative method by involving the

UICs. The sales representatives were allocated the UIC member numbers which are greater than 749 and the sales pricing card was altered to allow all users with privilege 3 and a higher UIC member number than 749 to create records and update them while allowing those with privilege 3 and a UIC number lower than 750 only to read. This effectively increases the number of levels to five while it eliminates the need to give privilege 2 to sales personnel which would allow them to create and update data in other files.

9.14 Shop Floor Computers

Chapter 5 explained that each company locates some data files in separate sub directories to be sent to the shop floor computers while chapter 7 explained that the VAX computer considers all line computers as individual users. It is necessary for these computers to log in to the specific directory which is allocated to the particular line and also it is necessary for each company sub directory to have four further branch directories, one for each line as [MASTER.BAY.1.LINE1], [MASTER.BAY1.LINE2], [MASTER.BAY.LINE.3] and [MASTER.BAY1.LINE4] so that they become the default directories for each corresponding line to log in to.

9.15 Backup

A good system should have a back up facility which will not allow the total data to be lost in the case of a problem. The company as a group should be doing this by backing up all files on to a tape but it is also useful to allow each subsidiary company to back up their own files on to a different area of the same disc. This would be quite useless in the case of a major problem like a hard disk error but will be very useful in recovering a lost file or files rather than mounting the magnetic tape and running a search which is time consuming.

A utility which is named 'BACKUPS' was created for this purpose and this saves all the data files of a specific company to another sub directory within the sub directory where its data files are located. These backup directories have names such as [MASTER.BAY1.BACKUP] and [MASTER.BAY2.BACKUP] etc. This utility can quite easily be run as a background job which automatically backs up all files in all the companies at a specified time every day or every week, but the disadvantage is that the corrupt files would then be backed up automatically which will

make the whole process quite useless. Therefore it has been configured to be manual and only the users who have the privilege level 1 are allowed to do this. It is their responsibility to do a periodical backup (preferably daily) after ensuring that there is no corrupt data. The module is executed via the master menu and the user has to type 'BACKUP' at the master menu to execute this.

9.16 Recovery

The backing up of files as explained in section 9.15 is useless unless there is some means of replacing individual current files by the old versions. A utility which is named 'RECOVER' was created to carry out this task and the user (once again with privilege 1) can either select and replace files or replace all the files by typing the word 'REPLACE' at the master menu which executes the module.

9.17 Non Traceable Errors

All the direct access data files have been designed in such a way that they follow the same principle as on a floppy disk[189]. The catalogue information such as the number of records in the file

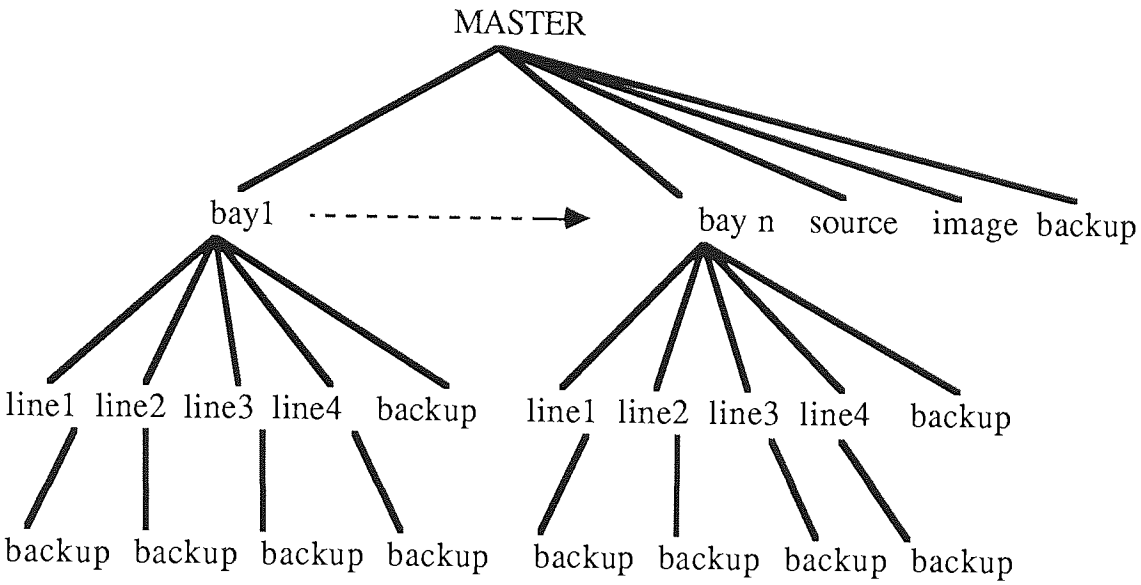
and last update details are kept in the first record while the other records consist of data. This method makes access very fast and efficient as opposed to reading through the complete file every time a new record has to be appended as done in some previous files for the company by the only person[115] who has used the direct access format in the past. (The sequential access method is even slower).

The catalogue information is read every time a user enters any module and is displayed above the main menu. A new record is appended to the file every time a user wishes and the catalogue information which is updated at the same time is used for this. The biggest problem in direct access files is that it sometimes sends spurious characters or control codes to records if an electrical fault or a voltage surge is occurred while the data is being written into a record which makes it unreadable. Sometimes the application software tries to read this record according to the catalogue information but finds it non existent and thus fails. This is a serious fault but can be eliminated easily by simply adjusting the catalogue information to read one less than the previous value of records which effectively 'deletes' the last record.

Fifteen modules were created to adjust the catalogue information on each module which uses the direct access format and were included in a single menu program called 'MENPROB'. This program is available to the users who have privilege 1 and can be accessed via the master menu by selecting an invisible option which is one higher than the maximum option as in the menu of any other program. The module names are then presented in an auxiliary menu from which the user can select one or exit to the master menu. Extreme caution must be taken as any selection made on this module would delete the last record of the corresponding data file which can cause a good record to be deleted if an incorrect selection is made. Such a record can however be recovered because the record is not actually deleted although the filing system is not aware of the presence of it due to the catalogue information. The source file should be altered to write the correct number of records to the catalogue to recover such a lost file. It must be remembered to re-adjust the source file to the original format as soon as it is done.

9.18 Tree Structure

The tree structure of the system is shown is figure 9.2



The tree structure of the system

Figure 9.2

Chapter 10

Networking

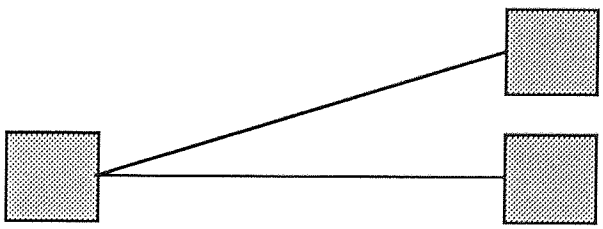
10.1 Introduction

The installation of shop floor computers and management terminals require a large number of ports on the host while the MicroVAX II computer provides only 8 ports for users to interact with it. This number can be increased to 16 with a special license but is not sufficient to accommodate all the subsidiary companies. Each company needs 8 ports at the very minimum since 4 have to be allocated to the shop floor computers and 1 for a printer which leaves only 3 ports available for the users. This means that one computer can be shared only between two subsidiary companies at present while even this arrangement is not sufficient in the future where each subsidiary company has to have it's own host computer. This needs some networking arrangement and this chapter explains how this can be done.

10.2 Network configurations

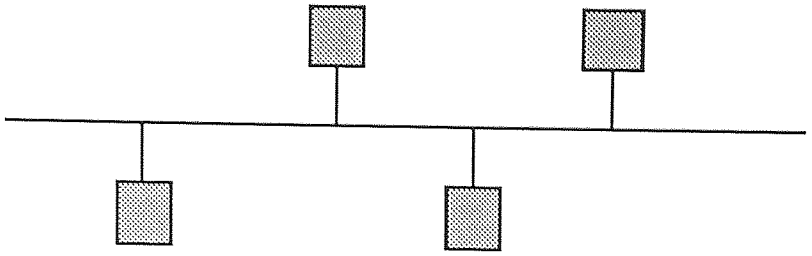
There are two basic types of networks which are known as point to point (figure 10.1) and multipoint or multidrop (figure 10.2) networks. All points are connected to each other in a point to point system but it is not practical to have such a network in large system. The multipoint nodes are connected together with a single cable.

The networks can either be centralised with one controlling node (figure 10.3) such as a dedicated communication processor or a switch, or be distributed (figure 10.2) which gives all nodes an equal chance to use the network.



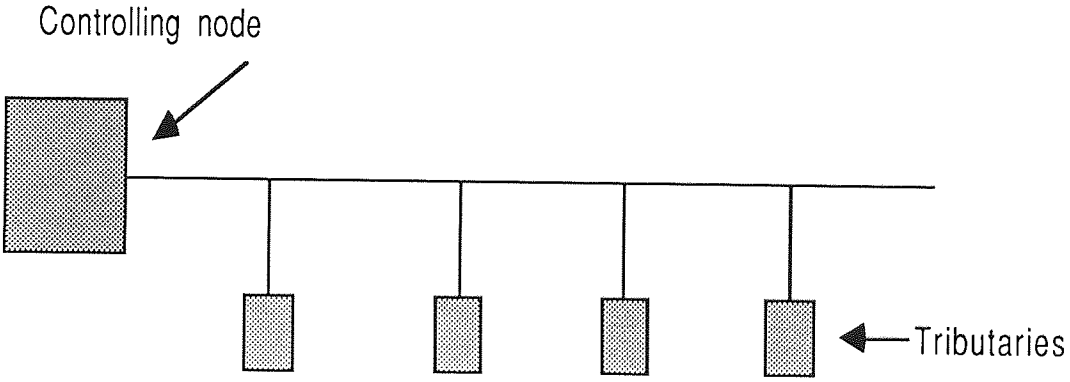
Point to point network

Figure 10.1



Multipoint network with distributed control

Figure 10.2



Multipoint network with centralised control

Figure 10.3

10.3 Network architectures

The number of connection architectures of computer equipment or topologies are numerous. Some of them are as follows.[190,191,192,193]

(1) Star

(2) Ring

(3) Loop

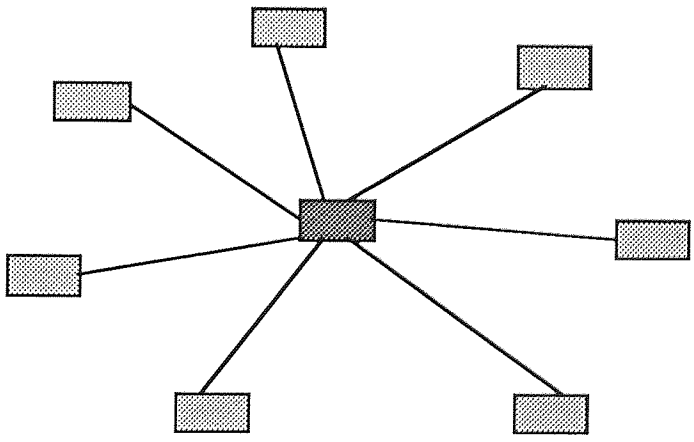
(4) Tree

(5) Mesh

(6) Bus

10.3.1 Star network

A star network is essentially a centralised network (figure 10.4) where one common device is shared by all other devices.

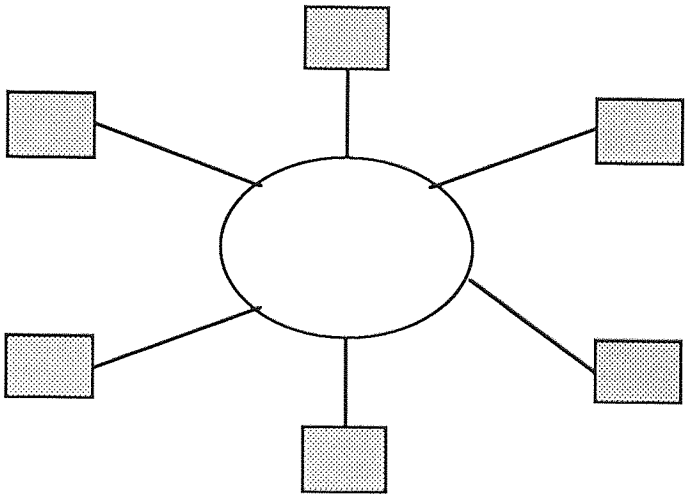


Star network

Figure 10.4

10.3.2 Ring network

A ring network is one which has all the devices connected in a ring (figure 10.5) where the communication passes one way round the ring. The communication can either reach the destination or arrive back at the source.

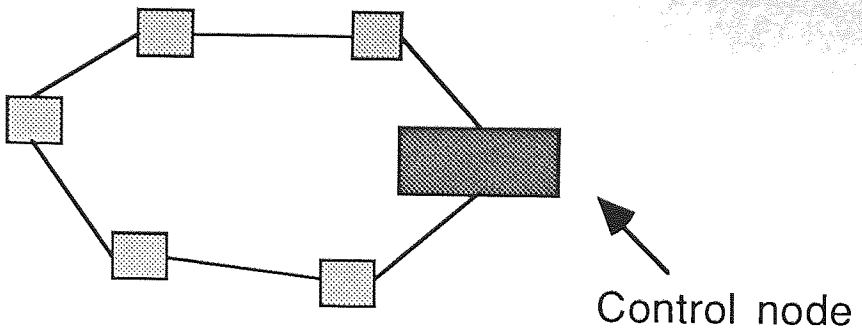


Ring network

Figure 10.5

10.3.3 Loop network

A loop network is very much similar to a ring network but with centralised control as in figure 10.6.

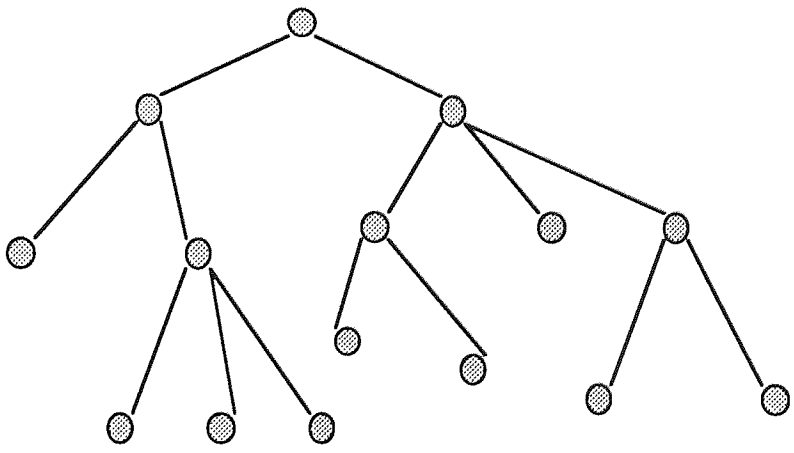


Loop network

Figure 10.6

10.3.4 Tree network

Tree network branches off from a root node (figure 10.7) similar to the directory structure in VMS as explained in chapter 9. A single root is available to travel from one node to another.

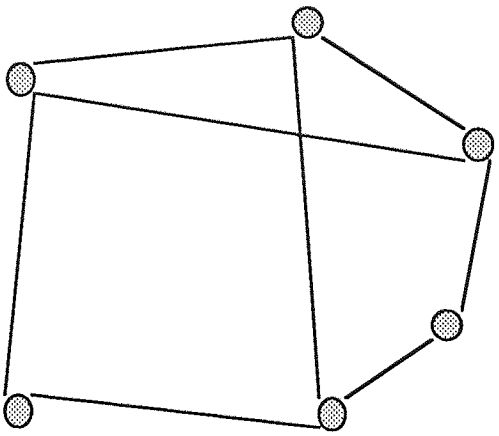


Tree network

Figure 10.7

10.3.5 Mesh network

A mesh network (figure 10.8) is sometimes used in cases where more than one root is required between nodes.



Mesh network

Figure 10.8

10.3.6 Bus network

A bus network is a multipoint network with distributed control as shown in figure 10.2.

10.4 Transmission media

Transmission media are either bounded media such as wires cables and optical fibres or unbounded media such as air waves. The main interest as applicable to this research is the former which consists of the following.

(1) Twisted pair

(2) Broadband coaxial cable

(3) Baseband coaxial cable

(4) Fibre optic cable

10.4.1 Twisted pair

This is the simplest transmission medium which consists of a pair if insulated wires which are twisted to form a helix. Many pairs of such wires can be used in one cable which is protected by shielding it along it's entire length. This shield is a wire mesh which can be earthed to eliminate electromagnetic interference.

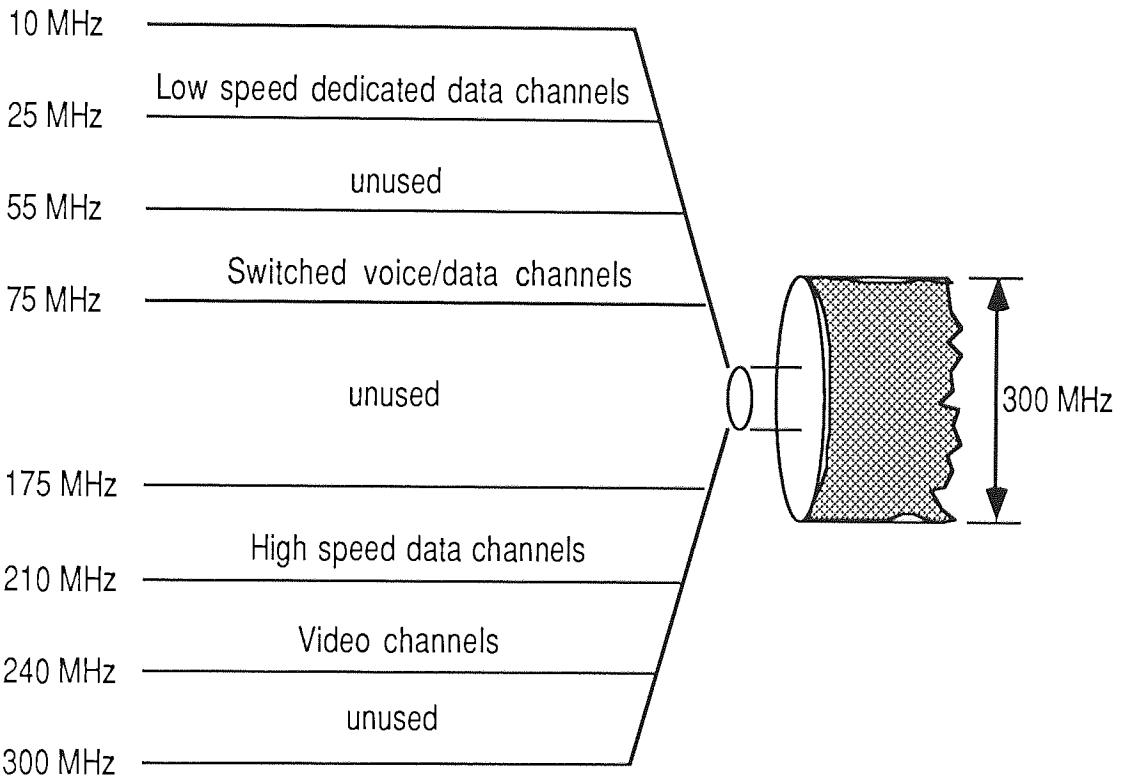
10.4.2 Broadband coaxial cables

Broadband coaxial cables consist of thick single insulated wire (typically 1.25cm) which are surrounded by a conducting shield which can be earthed as in the case of twisted pairs. These type of cables have bandwidths of between 0 to 300 MHz, are more expensive and more difficult to install than twisted pairs but can be used to achieve high speed data transmission with very low error rates such as 1 for every 10^8 to 10^{11} bits.

The data in a broadband cable can only travel in one direction. Therefore separate transmission and receive channels must be used by either folding the cable over itself to form a dual cable system or dividing the capacity of a single cable in half (i.e., 10 to 100 MHz for transmitting and 175 to 300 MHz for receiving) which is known as a mid split system. This requires a device called 'Head end' or a 'Central Retransmission Facility (CRF)' to modulate 'send signals' to 'receive signals' at the other end and vice versa.

Broadband cables are used to transmit a number of different analogue signals using the mid split principle by dividing a

single physical channel (figure 10.9) in to a number of frequency channels. The signals have to be modulated to form a carrier signal which can be transmitted at a different frequency and this can be done in 3 different ways.

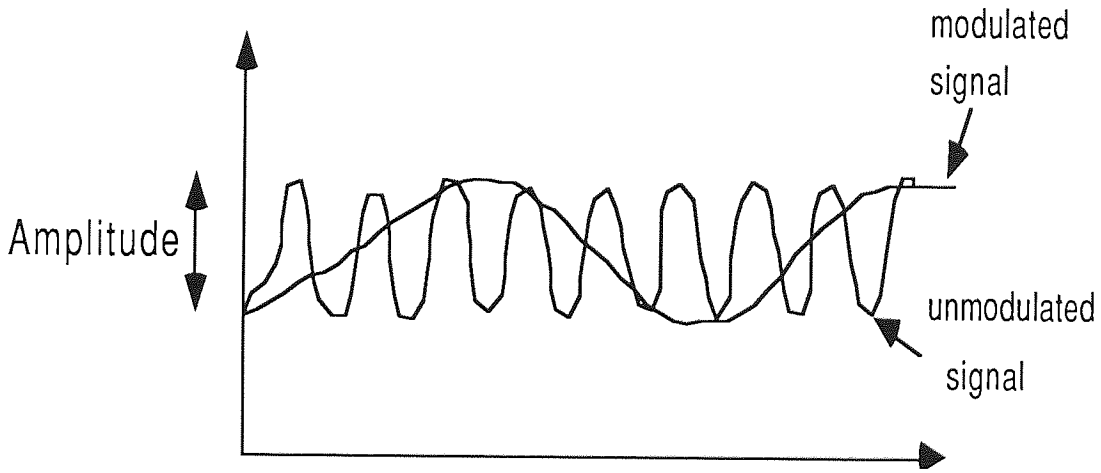


Broadband cable

Figure 10.9

10.4.2.1 Frequency modulation

The frequency of the carrier signal is altered to be sent (figure 10.10) at a different frequency while the strength and the amplitude are kept constant.

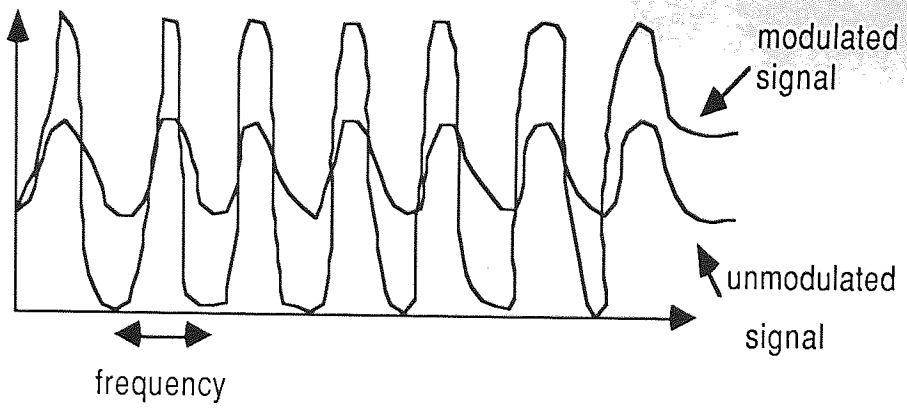


Frequency modulation

Figure 10.10

10.4.2.2 Amplitude modulation

The amplitude of the signal is altered (figure 10.11) with time without altering the frequency.

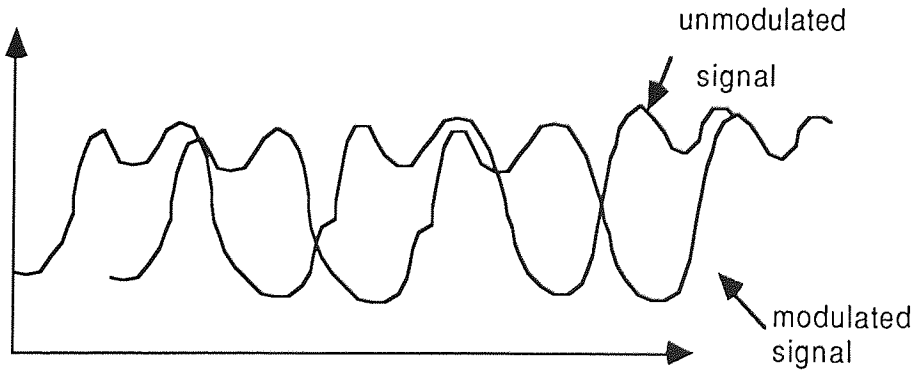


Amplitude modulation

Figure 10.11

10.4.2.3 Phase modulation

The phase of the signal is shifted (figure 10.12) without altering the frequency or the amplitude.



Phase modulation

Figure 10.12

10.4.3 Baseband coaxial cables

Baseband coaxial cables are physically similar to broadband cables with about 1 cm thickness but the essential difference of baseband transmission is that a signal can be transmitted at the original frequency without being modulated to form a carrier signal. Another difference is that a baseband signal is digital while a broadband signal is analogue.

10.4.4 Fibre optics cables

The fibre optics cable is the best bounded transmission medium which can have bandwidths of up to 3300 million Hz and can transmit data at speeds in excess of 10^9 bits per second. The error rates are in the range of 1 for 10^9 bits and the transmission is not affected by electrical or electromechanical interference. Furthermore, these cables are very small and light, thus have the capacity of hundreds of twisted pair cables.

The transmission is achieved by translating the electrical signals to light pulses by a modulator. These pulses are then

transmitted by a light source. These are detected and converted back to the original form at the destination by photoelectric diodes.

