ICL

Restricted publication

Introduction to 7904 and 7906

7900

ICL

Restricted publication

Introduction to 7904 and 7906

Restricted Publication 0365

First Edition July 1980

International Computers Limited
Head Office: ICL House, Putney, London SW15 1SW

This publication outlines the functions performed by 7904 and 7906 systems in a communications network. It introduces the software supplied by ICL for operating the various hardware configurations in the following ways:

1 As a front-end communications processor connected to a single parent mainframe and running under the Standard Operating System

2 As a front-end communications processor connected to one or more mainframes. This 7900 system runs under an Extended Operating System with the ICL supplied Non-dedicated Turnkey program for message switching and device control

3 As a local or remote communications processor connected to one or more mainframes. This 7900 system runs under an Extended Operating System with a user written program for message switching and device control

4 As a free standing computer system for user program development

As well as providing an introductory framework for the current suite of 7900 publications, detailed in Appendix 1, the text describes the different hardware and software products as an aid to planning a communications system. These descriptions assume that the reader has some knowledge of communications technology and of the concepts involved. A glossary of terms is, however, included in Appendix 2.

The publication as a whole is written for systems managers and network controllers, but Chapter 4 will be of interest to system programmers, and Chapter 6 to operators.

The text of this publication is divided into chapters in the normal way, and each chapter is subdivided into sections. A section's level in the hierarchy is indicated by its number. Therefore, within Chapter *n*, first level section headings are numbered *n*.1, *n*.2 and so on; second level headings are numbered *n*.1.1, *n*.1.2 ... *n*.2.1 and so on; third level headings are numbered *n*.1.1.1, *n*.1.1.2 ... *n*.1.2.1 and so on.

The contents list and index, and cross-references in the text, all refer to section numbers.

Pages are numbered within chapters, in the form *c-p*, where *c* is the chapter number and *p* the page number within that chapter. Figures and tables, where they appear, are also numbered within chapters, so that Figure *n*.2 is the second figure in Chapter *n*, and Table *n*.2 is the second table in that chapter.

Section numbers, page numbers and figure and table numbers in appendices are preceded by the letter A.

1.1 Communications processors

The technique of connecting a central computer to remote terminals and peripheral devices by telegraph , telephone lines or direct link is now well established. The equipment at the remote end of the communications link may be anything from a single teletypewriter to another computer with bulk peripherals of its own, for example, card reader, line printer and disc. The definition of the remote equipment arises out of the normal process of systems analysis and design.

There can be many such links between a central computer and its geographically dispersed terminals, with the result that an extra load is placed on the central processor in controlling the communications network and handling the messages (data) flowing between processor and terminals. It is therefore advantageous to insert a dedicated computer into the system, between the central processor and the communications links, to perform the tasks associated with the network. This frees the central computer for its proper application-oriented data processing function. In this case, the dedicated computer is known as a *front-end processor* (FEP), and the central computer is called the *mainframe*. The FEP is connected to the mainframe by a local cable. The functions performed by the FEP include:

1 Polling the remote devices to check whether any has a message waiting to be transmitted

2 Receiving, checking and buffering messages which are transmitted down the link in bit serial format

3 Converting messages from line format to internal machine format

4 Transmitting complete messages to the mainframe

5 Performing similar functions for out-going messages

6 Employing the correct line protocol, according to the type of terminal connected to the line

7 Handling the transfer of bulk input and output data

8 Handling correction of transient errors, and reporting error situations

The remote equipment could include one or more computers each with its own peripheral devices. Each may be acting as the controller of a cluster of video terminals or bulk input and output devices, transferring messages between them and the mainframe. The remote computer may be more sophisticated, for example acting as a *remote concentrator*, which allows a number of communications links to share a single higher speed link connected to a mainframe, or as a *message switch*, which routes each message received to one of a number of remote mainframes. A computer used in either or both of the above ways is called a *communications processor* or a *network node*. The communications processor is called *remote* when it is connected to at least one mainframe by a communications link rather than by a local cable.

This publication is concerned with the two types of processor referred to above as the FEP and remote network node. The hardware and software are described in general terms, with references in appropriate places to other sections or publications that contain more detailed information.

This publication is not concerned with the terminal devices at the end of the line; any messages to or from the allowed types of terminal can be handled, and the communications system is transparent to the user's applications.

1.2 7904 and 7906 communications processors

The ICL 7904 and 7906 computers offer a range of communications processors linked to ICL 1900 or 2900 Series mainframes. There are seven models which provide varying levels of communications capability. They also support a range of extra features. The models are pre-packaged to enable the user to select more easily the equipment most suited to his needs. The models are modular, to enable the user to upgrade easily from one to another as his requirements develop.

The 7904 and 7906 are powerful communication processors and can support a large number of terminals.

1.2.1 The 7904 communications processor

The 7904 is an FEP locally connected to a mainframe via a Local Processor Link and is not available as a remote processor. It offers:

1. A communications processor with a high rate of throughput
2. A physically compact configuration
3. A communications system which is compatible with ICL 1900 Series, DME/1900 on 2960, or VME/B on 2960, 2970, 2972, 2976 or 2980 and therefore remains compatible with upgrades in the user's mainframe

As an indication of the scale of these processors, the smallest model (7904/21) consists of:

1. 48 Kwords (2 bytes per word) of 800 nanosecond store
2. Executive console teletypewriter
3. Paper tape reader
4. Local Processor Link (LPL) to 1900 or 2900 mainframe
5. Communications Multiplexer (CMX) with 24 channels for connection of communications links

This may be enhanced to 56 Kwords of store and a 108 channel Communications Multiplexer or a Command Chain Unit can be used to replace the 24 channel CMX. In total extra communication link connections may be specified up to a maximum of 108 channels.

A diagram of a basic 7904 FEP is shown in Figure 1.1.
The 24 Communications Multiplexer (CMX) channels support either:

1. 24 asynchronous lines up to 300 bits per second
2. 12 synchronous lines up to 2400 bits per second
3. 6 synchronous lines up to 9600 bits per second
4. An appropriate combination of speeds

The maximum overall throughput rate is approximately 4.1 Kcharacters per second with a mill utilisation of 60%.

If the user's communications requirements outgrow a 7904, then the system can easily be enhanced to a 7906. Further details of upgrading 7904 and 7906 systems are given in Chapter 7.

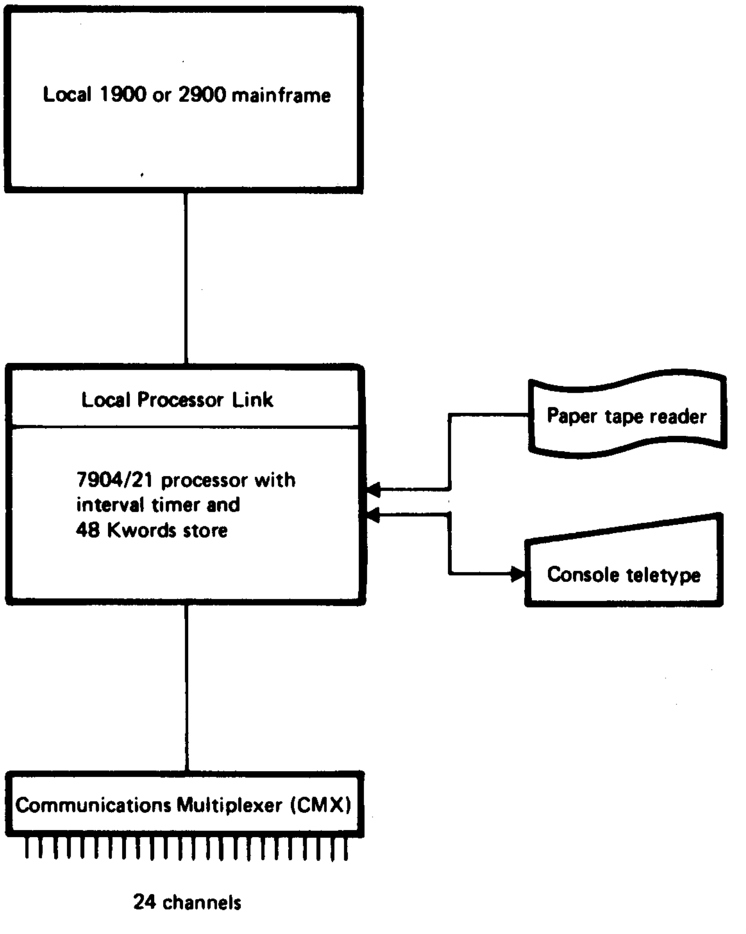

Figure 1.1 Basic 7904 system

1.2.2 The 7906 communications processor

The 7906 models are specifically designed for the user with advanced requirements such as very large communications networks, remote processing, user programming of the communications processor, spooling to disc, or support for local peripherals. The 7906 can be connected to ICL 1900 Series, DME/1900 on 2960, or VME/B on 2960, 2970, 2972, 2976 or 2980 as either a local or remote processor. The 7906 can be connected via Local Processor links to any of the mainframes listed above. A remote 7906 can be connected via communications links to several ICL mainframes, or the mainframes of other computer suppliers.