10.5 DECnet

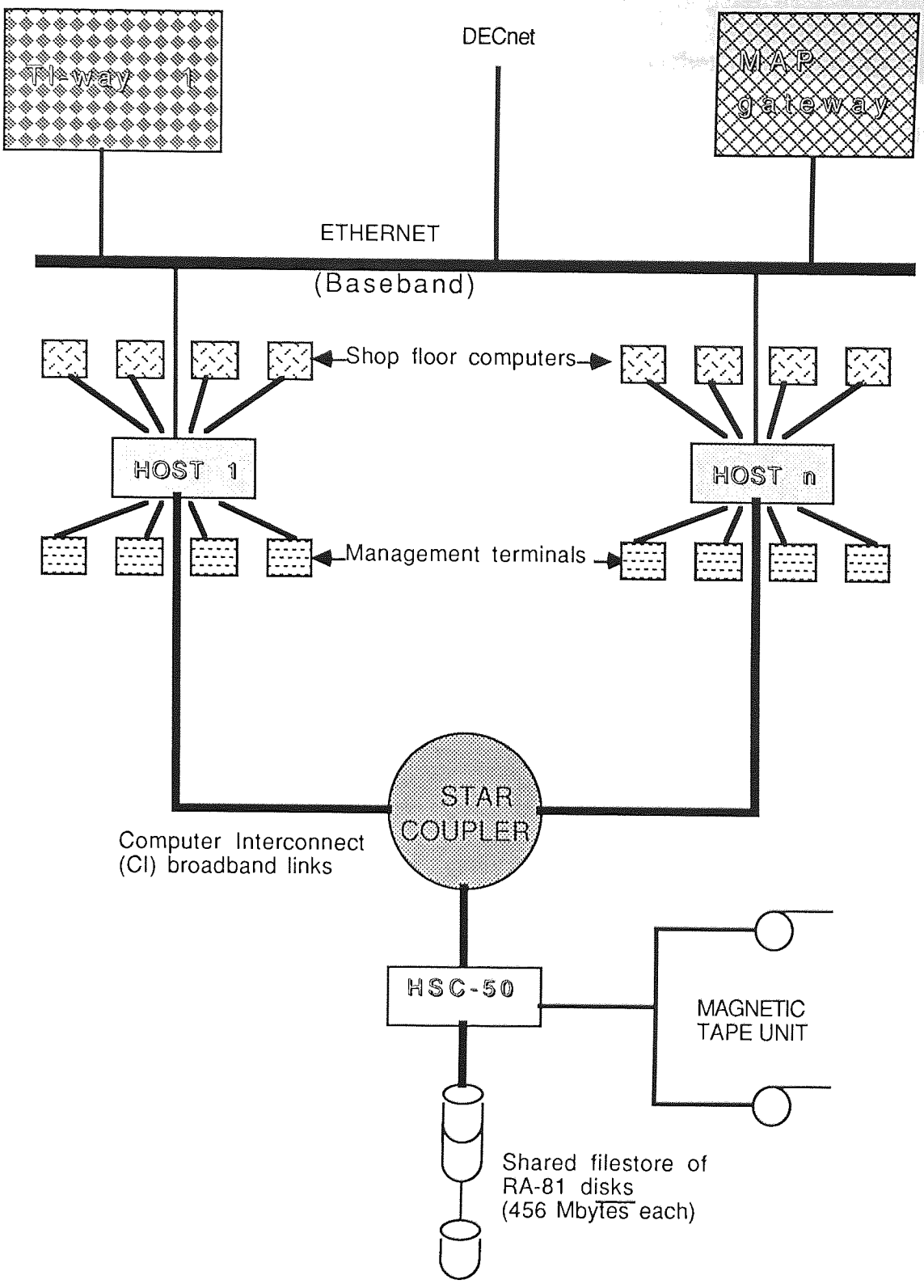
DECnet is a networking system which is used by the Digital Equipment Corporation to network their range of computers. Each node in a DECnet system has a unique address where the last two characters of this node address have to be two colons so that the address is distinguishable when used with a file name. All the files can be accessed quite easily by preceding the full file name such as 'KIRK::$DISK1:[MASTER.BAY2.LINE3]JOB5.DAT;2' where KIRK:: is the node address. The node address not necessary if the files are found in the current node.

10.6 Clustering

Clustering many similar computers together is more efficient than linking them via DECnet or any other networking protocol. The users in a cluster are able to use any processor with a centralised file store. The fact that there are many computers in the

cluster will be transparent to the users and it would appear that every user is on one computer. The software also need not have to be modified to include a node address while the policy makers of the group would have the same problem free access to all the companies as now. It is however recommended to read the manual Networks[194] for further information on networking.

All the hosts (nodes) can be linked via Computer Interconnect (CI) links which are dual path multidrop broadband links which have a 70 Mbit per second bandwidth. These can be coupled via a star coupler which serves as the common connecting point. A star coupler is a dual path RF-transformer coupled passive device which can connect up to 16 CI nodes.[195,196] These nodes can be either VAX processors or HSC50 high speed mass storage servers. The common file store which can include RA81 type disks and a tape backup unit can be connected to this HSC50 server. The full networking configuration is shown in figure 10.13 where a number of host computers (which in turn are connected to terminals and the shop floor as in the blue print 5) are connected together.



Complete system configuration

Figure 10.13

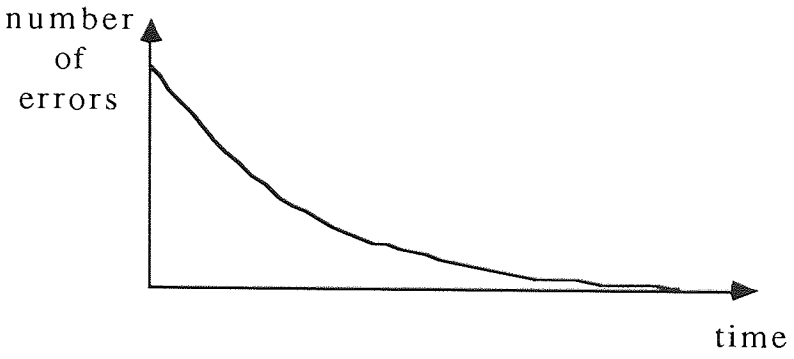
These nodes can also be connected either via ethernet or fibre optics to any other DECnet system or to a MAP gateway in the long term if the company wishes. This would increase the end to end distance[197] of the LAN (local area network) with CSMA/CD (Carrier Sense Multiple Access with Collision Detect)[198,199]

Chapter 11

Conclusions and corollary

Chapter 1 mentioned the essential requirements of the system while the design specification (appendix 4) gave more details and divided the work in to 6 phases out of which the first 5 have been fully developed. The management software was initially implemented in one subsidiary company (Sections and Profiles Ltd) and was later extended to another (Metal Ceilings Ltd) after the success of the former. As for phase 6, only one shop floor computer has been installed so far and more such computers will be included in the system as and when the company's finances permit.

Up to now several company personnel have been trained to use the system who have tested the software and hardware thoroughly by daily usage. Minor problems were discovered in the system as a result of such testing, as would be expected in any new system (figure 11.1). These were eliminated and more flexibility was introduced in the process. Sections and Profiles Ltd are currently doing away with the old paperwork system which has been running in parallel with the automated system during the development stage.



Error curve

Figure 11.1

One of the major achievements of this research is the usage of the design techniques which were described in chapter 3. The technique 'Fantasy' was particularly applied so that the shop floor part of the system could be built without considering it's limitations or the practicability of integrating it with the rest of the system. The integration was attempted only after both the shop floor and the management phases were completed and this effectively showed the advantage of a systematic, decision tree type of an approach. The other contributions to knowledge which were made as a result of this research are as follows.

(1) A new method was used in data base management which eliminated the problem of limiting the variable sizes in order

to retrieve them. This method allows the user to retrieve any record by supplying only a few characters of a variable. The looping procedure for this is also new and it enables him to examine a group of items at once without selecting each item individually.

(2) One of the main features in the data base management system is that no data is scrolled off the screen at any time unless instructed by the user. This is a valuable lesson to any potential programmer. The user is also allowed to toggle between different screens of data should he wishes. Another unusual feature is that the system allows the user to specify the required information in the production report at times a full report is not required.

(3) The technique which has been used for automatic updating of data by cross linking different modules is quite out of the ordinary and this was also extended to the user so that he is presented with all the incomplete records when he enters a module.

(4) A new technique has been used for maintaining the age of all stock and this allows the users to see the value and the age of all stock at a glance. This was extended to place invisible markers against different 'live' items to prevent them from being lost.

(5) The fast, direct access format has been used for storage and the method which has been used for locating records is very efficient because the catalogue information is maintained in the first record of all the files.

(6) New techniques have been used for scheduling and they take all the real time factors such as holidays, week ends, early/late hours and gaps between jobs in to consideration ignoring the limitations of the operating system. Furthermore, computerization of the line preference data allows the jobs to be scheduled on to the most economical lines.

(7) Research has been made in the area of tied jobs, and the new method which was used eliminates the setup and strip times where not necessary and thus increases the profitability.

(8) A new but a very simple method was used for the collection of data on the shop floor by generating a train of pulses and this method not only allows to calculate the actual production speed (which is more useful than the rolling speed of the mill) but also allows to count the number of produced units automatically.

(9) A new method has been used for choosing between different reasons of downtime and a new category called 'Downtime without a reason' has been introduced for extra efficiency.

(10) The technique which was used to utilize the existing resources of the VMS system in order to build the directory structure is also new and is quite efficient. This technique eliminates unnecessary duplication of files and allows the same set of image files to be used throughout the group while maintaining separate data files.

(11) The totally new techniques have been used for security and they are so powerful that different users are allowed different operations according to the pre specified privileges. It

was also explained that even the VMS system manager with his full set of privileges can not 'break in'. It can therefore be quite confidently stated that this system is more fool proof than any other comparable system.

(12) One revolutionary feature of this research is the data transfer between totally incompatible computers which was initially criticized as impossible by many people. The technique which was used is a very simple but a powerful technique which evolved as a result of this research and did not fail even in the presence of 4 entirely different software languages (Fortran 77, Basic, DCL and 6502 assembler) in the system although used simultaneously. This can be mentioned as a major contribution to knowledge.

It must also be mentioned that the formats of some manual reports were also used in the computerized system so that familiarity is introduced in order to eliminate the fear and dislike of working with a strange system. As a result, the system was firstly accepted by the users and then it became very popular amongst them before eventually becoming essential in their day to day work.

The system currently has 28 different users (appendix 24) including 4 shop floor computers for each subsidiary company.

The design specification became an invaluable aid during the research and each and every requirement of it has been fulfilled. The simplicity of operations has been given a high level of priority and the system has been designed to consist of menus and user friendly instructions throughout while utilities like backup, recovery and error elimination routines are provided, as already explained. All the help and documentation is already included in the system which consist of a number of stand alone modules for cross referencing which give brief descriptions of all the software programs (appendix 25), the subroutines which have been used (appendix 26), the variables in the programs (appendix 27) and the common blocks (appendix 28). The names of these programs are 'DIRECTORY', 'SUBROU', 'VARIAB' and 'BLOCK' respectively. The listings of all programs (Appendix 29) are also included. Data for a dummy company has also been provided in the software so that the personnel can check anything without corrupting the actual data in the files.

The design specification however is by no means a full and a complete description of the required system. This became apparent when the need for extra sections such as stock control and press data were discovered at a much later stage. As a result, the project could not be completed within the planned time limit. Furthermore, some purchased equipment was discovered to be faulty. These problems were unforeseen at the time of preparing the time schedule. Considering these problems and the volume of the previously unplanned work which had to be carried out, the time delay of only a few weeks is in fact a great achievement.

Every module of software has been developed to the highest professional standard as opposed for a purely educational purpose while every aspect of the research programme also has been planned out very carefully in order to produce a complete useable system with no loose ends. The only possible areas of future work are therefore to extend the system to finance and accounting (via the module 'periodical working details' which has already been prepared) and to link this system with the existing roll design and post processor programs so that the data from this database can be used for all the work in the company. One other small area can be to

implement the monitoring computers for each subsidiary company between the host and the process computers as shown in figure 4.13. Great care must however be taken in carrying out extensions and it is strongly recommended that only the highly professional personnel to be allowed to do such modifications.

The software must however be slightly modified in the event of adding more production lines to a company to include the new destinations of data. The software has been designed in such a way that these addresses are taken as variables in lines 20, 30 and 40 to be used in many places subsequently. The variable 'BAY' in line 20 points to the company number as in the company file while the variable 'LIN' in line 30 refers to the line number and the variable 'SUP$' in line 40 refers to the appropriate supervisory terminal on the host which the alarm messages are to be sent. Adding another company name is also quite simple and can be done at any time by using the module 'COMPANY'. The number of the company which is generated by the computer can be used in assigning users.

The next logical step of research, although not as an extension to the project, is to extend the technique of data transfer which was used in this research to form a general model to transfer data between any two computers rather than as a dedicated means of data transfer between two pre specified computers. Another area of future research can be to transfer not only data and text but also diagrams between different computers. A hint which can be given is, first to transform each pixel of the diagram to a format which can be passed on to another computer using the technique which was used in this research so that the second computer can translate those codes back to a picture on it's own screen, memory or filing system.

System security can be another area of research where all the existing facilities of the VMS system can be used. At present only a few users seem to know most of the facilities in the operating systems which they use and this would be a good opportunity look in to operating systems deeply, in order to write more efficient software in all areas.

Appendix 1

ASCII Codes

Code	Action	Description
0	NUL	null
1	SOH	start of heading
2	STX	start of text
4	EOT	end of transmission
5	ENQ	enquiry
6	ACK	acknowledge
7	BEL	bell
8	BS	back space
9	HT	horizontal tab
10	LF	line feed
11	VT	vertical tab
12	FF	form feed
13	CR	carriage return
14	SO	shift out
15	SI	shift in
16	DLE	data link escape
17	DC1	device control 1
18	DC2	device control 2
19	DC3	device control 3
20	DC4	device control 4
21	NAK	negative acknowledge
22	SYN	synchronous idle
23	ETB	end of transmission block
24	CAN	cancel
25	EM	end of medium
26	SUB	substitute
27	ESC	escape
28	FS	file separator
29	GS	group separator
30	RS	record separator
31	US	unit separator
32		space
33	!	
34	"	

Code	Action
35	£
36	$
37	%
38	&
39	'
40	(
41	)
42	*
43	+
44	,
45	-
46	.
47	/
48	0
49	1
50	2
53	5
54	6
55	7
56	8
57	9
58	:
59	;
60	<
61	=
62	>
63	?
64	@
65	A
66	B
67	C
68	D
69	E
70	F
71	G
72	H
73	I
74	J
75	K

Code	Action
76	L
77	M
78	N
79	O
80	P
81	Q
82	R
83	S
84	T
85	U
86	V
87	W
88	X
89	Y
90	Z
91	[
92	\
93	]
94	^
95	_
96	`
97	a
98	b
99	c
100	d
101	e
102	f
103	g
104	h
105	i
106	j
107	k
109	m
110	n
111	o
112	p
113	q
114	r
115	s

Code	Action	
====	======	
116	t	
117	u	
118	v	
119	w	
120	x	
121	y	
122	z	
123	{	
124	|	
125	}	
126	~	
127	DEL	delete

Appendix 2

Computer operating systems

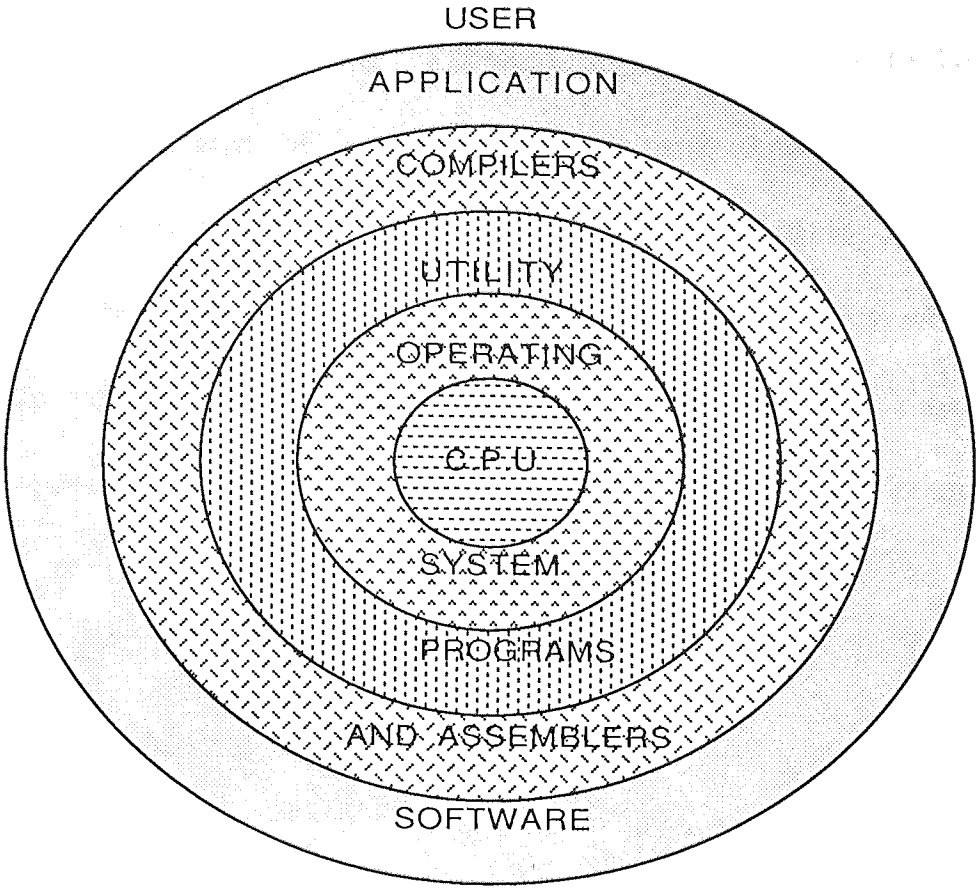
The operating system is the root of a computer's software. It is the operating system software which allows a user to interact[84,85] with a computer while it is not some type of a compiler.[86] Katzan[87] defines it as an organized collection of programs and data that is specifically designed to manage and utilize the resources of a computer system and to facilitate the creation of computer programs and control the access to that system. An operating system therefore looks after the inputs and outputs, messages, data storage, screen management, resource allocation, processing and various other tasks. This can be shown in a layered format[88] (figure A2.1) with the user in the outermost layer.

The operating systems can be classified in to 3 broad categories depending on the specific required applications as follows.

(1) Batch operating systems

(2) Time sharing operating systems

(3) Real-time operating systems



Layered representation of a computer system

Figure A2.1

A2.1 Batch operating systems

The jobs are not processed in the interactive mode in batch operating systems. The users first have to prepare the jobs and submit them for processing. The operating system will then form streams of these jobs in to batches of work which are processed in the best possible manner.

Normally the batch jobs are submitted as a deck of cards and have to be collected after a period of time. This has to be done even for compilation of a source program and therefore is a time consuming process and is not suitable for a real time application. However it is very useful for processing very long jobs where the users do not have to wait at a terminal. The computer time in a batch operating system is utilized in the most effective manner as opposed to other types of operating systems.

A2.2 Time sharing operating systems

The basic function of a time sharing operating system is to distribute computer time to a set of terminals in such a way that it seems to each user who is logged in via those terminals, that he has a dedicated computer which is providing him a service although in actual fact the computer attends to all the users' requests in turn.[89]

The terminals are multiplexed to the computer and each port or input/output line has to be serviced (polled) regularly without discrimination. Therefore the system may not utilize it's

hardware effectively. This type of a system is most suitable for short jobs.

A2.3 Real time operating systems

A real time operating system is essential when the processing has to be done within a rigid time limit. This processing activity is typically triggered by randomly accepted events and these events will receive data requests for service from an on line device[90] which is not operated by a human. The operating system can in this case request data directly from the device.

This type of an operating system is most suitable for real time on-line tasks where a minimum amount of human intervention is required. The computer can be a dedicated device which will carry out these tasks continuously and specialised interfaces must be available between the computer and the process as discussed above and the computer hardware too can be large or small depending on the task.

Microprocessor based systems are mainly used in such a system in many industrial applications and these can function over long periods of time without change. The computer sometimes become transparent to the user where the application program can reside in an EPROM (Erasable Programmable Read Only Memory). Usage of microprocessors this way is much cheaper than employing a conventional hard wired controller.

The most essential feature in a real time operating system is the quick response time and therefore all the data has to be on-line while all the on-line programs must be in an executable form.

The three types of above described operating systems have their advantages and disadvantages while each of them is suitable for a different application. Since the types of operating systems and the requirements are known, Morphological analysis can be used to list out these and select the most suitable combination. The table A2.1 has a list of popular operating systems and the relevant details.

System	Description	Usual machine	Application
Unix	———	IBM, ICL, DEC	single user + time sharing
MS/DOS	Micro Soft/ Disk Operating System	IBM compatible	single user
VOS	Vulcan Operating System	Harris	time sharing
VMS	Virtual Memory System	DEC compatible	time sharing
ELN	External Local area Networks	DEC compatible	Real time
MOS	Machine Operating System	Acron	single user
CP/M	Control Program/ Monitor	IBM, DEC, Amstrad	mainly single user
P/OS	Professional/ Operating System	DEC	single user

Operating systems

Table A2.1

A2.4 Unix

Unix operating system was developed at Bell laboratories[91] and is a derivative of the old Multics operating system. Unix at the early stages was only a single user system but now multi user systems often use Unix. The user work is carried out by a single sequence of events and this consists of some computer memory and the files being accessed. Another process is created by a copy of the process being made[92] and the two processes are only distinguished by the parent being able to wait for the child process to finish. A process may replace itself by another program to be executed. A unique feature of this operating system is that the priorities are assigned dynamically. A command language called 'SHELL' allows the user to interact with Unix.

A2.5 DOS

DOS is the abbreviation for Disc Operating System and is sometimes known as MS/DOS where MS stands for Micro Soft who are the manufacturers. This operating system is common in small IBM compatible computers like Apricot or Amstrad and the

operating system is kept on a floppy disk in the absence of hard disks. This operating system is suitable only for single user environments as the system start up procedure has to be automatic in a multi user operating system.

A2.6 VMS

VMS or the Virtual Memory System is quite popular on multiuser VAX machines which are manufactured by the Digital Equipment Corporation (DEC). The user has to interact with this system via a command language called DCL which is the abbreviation for Digital Command Language.

The principle of a virtual memory system is image invokation.[93] i.e., running a program from the scratch. The disadvantage of such a system is the large number of different programs which run in quick succession causes a severe degradation of response time. Each new image which is called for execution must be loaded to the physical memory to be executed and this will take a long time specially when the files are scattered all over the disc. This limits the usage of menu driven programs which call other menus to

run different programs. It has a range of privileges from 1 to 255 and these can be used to give preference to some jobs[94] over the others. The processing becomes even worse if privileges are increased for some jobs because the others will virtually come to a standstill. Therefore this system is not very suitable for real time applications.

A2.7 VOS

VOS is the abbreviation for Vulcan Operating System and it is found in Harris multi user computers. It is a multi-use, multi-programming and multi-lingual time sharing operating system. It supports up to 128 software priorities[95] and they are used by the operating system to determine which program or terminal will have control of the central processing unit at one time. The high priority jobs are executed before the others and all the equal priority jobs are executed by the technique of 'time slicing'. i.e., a small unit of time is allocated to each program and control is passed on to another program whenever this is used up. This goes on in a cyclic fashion but is transparent to the user. This too uses the principle of virtual memory.

A2.8 ELN

ELN or External Local area Networks operating system is also found on DEC's VAX range of machines. This is one of the very few real time operating systems which are available today. The manufacturers recommend it to be used with real time on-line systems[96] which run on VAX and MicroVAX processors. Unfortunately this is one of the new systems and the information which is available is not sufficient to have a full understanding about the system.

A2.9 MOS

The letters MOS stand for Machine Operating system and this is a very flexible operating system which work on Acorn's range of computers.[97] The access times are very fast because the operating system is kept on a PROM and one of the major advantages is that it can be accessed via service entry points through any language or program. The disadvantage is that it is available only in a single user range of micro computers.

A2.10 CP/M

CP/M is the abbreviation for Control Program/Monitor which uses the old name 'monitor' for a program. It is very popular today and is found on micro and small mini computers. CP/M is useful for data processing and management activities but is normally found on floppy disks[98] on small computers.

A2.11 P/OS

This operating system is found on DEC's 'Professional' range of computers[99] and is appropriately named Professional Operating System. This too is found on floppy disks in the Professional 325 computers and the system disk has to be inserted to start the system. It is found on a hard disk in the higher version (350) but the system initialisation is very slow. This too is suitable for a single user system.

Appendix 3

Computer languages

The computers handle data in a different way to the humans. The data in computers is handled as a series of ones and zeros. These are called binary digits or bits. The instructions within a computer in this form are called machine code instructions. It is very difficult for a programmer to provide instructions to a computer using the machine code as the programming errors or bugs will be very difficult to locate.

The assembly language programing is an easier way of programming as the instructions of the assembly language which are called mnemonics are like parts of English words. An assembler is a pre programmed code which resides in the computers and interprets assembly language. When a programmer writes a piece of code in assembly language it is interpreted or assembled by the resident assembler program in to a form of ones and zeros which can then be executed by the computer. this type of programming is called low level programming.

The usage of a language to program a computer is called high level programming and this type of programming is much easier than any other type due to the number of instructions which are available to the programmer. Most of these instructions are simple English words[101] and therefore error detection is easy. These programs will first have to be converted to a set of machine code instructions before being executed and this process is very similar to the usage of the assembler.

There are two ways of interpreting the instructions, the first being interpreting each line of instructions just before it is executed which takes relatively longer execution times and the second is to interpret the source program completely to form a set of executable machine code instructions. Obviously the execution time of the latter is quicker[102] because the instructions do not have to be interpreted as they are executed. Such a case is referred to as a compilation and the interpreter is called a language compiler. Any errors in the source program will halt the compiler in this case while such errors will be detected only when the corresponding line of instructions have been reached for execution in the earlier case. A language therefore is an accepted set of instructions which have to

be translated to machine code by using a previously written program.

Morphological analysis can once again be employed to look at the available languages and to select the most suitable one by comparing each language with the combination of requirements.

A3.1 Fortran

Fortran which is known as a scientific language was developed in the late 1950s by IBM workers[103] and was intended to perform FORmula TRANslation. Development of Fortran was the biggest step in the development of computers as the majority of programming was done using assembly language or machine code before Fortran.

There are different versions of Fortran, the earliest being Fortran 4. This was replaced by Fortran 66 which in turn was replaced by Fortran 77 later. Fortran 77 consists of a main program and a number of subprograms and it contains the essential features of a high level language such as GO TO statements to transfer control

to a different part of a program, IF ...THEN ...ELSE conditional statements and DO loops.[104] Fortran is normally a compiler based language and it is stated that Fortran is easily transportable.

There are some disadvantages in Fortran which show that it is rather an ageing language when compared with more modern languages. One of the disadvantages is that each character variable has to be declared at the beginning of the main program and all the subprograms must indicate the length of the character string. Any string which is shorter than the declared value is represented with spaces which have ASCII values of 32 rather than with null characters while any string which is longer than the declared value is truncated. This results in erroneous values in string conversions. Even the variables which are declared as integers can not be input as real values because the program simply crashes if such an attempt is made. This makes Fortran unsuitable to be used in a shop floor environment where many users are not computer literate.