The 7906 offers:

1 An FEP for complex networks (the ICL supplied Non-dedicated Turnkey for message switching and device control)

2 User programming of the 7906 for advanced requirements such as:

(a) Spooling input or output to disc

(b) Remote network node

(c) Connection to more than one mainframe (local or remote)

(d) Precise tailoring of the communications software to the user's requirement

User programming is supported by a comprehensive Program Development Environment for NAL and CORAL programming languages.

The smallest 7906 system (7906/21) consists of:

1 48 Kwords of 800 nanosecond store

2 Executive console teletypewriter

3 Paper tape reader

4 Communications multiplexer with 24 channels

The above system may be enhanced to a much greater degree than a 7904. For instance:

1 Storage may be increased to 112 Kwords

2 Up to four Local Processor Links may be specified

3 The full range of 7906 peripheral devices is available, for example, line printer, fixed and exchangeable discs, and terminals for user program development

4 One or two Command Chain Units (CCU), each with one or more scanners, may be specified. The CCU can handle up to a maximum of 256 channels connected to many hundreds of terminals. The maximum overall throughput rate is approximately 25 Kcharacters per second with a mill utilisation of 60%

The modules comprising a 7906 system are called *sub-systems*, and are illustrated in Figure 1.2 and described in Section 6.2.

An installation can develop easily up to the maximum configuration, because of the modular architecture of the hardware and software.

Further details of the hardware are given in Chapter 6.

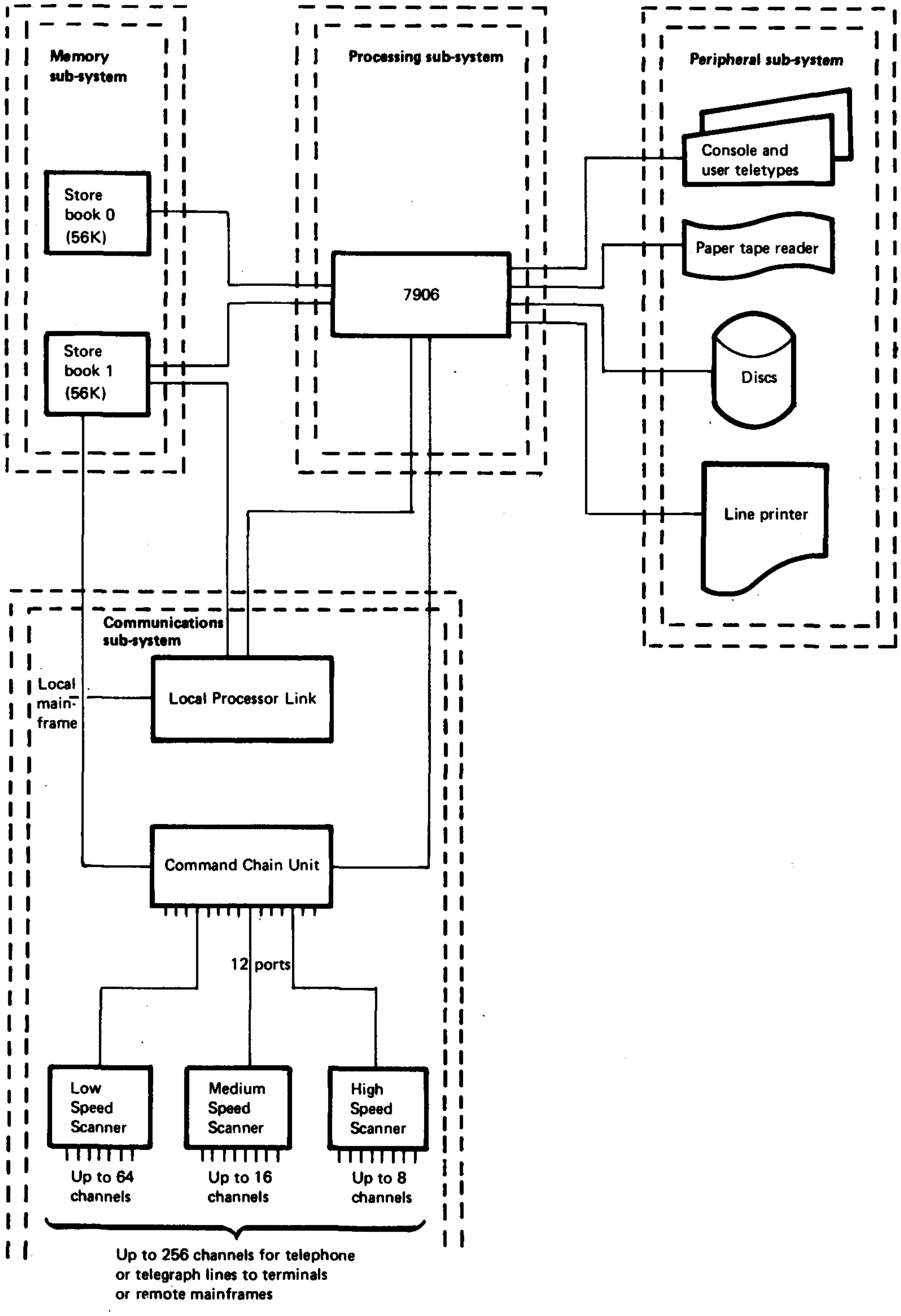

Figure 1.2 A large 7906 system

1.2.3 Models of 7904 and 7906

All models of 7904 and 7906 use the same processing unit. The models available are summarised below.

7904/21 Basic Turnkey System
Includes 7906/00 processor, 48 Kwords of store, 1900/2900 Local Processor Link and a 24 channel (12 port) Communications Multiplexer

7904/22 High Connectivity Turnkey System
Includes 7906/00 processor, 48 Kwords of store, 1900/2900 Local Processor Link and a 108 channel (54 port) Communications Multiplexer

7904/23 High Throughput Turnkey System
Includes 7906/00 processor, 48 Kwords of store, 1900/2900 Local Processor Link and a Command Chain Unit with one medium speed scanner

7906/21 Basic Programmable System
Included 7906/00 processor, 48 Kwords of store and a 24 channel (12 port) Communications Multiplexer

7906/22 High Connectivity Programmable System
Includes 7906/00 processor, 56 Kwords of store and a 108 channel (54 port) Communications Multiplexer

7906/23 High Throughput Programmable System
Includes 7906/00 processor, 56 Kwords of store and a Command Chain Unit with one medium speed scanner.

7906/24 Program Development System
Includes 7906/00 processor, 64 Kwords of store, Command Chain Unit with one medium speed scanner, and a fixed/exchangeable disc

Note that the above 7906 models are intended for both local or remote connection to a mainframe. For local connection a Local Processor Link must be added to the configuration.

1.3 Software

1.3.1 Types of 7900 operating system

The principle behind the design of the operating system for the 7904 and 7906 is that ICL provides the software as a set of modules. This technique enables the user to configure the operating system to his own requirements, and to change the system according to changing patterns of network traffic. The building of the operating system is done by using a Mainframe Support Environment (MSE) supplied by ICL, or by using the Program Development Environment (PDE) on a 7906.

There are two broad types of operating system for a 7900 working as a communications processor:

1. Standard Operating System (SOS) for 7904 or 7906 acting as a front end processor

2. Extended Operating System (XOS) for 7906 acting as an FEP or as a remote network node

These two types of operating system are supported by the same Mainframe Support Environment (MSE), which is available in three versions for 1900 Series George 3 and 4 users, for 1900 Series Executive users, and for 2900 Series VME/B users.

A third type of 7900 operating system is available called the Program Development Operating System (PDOS). It is used to control a 7906 when used as a stand-alone computer system for the development of communications programs written by the user. The range of software provided by ICL for use with a 7906 in this mode is called the Program Development Environment (PDE).

1.3.2 Standard Operating System (SOS)

SOS is designed for a 7904 or 7906 acting as a front end processor. It is the only operating system available to the 7904 user. SOS is dedicated, that is, the operating system cannot be modified at run time. This makes the SOS easy to load and run; the operating system is live and working as soon as the software has been succesfully loaded into the 7900. No user programming is required in order to use SOS, because SOS is a complete software package written by ICL to provide message handling facilities for most types of ICL terminal.

The SOS is configured, patched and assembled on the mainframe by using programs provided by the Mainframe Support Environment. The resultant SOS is loaded as required from the mainframe to the 7904 or 7906, using another program from the Mainframe Support Environment. MSE also provides dump and dump analysis facilities on the mainframe, see Figure 1.3. SOS is further described in Chapter 2.

1.3.3 Extended Operating System (XOS)

XOS is the alternative operating system to SOS for a 7906 communications processor. XOS is designed for a 7906 acting as a sophisticated FEP, or as a remote network node. XOS comprises an operating system base whose generation is specified by the user, plus at least one message buffering application program controlled by the operating system base.

The operating system base can be generated by the user on his 7906, and the message buffering application program can be developed by the user on his 7906. Both user system generation and application program development require the Program Development Environment described in the next section.

There is one use of XOS which does not require user programming. This is when XOS is used on a 7906 acting as a FEP, and employs a message buffering program provided by ICL. This optional message buffering program is called the *non-dedicated turnkey* (NDTKEY), and it provides all the facilities of SOS plus the facilities listed below.

1 Connection of more than one Local Processor Link

2 Connection to up to four mainframes

3 Selection of the devices required in the communications network from the full set of devices declared to the system generation process. The selection is performed by the system operator at the start of each days operation

Any use of XOS on a 7900 used as an FEP requires the Mainframe Support Environment (MSE). This environment provides facilities for patching, assembling, loading, dumping and dump analysis on a mainframe. ICL also provides an optional system generation service, which generates the XOS base to the user's specification, and the optional ND TKEY. These are not part of the MSE.

Figure 1.4 shows how the ICL software described above is used to build an Extended Operating System (XOS).

XOS is further described in Chapter 3.

1.3.4 Program Development Operating System (PDOS)

PDOS is designed for a free-standing model 7906/24, with discs and a line printer, dedicated to user system generation and program development. Communications facilities are not included in this operating system. It is intended for a 7906 used to develop communications programs written in CORAL 66 or 7900 Series Assembly Language (NAL).

PDOS incorporates software to handle the 7906 local peripherals and the operator interface. PDOS is used to run programs provided by the Program Development Environment (PDE). PDE provides the following facilities on a 7906:

1 CORAL 66 compiling system

2 NAL Assembler and link editor

3 Interactive Development Aid and editors

4 Advanced Operating Facilities (AOF)

5 Standard Disc Filing System (SDFS)

6 Utilities for loading, printing, dumping between the 7906 and its local peripherals

7 An operating system generation suite known as SYSGEN

PDE is shown in Figure 1.5. PDOS is further described in Chapter 4.