The biggest disadvantage in Fortran is that every variable has to be declared as 'common' in one or more blocks if it has to be passed from a subprogram to the main program. The

variables in a single block have to be in the same order in both the main program and the subprogram.[103] This annoying procedure makes it almost impossible use a lot of variables in a subroutine and therefore puts the programmer off using subroutines. The usage of Fortran is often unnecessarily lengthy and complicated for the experienced programmer as well as for the inexperienced one.

A3.2 Basic

Basic is by far the most popular language in micro computers[105] but not to that extent in larger computers. It too has the essential features like different types of loops, conditional statements, subroutines, transfer of control statements and a host of other utilities. When used in a micro, it can not only read the contents of a memory location but also write to any location directly, although this is impossible in a multi user system because of the system protection procedures.

Unlike Fortran, Basic allows the user to mix all variables other than character strings which are indicated by a '$' at the end of the variable name. The lengths of these however do not have to be

declared and therefore the complete string is saved as it is regardless of it's length. All variables which are used in the main program and the subprograms are regarded as common unless declared as local in any order. This eliminates the problem of common statements.

Basic can be found either as a compiler or an interpreter. The biggest disadvantage of Basic is the difficulty in transportation between different machines because of the number of versions which are available. Basic is regarded as a poor language in some circles and the main argument against it is that it is lacking suitability in developing programs in a systematic way and that it tends to produce programs that lack structure.(105) This in the author's opinion is a very poor argument because the systematic programing depends on the ability of the programmer. In a session where Dephi technique was used, more than 50% of the members claimed that Basic is not a suitable language but none were able to substantiate their claims when requested to do so. It was also found out that most of the claiments did not know programming in Basic while some others knew a little.

A3.3 Pascal

Pascal was developed in the Federal Institute of Technology, Zurich[105] and was intended as a simple language which is suitable for teaching purposes.[104] This is one of the reasons why Pascal is so popular in educational establishments but not to that extent in the industry. Pascal however is strongly supported to replace Basic as it is claimed that Pascal allows program development in a systematic way.

Pascal too can be obtained either as a compiler or a line by line interpreter and is available in most microcomputers as well as in larger machines. It also has a number of variations such as S-Pascal and Iso-Pascal. One of the biggest advantages of Pascal as opposed to other languages is that Pascal allows the separate definition of data types to be given by the programmer. Individual variables of this type may then be declared by giving only the name of the new type without replacing the definition. Another important feature is that a subroutine can be called within the same subroutine.

A3.4 Forth

This too is a strong contender to replace Basic for the simple reason that the programs which are written in Forth occupy less memory and run faster than the equivalent programs in Basic. The disadvantage of Forth is that it's different style of programming when compared with other conventional languages. This and the shortage of documentation makes Forth a less used language.

A3.5 Lisp

Lisp is the abbreviation for List Processor language and was first designed by John McCarthy at the Massachusetts Institute of Technology in 1960.[104] While Fortran, Basic, Pascal and Forth can be termed as general purpose languages, Lisp can be labeled as a special purpose language[105] which is intended for a specialised area[102] such as artificial intelligence.

The main difference between Lisp and the other languages is that the equivalence of programs and data which allows data structures to be executed as programs and programs to be

modified as data. Lisp is equally popular in academic institutions and the industry for heavy mathematical work.

A3.6 Cobol

Cobol or Common Business Oriented Language has been used since 1960s for business applications. It is widely used for business data processing, payroll and accounts while it gives a high priority for input output files which are most important in business. Another important characteristic is that it contains syntax which are closely related to English words which makes Cobol self documented. Cobol has not been known to be used for scientific or engineering purposes.

A3.7 Prolog

Prolog originated in the University of Aix-Marseilles in France in 1972 and the name Prolog is a contraction of the words Programming logic. Prolog is also a special purpose language like Lisp and is mainly used for artificial intelligence and can not be considered as a general purpose language.

A3.8 Algol

Algol was developed at the same time as Fortran and it has 3 versions to date namely; Algol 60, Algol 68 and S-Algol. An essential feature of Algol is that it is not a card or a line orientated language. i.e., one statement may occupy more than one card and more than one line may appear on one card.[102] Algol is not a very popular language today.

A3.9 Snobol

Snobol was originated in the Bell laboratories in 1972 and the latest version of it is Snobol4. Snobol4 shares with Lisp, the capability of translating and running programs which have been either constructed or read as data. Snobol is mainly used for scientific data processing and simulation purposes.

Snobol4 has an extensive selection of data types of which 'patterns' are exclusive to this language. The type of a variable need not be declared in the program and each variable name may refer to different objects of data at different points of a program. Multi

dimensional arrays and tables are also provided and the data type of each is allowed to vary dynamically at run time. Despite these advantages Snobol is not known to be widely used in the engineering industry.

A3.10 PL/I

PL/I is a large multipurpose language which was designed during the early 1960s by a committee organized by the IBM corporation. and the name PL/I is derived from the sequence of words 'Programming Language/IBM'. The aim of PL/I was to act as a successor to Fortran which did not have the capabilities of data structuring and a sophisticated operating environment.

Pl/I is applicable to a wide range of areas such as scientific and business applications but it seems to be somewhat analogous to Fortran with the formatted input/output, separately compiled subprograms and common blocks. It is understood that PL/I is more suitable for a less experienced programmer but it did not become popular as expected or a true successor to Fortran.

A3.11 APL

APL which is the abbreviation for A Programming Language was implemented at IBM's Watson Research centre on an IBM 360/50. APL is normally used as a time sharing system and it interacts with the user in a conversational style which allows the programmer to use it dynamically.

A useful feature in APL is that there is no concept of writing a main program. The subprograms can be called either from another subprogram or by the user via his terminal. The control therefore can be passed back and forth between the user and the program. Like many other languages which were discussed above APL too is not known to be used in the engineering industry to a great extent.

A3.12 Ada

Ada is a relatively new language which was developed in the 1970s by the American Department of Defence in order to replace several different languages which were being used. It was

intended for the use of large and diverse range of applications but the language itself turned out to be diverse.(106)

It has all the features of variable handling such as arrays, integers and strings while being capable of making calls to subprograms. The biggest disadvantage of the language is being not very popular and it can even be said as 'unknown' with respect to the engineering industry although it is available on VAX machines

Appendix 4

The design specification

A4.1 Introduction

An automated Computer Integrated Monitoring and production management system is required at Hadley Industries, Downing street, Warley, Smethwick to increase the overall efficiency in the production of cold rolled sections. The work is to be carried out over a period of two years and it consists of the following phases.

A4.2 Phase 1

Create a data base for job creation, historical product information including targets and overall production reports to suit the exact requirements of the company by designing software.

A4.3 Phase 2

Design an on line real time monitoring system which will feed automatic control signals which relate to speeds, and manual

signals from shop floor terminals which relate to different categories of downtime to a monitoring computer.

A4.4 Phase 3

Development of a system for data transfer communications between a host and a monitoring computer.

A4.5 Phase 4

Provision for three more systems as above to suit four parallel lines on the shop floor to work simultaneously and software design on scheduling for all four lines.

A4.6 Phase 5

System stabilization with back up power, alarms and corrective action (see special requirements) without affecting the degree of efficiency.

A4.7 Phase 6

The cost effective implementation of the system.

A4.8 Related documents

(1) Feasibility study on a computer aided production monitoring system ref 6013 - by Factory data systems[200]

A4.9 Units of measurements

Milimetres to be used as the shop floor measuring system of length but provide a facility to convert imperial units to milimetres only when a customer orders in imperial units.

Group production director

Group technical director

Production manager, Sections and Profiles

Production manager, Metal Ceilings

A4.11 Conditions and requirements

(1) The software is to be developed on a VAX computer and language to be used is FORTRAN 77

(2) The software has to be developed in such a way that some existing software on stock control can be linked.[115] Utilities like customer information and targets must be developed as separate programs but the main program should be able to extract any information from them.

(3) The monitoring system should be capable of monitoring all four production lines simultaneously.

(5) The operator should be able to send the reason via a shop floor terminal which is dedicated to that particular production line when a line is not operating.

(6) The calculation of the actual rolling speed of a line should not be affected by any stoppage.

A4.12 Performance and characteristics

(1) The software has to be user friendly and should have sufficient help and information for an untrained person to operate.

(2) The monitoring system should operate constantly, even if the display mode is different.

(3) The instructions should be clearly visible.

(4) All operations of the system should be silent and should not affect the productivity in any way.

(5) Production reports should be made automatically by using the information in a particular job or a batch of linked jobs as per instructions.

A4.13 Maintenance and reliability

(1) The hardware to be serviced as per instructions by the manufacturer.

(2) Reliability to be regarded an important feature, and the shop floor terminals must be very rugged to have a reasonable life under the shop floor environment.

(3) The degree of efficiency must be substantially higher than that of the existing system.

(4) The shop floor terminals should be able to withstand any shocks and vibrations. They should also be able to withstand an impact of 2N approximately.

(5) The shop floor computers should work independent of the host at all times except only for the transfer of data.

A4.14 Special requirements

(1) The system should have visual and audible alarms in case of,

(A) speeds being lower than a pre-determined value

(B) setup and strip times being higher than preset targets

(C) over produced quantities

(2) A back up power supply should exist and switch itself on in the case of power failures.

(3) A facility should exist to obtain hard copies of information when required.

(4) Networking need not be done with compatible computers.

(5) The system should have a fool proof security system in which different levels of access is available to different users depending on their seniority.

A4.15 Life span

The effective expected life must be at least 10 years.

A4.16 Safety

The system should be safe to be operated by any authorised person without losing any information and the system should be identify all users for this purpose.

A4.17 Measure of values

All requirements which are rated below (table A4.1) give the relative importance of themslves. The greater importance is indicated by the higher value.

simplicity and easy operateability	- 22
reliability	- 18
security	- 14
cost effectiveness	- 11
storage of information	- 10
durability	- 7
speed	- 6
visibility	- 5
instructions	- 4
size	- 3

Measure of values

Table A4.1

Appendix 5

Critical Path Analysis

Table A5.1 lists all the different events of the planned project and figure A5.1 shows how these are connected. The easiest way to find the critical path is to divide the critical path diagram in to 5 sections as follows. Then the critical path will be the sum of the critical paths of all the sections.

A5.1 Section 1 - Events A to D

1,2 = 1.5 months

1,3 = 1.5 months

1,4,5 = 2 months

The critical path of this section is 1,4,5 with a duration of 2 months.

A5.2 Section 2 - Events D to E

The only path is 6 which is the critical path and the duration is 0.5 months.

A5.3 Section 3 - Events E to L

7,9,11,15,16,17	= 5.5 months
8,10,11,15,16,17	= 6.75 months
8,12	= 2.0 months
8,13	= 2.0 months
8,14	= 2.0 months

The critical path for this section is 8,10,11,15,16,17 with a duration of 6.75 months.

A5.4 Section 4 - Events L to N

18	= 0.5 months
19,20	= 1.0 months
21	= 1.0 months

This section has two critical paths namely 19,20 and 21 and the duration on both paths is 1 month.

A5.4 Section 5 - Events N to Q

The only path of this section is 22,23,24 which has a duration of 8 months.

The summation of the critical paths of all the sections give two critical paths for the whole project as follows

(1) 1,4,5,6,8,10,11,15,16,17,18,22,23,24

(2) 1,4,5,6,8,10,11,15,16,17,19,29,22,23,24

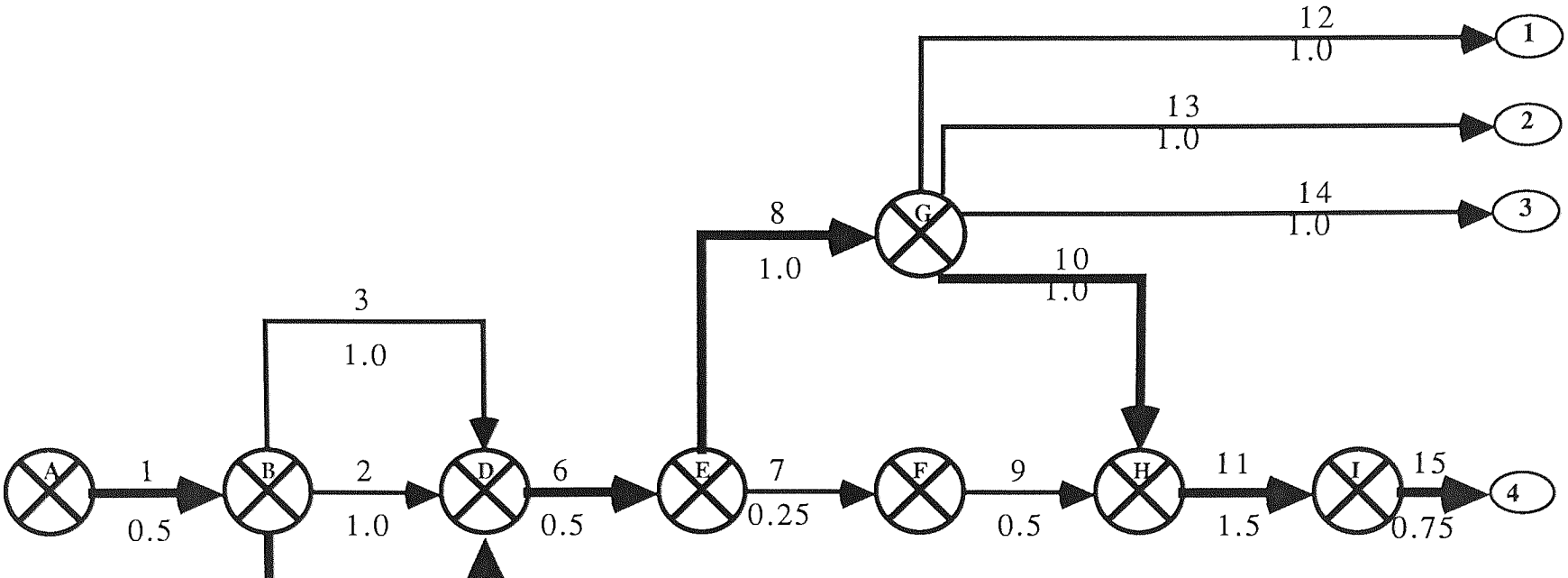
The total duration is 20.25 months.

According to the critical path analysis, the minimum period of time to complete the project is 20.25 months, providing that a sufficient number of people engaged in it in order do the possible parallel tasks at the same time. This will not be possible with only one person doing all the work since all the tasks will have to be attended one after the other. Therefore the minimum number of months to complete the project will be 24 as in the time schedule; and this critical path analysis will be used only to determine the best way to carry out the work rather than to determine the duration.

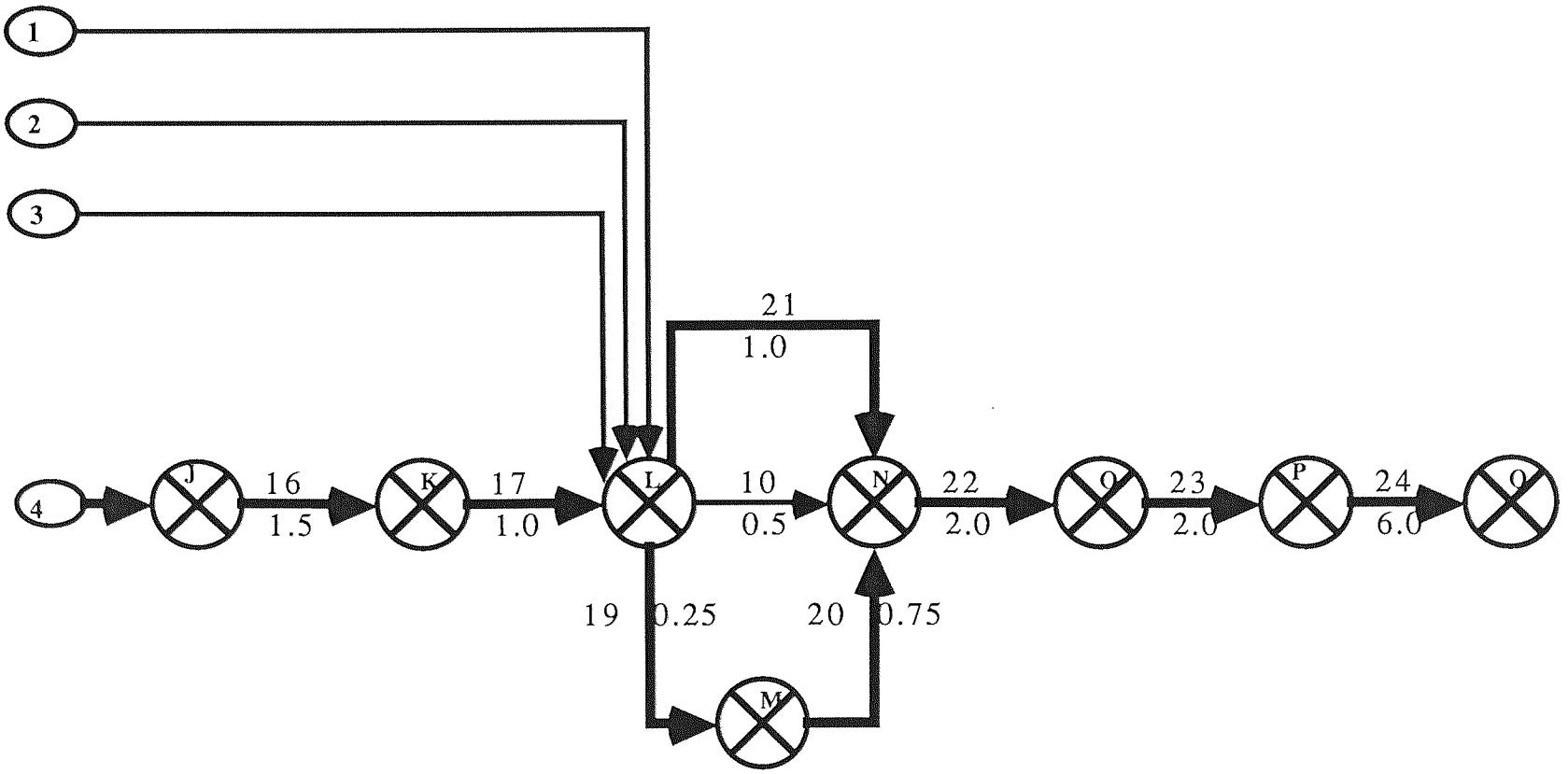
Activity	Description	Duration (months)
1	Establishment of need	0.5
2	Background study	1.0
3	Fortran programming	1.0
4	Goal recognition	0.5
5	Systems study	1.0
6	Initial ideas and market research	0.5
7	Background study on "on-line" systems	0.25
8	Jobcard creation	1.0
9	Study on sensors and monitoring	0.5
10	Target files	1.0
11	Monitoring system	1.5
12	Cost and sales pricing files	1.0
13	Market research on production reports	1.0
14	Customer data files	1.0
15	Hardware	0.75
16	Shop floor terminals	1.0
17	Data transfer and networking	1.0
18	Scheduling	0.5
19	Production report completion	0.25
20	Automatic target feed and comparisons	0.75
21	Staff training and system management	1.0
22	Feasibility study on implementation	2.0
23	System implementation and security	2.0
24	Corrections and report	6.0

Activities and durations

Table A5.1


A
B
C
D
E
F
G
H
I
1
2
3
4
1
0.5
2
1.0
3
1.0
4
0.5
5
1.0
6
0.5
7
0.25
8
1.0
9
0.5
10
1.0
11
1.5
12
1.0
13
1.0
14
1.0
15
0.75

continued ...