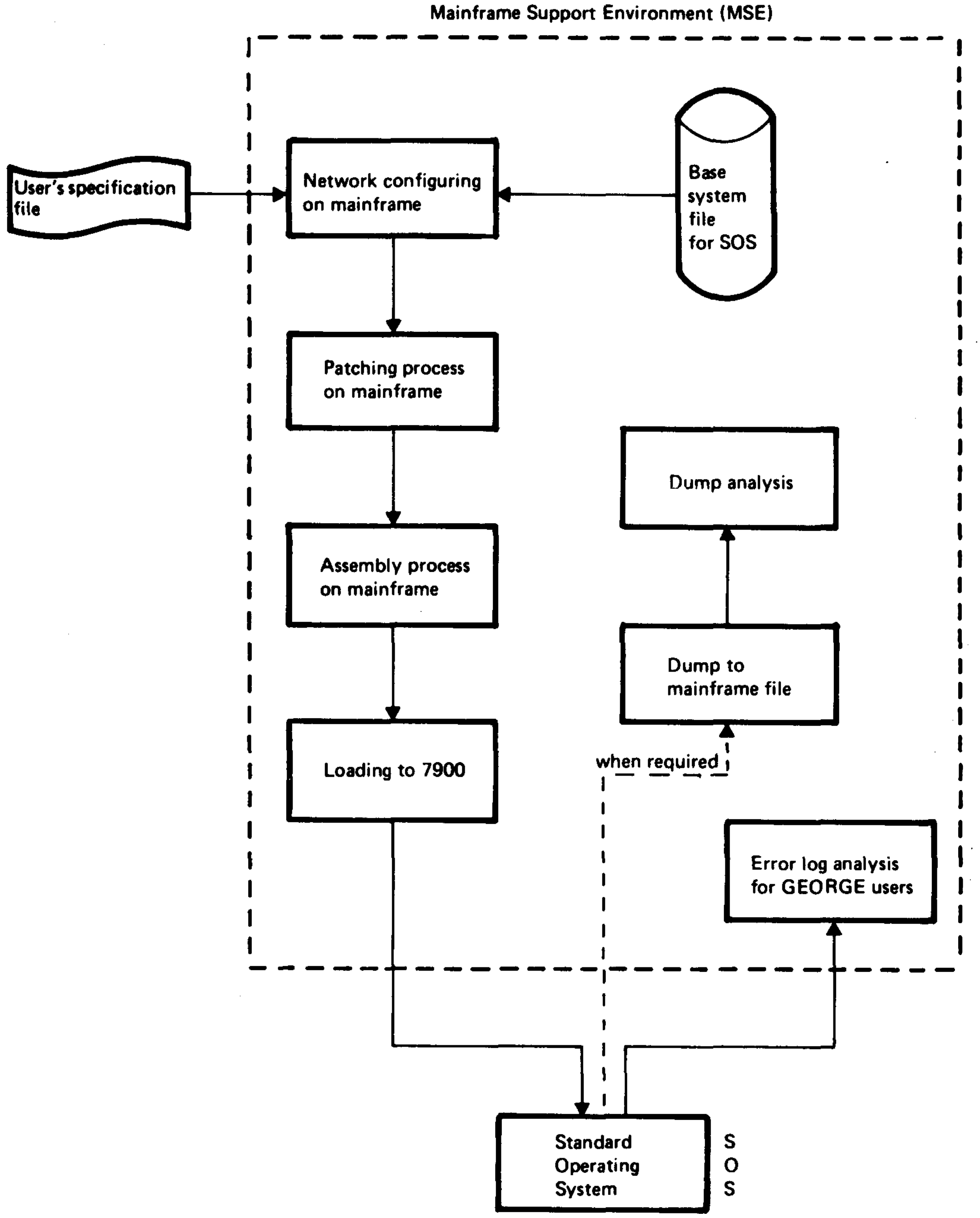

Figure 1.3 Building a Standard Operating System

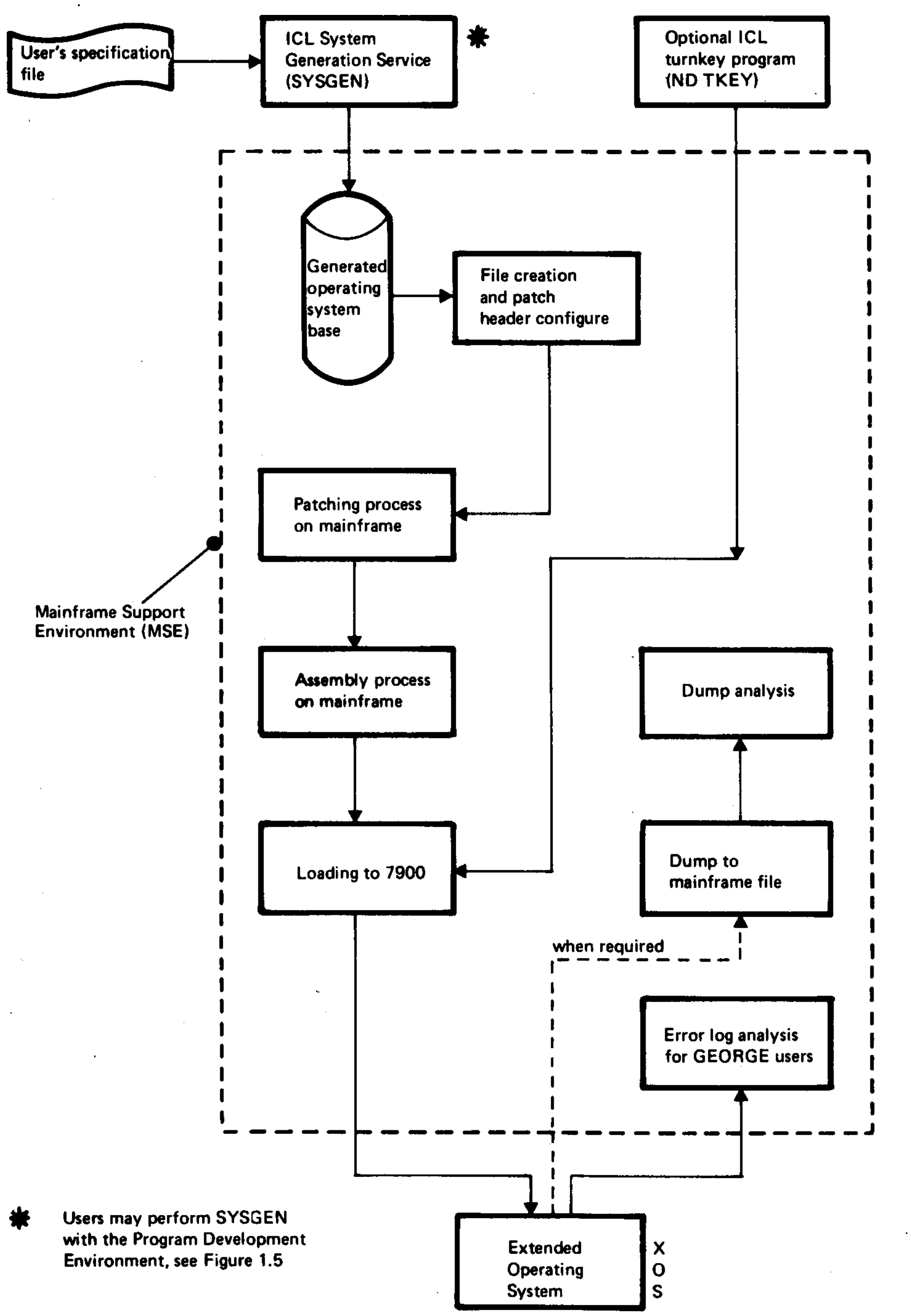

Figure 1.4 Building an Extended Operating System

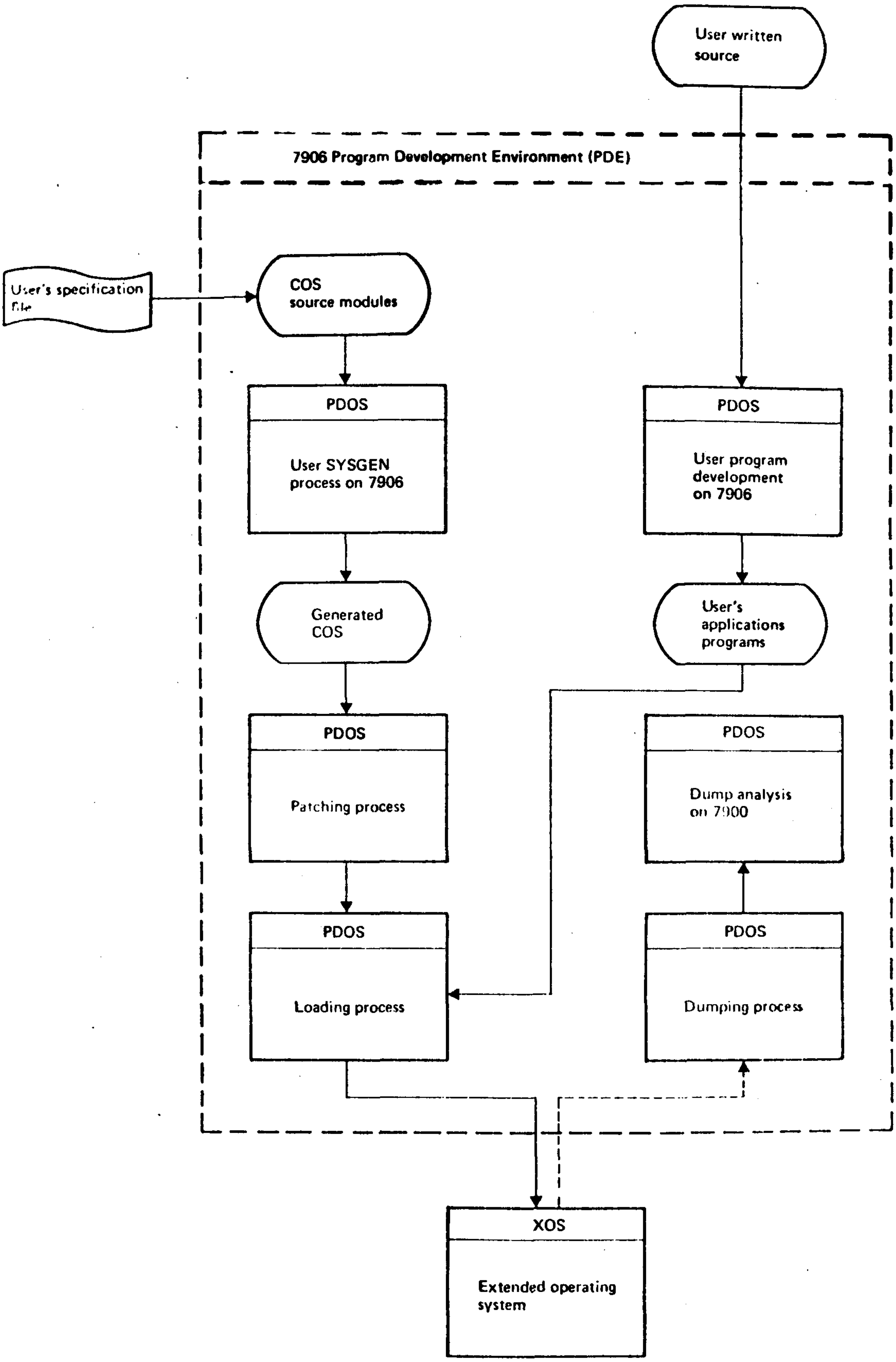

Figure 1.5 The Program Development Environment

1.4 User considerations

1.4.1 Assessing requirements

The following considerations need to be borne in mind when formulating the requirements for a communication system:

1 The number of messages per day, or hour, paying close attention to peak conditions

2 The number and type of terminal devices

3 Line speeds and types, for instance multi-drop, dial-up; for a heavy workload it may be advisable to use private wire circuits for greater reliability. Spare lines may be configured to enable swapping of lines when a fault develops on a line

4 The number and size of store buffers required to hold the messages

5 The processor, communications equipment and modems required

It is good practice to design a network map showing remote devices, lines, and line equipment such as modems, connected to the central processing equipment.

ICL strongly urges users to call on its support services for assistance and advice early in the planning stage. It is important that requirements are accurately known, specified and sized, in order to arrive at a configuration which will perform the required tasks and also allow for expansion and modification as future needs arise.

A brief outline of system resilience and recovery considerations is given in Chapter 5.