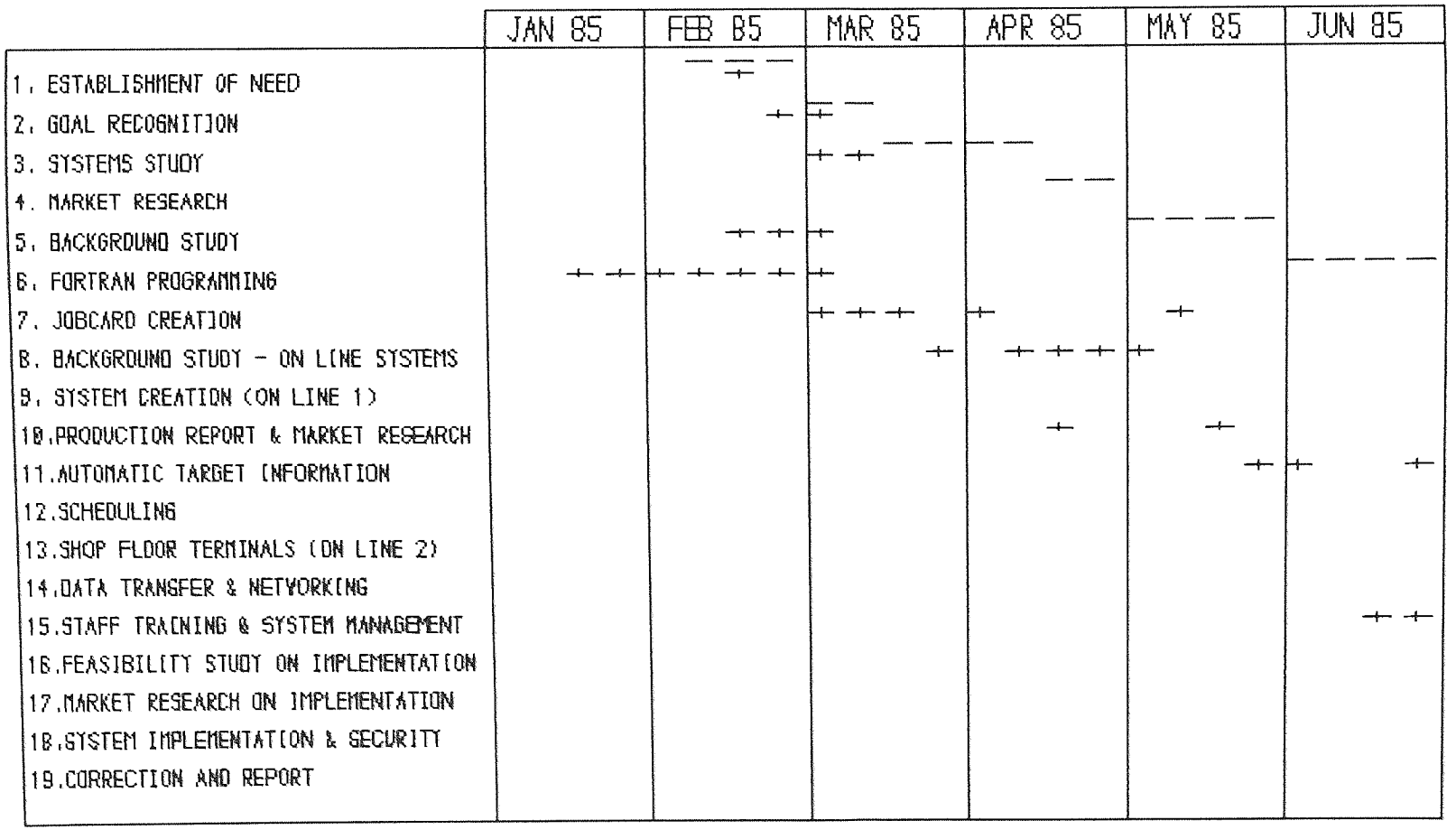
Critical path analysis

Figure A5.1

Appendix 6

The time chart

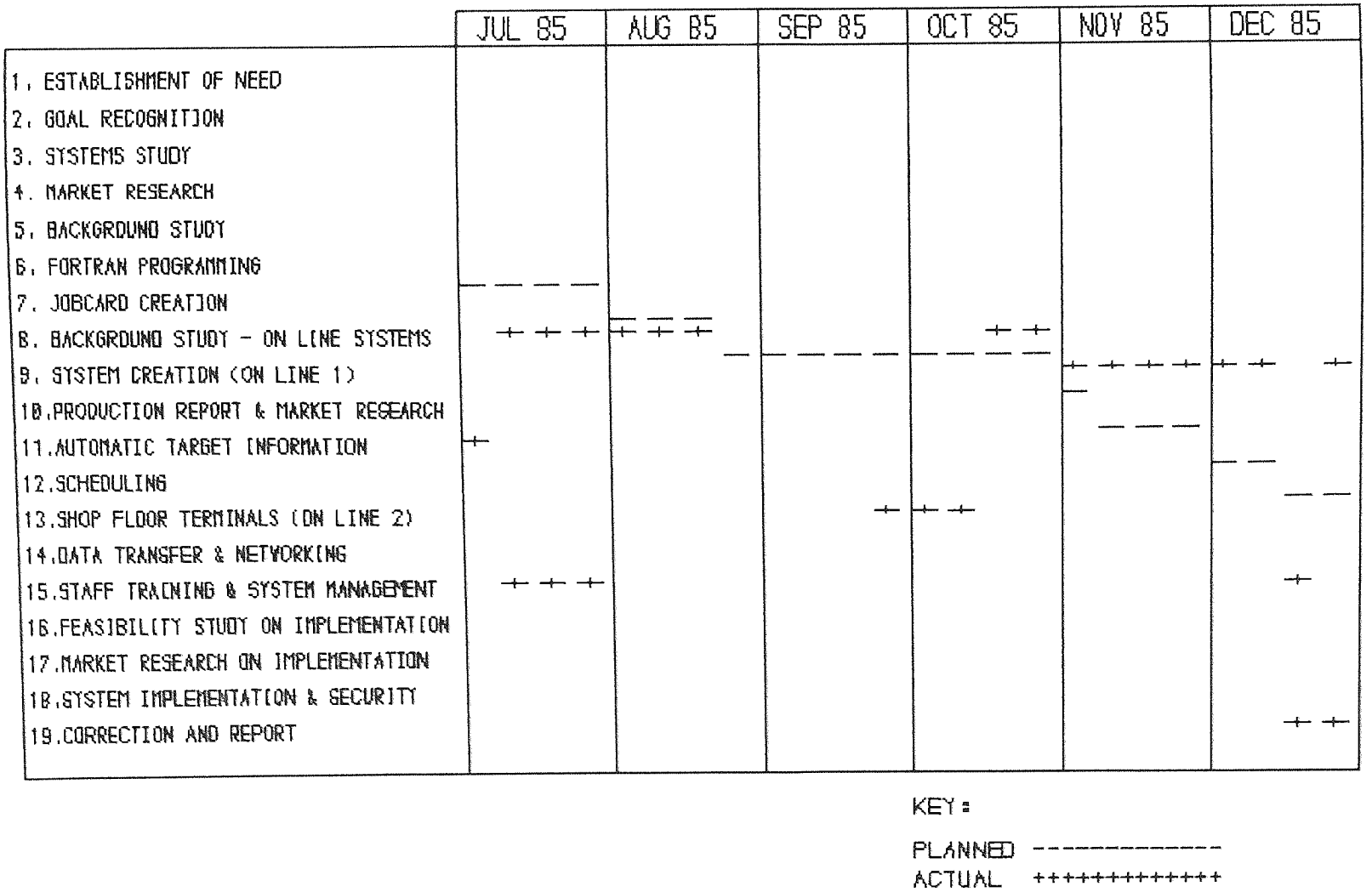
TIME CHART

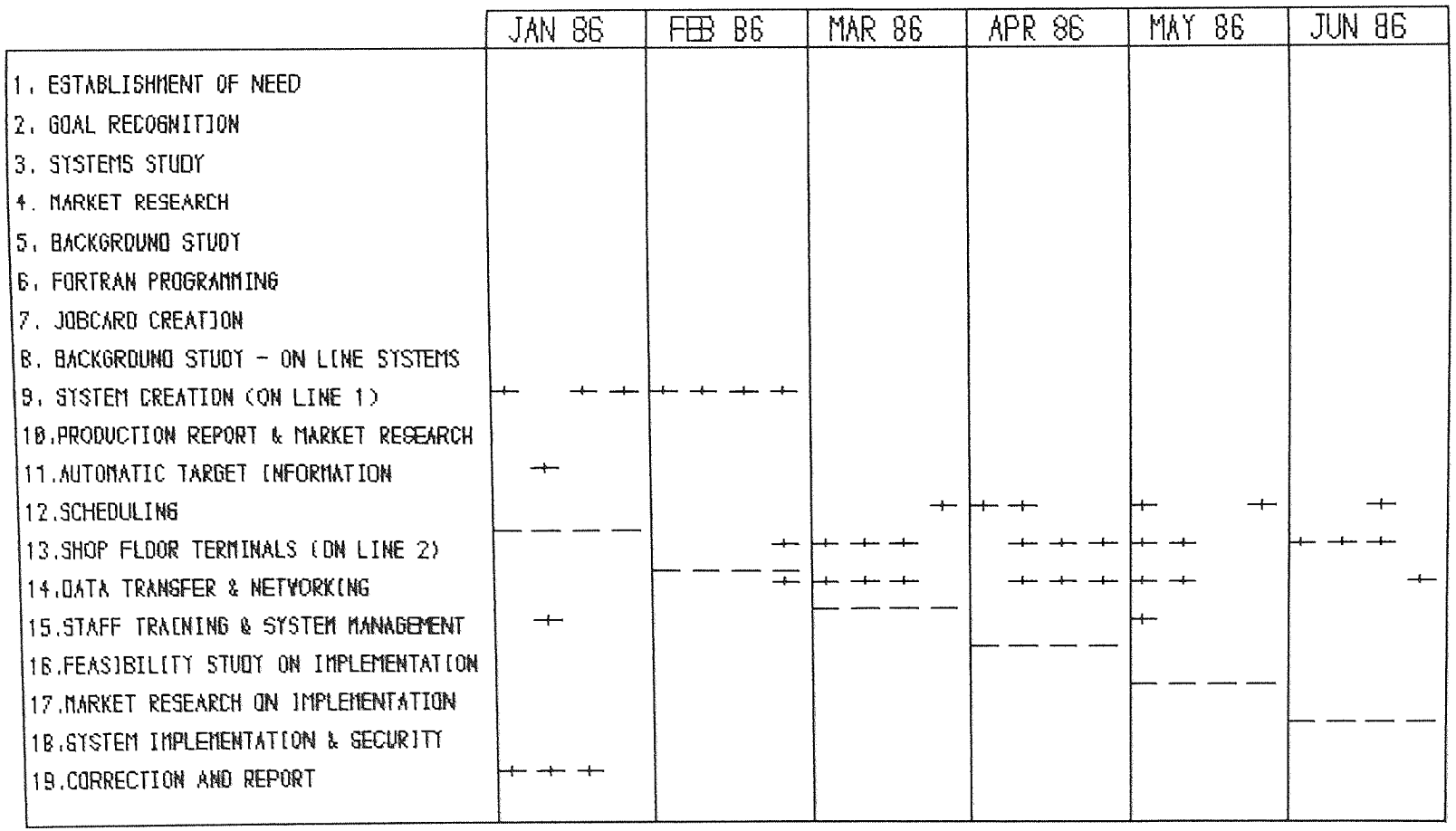


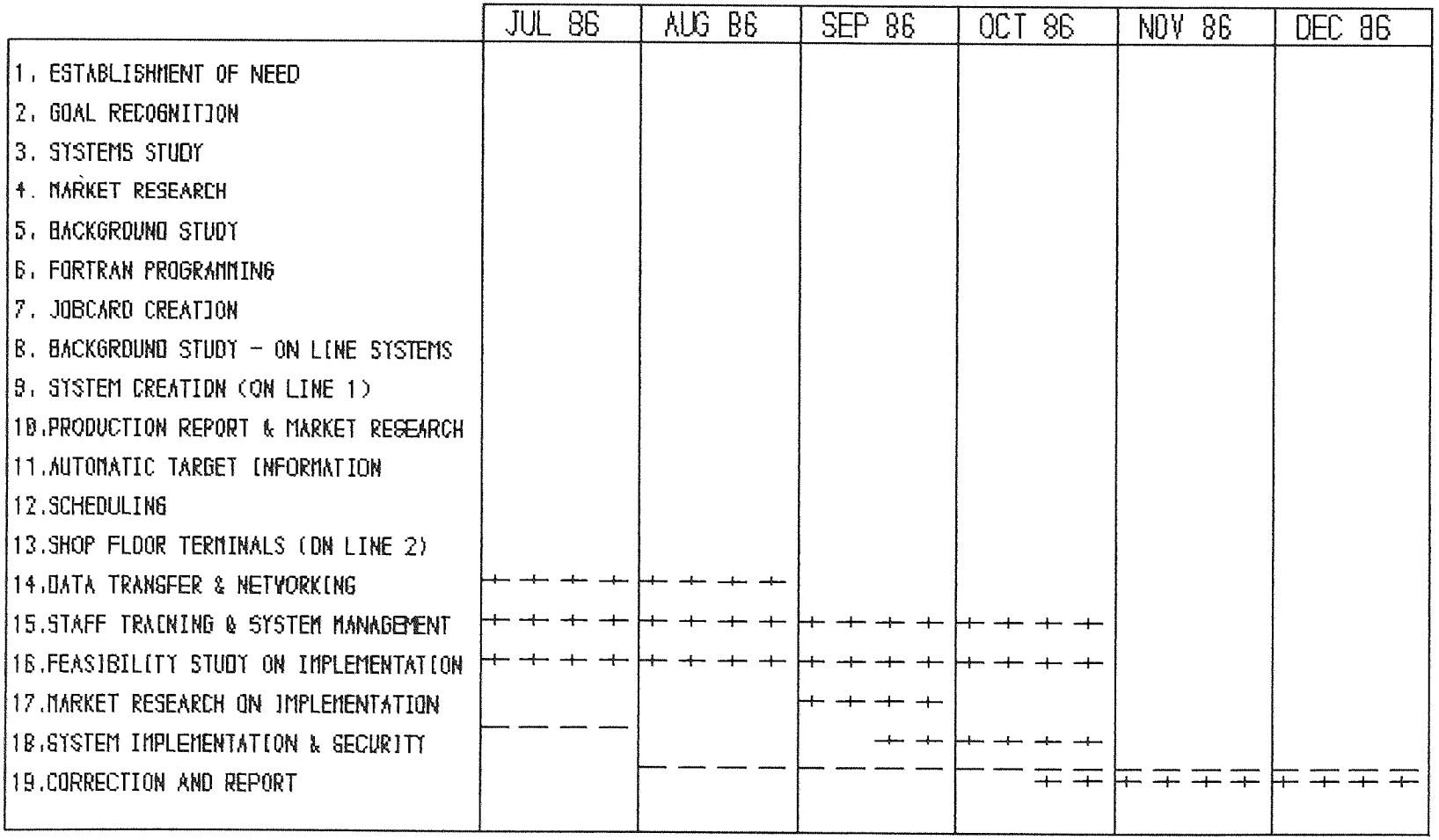
KEY:

PLANNED ----------

ACTUAL ++++++++++++


JUL 85
AUG 85
SEP 85
OCT 85
NOV 85
DEC 85
1. ESTABLISHMENT OF NEED
2. GOAL RECOGNITION
3. SYSTEMS STUDY
4. MARKET RESEARCH
5. BACKGROUND STUDY
6. FORTRAN PROGRAMMING
7. JOBCARD CREATION
8. BACKGROUND STUDY - ON LINE SYSTEMS
9. SYSTEM CREATION (ON LINE 1)
10.PRODUCTION REPORT & MARKET RESEARCH
11.AUTOMATIC TARGET INFORMATION
12.SCHEDULING
13.SHOP FLOOR TERMINALS (ON LINE 2)
14.DATA TRANSFER & NETWORKING
15.STAFF TRAINING & SYSTEM MANAGEMENT
16.FEASIBILITY STUDY ON IMPLEMENTATION
17.MARKET RESEARCH ON IMPLEMENTATION
18.SYSTEM IMPLEMENTATION & SECURITY
19.CORRECTION AND REPORT
KEY:
PLANNED ----------
ACTUAL ++++++++++++

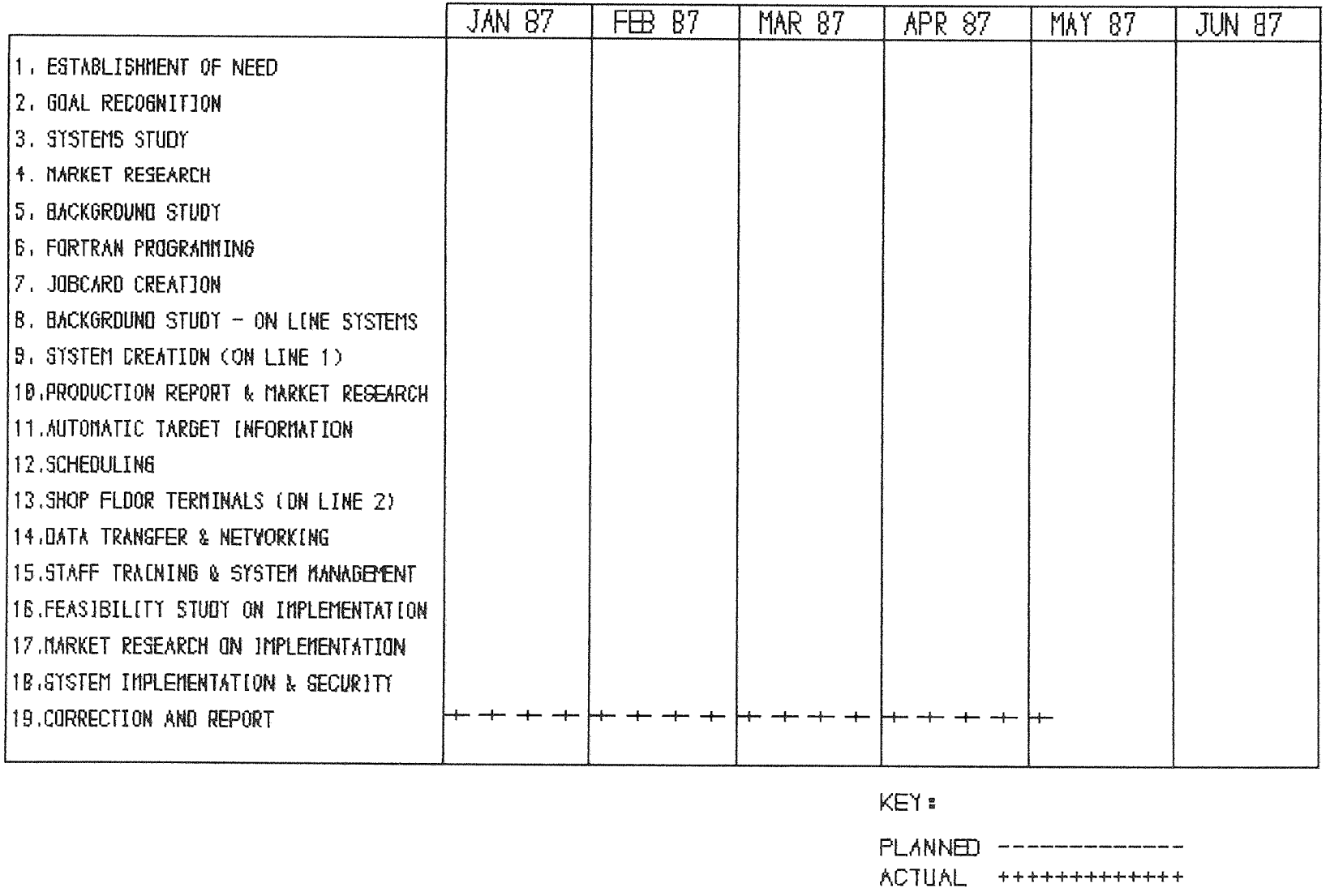

JAN 86
FEB 86
MAR 86
APR 86
MAY 86
JUN 86
1. ESTABLISHMENT OF NEED
2. GOAL RECOGNITION
3. SYSTEMS STUDY
4. MARKET RESEARCH
5. BACKGROUND STUDY
6. FORTRAN PROGRAMMING
7. JOBCARD CREATION
8. BACKGROUND STUDY - ON LINE SYSTEMS
9. SYSTEM CREATION (ON LINE 1)
10.PRODUCTION REPORT & MARKET RESEARCH
11.AUTOMATIC TARGET INFORMATION
12.SCHEDULING
13.SHOP FLOOR TERMINALS (ON LINE 2)
14.DATA TRANSFER & NETWORKING
15.STAFF TRAINING & SYSTEM MANAGEMENT
16.FEASIBILITY STUDY ON IMPLEMENTATION
17.MARKET RESEARCH ON IMPLEMENTATION
18.SYSTEM IMPLEMENTATION & SECURITY
19.CORRECTION AND REPORT
KEY:
PLANNED ----------
ACTUAL ++++++++++++



KEY:

PLANNED ----------

ACTUAL ++++++++++++


JAN 87
FEB 87
MAR 87
APR 87
MAY 87
JUN 87
1. ESTABLISHMENT OF NEED
2. GOAL RECOGNITION
3. SYSTEMS STUDY
4. MARKET RESEARCH
5. BACKGROUND STUDY
6. FORTRAN PROGRAMMING
7. JOBCARD CREATION
8. BACKGROUND STUDY - ON LINE SYSTEMS
9. SYSTEM CREATION (ON LINE 1)
10.PRODUCTION REPORT & MARKET RESEARCH
11.AUTOMATIC TARGET INFORMATION
12.SCHEDULING
13.SHOP FLOOR TERMINALS (ON LINE 2)
14.DATA TRANSFER & NETWORKING
15.STAFF TRAINING & SYSTEM MANAGEMENT
16.FEASIBILITY STUDY ON IMPLEMENTATION
17.MARKET RESEARCH ON IMPLEMENTATION
18.SYSTEM IMPLEMENTATION & SECURITY
19.CORRECTION AND REPORT
KEY:
PLANNED ----------
ACTUAL ++++++++++++

Appendix 7

Gauge tolerances

METAL CEILINGS LIMITED

GAUGE TOLERANCES as at - 29-APR-87 11:08 hrs
(Last updated on - 6-FEB-86 at 13:40 hrs)

No	Mean gauge	Min. gauge	Max. gauge	Type
1	0.450mm	0.390mm	0.510mm	ZIN
2	0.500mm	0.430mm	0.570mm	ZIN
3	0.550mm	0.480mm	0.620mm	ZIN
4	0.600mm	0.530mm	0.670mm	ZIN
5	0.700mm	0.620mm	0.780mm	ZIN
6	0.800mm	0.720mm	0.880mm	ZIN
7	0.900mm	0.810mm	0.990mm	ZIN
8	1.000mm	0.910mm	1.090mm	ZIN
9	1.100mm	0.990mm	1.210mm	ZIN
10	1.200mm	1.090mm	1.310mm	ZIN
11	1.500mm	1.370mm	1.630mm	ZIN
12	1.600mm	1.470mm	2.730mm	ZIN
13	1.800mm	1.650mm	1.950mm	ZIN
14	2.000mm	1.850mm	2.150mm	ZIN
15	2.400mm	2.230mm	2.570mm	ZIN
16	2.500mm	2.330mm	2.670mm	ZIN
17	2.600mm	2.400mm	2.800mm	ZIN
18	3.000mm	2.800mm	3.200mm	ZIN
19	3.100mm	2.850mm	3.350mm	ZIN
20	0.450mm	0.390mm	0.510mm	GAL
21	0.500mm	0.430mm	0.570mm	GAL
22	0.550mm	0.480mm	0.620mm	GAL
23	0.600mm	0.530mm	0.670mm	GAL
24	0.700mm	0.620mm	0.780mm	GAL
25	0.800mm	0.720mm	0.880mm	GAL
26	0.900mm	0.810mm	0.990mm	GAL
27	1.000mm	0.910mm	1.090mm	GAL
28	1.100mm	0.990mm	1.210mm	GAL
29	1.200mm	1.090mm	1.310mm	GAL

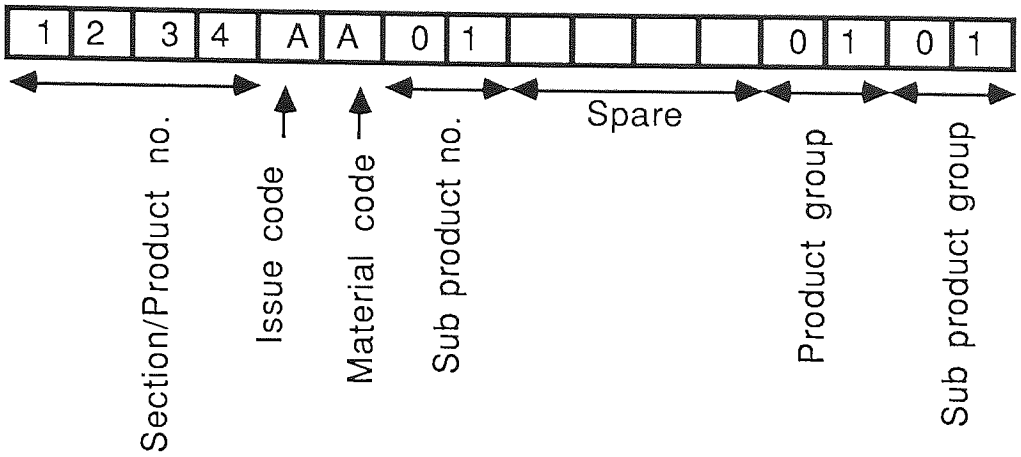
No	Mean gauge	Min. gauge	Max. gauge	Type
30	1.500mm	1.370mm	1.630mm	GAL
31	1.600mm	1.470mm	2.730mm	GAL
32	1.800mm	1.650mm	1.950mm	GAL
33	2.000mm	1.850mm	2.150mm	GAL
34	2.400mm	2.230mm	2.570mm	GAL
35	2.500mm	2.330mm	2.670mm	GAL
36	2.600mm	2.400mm	2.800mm	GAL
37	3.000mm	2.800mm	3.200mm	GAL
38	3.100mm	2.850mm	3.350mm	GAL
39	0.450mm	0.390mm	0.510mm	CR4
40	0.500mm	0.440mm	0.560mm	CR4
41	0.550mm	0.490mm	0.610mm	CR4
42	0.600mm	0.540mm	0.660mm	CR4
43	0.700mm	0.635mm	0.765mm	CR4
44	0.800mm	0.735mm	0.865mm	CR4
45	0.900mm	0.825mm	0.975mm	CR4
46	1.000mm	0.925mm	1.075mm	CR4
47	1.100mm	1.010mm	1.190mm	CR4
48	1.200mm	1.110mm	1.290mm	CR4
49	1.500mm	1.380mm	1.620mm	CR4
50	1.600mm	1.480mm	1.720mm	CR4
51	1.800mm	1.660mm	1.940mm	CR4
52	1.900mm	1.760mm	2.040mm	CR4
53	2.000mm	1.860mm	2.140mm	CR4
54	2.400mm	2.240mm	2.560mm	CR4
55	2.600mm	2.420mm	2.780mm	CR4
56	3.000mm	2.820mm	3.180mm	CR4
57	0.450mm	0.390mm	0.510mm	S/S
58	0.500mm	0.440mm	0.560mm	S/S
59	0.550mm	0.490mm	0.610mm	S/S
60	0.600mm	0.540mm	0.660mm	S/S
61	0.700mm	0.635mm	0.765mm	S/S
62	0.800mm	0.735mm	0.865mm	S/S
63	0.900mm	0.825mm	0.975mm	S/S
64	1.000mm	0.925mm	1.075mm	S/S
65	1.100mm	1.010mm	1.190mm	S/S
66	1.200mm	1.110mm	1.290mm	S/S
67	1.500mm	1.380mm	1.620mm	S/S

Gauge tolerances continued.....

No	Mean gauge	Min. gauge	Max. gauge	Type
68	1.600mm	1.480mm	1.720mm	S/S
69	1.800mm	1.660mm	1.940mm	S/S
70	2.000mm	1.860mm	2.140mm	S/S
71	2.400mm	2.240mm	2.560mm	S/S
72	2.600mm	2.420mm	2.780mm	S/S
73	3.000mm	2.820mm	3.180mm	S/S
74	1.200mm	1.030mm	1.370mm	HR
75	1.500mm	1.330mm	1.670mm	HR
76	1.600mm	1.430mm	1.770mm	HR
77	1.800mm	1.620mm	1.980mm	HR
78	2.000mm	1.820mm	2.180mm	HR
79	2.400mm	2.180mm	2.630mm	HR
80	2.500mm	2.280mm	2.720mm	HR
81	2.600mm	2.370mm	2.830mm	HR
82	3.000mm	2.770mm	3.230mm	HR

Appendix 8

Product code



Product code

Figure A8.1

A8.1 Section/Product number

The first four characters which are numeric characters are always the most important because they form the product number. If the product is a brand new one then it must have a new product number.

A8.2 Issue code

The issue code is relevant when there is either a modification or a minor dimensional change to an existing product. The issue code is usually an alphabetic character but is not compulsory when only one version of a product is available. The modifications can be given the issue codes 'A', 'B', 'C' etc.

A8.3 Material code

The material code should be shown as an alphabetic character and should follow the issue code. This will stand for a change in material used for the product or a change in colour. The material code would change when there is a material change to a product.

A8.4 Sub product code

Sub product code is used to define the different stages in production. For example, The first press operation after rolling will be 01 while the second will be 02.

A8.5 Product group

The product group is a numeric value between 1 to 99 and corresponds to the subsidiary company which manufactures the particular product.

A8.6 Sub product group

This is similar to the product group and refers to the sub groups within the product group.

Appendix 9

Sales pricing card

S/P ++++++++ SALES PRICING CARD ++++++++

1.PROD CODE :2222 2.DESCRIPT :CHICKEN TROUGH
3.CUSTOMER :BECLAWAT LIMITED 4.MATERIAL : 1.100mmX229.0mm ZIN

WEIGHT PER 100m: 5.NETT =197.69kg 6.SCRAP = 2.0% 7.GROSS =201.64kg
ESTD. ROLL INFO. : 8.SET UP = 1.0hrs 9.ROLL = 300.0m/hr10.STRIP = 1.0hrs

TOLERANCE -MAX :11.GAUGE = 1.210mm 12.WIDTH = 229.00mm 13.LENGTH=+1.0mm
-MIN : = 0.990mm = 229.00mm =-1.0mm

	Per ton	Per 100m
14.MATERIAL PRICE	:£ 300.00	£ 60.49
15.SET UP COST @ £ 43.00 PER HR	: -	£ 4.30
16.ROLLING COST @ £ 43.00 PER HR	: -	£ 14.33
17.STRIP COST @ £ 43.00 PER HR	: -	£ 4.30
18.PROFIT(@ 10.00% OF MAT COST)	: -	£ 6.05
19.ADDITIONAL OPERATIONS COST	: -	£ 0.00
20.TOOL AMORTISATION for 0m	: -	£ 0.00
21.TRANS. COST (£ 0.00 for job)	:£ 0.00	£ 0.00
22.COMMISSION 0.0	: -	£ 0.00

ACTUAL SELLING PRICES :

len-m	100m	100ft
25. 1000	£ 97.00	£ 30.00
26. 0	£ 0.00	£ 0.00
27. 0	£ 0.00	£ 0.00
28. 0	£ 0.00	£ 0.00
29. 0	£ 0.00	£ 0.00

23.SELLING PRICE BASED ON1000m:FOR 100m = £ 89.48 -> FOR 100ft =£ 27.27

24.ADDITIONAL INFORMATION : Roll with 2341

(The increase from original price is -47.6%)

Press 0 to go to menu, 1 to 29 to analyse, 30 to print or 31 for actual data

ROLLING DETAILS FOR 2222

No	Date	Machine	Setup time	Roll speed	Strip time	Comments
1	06.10.86	4	0.9 hrs	900.0 m/hr	2.0 hrs	
2	07.10.86	4	1.2 hrs	700.0 m/hr	2.5 hrs	Break down
3	15.11.86	4	1.0 hrs	835.0 m/hr	2.0 hrs	
4	21.12.86	3	1.1 hrs	770.0 m/hr	2.5 hrs	
TARGETS :			0.9 hrs	900.0 m/hr	1.0 hrs	

Press 0 to go to the menu or 1 to go back to analysis

Appendix 10

Jobcard

S/P +++++ JOB CARD 3 - Created on 16.10.86 at 13:35hrs +++++
(Tied with job number 1)

1. PRODUCT DESCRIPTION =CHICKEN TROUGH
2. PRODUCT CODE =2222
3. MATERIAL DESCRIPTION = 1.100mmX229.0mm ZIN

4. THICKNESS -MAX = 1.21mm -MIN = 0.99mm
5. WIDTH -MAX = 229.00mm -MIN = 229.00mm

WT OF RAW MAT : 6. ESTD: = 6.72kg 7. ACTUAL = -
ROLL SPEED : 8. TARGET = 900.0m/h 9. ACTUAL = -
SET UP TIME : 10. TARGET = 0.8hrs 11. ACTUAL = -
STRIP TIME : 12. TARGET = 1.0hrs 13. ACTUAL = -

STARTING TIME : -14. ESTD = MONDAY 25.08.1986 at 08:37hrs on line 3
-15.ACTUAL = -
FINISHING TIME : -16. ESTD. = MONDAY 25.08.1986 at 13:19hrs
-17.ACTUAL = -
(Deadline to finish -> THURSDAY 09.10.1986)

20. MATERIAL AVAILABILITY =NO

18. DETAILS ABOUT LENGTHS :

no	Length	Required Qty	Computed Qty	Actual Qty
1	1200.000mm	1000nos	-	-
2	2133.600mm	1000nos	-	-
3	2500.000mm	2000nos	-	-
4	2700.000mm	1000nos	-	-
5	2800.000mm	1200nos	-	-
6	3000.000mm	1000nos	-	-
7	3100.000mm	1100nos	-	-

19. DETAILS ABOUT CUSTOMERS :

CUSTOMER NUMBER 1 (Tolerance + 1.0mm,- 1.0mm)

A. NAME =BECLAWAT LIMITED
B. CODE =BECL01
C. ORDER NUMBER =SCV59 D. ORDER DATE =30. 09.1986
E. ACKNOW: N0: =SP1006UK F. DELIVERY DATE =10.10.1986
G. PACKING INST: =NONE
H. EXC: MAT: INS: =NONE
I. SPECIAL INSTR: =NONE
J. LEN/QUAN FOR CUS 1 : No: Length Qty
1 1200.00mm(0ft 0 0/ 0in) 1000nos
2 2500.00mm(0ft 0 0/ 0in) 2000nos
3 3100.00mm(0ft 0 0/ 0in) 1000nos

K. SELL. PRI. OF CUS 1 @ £ 49.48 per 100m =£ 1073.71

CUSTOMER NUMBER 2 (Tolerance + 1.0mm,- 1.0mm)

A. NAME =LINK51 LIMITED
B. CODE =LINK01
C. ORDER NUMBER =SVB456 D. ORDER DATE =31. 09.1986
E. ACKNOW: N0: =SP1007UK F. DELIVERY DATE =13.10.1986
G. PACKING INST: =NONE
H. EXC: MAT: INS: =NONE
I. SPECIAL INSTR: =NONE
J. LEN/QUAN FOR CUS 2 : No: Length Qty
1 2133.60mm(7ft 0 0/ 0in) 1000nos
2 2700.00mm(0ft 0 0/ 0in) 1000nos
3 3100.00mm(0ft 0 0/ 0in) 100nos

K. SELL. PRI. OF CUS 2 @ £ 56.64 per 100m =£ 2061.97

CUSTOMER NUMBER 3 (Tolerance + 1.0mm,- 1.0mm)

A. NAME =BIG DUTCHMAN LIMITED
B. CODE =BIGD01
C. ORDER NUMBER =BD4876 D. ORDER DATE =31. 09.1986
E. ACKNOW: N0: =SP1008UK F. DELIVERY DATE = -
G. PACKING INST: =PACK INTO UNIS OF 50
H. EXC: MAT: INS: =ROLL ALL COIL
I. SPECIAL INSTR: =DELIVER TO LONDON OFFICE
J. LEN/QUAN FOR CUS 3 : No: Length Qty
1 2800.00mm(7ft 0 0/ 0in) 1200nos
2 3000.00mm(0ft 0 0/ 0in) 1000nos

K. SELL. PRI. OF CUS 3 @ £ 58.60 per 100m =£ 2841.21

Appendix 11

Market research on hardware

Product	Features	Price
Forth Microboard (R.C.S. Microsystems)	Development system using a host for file storage and as a stand alone target system running in either FORTH or 6502 code.	
CMOS S.B.C. (R.C.S. Microsystems)	Wide range of data logging applications.	£165
Chemeleon (R.C.S. Microsystems)	Total of 54 I/O lines. 2 x 16 bit timer and counters with latches. RS422/423 interface, 'watchdog' and up to 65k RAM/EPROM	£169
Eleven Q Microsystem)	44 I/O lines, RS232, 8k RAM, 4 k battery backed RAM and 20 character display. 6502 assembler or FORTH.3 Timers	£225
Operator terminal O/T 86 (Newmen Ltd. 128 Alesk Rd. Moseley, 021-449-6342)	Full Alphanumeric keys, 20 user definable keys with label insert facility, RS232 serial port and 5 inch colour or monochrome monitor.	
Z80 systems Controller S.B.C.(Newmen)	Z80, 48k EPROM, 16k RAM, real time clock calender and 2 serial RS232s.	
S100 (H.T.E.)	3 serial I/O ports with separately programmable baud rates. (accessories available).	
Merlin P.L.C. (J.B. Microsystems)	Basic language, multiple task real time functions, 3 serial ports (RS232,422), 16 isolated digital inputs and 30 analog inputs.	£2295
CAD master CPU. (GNC Electronics)	Suitable as a single board controller or the heart of the CADmaster system. Multi tasking program and RS232.	£178+VAT
CMR 16 micro controller	Data acquisition and monitoring. Program development may be by an emulator plugged into the EPROM socket or on the host.	£220+VAT
1006 S.B.C. (R.C.S. Microsystems)	3 full duplex V24 ports. VIA 6522 with 40 way connector.	

Product	Features	Price
CMOS 65CO2 (R.C.S. Microsystem)	6502 and two 8 bit ports. 2 timers.	£183
W500 Machine Minder	Can be used as intelligent links. Can monitor and identify when performance is below target.	
Warwick Computers Ltd.	Can transfer information back to plant computer via RS232, RS422.	
Andelos 68000 S.B.C. (Andelos Systems)	24 line parallel interface and serial RS232. Sharp MHZ80A	£415
SDS Archer (Sherwood Data Systems Ltd.)	Z80A, 2RS232 port, 32 parallel I/O lines and 8 handshake signals.	£250
CMS 6809 Controller (Cambridge Micro Systems Ltd)	Based on BBC and 6502. Battery backed CMOS RAM	£500
IQ200 (Del Tek Electronics)	64k RAM Z80 based	£250
TDS 900 Forth Computer (Triangle Digital Services Ltd)	Up to 82 I/O, expansion, RS232, CMOS 12k RAM and 8k ROM	£99
8300 1 Micro com. (Automation and Technology Control)	8085 chip 2k x 8 bit RAM, 4k x 8 bit ROM, 16 character x 2 line LCD display, 16 button keyboard and 8 bit port plus 2 serial I/O lines 14 bit timers.	£199
8500 Processor Board (Automation and Technology Control)	Based on 8085 chip. 2k to 96k plus RS232.	£209
ARC40 (Arcom)	RS232, three 8 bit ports. 16 I/O lines of which 8 are bi-directional.	
ARC41 and ARC42	Data logging and real time monitoring.	
STEbus CP/M (Arcon Micro)	Based on Z80A. 64k RAM, 2 serial ports and 5 parallel I/O lines.	
ARC8000 28001 VME CPU	16/32 bit computer, dual RS232 and 34 way parallel I/O lines.	

Product	Features	Price
Eurobus (Micro Marketing (Electronics) Ltd)	64K RAM, RS232 and Floppy Disk controller for 5 $^{1}/_{4}$ or 8" drives.	
Exorbus	Triple programmable timer plus parallel I/O and RS232 ports and a floppy disc controller.	
Multibus	8 channel serial communication controller.	
Source SICs 208, 216, 280/BR (Perex)	4 line RS232 printer, I/O lines and 64 RAM second processor	
System 95 (Arthur Ford)	Uses TI9995. 16 bit $^{1}/4$ k RAM. 16 inputs and 16 outputs.	£9600+VAT
R6511Q Microprocessor & R6500/13micro computer (Pelco Electronics)	Rockwell's 6500 family of microprocessors.	
SBC6511 (Rockwell)	4 I/O ports, RS232 or 422, 2 16 bit timer counters plus TTL board.	
DAZ80 (Data Applications Ltd)	24 programmable I/O lines, TTL compatible serial Z80 DUART and full display.	£4990
Eurobeeb/CUBE (Control Universal Ltd)	6502 based development sysem with a 24 function keyboard and 1 or 2 line display. Battery backed RAM. Fully BBC compatible.	£450

Appendix 12

RS-232 standard

The letters RS stand for "Recommended Standard" and the RS-232 is by far the most popular standard for communication today. It is necessary to understand some widely used terms before understanding how this standard is used for communication.

A12.1 Data Terminal Equipment (DTE)

Data terminal equipment are usually computers, terminals and other peripherals.

A12.2 Data Communications Equipment (DCE)

Data Communications Equipment are MODEMs which help Data Terminal Equipment to communicate with each other. These equipment transfer the data in to a format which can be transmitted over long distances.

A12.3 Synchronous transmission

The speeds of transmission and the time between characters have to be pre set in order to use this type of transmission so that the receiving device can receive all the data without loss by using a timing sequence.

A12.4 Asynchronous transmission

Asynchronous transmission does not use a fixed time limit or a speed but uses the concept of enclosing each character with a start bit and a stop bit so that the receiving device can recognize all the data.

A12.5 Simplex transmission

The data can travel in only one direction in simplex transmission and this direction is fixed. The data can not travel in the other direction at any time.

A12.16 Half duplex transmission

The data can travel in both directions but not at the same time. Therefore the transmission in one direction must essentially finish before the transmission in the other direction can take place.

A12.7 Full duplex transmission

The data in this case can travel in both directions simultaneously.

A12.8 The principle of RS-232 transmission

The plugs and sockets which are used with the RS-232 communication normally comprise of 25 pins (or holes) of which all connections (table A12.1) are not used. The most important two out of these, are lines 2 and 3 which are used for transmitting and receiving data respectively. These two lines are crossed between two devices because one's transmitting line is the other's receiving line and vice versa.

Pin	Code	Description	Direction	Function
1	PG	protective ground	-	ground
2	TXD	transmitted data	from DTE	data
3	RXD	received data	from DCE	data
4	RTS	request to send	from DTE	control
5	CTS	clear to send	from DCE	control
6	DSR	data set ready	from DCE	control
7	SG	signal ground	-	ground
8	DCD	data carrier detect	from DCE	ground
9	-	reserved	-	-
10	-	reserved	-	-
11	-	unassigned	-	-
12	SDCD	secondary DCD	from DCE	control
13	SCTS	secondary CTS	from DCE	control
14	STXD	secondary TXD	from DTE	data
15	TT	transmit timing	from DCE	timing
16	SRXD	secondary RXD	from DCE	data
17	RT	receipt timing	from DCE	timing
18	-	unassigned	-	-
19	SRTS	secondary RTS	from DTE	control
20	DTR	data terminal ready	from DTE	control
21	-	signal quality detector	-	-
22	RI	ring indicator	from DCE	control
23	-	data signal rate detector	-	-
24	EXT-TT	external transmit timing	from DTE	timing
25	-	unassigned	-	-

RS-232 pin configuration

Table A12.1

The devices use 'dial up' connections when they are connected via MODEMs and the DTR (pin 20) of the devices are raised when they are switched on. The number of one DTE is dialled

(either manually or automatically) when the other wants to transmit and this is informed by raising the ring indicator (pin 22) of the far device. Once that device answers then each device raises it's DSR (pin 6). The devices can now transfer data until one of the DTRs is dropped at which time the connection is lost.

It is quite simple to transmit data this way if each device is capable of receiving any amount of data. The transmission fails if the receiver's buffer overflows and a technique called handshaking is used to avoid this.

A12.9 Handshaking

The basic principle of hardware handshaking is to inform the sender that the receiver's buffer is full and therefore to suspend transmission until further notice. The receiver sends another signal to the sender to continue transmission when there is enough room in the buffer. Table A12.2 shows the appropriate lines which are used for this.[199]

Pin no.	Abbreviation	Description
4	RTS	Request to send
5	CTS	Clear to send
8	DCD	Data carrier detect

Handshaking lines

Table A12.2

A12.9.1 Handshaking in a half duplex environment

When a device needs to transmit, it raises it's RTS (pin 4) high and this passes a signal to raise the DCD (pin 8) of the DCE. The DCE first checks to see whether it's DCD is already high and if not raises the CTS (pin 5) of the device which requested to send and also the DCD of the device which has to receive.

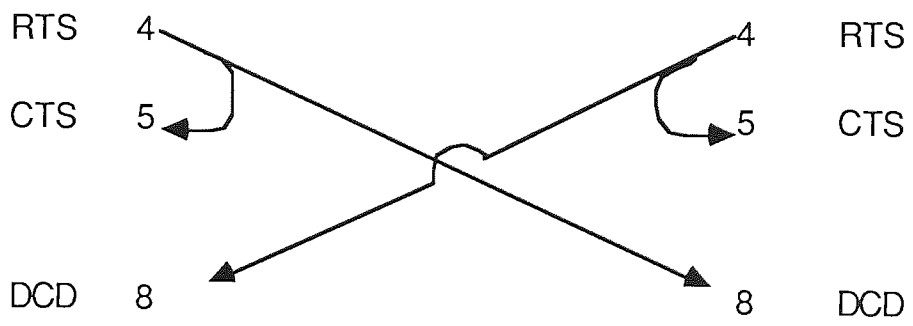
If the DCD of the DCE is already high, then it means that the second DTE is transmitting, in which case the DCE will not raise the CTS of the initial DTE. After transmission is complete, the device which transmitted drops it's RTS which in turn puts the DCD of the DCE and the DCD of the second DTE low. The DCE then waits until one device requests to transmit again.

A12.9.2 Handshaking in a half duplex environment

The DCE raises both CTSs if they request to send since both devices can transmit simultaneously.

A12.9.3 Direct connections

There is no DCE to raise the DCD of one device high in the case of a direct connection when the other device raises it's RTS. Therefore the two devices must do this task on their own and the connection in figure A12.1 is used.



Direct connection

Figure A12.1

All the parameters which can be set up in a RS-232 communication are shown in table A12.3

Parameter	Options
speed	50 to 19200 baud
flow control	Xon/Xoff, hardware
parity	odd, even, none
character length	5, 6, 7, 8
number of stop bits	1, 1.5, 2
mode	simplex, half duplex, full duplex
local echo	yes, no
line feeds	0, 1, 2
transmission mode	asynchronous, synchronous
polarity	positive, negative

Communication parameters

Table A12.3

Appendix 13

Communications standards

RS-232	Interface standard for Data Terminal Equipment and Data Communication Equipment (Note this is a Recommended Standard of Electronic Industries Association (EIA)
RS-422	Electrical standard for interfacing balanced circuits.
RS-423	Electrical circuit for interfacing unbalanced circuits.
RS-449	Mechanical standard for connector pin assignments
X3.28	Character code control standards (American National Standards Institute - ANSI)
1748	Basic mode control standard (International Standards Organization - ISO)
V.35	48k bit/sec data transmission standard (International Consultive Committee for Telegraphy and Telephony- CCITT)
X.21	General purpose DTE to DCE standard for public networks (CCITT)
X.25	Interface standard for packet switched service on public networks (CCITT)
SDLC	Synchronous Data Link Control (IBM)
SNA	Systems Networks Architecture (IBM)
DNA	Digital Equipment Company Network Architecture
UART	Universal Asynchronous Receiver/Transmitter

USRT	Universal Synchronous Receiver/Transmitter
USART	Universal Synchronous/Asynchronous Receiver/ Transmitter
NCP	Network Control Program (IBM)
NRZI	Non Return Zero Inverted (magnetic tape)
NSP	Network Services Protocol (DEC)
HDLC	High level Data Link Control
DLC	Data Link Control
BISYNC	Binary Synchronous Communication (IBM)
DDCMP	Digital Data Communications Message Protocol
DDD	Direct Distance Dialling
ADCCP	Advanced Data Communications Control Procedure
BOP	Bit Oriented Protocol
CCP	Character Controlled Protocol
CRC	Cyclic Redundancy Check
PAD	Packet Assembler/Disassembler

Appendix 14

Manufacturing Automation Protocol

MAP is in fact a collection of various standards brought together to define a broadband Local Area Network through devices such as computers and controllers which can communicate in real time. Since it is a Broadband or a high frequency (6 to 12 MHz) data transmission system which is similar to certain types of cable TV, it can transmit data at high speeds (5 to 10 Mbps), in several channels as opposed to the relatively slow low frequency baseband transmission. The conceptual model of MAP divides it's functions to seven different layers, some of which contain sub layers. MAP is still in it's development stage and there is quite a lot of work to be done in the upper layers while the lower layers are getting near completion. These layers are expected to be put on an interface board which can be plugged in to the appropriate networks but it is still not understood how much of it will have to be downloaded from the host on to the board for temporary residence.

A factory network can link to a MAP network using either a bridge or a gateway. The former can link OSI standard

networks to MAP networks while the latter can link non-OSI networks. The seven layers of MAP and their functions can be discussed briefly as follows.[2,3]

A14.1 Layer 1 - Physical (IEEE 802.4)

This is the bottom layer of the hierarchy. This layer deals with the electrical signalling among the nodes and specifies the connections between each station and the transmission medium. In effect it delivers the electrical signal (i.e., the bit stream) between stations but essentially in a single local area network. It uses a single cable and a MODEM. The method that is used is frequency division multiplexing, and hence the frequency converter and the MODEM at each end (including specifications) are included.

A14.2 Layer 2 - Data Link (IEEE 802)

A14.2.1 Sublayer Logical Link Control-LLC(IEEE802.2 type1)

This is an unacknowledged connectionless link control and is a null protocol at present.

A14.2.2 Sublayer Medium Access Control- MAC (IEEE 802.4)

It is a token bus or a token ring which transfers frames across a single network using the standard transmission medium. It selects and delimits the required frames of data which are addressed to a particular station. Correction of errors by noise detection (by checking each frame in sequence) is also an important function of this sublayer.

A14.3 Layer 3 - Network (ISO 8483)

This delivers packets of data between OSI end systems across an arbitrary set of sub networks by routing and sequencing the data. The process is similar to an X25 Packet switching exchange which break long messages in to small blocks of data for onward transmission to avoid congestion at different parts of the network. The difference between this layer and the two lower layers is that this deals with a set of sub networks while the other two deal in a single network. Hence globally unique node addresses have to be used as opposed to local addresses. These global addresses are used by the relay stations to determine the route of the data packets.

A14.4 Layer 4 - Transport (ISO 8073)

This is the most complex and the most capable transport protocol defined yet by the ISO and it's services include establishment of connection between two OSI end systems, reliable transfer of unlimited sized data packets and the termination of connection.

What this layer delivers to an OSI end system is the data that was brought to another OSI end system by the three lower layers. (i.e., it provides end to end data delivery between two OSI end systems.)

A14.5 Layer 5 - Session (ISO 8237)

The Session layer allows a systematic dialogue between application programs. The structure can be either two way simultaneous (full duplex) or two way alternative (half duplex). In addition it provides a check pointing and re-synchronizing capability which will help the error recovery in the FTAM sublayer in the top layer.

A14.6 Layer 6 - Presentation (ISO 8823)

This layer is null at present but it is expected that it would consist of the following features when developed. The formatting of data (including graphical models and their co-ordinates) will be the most important feature of the presentation layer. The reason is that this data from one system representation has to be transformed in to a universal format before being transmitted to another system because the second system would not understand the format of data in the first system. This layer will then have to re-format the data in such a way that it can be understood by the second system. In other words the Session layer has to translate data from one system to another by translating it in to it's own format first. The ASN (Abstract Syntax Notation) basic encoding rules (ISO 8825) are used for the transfer syntax.

A14.7 Layer 7 - Application

The Application layer is the top layer of the system and is the most complex. It consists of five different sublayers and each handles a different function.

A14.7.1 Sublayer CASE

CASE is the abbreviation for Common Application Service Element and it links OSI application protocols or user software that use services of exchanging information (not data). It switches between file transfer and terminal emulation. Mutinational funds transfer is an example of such a service.

A14.7 2 Sublayer SASE

This is the Specific Application Service Element and have functions like File Transfer Access and Management (FTAM), Virtual Terminal Protocol (VTP), Job Transfer Manipulation (JTM) and Message Handling Systems (MHS).

In the FTAM section it allows creation and deletion of files in a remote system while VTP allows reading and updating remote files. JTM is also an important feature which deals with remote jobs while MHS is an electronic mail system.

A14.7.3 Sublayer Management (IEEE 802.1)

This protocol allows a management node to collect statistics related to network activity. The purpose of this sublayer is to aid problem detection and isolation. It is expected that altering network configuration and performance parameters such as transmission speeds, flow control, delay between characters etc. will also be provided when the model is fully developed.

A14.7.3 Sublayer Directories

This is currently a General Motors specification and is expected to be replaced by the equivalent OSI protocol when defined. It is expected that this sublayer will provide the necessary information to access a user software program within the global network.

A14.7.5 Sublayer MMS

Sublayer Manufacturing Message Service is the current specification of the old Manufacturing Message Format Standard.

A14.8 RS-511

The RS511 protocol[164] has been under development for the last 5 or 6 years as the rest of the Manufacturing Automation Protocol. It provides all the basic functions which are required by the network control applications such as,

(1) Initialisation and management of dialogue between two application software programs.

(2) Starting and stopping services at a remote node or a device.

(3) Determination of an operational service of a device.

The most important services which are provided by MAP will be using the RS-511 protocol. (within the top layer.) The main RS-511 services can be divided in to seven different categories and can be explained as follows.

A14.8.1 Synchronization

This is a very important aspect in communication as the sender has to know whether the receiver is ready to receive before the next lot of data is sent, and the receiver has to inform the sender when ready to receive. This is conventionally known as handshaking and the reliability of the transfer depends on this. This is also useful in multi tasking jobs.

A14.8.2 Operaror communication

The operators at each end can communicate with each other, either via a software program or directly.

A14.8.3 Remote memory access

This is by far the most important function which the RS-511 or the whole of Manufacturing Automation Protocol can provide. This function permits access to variables stored in a remote device and also allows operations like read, write, define type and define variable name. The services therefore maintain information

such as data type (i.e., integer, array character etc.) so that the transferred data can be stored in the device without the operator intervention. The user therefore does not have to be concerned about either the different floating point representations or the difference between processors.

A14.8.4 Job scheduling

This is particularly a useful function in Computer Integrated Manufacture where the management need to schedule or re-schedule jobs at another plant by using a central computer.

A14.8.5 Robotic control

Robotic control is a function that is found even in small mini and micro computers. The difference in the RS-511 equivalent service is that all the software is provided in the protocol where in most the other cases the user has to write his own software. It's typical usage will be in a materials handling environment where tool changes, positioning of working surfaces and cutter paths are determined.

A14.8.6 File transfer and management

This function has it's advantages while being not as important as remote memory access. It allows the transfer of sequential access files or part programs and helps a host system to manage a file store of a remote device.

File transfer and management also enables a user to see an operator display (for example a shop floor terminal) and also to write instructions to it.

A14.8.7 Remote event definition and management

An event on a remote device will be able to be created or defined and also to be managed by the user or a software program with the aid of this function. Therefore the management can simply start and manage a new function at a remote plant from their own location.