1.4.2 Transition and growth

The ICL 7904 or 7906 provides a transition route from a 1900 environment to DME/1900, or to VME/B on 2960, 2970, 2972, 2976 or 2980. This approach safeguards the user's investment in the network.

As traffic volume increases and the user's requirements grow, there arises a demand for more terminals. A 7904 may be expanded from 7904/21 to models 22 or 23, to handle the higher number of lines, or the increased message rate.

If special requirements arise, such as spooling (the storage of messages on disc), use of a remote network node, or connection to more than one mainframe, a 7904 may be upgraded to a 7906. The 7906 SOS caters for larger systems, the 7906 XOS caters for more than one parent mainframe and more than one Local Processor Link, while the 7906 with user written communications programs can cater for spooling and remote network nodes.

The 7906 also has three models for use as a communications processor, numbers 21, 22 and 23, to enable the configuration to expand in step with increasing line activity or number of links. There is also a considerable range in the size of communications network that can be supported by each model, because further modules of store and line connection equipment can be added to each model.

The software is also modular and capable of modification to match the user's changing requirements. The Standard Operating System (SOS) is intended for FEPs and can support straightforward networks of considerable size. Changes in the number and type of terminals connected, and in the mix of line protocols or line speeds used in the network, can often be handled by a simple reconfiguration of the SOS by the user.

When the user's communications network becomes more complex the 7906 can be retained by switching to the Extended Operating System (XOS). Because the message handling part of the XOS can be written by the user, additional facilities such as remote connection to several mainframes, spooling, handling of non-standard terminals, or local connection of the 7906 as an FEP to several mainframes, can be provided. The generalised nature of the 7906 hardware and operating system, combined with access to the operating system routines from a user written program, means that many varied facilities can be developed by user programming.

Details of 7904 and 7906 upgrading and transition facilities are given in Chapter 7.

2.1 Introduction

The Standard Operating System (SOS) runs a 7904 or 7906 system as a front end processor to a single mainframe. It is designed for potentially large scale but straightforward communications networks.

The SOS can be run on 7904 systems with up to 56 Kwords of store and on 7906 systems with up to 112 Kwords of store.

The SOS is configured by the user at the mainframe (1900, DME/1900 or 2900 VME/B) using ICL-supplied software. Only the software modules required for the specified network are consolidated into the SOS which is therefore a compact, efficient, dedicated system. The SOS is loaded across The Local Processor Link (LPL) from the mainframe to the FEP.

The remainder of this chapter describes the main features of the SOS. Full details are in the SOS manuals *1900 Series: Using 7900 Standard Operating Systems*, and its equivalent publication *VME/B: Using 7900 Standard Operating Systems*.

2.2 Functions of SOS

The functions of the SOS are:

1 To interface, via Communications Multiplexer or Command Chain Unit and scanners, with most ICL communications devices (see section 2.3.2)

2 To interface with one mainframe via a Local Processor Link (LPL), using code translation appropriate to the parent mainframe

3 To control the Executive console teletypewriter and local paper tape reader

4 To provide software control of line protocol, polling, clocking and interrupt timing

5 To assemble the characters received from the terminals into messages, and to pass on these messages to the parent mainframe

6 To transmit messages received from the mainframe to the destination terminals

7 To provide the TRACE facility to check that all units are functioning normally while the SOS is controlling the communications network

8 To allow the ICL engineer to run a diagnostic program while the SOS is controlling the communictions network

9 To provide an Error Manager facility, in conjunction with a mainframe with GEORGE 3 or GEORGE 4 operating system, to record errors and transmit them to the mainframe for later analysis

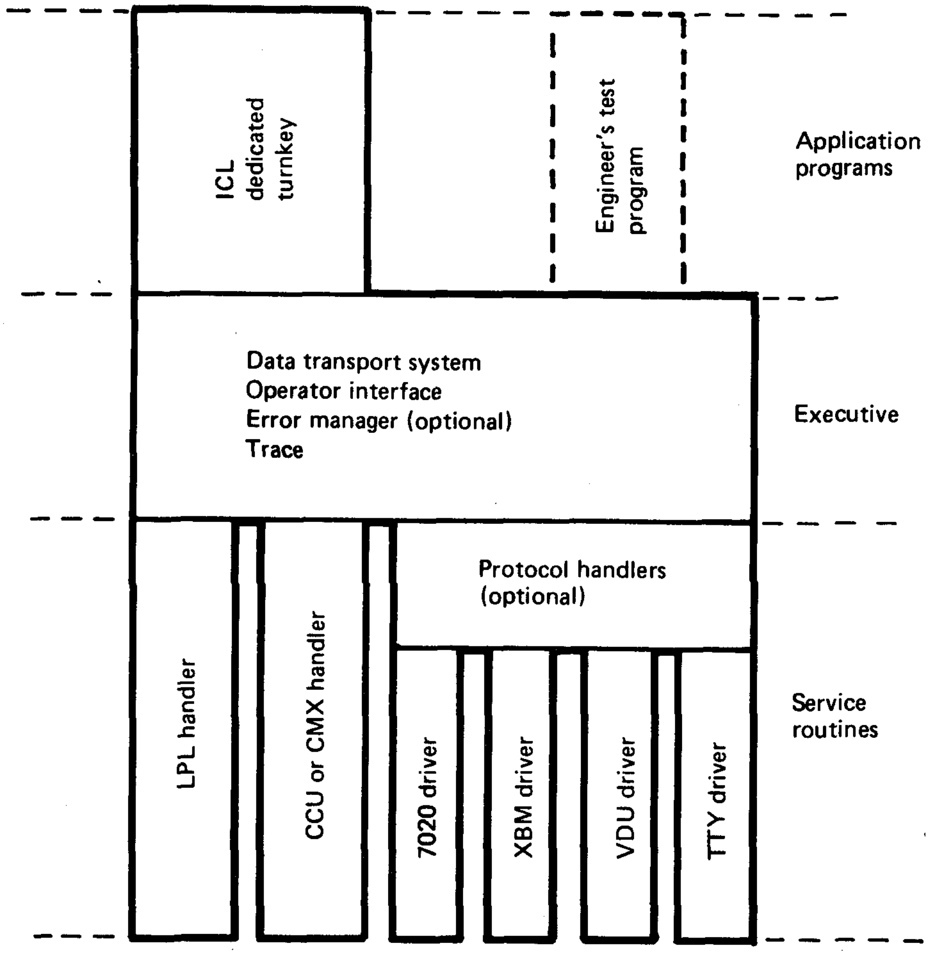

Figure 2.1 Components of the Standard Operating System

2.3 Structure of SOS

Figure 2.1 shows the major components of the SOS. The central portion, the *Executive*, is common to all versions of SOS. The lower area, labelled *service routines*, shows the communication modules from which a selection is made at the configuration stage. The upper area, labelled *application programs*, shows the two operational programs which can be run concurrrently by Executive.

2.3.1 Executive

The major components within the Executive are:

1 Data Transport System
Updates the hierarchy of controls within the system and passes control to the appropriate driver routine

2 Operator interface
The system operator issues commands to the operating system at the Executive Console and receives replies and other messages. The part of the Executive that is responsible for this communication is the Operator Interface. The system operator can control the following functions:

(a) Switching a communications line to its standby speed of transmission, or reverting to normal speed

(b) Exchanging the terminals on two communications lines, or moving the terminals on a failed line to an alternative line

(c) Broadcast of a user specified message to a terminal if it inputs a message when the mainframe is not available

(d) Control of the Trace facility

(e) Use of the Error Manager facility by George 3 and George 4 users

3 Error Manager
This module is available only on systems with mainframe operating under GEORGE 3 or GEORGE 4. It reports errors, such as re-tries on a faulty communications line, to the mainframe for subsequent analysis

4 Tracing is carried out by the system, to provide a detailed record of the activities performed by the main software components of SOS, and therefore helps with the diagnosis of errors. The engineers on-line test program slot is used to store the trace information. Tracing is switched off when a test program is loaded

2.3.2 Service routines

The modules in this section are:

1 The LPL Handler which controls the connection to a local mainframe.

The line control procedures used for LPL connections to 1900 Series and 2900 Series (DME/1900 or VME/B) systems are based on those used to communicate with a Dedicated Control Program (DCP) running a 7903 FEP.

2 The CCU or CMX Handler which contains routines for receiving data from, and transmitting data to, a Command Chain Unit (CCU) or Communications Multiplexer (CMX)

3 Line drivers which service the devices at the ends of the communications links. Different types of device need different contol procedures (protocols). The four types provided, from which a selection is made at the configuration stage, are:

(a) Asynchronous (TTY) devices:
teletypewriters, termiprinters, keyboard printers

(b) Synchronous basic mode bulk (7020 type) devices:
ICL 7020 and 7503 (using T3Ax) Remote Job Entry (RJE) clusters with bulk peripherals; remote 2903 Range and 1500 Series computers emulating basic mode bulk protocol

(c) Synchronous basic mode interactive (7181 type) devices:
Line Sharing Adaptor (TLSA and QLSA) controlling 7181 or 7184 videos; video clusters, with hard-copy printers for direct output, controlled by 7501 (using T2B5), 7502 (using T2Bx) and 7503 (using T3Bx); remote 1500 Series computers emulating basic mode interactive protocol

(d) Extended Basic mode (XBM) synchronous devices:
clusters, with bulk and interactive peripherals controlled by 7501 (using T2C5, T2SAxI, T2AP01, T2ASx) and 7502 (using T2Cx, T2SAxI, T2RBxx, T2AP01, T2IPxx or T2ASx), and 7503 (using T3Cxx or T3D1); remote 2903 Range andd 1500 Series processors emulating XBM protocol

All the above devices can be connected to a 7900 SOS, but the facilities of some of these devices may not be supported by the particular mainframe system to which the 7900 is connected, see section 2.4

2.3.3 Application programs

The ICL dedicated turnkey program is the software element which initiates the service routines as required, and is responsible for routing and buffering messages between terminals and mainframe.

The turnkey program also actions commands input by the console operator to initiate the facilities mention in section 2.3.1.

2.4 Mainframe environments

2.4.1 Mainframe Support Environments

The concept of the Mainframe Support Environment (MSE) is illustrated in Figure 2.2. The MSE represents all the facilities supplied by ICL to support a 7900 operating system; the following aspects of this support for SOS are illustrated:

1 Creation of the SOS by specifying the user's configuration to a Network Configuring Program (NCP), followed by a patching process to incorporate the latest amendments, followed in turn by assembly of the SOS.

2 Loading the complete SOS from mainframe disc file to the 7900

3 In the event of a system failure, the contents of the FEP store can be dumped to the mainframe disc backing store for later analysis. The system can be re-loaded quickly into the FEP, and can therefore be back in use while the dump is being analysed, printed and examined

4 An error log maintained by George 3 or George 4 in conjunction with the 7906 module Error Manager described in Chapter 5. The log can be analysed to produce statistics on the performance of lines and devices so that repeated failures can be high-lighted and corrective action take.

Note that for George 3 and 4 users, and for VME/B users, a set of macros is provided by ICL to run the mainframe utility programs included in the MSE.

The user must employ the version of Mainframe Support Environment appropriate to the type of parent mainframe system in use. The mainframe systems, and the names of the software sets that provide the associated Mainframe Support Environment (MSE), are given in the following table.

Mainframe System	*Name of software set for MSE*
1900 Series or DME/1900 using Multiprogramming Overlaid Executive (MPOE)	MME
1900 Series or DME/1900 using GEORGE 3 or 4	MG3
2900 Series using VME/B old KOMMFY	MVB

The communications facilities of the above mainframe systems that can be used with devices connected through 7900 SOS are given in the following sections.

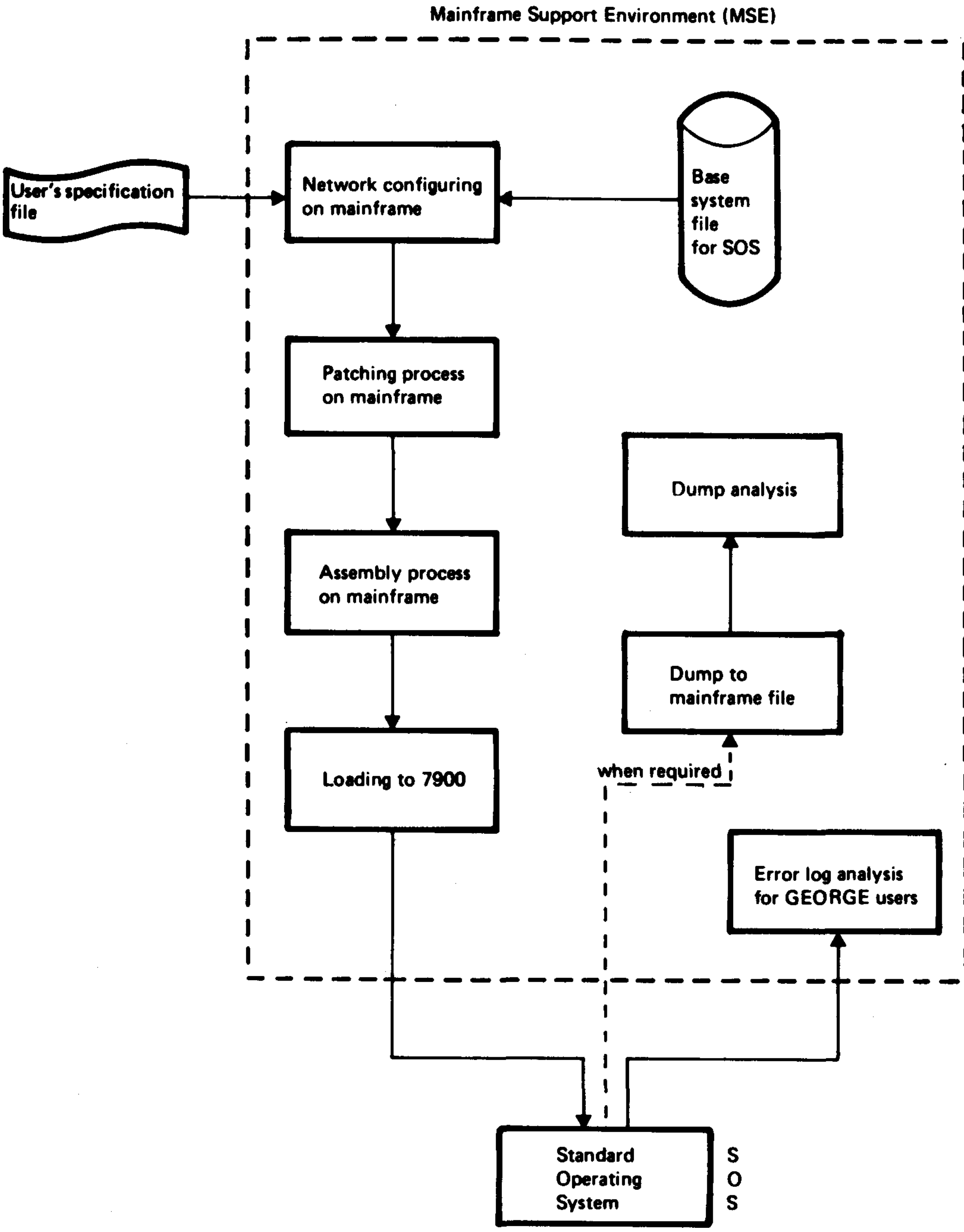

Figure 2.2 Building a Standard Operating System

2.4.2 1900 Series MPOE (Multi-programming Overlaid Executive) and GEORGE 2 systems

1 Under both MPOE and GEORGE 2, when used with Communications Manager, mainframe application programs written to the appropriate standards (but with buffer size restricted to 597 words maximum) can communicate with devices using asynchronous, basic mode interactive or XBM protocols.

Further details are given in *Communications Manager* (Edition 1, TP4420)

2 Under MPOE the MAXIMOP system can be used with Communications Manager to support data input and output and interactive program control from devices using asynchronous, basic mode interactive or XBM protocols. The MAXIMOP text editor may be used, and the same terminals can be used to access the GEORGE 2 RJE system (see below).