Appendix 15

Production report

S/P PRODUCTION REPORT 1 - Created on 16.10.86 at 13:35hrs

1. PRODUCT DESCRIPTION =CHICKEN TROUGH
2. PRODUCT CODE =2222
3. MATERIAL DESCRIPTION = 1.100mmX229.0mm ZIN

4. THICKNESS -MAX = 1.21mm 5. WIDTH -MAX = 229.00mm
-MIN = 0.99mm -MIN = 229.00mm

WT OF RAW MAT	:	6. ESTD:	= 4.99kg	7. ACTUAL	= 5.20kg
ROLL SPEED	:	8. TARGET	= 200.0m/h	9. ACTUAL	= 220.0m/hr
SET UP TIME	:	10. TARGET	= 1.0hrs	11. ACTUAL	= 1.4hrs
STRIP TIME	:	12. TARGET	= 0.0hrs	13. ACTUAL	= 0.0hrs

STARTING TIME : -	14. ESTD.	= FRIDAY	22.08.1986 at 11:15hrs on line 3
	15. ACTUAL	= FRIDAY	22.08.1986 at 09:18hrs
FINISHING TIME :	16. ESTD.	= MONDAY	25.08.1986 at 08:37hrs
	17. ACTUAL	= TUESDAY	29.04.1986 at 11:33hrs
	(Deadline to finish ->	FRIDAY	26.09.1986)

18. DETAILS ABOUT LENGTHS :

no	Length	Reqd Qty	Compd Qty	Actual Qty
1	1000.000mm	1100nos	1103nos	1100nos
2	1100.000mm	1250nos	1255nos	1255nos

19. DETAILS ABOUT CUSTOMERS :

CUSTOMER NUMBER 1 (Tolerance + 1.0mm,- 1.0mm)

A. NAME =LINK51 LIMITED
B. CODE =LINK01
C. ORDER NUMBER =WT481 D. ORDER DATE =23. 9.1986
E. ACKNOW: N0: =SP1000UK F. DELIVERY DATE =28. 9.1986
G. PACKING INSTR: =NONE
H. EXC: MAT: INS: =NONE
I. SPECIAL INSTR: =NONE

Appendix 15

Production report

S/P PRODUCTION REPORT 1 - Created on 16.10.86 at 13:35hrs

1. PRODUCT DESCRIPTION =CHICKEN TROUGH
2. PRODUCT CODE =2222
3. MATERIAL DESCRIPTION = 1.100mmX229.0mm ZIN

4. THICKNESS -MAX = 1.21mm 5. WIDTH -MAX = 229.00mm
-MIN = 0.99mm -MIN = 229.00mm

WT OF RAW MAT	:	6. ESTD:	= 4.99kg	7. ACTUAL	= 5.20kg
ROLL SPEED	:	8. TARGET	= 200.0m/h	9. ACTUAL	= 220.0m/hr
SET UP TIME	:	10. TARGET	= 1.0hrs	11. ACTUAL	= 1.4hrs
STRIP TIME	:	12. TARGET	= 0.0hrs	13. ACTUAL	= 0.0hrs

STARTING TIME : - 14. ESTD. = FRIDAY 22.08.1986 at 11:15hrs on line 3
- 15. ACTUAL = FRIDAY 22.08.1986 at 09:18hrs
FINISHING TIME : - 16. ESTD. = MONDAY 25.08.1986 at 08:37hrs
- 17. ACTUAL = TUESDAY 29.04.1986 at 11:33hrs
(Deadline to finish -> FRIDAY 26.09.1986)

18. DETAILS ABOUT LENGTHS :

no	Length	Required Qty	Computed Qty	Actual Qty
1	1000.000mm	1100nos	1103nos	1100nos
2	1100.000mm	1250nos	1255nos	1255nos

19. DETAILS ABOUT CUSTOMERS :

CUSTOMER NUMBER 1 (Tolerance + 1.0mm,- 1.0mm)

A. NAME =LINK51 LIMITED
B. CODE =LINK01
C. ORDER NUMBER =WT481 D. ORDER DATE =23. 9.1986
E. ACKNOW: N0: =SP1000UK F. DELIVERY DATE =28. 9.1986
G. PACKING INSTR: =NONE
H. EXC: MAT: INS: =NONE
I. SPECIAL INSTR: =NONE

J. LEN/QUAN FOR CUS 1 : No: Length Qty
1 1000.00mm(0ft 0 0/ 0in) 1100nos
2 1100.00mm(0ft 0 0/ 0in) 1250nos

K. SELL. PRI. OF CUS 1 @ £ 96.64 per 100m =£ 2391.91

20. DOWN TIME ANALYSIS:

A. SETTING TIME	= 1.38hrs
B. STRIP TIME	= 0.00hrs
C. LENGTH CHANGE	= 2.30hrs
D. COIL CHANGE	= 0.00hrs
E. WAITING FOR INFORMATION	= 0.00hrs
F. FAULTY ROLLS	= 0.00hrs
G. DEFECTIVE TOOL	= 0.00hrs
H. MACHINE FAULT	= 0.00hrs
I. TOOL TRIALS	= 0.00hrs
J. REJECT MATERIAL	= 0.00hrs
K. MATERIAL SHORTAGE	= 0.00hrs
L. MEAL BREAK	= 0.00hrs
M. DOWNTIME WITHOUT A REASON	= 1.50hrs

---------> TOTAL DOWNTIME	= 5.18hrs

TOTAL JOB TIME	:	21. ESTD:	= 13.38hrs	22. ACTUAL	= 18.50hrs
TOT. ROLL TIME	:	23. ESTD:	= 12.38hrs	24. ACTUAL	= 13.32hrs
TOT. PROD.TIME	:	25. ESTD:	= 13.38hrs	26. ACTUAL	= 14.70hrs
EFFICIENCY	:	27. ESTD:	= 100.00%	28. ACTUAL	= 79.46%
UNIT PRI. MAT.	:	29. ESTD	=£300.00/ton	30. ACTUAL	=£200.00/ton
UNIT OVERHEAD	:	31. ESTD	=£ 43.00/hr	32. ACTUAL	=£ 35.00/hr
TOT. MAT. COST	:	33. ESTD:	=£ 1497.21	34. ACTUAL	=£ 1 040.00
TOT. O/H COST	:	35. ESTD:	=£ 575.13	36. ACTUAL	=£ 647.50
ADD. OPP. COST	:	37. ESTD:	=£ 0.00	38. ACTUAL	=£ 12.00
TOOL AMO. COST	:	39. ESTD:	=£ 0.00	40. ACTUAL	=£ 20.00
TRANS. COST	:	41. ESTD:	=£ 0.00	42. ACTUAL	=£ 0.00
COST	:	43. ESTD:	=£ 0.00	44. ACTUAL	=£ 0.00
COMMISSION	:	45. ESTD:	=£ 0.00	46. ACTUAL	=£ 0.00
TOTAL COST	:	47. ESTD:	=£ 2072.34	48. ACTUAL	=£ 1719.50
SELLING PRICE	:	49. ESTD:	=£ 2391.91	50. ACTUAL	=£ 2400.75
PROFIT/(LOSS)	:	51. ESTD:	=£ 319.57	52. ACTUAL	=£ 681.25
PROFIT PER HR	:	53. ESTD:	=£ 23.89	54. ACTUAL	=£ 36.82
PERCE: PROFIT	:	55. ESTD:	= 15.42%	56. ACTUAL	= 39.62%

Appendix 16

Weekly downtime record

SECTIONS AND PROFILES LIMITED

All downtime details for week ended Sunday 14.09.86

Code	Downtime reason	Total hrs
1	SETTING UP	21.0800
2	STRIP DOWN	0.0000
3	COIL CHANGE	19.1200
4	LENGTH CHANGE	0.0000
5	WAITING FOR INFORMATION	0.0000
6	FAULTY ROLLS	0.0000
7	DEFECTIVE TOOL	2.3450
8	MACHINE FAULT	0.0000
9	TOOL TRIALS	0.0000
10	REJECT MATERIAL	0.0000
11	MATERIAL SHORTAGE	0.0000
12	MEAL BREAK	0.0000
13	DOWNTIME WITHOUT A REASON	1.2300
14	NO WORK	0.0000
15	NO LABOUR	0.0000

TOTAL DOWNTIME FOR THE WEEK IS 43.7750 hrs

Appendix 17

Operators' working details

METAL CEILINGS LIMITED

All operator working details for week ended Sunday 21.09.86

Num	Clock	Operator's name	Leng(m)	Bonus(m)	Tim(Hrs)
1	1147		425.00	601.65	36.20
2	1156		374.70	847.15	39.16
3	1157		231.12	231.12	37.01
4	1161		235.23	602.34	34.04
5	1164		345.67	700.23	38.31
6	1165		235.56	502.12	37.34
7	1166		412.27	700.26	39.45

Do you want a hard copy (Y/N) ?

Appendix 18

Status file

SECTIONS AND PROFILES LIMITED

Line status as on 29-APR-87 at 13:14hrs

Job number	- 2	Line	- 1	Comp. time	- 8.8335hrs
Length number	- 1	Length-	1000 mm	Qty produced	- 345
Current Speed - 220.4 m/hr				Target	- 200 m/hr

No job on line 2

Job number	- 1	Line	- 3	Comp. time	- 124.52hrs
Length number	- 1	Length-	1000 mm	Qty produced	- 0
Current Speed	- 304.5 m/hr			Target	- 200 m/hr

No job on line 4

Appendix 19

History of a previous job

SECTIONS AND PROFILES LIMITED

Job number 2

1147 STARTED AT 18:23hrs ON 5.9.86
DOWNTIME 1 Started AT 18:24 hrs ON 5.9.86
DOWNTIME 1 finished AT 19:46 hrs ON 5.9.86
DOWNTIME 13 Started AT 19:46 hrs ON 5.9.86
DOWNTIME 13 finished AT 19:47 hrs ON 5.9.86
ROLLING started AT 19:47 hrs ON 5.9.86
DOWNTIME 3 Started AT 20:28 hrs ON 5.9.86
DOWNTIME 3 finished (gau. 55mm) AT 20:35 hrs ON 5.9.86
DOWNTIME 13 Started AT 20:35 hrs ON 5.9.86
DOWNTIME 13 finished AT 20:35 hrs ON 5.9.86
ROLLING started AT 20:36 hrs ON 5.9.86
JOB SUSPENDED AT 21:56 hrs ON 5.9.86
1147 LOGGED OUT & SHUT DOWN AT 21:57 hrs ON 5.9.86
1147 RE-STARTED THE JOB AT 05:59 hrs ON 6.9.86
DOWNTIME 4 Started AT 07:05 hrs ON 6.9.86
DOWNTIME 4 finished AT 07:31 hrs ON 6.9.86
DOWNTIME 13 Started AT 07:31 hrs ON 6.9.86
DOWNTIME 13 finished AT 07:32 hrs ON 6.9.86
DOWNTIME 6 Started AT 07:32 hrs ON 6.9.86
DOWNTIME 6 finished AT 07:45 hrs ON 6.9.86
DOWNTIME 13 Started AT 07:46 hrs ON 6.9.86
DOWNTIME 13 finished AT 07:46 hrs ON 6.9.86
ROLLING started AT 07:47 hrs ON 6.9.86
1147 LOGGED OUT AT 14:02 hrs ON 6.9.86
1156 LOGGED IN AT 14:03 hrs ON 6.9.86
OVER PRODUCED AT 15:04 hrs ON 6.9.86
OVER PRODUCED - not accepted AT 15:04 hrs ON 6.9.86
DOWNTIME 2 Started AT 15:40 hrs ON 6.9.86
DOWNTIME 2 finished AT 16:21 hrs ON 6.9.86
DOWNTIME 13 Started AT 16:21 hrs ON 6.9.86
JOB FINISHED AT 16:22 hrs ON 6.9.86

Appendix 20

Raw material card

S/P ++++++++++ RAW MATERIAL CARD ++++++++++

P.CODE :22223416 0101 CUSTOMER : LINK51 LIMITED
MAT. : 1.100mmX229.0mm ZIN

No	Date	Ref.	Supplier	Pri/ton	Quantity	Status	Balance
1	2.11.86	ZXC	WEST	£ 210.00	3.600tons	RECEIVED	6.300tons
2	5.11.86	ASD	ASK	£190.00	14.500tons	RECEIVED	20.800tons
3	13.11.86	AS	many	£ 210.00	2.100tons	ISSUED	18.700tons
4	23.11.86	XCV	ASK	£ 185.00	4.000tons	PLACED	18.700tons
5	24.11.86	XCV	ASK	£ 185.00	2.500tons	RECEIVED	21.200tons
6	25.11.86	XCV	ASK	£ 185.00	2.500tons	RECEIVED	23.700tons
7	25.11.86	ASD	ASK	£ 185.00	1.200tons	REJECTED	22.500tons
8	26.11.86	XCV	ASK	£ 185.00	2.400tons	RECEIVED	24.900tons
9	30.11.86	SP23	many	£ 210.00	3.200tons	ISSUED	21.700tons
10	1.12.86	AZXC	WEST	£ 210.00	23.000tons	RECEIVED	44.700tons
11	2.12.86	WE1	WEST	£ 210.00	2.100tons	REJECTED	42.600tons
12	3.12.86	WES2	WEST	£ 210.00	2.300tons	RECEIVED	44.900tons
13	4.12.86	SDG	WEST	£ 210.00	3.400tons	PLACED	44.900tons
14	5.12.86	SDFG	WEST	£ 210.00	1.100tons	RECEIVED	46.000tons
15	6.12.86	SDF	WEST	£ 210.00	2.200tons	RECEIVED	47.200tons

Press <RETURN> to go to the next page

S/P RAW MATERIAL FOR 22223416 0101 ------- page 2

No	Date	Ref.	Supplier	Pri/ton	Quantity	Status	Balance
16	7.12.86	SDF	WEST	£ 210.00	1.300tons	RECEIVED	48.500tons
17	7.12.86	SDF	WEST	£ 210.00	0.500tons	REJECTED	48.000tons
18	8.12.86	SDF	WEST	£ 210.00	0.500tons	RECEIVED	48.500tons
19	12.12.86	BN	ASK	£ 185.00	2.100tons	PLACED	48.500tons
20	13.12.86	BN	ASK	£ 185.00	2.100tons	RECEIVED	50.600tons
21	14.12.86	ER	WEST	£ 210.00	1.200tons	PLACED	50.600tons
22	15.12.86	ER	WEST	£ 210.00	1.200tons	RECEIVED	51.800tons
23	15.12.86	FG	ASK	£ 190.00	2.100tons	PLACED	51.800tons
24	17.12.86	ADF	ASK	£ 190.00	1.100tons	RECEIVED	52.900tons
25	18.12.86	SDF	WEST	£ 210.00	1.200tons	RECEIVED	54.100tons
26	19.12.86	WXC	ASK	£ 190.00	2.300tons	PLACED	54.100tons
27	19.12.86	ASD	ASK	£ 190.00	1.300tons	RECEIVED	55.400tons

Press 1 to go to the menu or 0 to try again

Appendix 21

Finished product file

FINISHED PRODUCTS as at 29-APR-87 13:20 hrs

(Last updated on - 4-FEB-87 at 17:29 hrs)

S/P ++++++ FINISHED PRODUCT DETAILS FOR 11113456 0101++++++

CUSTOMER:LINK51 LIMITED MATERIAL : 1.100mmX229.0mm ZIN
LENGTH : 1000.00mm DESCRIP. : CHICKEN TROUGH

No	Date	Reference	Quantity	Status	Balance
1	0. 0. 0	INI. BAL	0 nos	AUDITED	0 nos
2	2. 9.86	ASD	200 nos	PRODUCED	200 nos
3	4. 4.86		10 nos	AUDITED	210 nos

S/P ++++++ FINISHED PRODUCT DETAILS FOR 2222 ++++++

CUSTOMER:LINK51 LIMITED MATERIAL : 1.100mmX229.0mm ZIN
LENGTH : 1200.00mm DESCRIP. :CHICKEN TROUGH

No	Date	Reference	Quantity	Status	Balance
1	0. 0. 0	INI. BAL	200 nos	AUDITED	200 nos
2	23. 1.87	-	0 nos	AUDITED	200 nos
3	24. 1.87	QASD	234 nos	PRODUCED	434 nos
4	25. 1.87	WEST	267 nos	DESPTCHD	167 nos
5	30. 1.87	WEST1	23 nos	REJECTED	190 nos
6	2. 2.87	SP101	400 nos	PRODUCED	590 nos

Press <RETURN> to continue

Appendix 22

Instantaneous stock value

METAL CEILINGS LIMITED

RAW MATERIAL POSITION as at - 30-APR-87 16:12 hrs

Prd code	Material description	Amt-tn	P/tn£	Val(£)	Date	C
22223416	1.100mmX229.0mm ZIN	2.400	185.00	444.00	26.11.86	A
22223416	1.100mmX229.0mm ZIN	20.900	210.00	389.00	1.12.86	A
22223416	1.100mmX229.0mm ZIN	2.300	210.00	483.21	3.12.86	B
22223416	1.100mmX229.0mm ZIN	1.200	210.00	252.00	6.12.86	B
22223416	1.100mmX229.0mm ZIN	0.800	210.00	168.34	7.12.86	B
13478213	1.300mmX261.0mm GAL	0.500	210.00	105.61	8.12.86	B
13478213	1.300mmX261.0mm GAL	1.200	210.00	252.00	15.12.86	B
13478213	1.300mmX261.0mm GAL	1.200	210.00	252.00	18.12.86	B
13478213	1.300mmX261.0mm GAL	1.300	190.00	247.08	19.12.86	B
13478213	1.300mmX261.0mm GAL	2.100	210.00	441.77	22.12.86	B
3333	1.100mmX165.0mm S/S	4.300	300.00	290.00	24.10.86	C

TOTAL VALUE OF RAW MATERIAL as at 30-APR-87 16:12 hrs = £ 4235.01

Key : A - less than 3 month old ; B - 3 to 6 months old
C - 6 to 12 months old ; D - over 12 months old

SECTIONS AND PROFILES LIMITED

FINISHED PRODUCT POSITION as at - 30-APR-87 16:13 hrs

Prd.Code	Cust.name	Description	Len-mm	Qty	£/100m	Val(£)	Date	C
2222	LINK51 LTD	CHICKEN TRO	1200.00	400	97.00	465.60	2. 2.87	A
2222	LINK51 LTD	CHICKEN TRO	1200.00	23	97.00	26.77	30. 1.87	B
2222	LINK51 LTD	CHICKEN TRO	1200.00	167	97.00	194.39	24. 1.87	B
11113456	LINK51 LTD	CHICKEN TRO	1000.00	200	0.00	0.00	2. 9.86	C
1111	LINK51 LTD	CHICKEN TRO	2000.00	300	78.60	471.60	3. 5.86	C
11113456	LINK51 LTD	CHICKEN TRO	1000.00	10	0.00	0.00	4. 4.86	D

TOTAL VALUE OF FINISHED PRODUCTS as at 30-APR-87 16:13 hrs = £ 1158.36

Note - Price per 100m $0.00 indicates pricing card incomplete.

Key : A - less than 3 month old ; B - 3 to 6 months old
C - 6 to 12 months old ; D - over 12 months old

Appendix 23

Master menu and secondary menu

MASTER MENU
********* ******

1. SALES PRICING CARD
2. RAW MATERIAL CARD
3. FINISHED PRODUCT CARD
4. STOCK VALUE
5. REJECT FILE
6. JOB CARD
7. SCHEDULING
8. PRODUCTION REPORT
9. WEEKLY WORKING DETAILS
10. WEEKLY DOWN TIME RECORD
11. JOB MONITORING
12. LOG FILES OF PREVIOUS JOBS
13. GO TO THE SECONDARY MENU
14. CHANGE THE PASSWORD
15. EXIT

Select from 1 to 15 above :

SECONDARY MENU
********** ******

1. GAUGE TOLERANCES
2. LINE PREFERENCE
3. COMPANY HOLIDAYS
4. PRESS OPERATIONS DATA
5. CUSTOMERS
6. OPERATORS
7. EXIT TO MASTER MENU

Select from 1 to 7 above : 7

Appendix 24

Users

HADLEY INDUSTRIES PLC

AUTHORIZED USERS OF THE SYSTEM as on 30-APR-87 at 16:21hrs

(Last updated on 16-MAR-87 at 13:50hrs)

No	UIC	Owner's name	Default directory	User name	P	C
1	[201,1]		[MASTER]			¥
2	[201,2]		[CAM]			¥
3	[201,3]		[CAM]			¥
4	[201,4]		[CAM]			¥
5	[201,20]		[CAM]			2
6	[201,21]		[CAM]			2
7	[201,22]		[CAM]			2
8	[201,23]		[CAM]			2
9	[201,24]		[CAM]			2
10	[201,25]		[CAM]			2
11	[201,30]		[CAM]			3
12	[201,31]		[CAM]			3
13	[201,32]		[CAM]			3
14	[201,33]		[CAM]			3
15	[201,111]		[MASTER.BAY1.LINE1]			1
16	[201,112]		[MASTER.BAY1.LINE2]			1
17	[201,113]		[MASTER.BAY1.LINE3]			1
18	[201,114]		[MASTER.BAY1.LINE3]			1
19	[201,121]		[MASTER.BAY2.LINE1]			2
20	[201,122]		[MASTER.BAY2.LINE2]			2
21	[201,123]		[MASTER.BAY2.LINE3]			2
22	[201,124]		[MASTER.BAY2.LINE4]			2
23	[201,131]		[MASTER.BAY3.LINE1]			3
24	[201,132]		[MASTER.BAY3.LINE2]			3
25	[201,133]		[MASTER.BAY3.LINE3]			3
26	[201,133]		[MASTER.BAY3.LINE4]			3
27	[201,752]		[CAM]			2
28	[201,762]		[CAM]			2

Note: ¥= special users who can select any company

Appendix 25

Directory of modules

ALL FILES IN THE SYSTEM - as on 30-APR-87 at 16:20 hrs

(Last updated on - 7-JAN-86 at hrs)

No	File name	Description
1	AUTHORIZE	USER AUTHORIZATION UTILITY
2	AUTOLOGO	AUTOLOGOUT BACKGROUND JOB
3	BACKUPS	FILE BACKUP UTILITY
4	BACKSCHE	GAP ELIMINATION FOR SCHEDULING
5	BAY	COMPANY SELECTION UTILITY FOR TOP USERS
6	BAY3	A SECURITY ROUTINE
7	BONUS	BONUS CALCULATION PROGRAM
8	BRIEF	TO PERFORM BRIEF OR A FULL LISTING
9	BYE	TO WISH BYE AT THE END OF A SESSION
10	CODE	A SECURITY ROUTINE
11	COMPANY	SUBSIDIARY COMPANIES IN THE GROUP
12	CHEPUSH	TO PUSH ALL LATER WORK TO CLOSE GAPS IN SCHEDULING
13	CHESCHE	TO CHECK THE SCHEDULING CAPABILITY
14	CUSPROB	DELETES THE LAST RECORD OF THE CUSTOMER FILE
15	CUSTOMER	CUSTOMER DATA FILE
16	DESCRIP	MATERIAL DESCRIPTION
17	DIRECTORY	THIS FILE
18	DOWNDET	DOWNTIME DETAILS
19	DOWNTIME	DOWNTIME RECORD FROM THE SHOP FLOOR
20	DUMMY	RESET THE No. OF RECORDS IN ABORTING A NEW ENTRY
21	EARLDEL	DELIVERY DATE
22	FINPROB	DELETES THE LAST RECORD OF THE FINISHED GOODS FILE
23	FINPROD	FINISHED GOODS FILE
24	GAUGE	GAUGE TOLERANCES FILE
25	GAUPROB	DELETES THE LAST RECORD OF THE GAUGE TOLERANCES FILE
26	GET	A SECURITY ROUTINE
27	GETTIME	TIME OF AN OPERATION
28	HELLO	TO GREET HELLO AT THE START OF A SESSION
29	HOLIDAYS	COMPANY HOLIDAYS FILE
30	HOLPROB	DELETES THE LAST RECORD OF THE HOLIDAYS FILE
31	INVALID	TO TRAP AN INVALID SELECTION
32	JOBCARD	JOBCARD FILE
33	JOBPROB	DELETES THE LAST RECORD OF THE JOBCARD FILE
34	KEY	A SECURITY ROUTINE
35	LENGTH	LENGTH OF A SECTION
36	LINE	LINE PREFERENCE DATA
37	LINPROB	DELETES THE LAST RECORD OF LINE PREFERENCE DATA FILE
38	LOGFILE	LOGFILES OF PREVIOUS FILES
39	MAT	MATERIAL DESCRIPTION
40	MATERIAL	EXTRA MATERIAL DESCRIPTION
41	MAXTHICK	MAXIMUM GAUGE
42	MINTHICK	MINIMUM GAUGE
43	MEANTHICK	MEAN GAUGE

No	File name	Description
44	MENU	MASTER MENU
45	MENPROB	ACCESS FOR THE LAST RECORD DELETION FACILITY
46	NAME	CUSTOMER DETAILS SETTING AND EXTRACTION
47	NUMPASS	NUMBER OF PASSES
48	OPERATOR	OPERATOR DATA FILE
49	OPPROB	DELETES THE LAST RECORD OF THE OPERATOR FILE
50	OPERDAT	OPERATORS' WORKING DETAILS FOR THE PERIOD
51	OPNAME	OPERATOR NAME EXTRACTION FOR THE SHOP FLOOR
52	ORDERNO	REFERENCE NUMBER
53	PASS	RESETS PASSWORD OF A USER
54	PREFERENCE	ACCESS ROUTINE FOR LINE PREFERENCE DATA
55	PRESSDATA	ACCESS ROUTINE FOR PRESS OPERATIONS DATA
56	PRESS	PRESS OPERATIONS DATA FILE
57	PRESPROB	DELETES THE LAST RECORD OF THE PRESS DATA FILE
58	PRICARD	SALES PRICING CARD
59	PRICE	PRICE OF A MATERIAL
60	PRICING	ACCESS ROUTINE TO SALES PRICING CARD
61	PRIPROB	DELETES THE LAST RECORD OF THE PRICING CARD FILE
62	PRODCODE	PRODUCT CODE
63	PRODCREA	AUTOMATIC PRODUCTION REPORT CREATION
64	PRODMAN	MANUAL PRODUCTION REPORT CREATION
65	PRODREP	PRODUCTION REPORT
66	PRODPROB	DELETES THE LAST RECORD OF THE PRODUCTION REPORT
67	PRODST	START OF PRODUCTION OF A JOB
68	PROPS	NUMBER OF PRESS OPERATIONS
69	PULLOTHER	TO PULL OTHER JOBS TO CLOSE GAPS IN SCHEDULING
70	QUANTITY	QUANTITY OF A RAW MATERIAL
71	QUOTED	THE LENGTH ON WHICH THE CALCULATIONS ARE BASED
72	RAWMAT	RAW MATERIAL CARD
73	RAWPROB	DELETES THE LAST RECORD OF THE RAW MATERIAL CARD
74	RAWCHECK	QUICK CHECKING FACILITY FOR THE RAW MATERIAL FILE
75	REJECT	REJECT GOODS FILE
76	REJPROB	DELETES THE LAST RECORD OF THE REJECT GOODS FILE
77	RESCHE	MANUAL RE-SCHEDULING ROUTINE
78	REPLACE	UTILITY TO REPLACE CORRUPT FILES
79	ROLLEST	ESTIMATED ROLL SPEED
80	SCALE	SCALE FACTOR
81	SCHE	AUXILIARY ROUTINE FOR SCHEDULING
82	SCHEDULE	MAIN SCHEDULING FILE
83	SCHEPROB	DELETES THE LAST RECORD OF THE SCHEDULING FILE
84	SETEST	ESTIMATED SET UP TIME
85	SETTIME	SYNCHRONIZES TIME WITH THE SHOP FLOOR COMPUTERS
86	SPAWN	TO SPAWN A JOB TERMINATION MESSAGE
87	STATUS	JOB STATUS FILE
88	STRIPEST	ESTIMATED STRIP TIME
89	SET	AUXILIARY ROUTINE TO SET A PASSWORD
90	SUPPLIER	RAW MATERIAL SUPPLIER
91	SURE	FOR DATA CONFIRMATION
92	SECONDARY	ACCESS ROUTINE TO SECONDARY MODULES
93	TIED	TIED JOBS ATTACHMENT
94	TYPE	RAW MATERIAL TYPE
95	UPSCHE	TO SCHEDULE AFTER CLOSING THE GAPS

No	File name	Description
96	UTILS	ACCESS ROUTINE TO THE SYSTEM UTILITIES
97	UNAUT	UNAUTHORIZED USERS' FILE
98	VALUE	INSTANTANEOUS STOCK VALUES
99	WEEKDAY	DAY OF THE WEEK
100	WIDE	WIDTH OF A RAW MATERIAL
101	WORKDET	PERIODICAL WORKING DETAILS
102	WRISCHE	WRITES SCHEDULING DETAILS TO THE FILE

Appendix 26

System subroutines

SUBROUTINES IN THE SYSTEM - as on 30-APR-87 at 16:16 hrs

(Last updated on - 18-SEP-86 at 17:21 hrs)