Further details of 1900 MAXIMOP are given in *Maximop System* (Edition 1, TP4399) and *Using Maximop* (Edition 1, TP4398)

3 Using GEORGE 2, #XKVB and Communications Manager, Remote Job Entry facilities are supported for 7020, 7503 and 7502 terminal clusters using basic mode bulk or XBM protocols. Job descriptions and data are input off-line, processed using the GEORGE 2 central module and the results printed off-line.

Further details of GEORGE 2 RJE are given in *GEORGE 2 Disc Based Operating* (Edition 1, TP4432)

2.4.3 1900 Series GEORGE 3 or 4 systems with integral MOP and RJE facilities

1 GEORGE 3 or 4 MOP facilities support data input and output, and interactive job and program control from devices using asynchronous basic mode interactive or XBM protocols. The GEORGE 3 or 4 text editor can also be used

2 GEORGE 3 or 4 Remote Job Entry (RJE) facilities support 7020, 7503 and 7502 terminal clusters using basic mode bulk or XBM protocols. Job descriptions and data files are input off-line to filestore, processed under control from the cluster console, and the results printed off-line

3 Mainframe application programs written to the appropriate standards can communicate via Communications Manager with devices using asynchronous, basic mode interactive or XBM protocols. Further details are given in *Communications Manager* (Edition 1, TP4420).

Any SOS running in a processor that is MOPPED OFF can be connected on-line to an application program. Mainframe identifiers known to a SOS may be attached (AH command) to a GEORGE message-buffering conceptual; the conceptual can then be connected on-line (OL command) to an application program.

Note: When GEORGE 3 or 4 requests a configuration report at load-time, a cluster of devices connected to a terminal processor is represented by the SOS as though the devices are multidropped along the line (see figure 2.3).

Further details of GEORGE 3 or 4 commands and facilities are given in *GEORGE 3 and 4 Operation Management* (Edition 6, TP4438) and *Operating Systems GEORGE 3 and 4* (Edition 5, TP4345)

2.4.4 2960 Processors with DME/1900

The facilities provided are as described in sections 2.4.2 and 2.4.3

2.4.5 2900 Series VME/B old KOMMFY systems

1 Under VME/B TP option, mainframe application programs written to the appropriate standards can communicate with interactive devices using asynchronous, basic mode interactive, or XBM protocols

2 Under VME/B the MAC service can be used to support data input and output, and interactive job and program control from interactive devices using asynchronous, basic mode interactive or XBM protocols

3 Under VME/B the RJE service can be used to support Remote Job Entry facilities for 7020 or 7503 systems using basic mode bulk protocol, for 7503 systems using PCT2 protocol, and for 7502 and 7503 systems using XBM protocol. Job descriptions and data files can be input to the mainframe, jobs processed under control from the terminal system console, and results output

Note that support for XBM protocol (and its sub-set PCT2 protocol) is not a standard facility of VME/B old KOMMFY when the lines are connected via a 7900. The user should consult his ICL suppport representative if he wishes to use these protocols.

2.5 User considerations

The preliminary work leading up to the final specification of the network to be handled by the operating system should include, as a minimum, the following steps. These would usually follow a systems analysis and design function.

1. Definition of the terminal devices required
2. Definition of any clusters of devices (that is, using the same communications line to the FEP)
3. In the case of ICL 7500 Series terminal systems, the hardware model and terminal executive in use should be defined
4. Definitions of lines required and their characteristics, such as telegraph or telephone, private wire or dial-up, modems, protocols and speeds
5. Definition of CCU channels or CMX ports
6. A network map should be drawn up. An example is given in Figure 2.3
7. A sizing operation should be performed on the resultant configuration. This should be done in conjunction with ICL support staff. The performance of a communications system is dependent on many factors, including:

   (a) Mainframe hardware, software and loading

   (b) 7904 or 7906 hardware and software

   (c) The number and types of terminal, their message sizes and polling rates

   The importance of sizing therefore, cannot be over-stressed.
8. Bearing in mind possible future requirements of the network, a decision can be taken as to whether the SOS is suited to the task or whether the capabilities of the XOS should be examined. XOS is described in Chapter 3

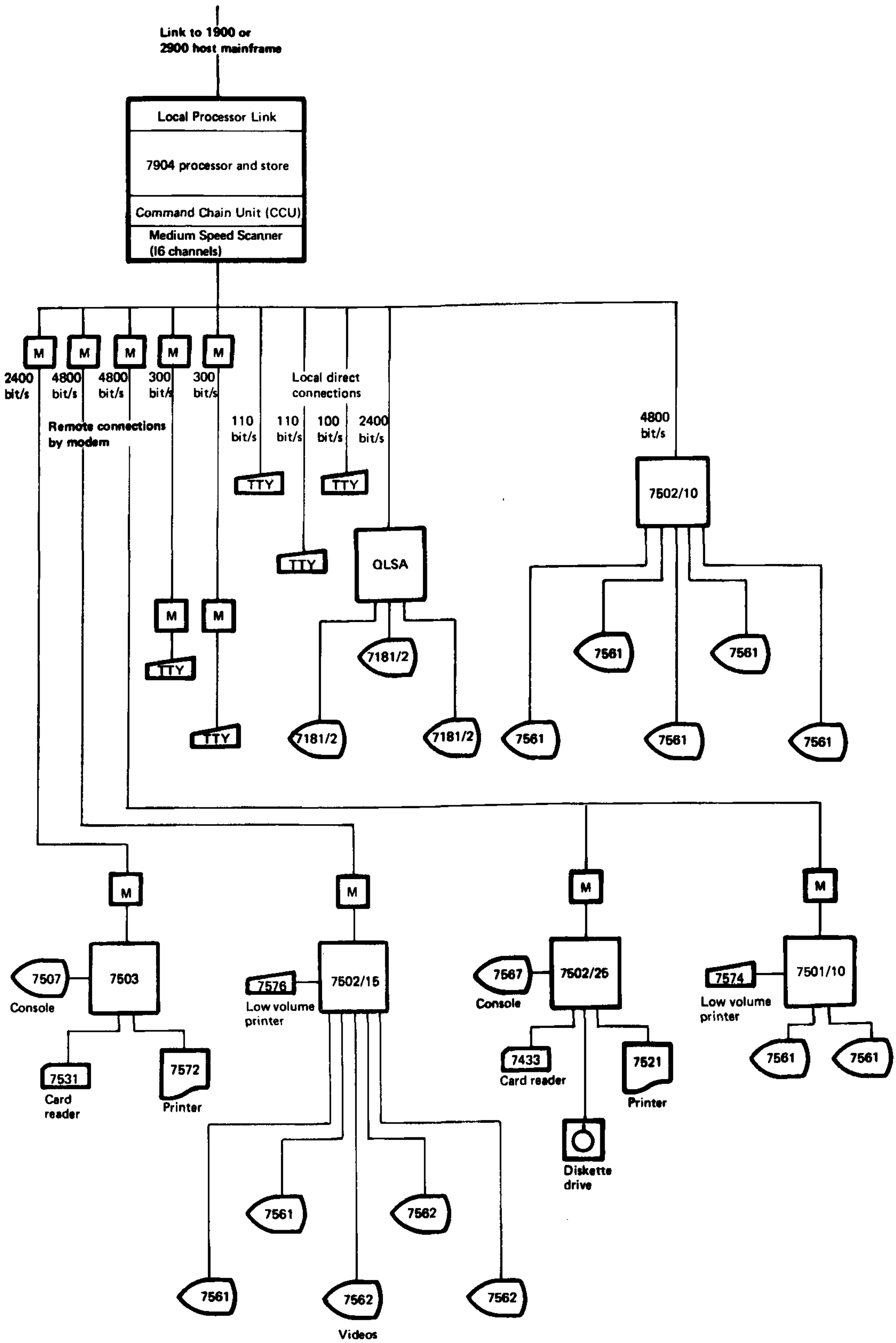

Figure 2.3 Example network map for a 7904/23 front end processor

3.1 Introduction

The Extended Operating System (XOS) is a communications operating system used in front-end or remote 7906 processors. XOS is intended for the customer with a large-scale or complex communications network whose needs cannot be fully satisfied by the Standard Operating System described in Chapter 2. XOS provides precise tailoring of software to the user's requirements.