No	Name	Bk	Variables	Description
1	JOBNUM(M)	3	BAR	ACKNOWLEDGEMENT NUMBER
2	PRODCODE(M)	7	PC	PRODUCT CODE (TEMPORARY)
3	CUSDET(M)	-		HEADING FOR CUSTOMERS
4	NAME(M)	¥	(MANY)	CUSTOMER DATA EXTRACTION
5	ORDERNUM(M)	13	XAR	OWN ORDER NUMBER FOR CUSTOMER
6	MATERIAL	¥	(MANY)	MATERIAL DESCRIPTION
7	LENG(M,N)	2	(MANY)	LENGTH NUMBER FOR A CUSTOMER
8	SETEST	1	SET,SETC,UC,MET	ESTIMATED SET TIME
9	ROLLEST	36	ROT,ROTC,UC	ESTIMATED ROLL SPEED
10	STRIPEST	37	STT,STTC,UC,MET	ESTIMATED STRIP TIME
11	SETACT(M)	45	SEAC	ACTUAL SET TIME FOR OPERATION
12	ROLLACT(M)	46	ROAC	ACTUAL ROLL SPEED FOR OPERATION
13	STRIPACT(M)	47	STAC	ACTUAL STRIP TIME FOR OPERATION
14	QUEST	4	ANS	TO ACCEPT DATA OR REJECT
15	CORRECT	5	COR	SELECTION OF THE ERRONEOUS VALUE
16	PASS	10	NPS	NUMBER OF PASSES
17	DESCRIP	11	DES	DESCRIPTION OF THE FINAL PRODUCT
18	SCALE	12	SC	SCALE FACTOR
19	SURE	4	ANS	DATA CONFIRMATION
20	THICKNESS	¥	(MANY)	READ GAUGE FROM FILE
21	WIDE	5	(MANY)	TO CALCULATE MEAN WIDTH
22	SPINST(M)	16	DAR	RAW MATERIAL AVAILABILITY
24	SELLPRI(M)	21	CAR	ACTUAL SELLING PRICE
25	CODE	24	(MANY)	PASSWORDS
26	ALTER	24	(MANY)	PASSWORDS
27	INCOMP	-	-	VALUES CANNOT BE CHANGED
28	LEN	-	-	ROLLING LENGTHS
29	EXCESS(M)	25	EAR	EXCESS MAT INSTRUCTIONS
30	INVALID	-	-	TO TRAP AN INVALID ENTRY
31	PRICING(M)	¥	(MANY)	TO READ SALES PRICING CARD
32	TONPRI	30	PT	PRICE PER TON
33	QUOTED	31	MET	NO OF METRES TO BASE THE PRICES
34	UNITWT	32	(MANY)	CONVERSION FACTOR FOR WEIGHT
35	SCRAP	33	NSCR	PERCENTAGE OF SCRAP
36	MATPRICE	¥	(MANY)	PRICE PER 100m
37	PROFQUES	38	(MANY)	TO BASE THE PROFIT ON
38	PROFSORT	¥	(MANY)	TO SORT PROFIT
39	COMMISSION	39,	COM,CPER,CDET	COMMISSION PER 100m
40	TOOL	40	TOOM	TOOL AMORTIZATION PER 100m
41	TRANSPORT	41	(MANY)	TRANSPORT COST
42	SELLPRI	¥	(MANY)	SELLING PRICE
43	NOCHANGE	-	-	NON CHANGEABLE VALUES
44	OVERHEAD	50	UC	OVERHEAD COST PER HOUR
45	ROLLDATE	43	RD	DATE OF OPERATION
46	MACHINE(KK	44	ACH(M)	MACHINE USED FOR OPERATION

No	Name	Bk	Variables	Description
47	COMMENTS	48	OP	COMMENTS
48	TARGETS	¥	(MANY)	TARGET CALCULATION
49	LENTOL	73	XLMAX,XLMIN	LENGTH TOLERANCE
50	ADDITIONAL	74	(MANY)	ADDITIONAL OPERATIONS AND COSTS
51	MEANTHICK	15	THICK	MEAN THICKNESS
52	MUNTHICK	76	THIMIN	MINIMUM THICKNESS
53	MAXTHICK	77	THIMAX	MAXIMUM THICKNESS
54	TYPE	75	TY	TYPE OF MATERIAL
55	ORDDATE(M)	63	(MANY)	DATE OF ORDER
56	ORDERNO	64	ORD	ORDER NUMBER
57	QUANTITY	65	QUAN(M)	TRANSACTED QUANTITY
58	PRICE	66	PRI(M)	PRICE PER TON FOR A TRANSACTION
59	BALANCE	67	(MANY)	BALANCE OF A TRANSACTION
60	STATUS(M)	69	STST(M)	STATUS OF KK'TH OPERATION
61	SUPPLIER	68	SU	SUPPLIER OF RAW MATERIAL
62	DUMMY	27	I,MAX	DUMMY TO RESET
63	ACTUALSEL	17	(MANY)	ACTUAL SELLING PRICES AND LENGTHS
64	INFORM	82	ADINF	ADDITIONAL INFORMATION
65	BRIEF	29	IBRIEF	BRIEF DETAILS OR NOT IN A LISTING
66	ORDERDAT	51	(MANY)	ORDER DATE OF A CUSTOMER
67	DELIVARYDAT	52	(MANY)	DELIVERY DATE FOR A CUSTOMER
68	IDENTIFI	56	UIC	USER IDENTIFICATION CODE
69	OWNER	58	PER	OWNER'S NAME
70	USER_NAME	59	UNAM	USERNAME
71	PRIORITY	58	PASS	PRIORITY LEVEL PASSWORD
72	COMPANY	19	BAY	COMPANY NUMBER
73	DIRECTORY	60	FIR	FIRST NAME
74	EARL_DEL	¥	(MANY)	EARLIEST REQD DELIVERY DATE
75	RE_SCHE	¥	(MANY)	SCHEDULING
76	CHE_SCHE	¥	(MANY)	CHECK SCHEDULE
77	WRI_SCHE	¥	(MANY)	WRITE THE SCHEDULED DATA
78	WEEKDAY	49	RQDAY,DA	WEEKDAY OF A GIVEN DATE
79	REASON	94	REA	DOWNTIME REASON

Note: ¥ = more than 1 common block in the subroutine

Appendix 27

System variables

VARIABLES IN THE SYSTEM - as at 30-APR-87 at 16:17 hrs

(Last updated on - 22-SEP-86 at 10:40 hrs)

No	Variable	Type	Description
1	I	I6	RECORD NUMBER
2	MAX	I6	NO. OF RECORDS THAT EXIST
3	XJN(M)	A16	ACKNOWLEDGEMENT NUMBER FOR Mth CUSTOMER
4	PC	A16	PRODUCT CODE
5	NPS	I2	NUMBER OF PASSES
6	DES	A24	JOB DESCRIPTION
7	SC	F4.2	SCALE FACTOR
8	K(M)	I2	NO. OF LENGTHS FOR Mth CUSTOMER (UP TO 15)
9	QLEN(M,N)	F8.2	Nth LENGTH FOR Mth CUSTOMER (BEFORE SORTING)
10	LQTY(M,N)	I5	Nth QTY FOR Mth CUSTOMER (BEFORE SORTING)
11	YLEN(M,N)	F8.2	TEMPORARY LENGTH VALUE FOR SORTING
12	JQTY(M,N)	I5	TEMPORARY QUANTITY VALUE FOR SORTING
13	XLEN(M)	F8.2	SORTED LENGTH (FOR ROLLING)
14	IQTY(M)	I5	SORTED QUANTITY VALUE FOR ROLLING
15	QL	F8.2	LENGTH (TEMPORARY)
16	NQ	I5	QUANTITY (TEMPORARY)
17	NCUS	I1	NUMBER OF CUSTOMERS (UP TO 6)
18	XNAME(M)	A46	NAME OF Mth CUSTOMER
19	ORNO(M	A16	OWN ORDER NUMBER OF Mth CUSTOMER
20	PAC(M)	A46	PACKING INSTRUCTIONS FOR Mth CUSTOMER
21	XMAT	A24	MATERIAL DESCRIPTION
22	THIMAX	F6.3	MAXIMUM THICKNESS
23	THIMIN	F6.3	MINIMUM THICKNESS
24	WIDMAX	F7.2	MAXIMUM WIDTH
25	WIDMIN	F7.2	MINIMUM WIDTH
26	SPI(M)	A46	SPECIAL INSTRUCTIONS FOR Mth CUSTOMER
27	ROTG	F6.1	ROLLING TARGET
28	SETG	F6.1	SETTING TARGET
29	STTG	F6.1	STRIPPING TARGET
30	ROA(KK)	F6.1	ACTUAL ROLLING TIME OF KKth OPERATION (UP TO 15)
31	SEA(KK)	F6.1	ACTUAL SET TIME OF KKth OPERATION (UP TO 15)
32	STA(KK)	F6.1	ACTUAL STRIP TIME OF KKth OPERATION (UP TO 15)
33	ESWT	F7.2	ESTIMATED WEIGHT OF RAW MATERIAL
34	AVA	A25	MATERIAL AVAILABILITY (YES/NO/PART)
35	VAR	F7.2	TEMPORARY ACTUAL AND TARGET VALUES
36	WAR	A46	CUSTOMER NAME (TEMPORARY)
37	XAR	A16	ORDER NUMBER (TEMPORARY)
38	PRD	A16	PRODUCT CODE
39	ZAR	A46	PACKING INSTRUCTIONS (TEMPORARY)
40	AAR	A16	CUSTOMER CODE (TEMPORARY)
41	BAR	A16	ACKNOWLEDGEMENT NUMBER (TEMPORARY)
42	CAR	-	DUMMY FOR CALCULATING SELLING PRICE SP(M)
43	ANS	A1	VARIABLE FOR YES/NO (Y/N) - COMMON
44	BNS	A1	JOBFILE OR PROD REPORT MENUS SELECTION VARIABLE

No	Variable	Type	Description
45	CNS	A1	SELECTION VARIABLE FOR A SMALL MENU
46	COR	I2	TO ASK WHAT VALUE IS WRONG
47	M	-	DUMMY VARIABLE FOR COUNTING (RECORDS ETC)
48	MA	-	DUMMY VARIABLE FOR SORTING RECORDS
49	J	-	VARIABLE FOR MENU OPTIONS
50	UC	F6.2	UNIT OVERHEAD COST
51	PT	F7.2	UNIT COST OF RAW MATERIAL
52	EMC	F8.2	ESTIMATED TOTAL RAW MAT COST
53	PRI	F8.2	ESTIMATED TOTAL OF SELLING PRICES
54	N	-	NUMBER OF LENGTHS FOR A CUSTOMER
55	IJ	-	NUMBER OF LENGTHS OR QUANTITIES FOR SORTING
56	IL	-	NUMBER OF LENGTHS OR QUANTITIES AFTER SORTING
57	XNAM	A4	EXTRACTION VARIABLE OF A CUSTOMER NAME
58	CODD(M)	A16	CODE FOR Mth CUSTOMER
59	EXE(M)	A46	EXCESS MAT. INSTRUCTIONS FOR Mth CUSTOMER
60	BOK	A2	MAIN MENU SELECTION
61	PER	F5.1	PERCENTAGE INCREASE OF ROLL TARGET
62	DIR	A19	DEFAULT DIRECTORY
63	PASS	A20	PRIVILEGE LEVEL PASSWORD
64	COD	A20	VARIABLE FOR CHECKING THE PASSWORD
65	MR	-	NUMBER OF USERS
66	DTC	F6.2	DOWN TIME FOR COIL CHANGE
67	DTD	F6.2	DOWN TIME-WAITING FOR INFORMATION
68	DTE	F6.2	DOWN TIME-FAULTY ROLLS
69	DTF	F6.2	DOWN TIME-DEFECTIVE TOOL
70	DTG	F6.2	DOWN TIME-MACHINE FAULT
71	DTH	F6.2	DOWN TIME-TOOL TRIALS
72	DTI	F6.2	DOWN TIME-REJECT MATERIAL
73	DTJ	F6.2	DOWN TIME-MATERIAL SHORTAGE
74	DTK	F6.2	DOWN TIME-MEAL BREAK
75	DTN	F6.2	DOWN TIME-NO WORK
76	DTM	F6.2	DOWN TIME-NO LABOUR
77	TJTE	F6.2	ESTIMATED TOTAL JOB TIME
78	TJTA	F6.2	ACTUAL TOTAL JOB TIME
79	TRTE	F6.2	ESTIMATED TOTAL RUN TIME
80	TRTA	F6.2	ACTUAL TOTAL RUN TIME
81	TPT	F6.2	TOTAL PRODUCTIVE TIME
82	EF	F6.2	EFFICIENCY
83	AMC	F8.2	ACTUAL MATERIAL COST
84	OHE	F8.2	ESTIMATED TOTAL OVERHEAD COST
85	OHA	F8.2	ACTUAL OVERHEAD COST
86	XIS1E	F8.2	ESTIMATED MISCELLANEOUS COST 1
87	XIS1A	F8.2	ACTUAL MISCELLANEOUS COST 1
88	XIS2E	F8.2	ESTIMATED MISCELLANEOUS COST 2
89	XIS2A	F8.2	ACTUAL MISCELLANEOUS COST 2
90	XIS3E	F8.2	ESTIMATED MISCELLANEOUS COST 3
91	XIS3A	F8.2	ACTUAL MISCELLANEOUS COST 3
92	TJCE	F8.2	ESTIMATED TOTAL COST
93	TJCA	F8.2	ACTUAL TOTAL COST
94	PRIA	F8.2	ACTUAL TOTAL OF SELLING PRICES
95	PLE	F8.2	ESTIMATED PROFIT/LOSS
96	PLA	F8.2	ACTUAL PROFIT/LOSS
97	UPLE	F8.2	ESTIMATED PROFIT PER HOUR

No	Variable	Type	Description
98	UPLA	F8.2	ACTUAL PROFIT PER HOUR
99	PPLE	F6.2	ESTIMATED PERCENTAGE PROFIT
100	PPLA	F6.2	ACTUAL PERCENTAGE PROFIT
101	IFE(M,N)	I3	NO OF FEET IN LEN. N FOR Mth CUSTOMER
102	INS(M,N)	I2	NO OF INCHES IN LEN N FOR Mth CUSTOMER
103	FT(M,N)	A1	FEET OR mm FOR Nth LENGTH OF Mth CUSTOMER
104	P(M,N)	F7.3	UNIT PRICE OF LEN N FOR Mth CUSTOMER
105	PR	F7.3	UNIT PRICE OF A LENGTH (TEMPORARY)
106	MM	-	TOTAL SELLING PRICE (TEMPORARY)
107	EAR	A46	EXCESS MATERIAL INSTRUCTIONS (TEMPORARY)
108	DAR	A46	SPECIAL INSTRUCTIONS (TEMPORARY)
109	NNU(M,N)	I2	NUMERATOR OF INCH FRACTION
110	NDE(M,N)	I2	DENOMINATOR OF INCH FRACTION
111	SP(M)	F8.2	TOTAL SELLING PRICE FOR Mth CUSTOMER
112	PNE	A20	NEW USER'S NAME
113	DTL	F6.2	DOWN TIME - LENGTH CHANGE
114	ACWT	F7.2	ACTUAL WEIGHT OF RAW MATERIAL
115	THICK	F6.3	MEAN THICKNESS
116	WIDTH	F5.1	MEAN WIDTH
117	XLMAX	F4.1	LENGTH TOLERANCE (+ VALUE)
118	XLMIN	F4.1	LENGTH TOLERANCE (-VALUE)
119	TMAX(M)	F4.1	LENGTH TOLERANCE (+ VALUE) FOR Mth CUSTOMER
120	TMIN(M)	F4.1	LENGTH TOLERANCE (-VALUE) FOR Mth CUSTOMER
121	ACWT	F7.2	ACTUAL WEIGHT OF RAW MATERIAL
122	CUS	A46	CUSTOMER NAME
123	MET	I6	LENGTH WHICH THE PRICE IS BASED ON
124	SCR	F4.1	SCRAP PERCENTAGE OF RAW MATERIAL WEIGHT
125	MATPRI	F6.2	PRICE OF MATERIAL FOR 100m
126	SET	F4.1	ESTIMATED SET UP TIME
127	ROT	F4.1	ESTIMATED ROLL SPEED
128	STT	F4.1	ESTIMATED STRIP TIME
129	PERC	F4.1	PERCENTAGE PROFIT
130	PHR	F6.2	PROFIT PER HOUR
131	PROF	F6.2	PROFIT PER 100m
132	UWT	F6.2	WEIGHT OF MATERIAL WITHOUT SCRAP
133	GWT	F6.2	WEIGHT OF MATERIAL WITH SCRAP
134	ADI	F6.2	ADDITIONAL OPERATIONS COST PER 100m
135	COM	F6.2	COMMISSION PER 100m
136	TOOM	F6.2	TOOL AMORTIZATION PER 100m
137	TRAT	F6.2	TRANSPORT COST PER TON (NOT FOR THE SCRAP)
138	TRAM	F6.2	TRANSPORT COST PER 100m
139	SELM	F6.2	SELLING PRICE PER 100m
140	SELF	F6.2	SELLING PRICE PER 100ft
141	SEM(M)	F6.2	SELLING PRICE FOR Mth CUSTOMER
142	SEF(M)	F6.2	SELLING PRICE FOR Mth CUSTOMER
143	KK	-	COUNTING VARIABLE OF OPERATIONS
144	MV	-	DUMMY
145	MW	-	DUMMY
146	IOK	I2	MENU SELECTION VARIABLE
147	IOB	I2	MENU SELECTION VARIABLE
148	IANS	I2	ERROR SELECTION VARIABLE OF OPERATIONS 1TO 15
149	AAR	A16	CUSTOMER CODE (TEMPORARY)
150	CLA	A80	ADDRESS OF A CUSTOMER

No	Variable	Type	Description
151	COM	A105	COMMENTS ABOUT A CUSTOMER
152	OPER(LM)	A20	COMMENTS FOR LMth OPERATION
153	RDAT(LM)	A9	DATE OF LMth OPERATION
154	ACH(LM)	A1	MACHINE NUMBER OF LMth OPERATION
155	YMAT	A19	A PART OF THE MATERIAL DESCRIPTION
156	TY	A4	MATERIAL TYPE
157	SEAC(LM)	F4.1	ACTUAL SETTING TIME FOR LMth ROLLING
158	ROAC(LM)	F6.1	ACTUAL ROLL SPEED FOR LMth ROLLING
159	STAC(LM)	F4.1	ACTUAL STRIP TIME FOR LMth ROLLING
160	DE	A10	TO PRINT OUT THE SELECTED OPTION
161	DMEAN	F6.3	MEAN GAUGE (TEMPORARY)
162	DMIN	F6.3	MINIMUM GAUGE (TEMPORARY)
163	DMAX	F6.3	MAXIMUM GAUGE (TEMPORARY)
164	DTY	A4	MATERIAL TYPE (TEMPORARY)
165	DAT	A9	LAST UPDATED ON
166	DD	A9	TODAY'S DATE
167	SETC	F6.2	SET UP COST FOR 100m
168	ROAC	F6.2	ROLLING COST FOR 100m
169	STAC	F6.2	STRIP COST FOR 100m
170	ENS	A1	PROFIT TYPE
171	OP	A20	COMMENTS (TEMPORARY)
172	RD	A9	DATE OF OPERATION (TEMPORARY)
173	REQ	A12	RAW MATERIAL REQUIREMENT FOR A JOB
174	TAGE	F4.1	PERCENTAGE INCREASE IN ANALYSING
175	STAT(16)	-	STATUS
176	TC	-	TYPE OF TRANSPORT COST
177	TRAJ	F6.2	TRANSPORT COST FOR JOB
178	PRR	F6.2	PROFIT PER MACHINE HOUR
179	PAR	F5.1	PERCENTAGE INCREASE OF SETUP TARGET IN JOBCARD
180	SDAT(16)	A9	TRANSFER VARIABLE OF ROLL DATE
181	MDA(KK)	I2	DAY OF OPERATION
182	MMO(KK)	I2	MONTH OF OPERATION
183	MYE(KK)	I2	YEAR OF OPERATION
184	ST	-	STATUS (TEMPORARY)
185	ACTM(5)	F6.2	ACTUAL SELLING PRICE FOR 100m
186	ACTF(5)	F6.2	ACTUAL SELLING PRICE FOR 100ft
187	PRVAR	F6.2	DETERMINES TYPE OF PROFIT
188	PRDET1	A3	OUTPUT FORMAT VARIABLE FOR PERCENTAGE PROFIT
189	PRDET2	A14	OUTPUT FORMAT VARIABLE FOR PERCENTAGE PROFIT
190	III	I2	COUNTING VARIABLE
191	SE	F4.1	SETUP TARGET (TEMPORARY)
192	RO	F6.1	ROLL TARGET (TEMPORARY)
193	ST	F4.1	STRIP TARGET (TEMPORARY)
194	IACTL(5)	I5	LENGTH BASED ON TO GIVE ACTUAL SELLING PRICES
195	ADINF	A40	ADDITIONAL INFORMATION
196	ADIOP(5)	A46	ADDITIONAL OPERATION DESCRIPTION
197	ADC(5)	F6.2	COST FOR A PARTICULAR ADDITIONAL OPERATION
198	CPER	F4.1	PERCENTAGE COMMISSION
199	CDET	A1	OUTPUT FORMAT FOR COMMISSION
200	BAY	A1	COMPANY NUMBER
201	KEY1	A20	PASSWORD OF THE TOP LEVEL
202	KEY2	A20	PASSWORD OF THE MIDDLE LEVEL
203	KEY3	A20	PASSWORD OF THE BOTTOM LEVEL

No	Variable	Type	Description
204	IBRIEF	A1	BRIEF DETAILS OR NOT IN A LISTING
205	IORDA(M)	I2	DAY OF THE CUSTOMER M'S ORDER
206	IORMO(M)	I2	MONTH OF THE CUSTOMER M'S ORDER
207	IORYE(M)	I2	YEAR OF THE CUSTOMER M'S ORDER
208	IDEDA(M)	I2	DAY OF CUSTOMER M'S DELIVERY DATE
209	IDEMO(M)	I2	MONTH OF CUSTOMER M'S DELIVERY DATE
210	IDEYE(M)	I2	YEAR OF CUSTOMER M'S DELIVERY DATE
211	MDA	I2	DAY OF JOBCARD CREATION
212	MMO	I2	MONTH OF THE JOBCARD CREATION
213	MYE	I2	YEAR OF THE JOBCARD CREATION
214	INO	I2	NUMBER OF CARDS TO BE DUPLICATED
215	INUM(25)	I2	CARD NUMBER OF EACH CARD TO BE DUPLICATED
216	IUPDUP	-	DUPLICATION IN UPDATING
217	ICRDUP	-	DUPLICATION IN NEW CARD CREATION
218	IST	-	SELECTED CARD NUMBER IN STOCK
219	ISCHE	-	SCHEDULING 0,1,2
220	UNAM	A14	USER NAME
221	UIC	A12	USER IDENTIFICATION CODE
222	USER	A12	USER IDENTIFICATION CODE (TEMPORARY)
223	TOTTIM	F4.1	TOTAL TIME FOR THE JOB
224	SM(4)	A1	LINE PREFERENCE
225	FINDAY	A12	FINISHING DAY OF A JOB
226	IFNDAT	I2	FINISHING DATE OF A JOB
227	IFNMON	I2	FINISHING MONTH OF A JOB
228	IFNYER	I4	FINISHING YEAR OF A JOB
229	FINTIM	A9	FINISHING TIME OF A JOB
230	STDAY	A12	STARTING DAY OF A JOB
231	ISTDAT	I2	STARTING DATE OF A JOB
232	ISTMON	I2	STARTING MONTH OF A JOB
233	ISTYER	I4	STARTING YEAR OF A JOB
234	STTIM	A9	STARTING TIME OF A JOB
235	JDEDA	I2	REQUIRED DATE TO FINISH A JOB
236	JDEMO	I2	REQUIRED MONTH TO FINISH A JOB
237	JDEYE	I4	REQUIRED YEAR TO FINISH A JOB
238	RQDAY	A12	REQUIRED WEEKDAY
239	DA	A11	CHARACTER REPRESENTATION OF A DATE
240	REA	A45	DOWNTIME REASON

Appendix 28

System common blocks

COMMON BLOCKS IN THE SYSTEM - as at 30-APR-87 at 16:18 hrs
(Last updated on - 22-SEP-86 at 10:37 hrs)

Blk	Variables	Comments
1	SET,SETG	ESTIMATED SETUP TIME AND TARGET
2	FT(6,15),IFT(6,15), IFS(6,15)	LENGTH, QUANTITY AND SORTING DETAILS
3	BAR	JOB NUMBER (TEMPORARY)
4	ANS	ANSWER TO A QUESTION Y/N
5	COR	ERROR SELECTION VARIABLE
6	I	DUMMY VARIABLE FOR COUNTING
7	PC/PRD	PRODUCT CODE
8	WAR/CUS	CUSTOMER NAME EXTRACTION
9	XMAT	MATERIAL DESCRIPTION
10	NPS	NUMBER OF PASSES
11	DES	DESCRIPTION OF THE FINAL PRODUCT
12	SC	SCALE FACTOR
13	XAR	ORDER NUMBER (TEMPORARY)
14	THICK	MEAN THICKNESS
15	WIDTH	MEAN WIDTH
16	DAR	SPECIAL INSTRUCTIONS (TEMPORARY)
17	IACTL(5),ACTM(5),ACTF(5)	ACTUAL SELLING PRICES
18	ESWT	ESTIMATED WEIGHT OF RAW MATERIAL
19	BAY	COMPANY NUMBER
20	AVA,REQ	MATERIAL AVAILABILITY AND AMOUNT REQD.
21	CAR	SELLING PRICES (TEMPORARY)
22	YAR	QUAN/LEN (TEMPORARY)
23	ZAR	PACKING INSTRUCTIONS (TEMPORARY)
24	BOK	SELECTION VARIABLE IN MAIN MENU
25	EAR	EXCESS MATERIAL INSTRUCTIONS (TEMPORARY)
26	AAR	CUSTOMER CODE (TEMPORARY)
27	I,MAX	COUNTER FOR RECORDS
28	YMAT	PART OF THE MATERIAL DESCRIPTION
29	IBRIEF	BRIEF DETAILS OR NOT IN A LISTING
30	PT	PRICE PER TON
31	MET	THE LENGTH IN METRES TO BASE THE PRICES
32	UWT	WT OF MATERIAL FOR 100m WITHOUT SCRAP
33	NSCR	PERCENTAGE OF SCRAP
34	GWT,PRIMAT	GROSS WT WITH SCRAP AND MATERIAL PRICE
35	ISCHE	FOR SCHEDULING
36	ROT,ROTC	ESTIMATED ROLL SPEED AND COST PER 100m
37	STT,STTC	ESTIMATED STRIP TIME AND COST PER 100m
38	(MANY)	PERCENTAGE PROFIT
39	COM	COMMISSION PER 100m
40	TOOM,ITOOB	TOOL AMORTIZATION
41	TRAT,TRAL,TRAJ,TC	TRANSPORT COSTS
42	SELM,SELF	CALCULATED SELLING PRICE
43	RD	DATE OF OPERATION
44	ACH(16)	MACHINE NUMBER
45	SEAC(16)	ACTUAL SET UP TIME

System common blocks continued ...

Blk	Variables	Comments
46	ROAC(16)	ACTUAL ROLL SPEED
47	STAC(16)	ACTUAL STRIP TIME
48	OP	COMMENTS FOR THE OPERATION
50	UC	UNIT OVERHEAD COST PER HOUR
51	(MANY)	DATE OF ORDER
52	(MANY)	DELIVERY DATE
53	SM(4)	MACHINE PREFERENCE
54	(MANY)	FINISHING TIMES OF THE JOB
55	(MANY)	STARTING TIMES OF JOB
56	UIC	USER IDENTIFICATION CODE
57	PER	PERSON'S NAME
58	PASS	PASSWORD FOR PRIVILEDGE LEVEL
59	UNAM	USER NAME
60	DIR	DEFAULT DIRECTORY
61	TOTTIM	TOTAL TIME FOR THE JOB
62	JDEDA,JDEMO,JDEYE	COMPLETION DATE AND DEADLINE
63	MDA(16),MMO(16),MYE(16)	DATE OF OPERATION
64	ORD	ORDER NUMBER
65	QUAN(16) OR IQTY(16)	QUANTITY FOR THE OPERATION
66	PRI(16)	PRICE PER TON OF THE OPERATION
67	BAL(16) OR IBAL(16)	BALANCE AFTER THE OPERATION
68	SU	SUPPLIER FOR THE OPERATION
69	STAT(16)	STATUS
70	XLEN	LENGTH IN FINPROD
71	ENS	PROFIT TYPE
72	SETG,ROTG,STTG	TARGETS
73	XLMAX,XLMIN	LENGTH TOLERANCES
74	ADI	ADDITIONAL OPERATIONS COST PER 100m
75	TY	TYPE OF MATERIAL
76	THIMIN	MINIMUM THICKNESS
77	THIMAX	MAXIMUM THICKNESS
78	WIDMIN	MINIMUM WIDTH
79	WIDMAX	MAXIMUM WIDTH
80	CPER,CDET	PERCENTAGE COMMISSION
81	ADIOP(5),ADC(5)	ADDITIONAL OPERATIONS AND COSTS
82	ADINF	ADDITIONAL INFORMATION
84	NCUS	NUMBER OF CUSTOMERS
85	JBN	JOB NUMBER
86	IBNS	VARIABLE TO ABORT OPERATION IN SCHEDULING
87	IPUSH	TO SEE WHETHER JOBS HAVE TO BE PUSHED
88	MAC	LINE NUMBER
89	XLEN(90)	LENGTH FOR THE SHOP FLOOR
90	IPR(90)	NO OF PRESS OPERATIONS FOR THE LENGTH
91	ITIED	JOB NUMBER OF A TIED JOB TO A JOB
93	SHSETG,SHROTG	TARGETS FOR THE SHOP FLOOR
94	REA	DOWNTIME REASON

Appendix 29

Listings of all programs

The microfiche which contains the listings of all programs is attached to the back cover.

References

1 Meegoda, R — "Why you should use computers", pp 29 to 31, Computerised Manufacturing, June 1986.

2 Crowder, R — "The MAP specification", pp 22, Control Engineering, October 1985.

3 Laduzinsky, A.J — "The MAP to automation", pp 19, Control Engineering, October 1985.

4 Meegoda, R and Milner, D.A — "Communication, Where does MAP fit in ?", to be published.

5 Vasiliou, V.C — "Computer integrated system for form-roll manufacture", Ph D thesis, University of Aston in Birmingham, 1985.

6 Wong, T.N — "Computer Aided Design and Manufacture of form rolls", Ph D thesis, University of Aston in Birmingham, 1983.

7 Ng, K.H — "Computer aided geometric design of form rolls for NC manufacture", University of Aston in Birmingham, 1981.

8 Kirkland, W.G — "Cold roll forming practice in the United States", American Iron and Steel Institute, 1966.

9 Roberts, W.L — "Cold rolling of Steel", Marcel Dickins Inc., 1978.

10 Walker, A.C — "Design and Analysis of Cold Formed Sections", Intertext Books, 1975.

11 — "Drawer runner rail", Product code 1761, Sections and Profiles Ltd.

12 "Stock 420", Product code 1380, Sectons and Profiles Ltd.

13 Cotterell, A "An introduction to metallurgy", 2nd edition, Edward Arnold Publishers Ltd, 1975.

14 Tweedale, J.G "Metallurgical Principles for Engineers", Iliffe books Ltd, 1962.

15 Nutt, M.C "Principles of Modern Metallurgy", Charles E Merril Publishing Company, 1968.

16 Pacoe, K.J "An introduction to the properties of engineering materials", 3rd edition, Van Nostrand Reinhold Company, 1978.

17 "B.S. 1400 - Copper and copper alloy castings", The Association of Bronze and Brass Founders, 1961.

18 Higgins, R.A "Engineering Metallurgy Part2", English Universities Press, 1970.

19 Johnson, C.G and Weeks, W..R "Metallurgy", 5th edition, American Technical Society, 1977.

20 Burton, M.S "Applied Metallurgy for Engineers", Mcgraw Hill, 1956.

21 Morrison, D "Engineering Design", Mcgraw Hill, 1968.

22 Archer, L.B "Systematic methods for designers", Council of Industrial Design, 1965.

23 Farr, m "Design Management", Huchingson, 1961.

24 Reswick, J.B "Prospectus for Engineering Design Centre", Case Instute of Technology, 1966.

25 Dixon, R "Design Engineering", Mcgraw Hill, 1966.

26 Groover, M.P — "Automation Production Systems and Computer Aided Manufacturing", Prentice Hall Inc., 1980.

27 Webster, N — "Webster's 3rd new dictionary of English language", pp 266, Vol 11, A to F, London Bell Sprinfield Mas Merican, 1961.

28 Osborn, A.F — "Applied Imagination Principles and Procedures of Creative Problem Solving", New York Scribners, 1953.

29 Von Fange, E.K — "Professional Creativity", Prentice Hall, 1959.

30 Askimow, M — "Introduction to Design", Prentice Hall, 1962.

31 Gordon, W.J.J — "Synectics - The development of creative capacity", Harper & Row Publishers, 1961.

32 Jones, J.C — "Design methods", John Willey & sons, 1980.

33 Prince, G.M — "Synectics - A method of creative thought", pp 805 to 806, Engineering Education, March 1968.

34 Yokel, F.Y — "Synectics - A new experience in creative thinking", pp 810 to 811, Engineering Education, March 1968.

35 — "Synectics Inc., 28, Church Street, Cambridge, Messachusetts, U.S.A.

36 French, M.J — Engineering design: The conceptual stage", Pitman press, 1971.

37 Norris, K.W — "The Morphological Approach to Design Engineering", pp115 to 140, Conference on design methods, Sept 1962, Pergamon press 1963.

38 Gregory, S.A — "The Design Method", London Butterworths, 1966.

39 Gosling, W — "The Design Engineering Systems", Ballantyne and Company Ltd, 1962.

40 Meegoda, R — "Computer Integrated Monitoring", First year research report, January1986.

41 Smith, W.A — "A guide to CAD/CAM", Institution of Production Engineers/Numerical Engineering Society, 1985.

42 Barr, P.C, Krimper, R.L, Luzear, M.R and Stammen, C — "CAD Principles and Applications", Prentice Hall Inc., 1985.

43 Besent, C.B, Lui, G.W.K — "Computer Aided Design and Manufacture", 3rd edition, Ellis Horwood Ltd, 1986.

44 Leventhal, L.A — "Introduction to mocroprocessors: software and hardware programming", pp 543 to 548, Prentice Hall Inc., 1978.

45 Hebditch, D.L — "Data Communications, an introductory guide", pp 49 to 60, Pliener Associates Ltd, 1975.

46 Lipschutz, M.D, Lipschutz, S — "Data Processing", pp 27 to 36 and 130 to 132, Schaum's outline series in computers, Mcgraw Hill, 1981.

47 Flores, I and Terry, C — "Microprocessor Systems", pp 82 to 100 and 173 to 193, Van Nostrand Reinholt Company, 1982.

48 Colley, P — "Hardware for interactive computer graphics", M.Sc in CAD/CAM notes, University of Aston in Birmingham.

49 Coll, J — "The BBC Microcomputer User Guide",British Broadcasting Corporation, 1982.

50 — "Acorn Electron user guide", Acorn Computers Ltd, 1983.

51 Favert, A.G "Digital Computer Principles and Applications", Van Nostrand Reinhold Company, 1972.

52 Butler, K.J "Introduction to data handling and computer selection", University of Aston in Birmingham, 1981.

53 "Cumana Disk Drive Guide", Cumana Ltd, 1984.

54 Ward, B.J "Walters Computer Systems", Acorn Computers Ltd.

55 Acorn computers ltd "Electron plus3 - new dimensions in capacity", APM 56, Acorn computers.

56 Pharo, C "The Advanced Disc User Guide for the BBC Micro", Cambridge Microcomputer Centre, 1985.

57 Dutt, C.B "Introducing the Macintosh", pp 5 to 7, Mcgraw Hill, 1984

58 Morris, S "Getting to know your Apricot", pp 14 to 17, Newtech Publishing Ltd, 1984.

59 "Trackerball", Wigmore House Ltd, 32, Saville Row, London W1X 1AG, Tel 01- 734 0173.

60 "Tektronix 4107/4109 Computer Display Terminal", Tektronix Inc., 1983.

61 Milner, D.A "Some Aspects of Numerical Control and Adaptive Control", University of Aston.

62 Billingsley, J "Robots and Automated Manufacture", Peter Peregrinus Ltd, 1985.

63 IIT Rresearch Institute "APT part programming", Mcgraw Hill, 1967.

64 Raouf, A and Ahmed, S.I — "Flexible Manufacturing - Recent developments in FMS, Robotics, CAD/CAM, CIM", Elsevier Science Publishers B.V, 1985.

65 Orlicky, J — "Material Requirements Planning", Mcgraw Hill, 1975.

66 Anderson, J.C — "Material Requirements Planning", American Production and Inventory Control Society, 1981.

67 Wallace, T.T — "MRP 2", Oliver Wight publications Ltd, 1985.

68 Martin, A.J — "DRP - Distribution Resource Planning", Oliver Wight Ltd Publications Inc., 1983.

69 Groover, M.P and Zimmers jnr, E.W — "CAD/CAM - Computer Aided Design and Manufacturing", Prentice Hall Inc., 1984.

70 Thomas, A.B — "Stock Control in Manufacturing Industries", Gower press, 1980.

71 Morrison, A — "Storage and Stock Control for Industry and Public Undertakings", Camelot Press Ltd, 1970.

72 Tudor, P — "CAPP - the Engineering Database", Effective CADCAM 1985, pp 44 to 51, I Mech E Conference publications 1985-87.

73 Mayer, R.R — "Production Management", Mcgraw Hill, 1968.

74 Shaw, R — "Process Control Systems - A case study of continuous creation", pp 5 to 13, Process Measurement and Control Applications, Institute of Measurement,1984

75 Brown, J — "Choosing between a Data Logger (DL) and a Data Acquition System (DAS)", pp 33 to 38, Instuments and Control Systems, Sept 1976.

76 Jovic`, F — "Process Control Systems, Principles of design and operation", Kogan Press Ltd, 1986.

77 Hunter, R.P — "Automated Process Control Systems Concepts and Hardware", Prentice Hall Inc., 1978.

78 Young, R.E — "Supervisory Remote Control Systems", pp 6 and 136, Peter Peregrinus Ltd, 1977

79 Tanenbaum, A — "Structured Computer Organization", 2nd edition, Prentice Hall Inc., 1984.

80 Lesea, A
Zacks, R — "Microprocessor Interfacing Techniques", 2nd edition, Sybex Inc., 1978.

81 Betts, B — "VAX 8800 series", pp 52 to 56, DEC user, October 1986.

82 — "The VAX 8650: last of the unibus line", pp 55, DEC user, October 1986.

83 — "Poly puts big VAX to work", pp 52, DEC user, October 1986.

84 Kitt, N — "Pick your system", Sigma Press Publications, 1986.

85 — "Introduction to system 4 operating Systems", International Computers Ltd, 1972.

86 Comer, D — "The Operating System Design - the Xinu approach", Prentice Hall Inc., 1984.

87 Katzan Jr, H — "Operating Systems, a pragmatic approach", 2nd edition, Van Nostrand Reinhold Company, 1980.

88 Lister, A.M — "Fundamentals of Operating Systems", 3rd edition, Camelot Press Ltd, 1984.

89 Barron, D.W "Computer Operating Systems", Cox and Wyman Ltd, 1971.

90 Lorin, H
Deitel, H.M "Operating Systems", Addison Wesley Publishing Company, 1981.

91 Bourne, S.R "The Unix System", Addison Wesley Publishing Company, 1982.

92 Banham, M and
Rutter, A "Unix the book", Sigma Technical Press, 1982.

93 Lush, A
Kenny, M "An introduction to VMS", DEC user conference, 1986.

94 "MicroVMS user's manual", AA-Z209B-TE, Digital Equipment Corporation, September 1984.

95 "Vulcan concepts and features", Harris Corporation, 1978.

96 "VAX Ada and VAXELN Ada technical summary", pp 2-3, Digital Equipment Corporatin, 1985.

97 "The BBC microcomputer system MASTER series reference manual Parts 1 and 2, Acorn Computers Ltd, 1986.

98 Cortesi, D.E "Inside concurrent CP/M - A guide for users", CBS college publishing, 1984.

99 "Professional 300 series user's guide to diskette system", Digital Equipment Corporation, 1983.

100 Cavill, J "Putting intelligence in to I/O boards", pp 91 to 97, Control and Instrumentation, November 1985.

101 McGettrick, A.D — "The Definition of Programming Languages", Cambridge University Press, 1984.

102 Peterson, W.W — "Introduction to Programming Languages", Prentice Hall Inc., 1974.

103 Malone, J.R — "Comparative Languages", Chartewell-Bratt (Publishing and Training) Ltd, 1984.

104 Pratt, T.W — "Programming Languages, design and implimentation", 2nd edition, Prentice Hall Inc., 1984.

105 Marshal, G — "Computer Languages and their uses", Granada Publishing Ltd, 1983.

106 Barmes, J.P — "An overview of ADA", pp 322 to 354, Programming Languages - a grand tour, Computer Science Press Inc., 1983.

107 Laurie, P — "Databases, how to manage information on your micro", Chapman and Hall Ltd, 1985.

108 Date, C.J — "Databases, A primer", Addison Wesley Publishing Company, 1984.

109 Mayne, A — "Database management systems: A technical review", National Computing Centre, 1981.

110 Remenyi, D — "dBase 11 and dBASE 111", Harlow Longman, July 1985.

111 Brooks, F.D — "INFO Primer", Waltham (mass) Henco Software, 1980.

112 Howe, R — "Data Analysis for Data Base Design", Edward Arnold (Publishers) Ltd, 1983.

113 — "VAX Fortran user's guide", pp 3-9 to 3-20, Digital Equipment Corporation, April 1982.

114 — "VAX/VMS Primer", Digital Equipment Corporation, May 1982.

115 Liossis, G — "Computer Aided Stock Control System for Factory Automation", M.Sc thesis, University of Aston, 1983.

116 — "VAX Fortran user's guide", pp 5-8 and 7-13, Digital Equipment Corporation, April 1982.

117 — "VAX Basic user's guide", pp 4-7, Digital Equipment Corporation, November 1982.

118 Burch, M — "Changing the face of Data Acquisition", pp 71 to 74, Control and Instrumentation, November1985.

119 Wallace, B — "Using 32 Bit Minis for High Speed Acquisition", pp 79 to 81, Control and Instrumentation, November 1985.

120 Macdonald, N — "Low Cost Networks and Data Acquisition", pp 87 to 89, Control and Instrumentation, November 1985.

121 Brighton, M — "Putting personality into modular loggers", pp 99, Control and Instrumentation, November 1985.

122 Lee, G — "Acquisition and analysis gives greater power", pp 107 to 110, Control and Instrumentation, November 1985.

123 Usher, M.J — "Sensors and Transducers", Macmillian Press, 1985.

124 Seippel, R.G — "Transducers, Sensors and Detectors", Reston Publishing Co., 1983.

125 Brown, K — "More intelligence for I/O sub-systems", pp 101 to 105, Control and Instrumentation, November 1985.

126 "Micross shop floor data collection", (data sheet), Kewill Systems, 6th floor, Epic House, Charles Street, Leicester LE1 3SH, Tel (0533) 538915.

127 "TM 900 factory data collection systems", (data sheet), Data acquisition and control division, Burr Brown, 3631E, 44th Street, Tuscon, Arizona 85713, Tel (602) 747-07111.

128 "An introduction to PMS production monitoring systems", (data sheet), Retriever Technology Mill Lane, Dunsley, Kinver, Tel (0384) 873842.

129 Zaks, R "Programming the 6502", 4th edition, Sybex Inc., 1983.

130 Mac Gregor, M "The evolution of the controller", pp 21 to 26, Industrial Computing, February 1987.

131 Jones, N "Programming for all deciplines", pp25, Automation, January 1986.

132 Wain. L "Progress in process", pp 14 to 18, Industrial Computing, October 1986.

133 "programmable controllers", Catalogue S-DS 0048 8182, IMO Precision Controls Ltd, North Circular road, Staples Corner, London NW2 7JP, Tel. (01) 452-6444.

134 "Hitachi programmable control system", Catalogue SI_EO74, Hitachi Ltd, Old Bridge Road, Radlett House, West Hill, Aspley Guise, Milton Keynes MK17 8DT, Tel (0908) 584868.

135 "Mechman IPC 500 Industrial Programmable Controller", (data sheet), Mechman Ltd Mechman House, Suttens Park Avenue, Readding RG6 1AZ, Tel (0734) 664456.

136 "Omron System C services programmable controller", Catalogue 455-3M, Omron Tateisi Electronics 9th floor, Osaka Centre Building, 4-68, Kitakyutaro, Higashiku, Osaka 541, Japan, Tel (06) 282-2706.

137 Ward, N "Monitoring with the programmable controller", pp17, Automation, January 1985.

138 "Terminals give APT 'foolproof' data collection", pp 33 to 34, Production Engineer, February 1985.

137 Rao, G.V "Microprocessors and microcomputer systems", Van Nostrand Reinhold Company, 1982.

138 Haines, G "BBC micro and Electron book", Mcgraw Hill, 1985.

139 Bray, A.C
Dickens A.C
Holmes, M.A "Advance user guide for the BBC micro", Cambridge Microcomputer Centre, 1983.

140 "Stock 411, 2 inch inner channel", Product code 1430, Sections and Profiles Ltd.

141 "Jobber interface keyboard/ LCD/ Printer driver", Issue ID CB490AA/5D1, Control Universal Ltd, 137, Ditton Walk, Cambridge CB5 8QF, Tel (0223) 244447.

142 "The Euro Beeb/ EuroCUBE-65 operating system", Issue ID CB664AB-1/12086, Control Universal Ltd, 137, Ditton Walk, Cambridge CB5 8QF, Tel (0223) 244447.

143 "Viewline Technical manual", CB480FP-5G/1, Control Universal Ltd, 137, Ditton Walk, Cambridge CB5 8QF, Tel (0223) 244447.

144 "CU-KEY-25 Non staggerd keyboard", Control Universal Ltd, 137, Ditton Walk, Cambridge CB5 8QF, Tel (0223) 244447.

145 "6502 2nd processor for the BBC micro", Cambridge Microsystems Ltd, Brookfield Business Centre, Cottenham, Cambridge, Tel (0954) 51122.

146 "CMS 6502 Operating system Advanced user guide", Cambridge Microsystems Ltd, Brookfield Business Centre, Cottenham, Cambridge, Tel (0954) 51122.

147 "CMS 6502 Network managers guide", Cambridge Microsystems Ltd, Brookfield Business Centre, Cottenham, Cambridge, Tel (0954) 51122.

148 "CMS hrdware/software 1985", Cambridge Microsystems Ltd, Brookfield Business Centre, Cottenham, Cambridge, Tel (0954) 51122.

149 "The PM 10 Panel Mounted Terminal", Catalogue SM/586, Star Microterminals Ltd, Harman House, Andover Rd, Winchester, Hampshire SU23 7BS, Tel (0962) 51422.

150 "Panel mounted terminal (PM10)", (data sheet), Star Microterminals Ltd, Harman House, Andover Rd, Winchester, Hampshire SU23 7BS, Tel (0962) 51422.

151 "4 Channel Serial Card", Cambridge Microsystems Ltd, Brookfield Business Centre, Cottenham, Cambridge, Tel (0954) 51122.

152 "Dual Asynchronous Receiver/Transmitter (DUART) SCN 2681 series", (data sheet) Cambridge Microsystems Ltd, Brookfield Business Centre, Cottenham, Cambridge, Tel (0954) 51122.

153 Hartland-Swann, J "How Much Integration", pp 12 to 14, Industrial Computing, February 1987.

154 Hartland-Swann, J "CIM: Hit or myth ?", pp 22 to 23, Industrial Computing, February 1986.

155 Brooks, T "Micros and Local Networks: Making the marriage succeed", pp 81 to 90, Networks 85, British Library Cataloguing in Publication, June 1985.

156 Looney, M "Integration is the key", pp 31 to 34, Computerised Manufacturing, June 1986.

157 "W.T.Avery Ltd", Smethwick, Warley, West Midlands, Tel (021) 558-1112.

158 "Teletext Adaptor", Morley Electronics, 1, Morley Place, Shiremoor, Tyneside NE27 0QS, Tel (091) 251-3883.

159 Peters, N "Gear up your Electron and communicate", pp 14 to 15, Electron User, February 1986.

160 Campt, D.W "Missing: the universal micro-to-mainframe link", pp 125 to 130, Data Communications, March 1984.

161 Bryant, S.F "Micro-to-Mainframe Links", pp 162 to 183, Computer Decisions, July 1984.

162 Benoit, W "Micro-to-Mainframe Harmony", pp 17 to 19, Computerworld on Communications, August 1984.

163 Feltman, C — "Local Networks and Micro Mainframe Integration", pp 73 to 79, Networks 85, British Library Cataloguing in Publication, June 1985.

164 Moon, D — "Developing standards smooth the intrgration of programmable factory floor devices", pp 49, Control Engineering, Vol 32, October 1985.

165 Dwyer, J — "MAP bandwagon poised to run over the unwary", pp 26 to 27, Automation, March 1986.

166 — "MAP bandwagon passes point of no return at US automation show", pp 3, Automation, November/December 1985.

167 — "DTI wants a MAP demo in October", pp 10 Automation, May 1986.

168 Charlish, G — "Major players line up behind automation standards initiative", pp 16, Financial times, 23rd November 1986.

169 Dwyer, J — "Motorola first in with multilevel MAP chips", pp 23, Automation, May 1986.

170 — "Coup imminent for Aldershot MAP implementers", pp 5, Automation, January 1986.

171 — "MAP testing is now the big obstacle to it's success", pp 19, Automation, May 1986.

172 Dwyer, J — "MAP message format hitch", pp 12, Automation, January 1987.

173 Dwyer, J — "CIMAP - the verdict", pp 16, Automation, January 1987.

174 Dominico, T — "The MAP to the TOP", pp 15, The Guardian, 6th November 1986.

175 Housley, N — "Merging Diverse Network Technologies", pp 605 to 625, Networks 84, British Library Cataloguing in Publication, July 1984.

176 Grant, K — "Successful networking of diverge personal computers", pp 285 to 292, Networks 84, British Library Cataloguing in Publication, July 1984.

177 Stanton, C — "Texas' direct attack on machine interfacing", pp 4, Engineering News, May 1986.

178 Leach, A — "PLC design evolves to meet networking needs", pp 27 to 28, Automation, January 1986.

179 Williams, P — "Programming the BBC micro", Butterworth and Co (publishers) Ltd, 1983.

180 — "MicroVMS release notes Version 4.1", pp 4-1, and 7-1 to 7-28, Digital Equipment Corporation, May 1984.

181 Meegoda, R
Milner, D.A — "Data transfer communication between a host and a micro", to be published.

182 — "VAX/VMS Command language user guide", PP 87, 88 and 331 to 333, Digital Equipment Corporation, May 1982.

183 — "VAX/VMS Guide to using command procedures, pp 8-1 to 8-8, Digital Equipment Corporation, May 1982.

184 — "DHV11 Technical Manual", EK-DHV11-TM -001, Digital Equipment Corporation, 1983.

185 — "32k Shadow RAM Card", Watford Electronics Ltd, Jessa House, Lower High Street, Watford WD1 2AN, Tel (01) 950-8989.

186 "Master RAM Board", Slogger Ltd, 107, Richmond Rd, Gillingham, Kent, Tel (0604) 52203.

187 "Sprinter", Mutek, Quarry Hill, Box, Wiltshire, Tel (025) 743289.

188 "VAX/VMS Run Time Library Routines Reference Manual", AA-Z502A-TE, Digital Equipment Corporation, September 1984.

189 Waddilove, R "Make a call to the lost and found department - Disc filing system-part V11", PP 20 to 21, Electron User, April 1986.

190 Fortier, P.J "Design and Analysis of Distributed Real-time Systems", Intertext Publications Inc., 1985.

191 Jesty, P.H "Networking with Microcomputers", Blakewell Scientific Publications, 1985.

192 Milner, D.A Vasiliou, V.C "Computer Aided Engineering for Manufacture", Kogan Press Ltd, 1986.

193 Bennett, S "Computer Control of Industrial Processes", Institute of Electrical Engineers, 1982.

194 "Networks and Communications buyers guide", Digital Equipment Corporation, April - June 1986.

195 "VAX Enterprise Custer introductory user guide", Aston University Computing Service, VCL86 001, October 1986.

196 "VAX Enterprise Cluster introductory user guide", Aston University Computing Service VCL86 014, November 1986.

197 "A proposal for a campuswide network", REG/86/333, Aston University Computing Service June 1986.

198 Carter, G "Local Area Networks - The principles of data communications", Hineman Computers in Education Ltd, 1984.

199 Sayer, M.D "RS-232 made easy", Prentice Hall Inc.,1984.

200 "Feasibility Study on a Computer Aided Production Monitoring System, Ref 6013, Factory Data Systems, July 1984.