The XOS has two major divisions:

1 An operating system base, generated by means of the SYSGEN suite of programs, as described in Chapter 4.

2 One or more communications applications programs.

The SYSGEN procedure may be carried out by the user or by ICL. Similarly, the applications programs may be user written, in NAL or CORAL, or the ICL non-dedicated turnkey program ND TKEY may be used. ND TKEY is only applicable to the use of a 7906 as a front-end processor.

There are therefore two methods of producing an XOS which can be summarised as:

1 User-developed, where the user generates the XOS base by means of SYSGEN on a 7906, and develops his application programs, also on a 7906. For this, the Program Development Environment (PDE) described in Chapter 4 is required. The final system is either dumped to the mainframe disc store and re-loaded to the FEP as required, or it is loaded from 7906 disc to store.

   This method of producing an XOS is described further in Sections 3.3.2 and 3.3.3.

2 Specification of the SYSGEN steering file, followed subsequently by the loading of the generated system into the 7906, together with the ICL non-dedicated turnkey program ND TKEY. The communications configuration can then be specified to ND TKEY. This method has the advantage that the user need not concern himself with programming an FEP system

   This method of producing an XOS is described further in Sections 3.3.4 and 3.3.5.

Installing, running and maintaining an XOS are described in the publications *1900 Series: 7906 Operation* and *VME/B: 7906 Operation*.

3.2 Facilities

The facilities provided by an individual XOS depend on which method is used to produce the XOS. The facilities are described in the following two sub-sections.

3.2.1 User-developed XOS facilities

For the user who develops his own communications programs, XOS can provide all that SOS provides, plus the following features:

1 Precise tailoring of the operating system to user requirements at load time

2 User choice of buffer sizes and priorities for different terminals and message types

3 Up to four Local Processor Links can be handled

4 Multi-programming is supported

5 The programs can take full advantage of all the facilities offered by a 7906 processor, such as the use of local peripherals (discs and line printer), or operation as a remote communications processor, by means of the Communications User Interface (CUI). The CUI is described in the publication *7905: Communications User Interface*

6 The use of a timing mechanism known as the Interval Timer

These facilities mean that an extensive range of communications requirements can be satisfied, including: local or remote intelligent concentrator; a network node; spooling of data received during a time of non-availability of the mainframe or a time of peak activity; and routing of messages by type or message content.

3.2.2 Facilities available to the ND TKEY user

The user who elects to use the program ND TKEY has the facilities offered by SOS (see Section 2.3) plus the following features:

1 Connection of more than one Local Processor Link

2 Connection to more than one parent mainframe

3 Selection of devices to be used in the communications network by means of commands issued to ND TKEY at run time, provided these devices have been declared to SYSGEN

ND TKEY is only available for a 7906 used as a front-end processor. The ND TKEY user has no involvement in the complexities of user program development.

3.3 Structure of XOS

3.3.1 Components of XOS

Figure 3.1 shows the major components of XOS. The *Executive* is very similar to the Executive used in Standard Operating Systems, but can also provide a general multi-programming facility. The *service routines* required are selected by the system generation process, and the resultant combination of Executive and service routines is called the *operating system base* or *Communications Operating System* (COS). Note that in addition to the service routines available with SOS, XOS can also support local peripherals, several Local Processor Links, and connection to remote mainframes.

The operating system base can support several *application programs* provided that sufficient store is available in the 7906. If Fixed/Exchangeable Discs are available on the 7906, then the virtual store facility can be used to increase the effective store available to application programs.

The ICL non-dedicated turnkey program ND TKEY can be used, and user-written programs can be used. Note that usually either ND TKEY or user programs are used, but if store size permits both can be used simultaneously.

User-written programs make use of the facilities offered by the operating system base by issuing commands from the user-written program. There is a very wide range of commands available, giving detailed control over the operating system facilities. The commands form a set called the Communications User Interface (CUI), described in the publication *7905: Communications User Interface*.

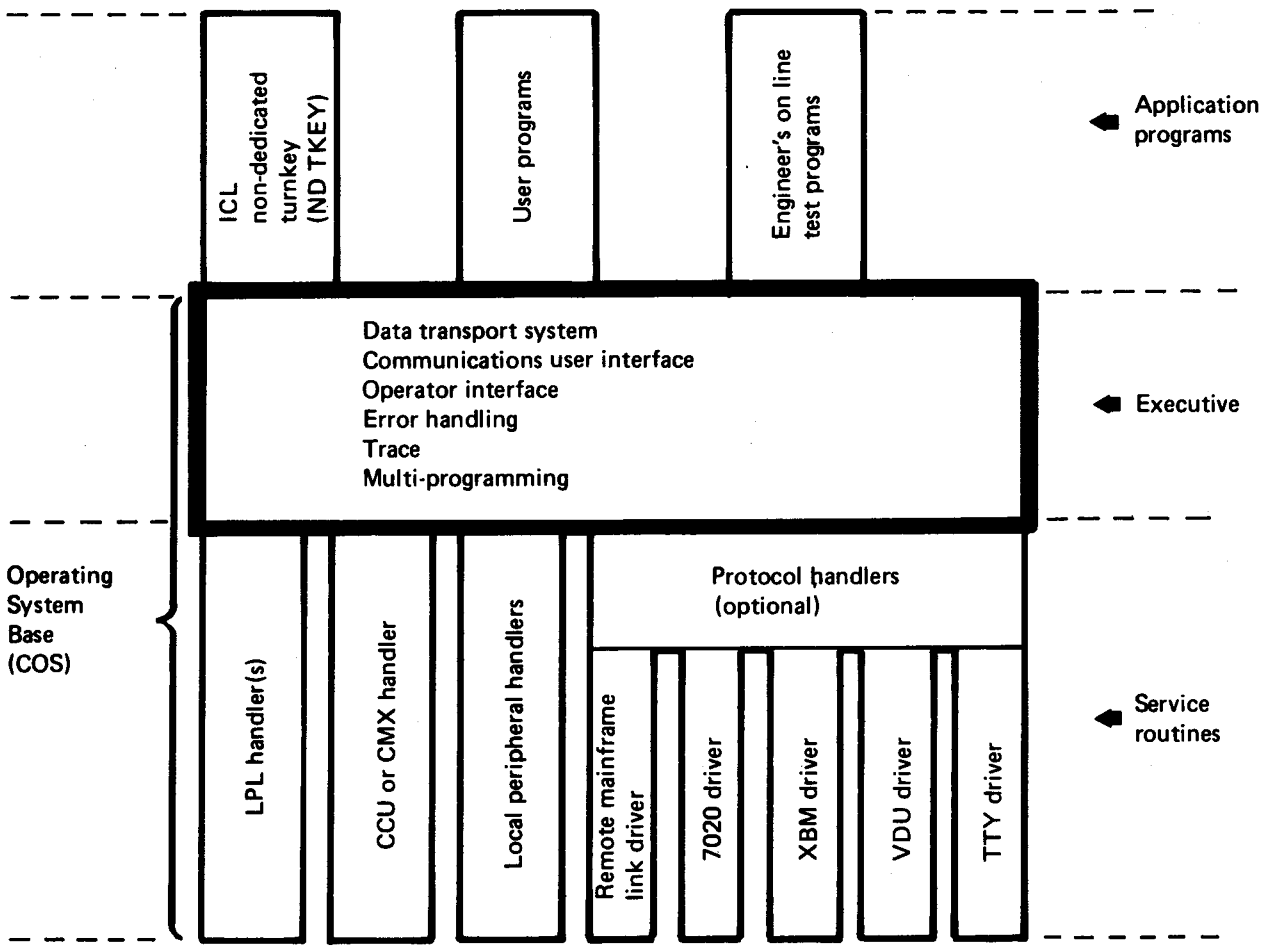

Figure 3.1 Components of the Extended Operating System (XOS)

3.3.2 System generation by the user

The user can generate an XOS on a 7906/24 with a Program Development Environment (PDE), as described in Chapter 4.

Generation is performed by specifying the communications system in a steering file to the SYSGEN suite of programs. By this means the generated system may be tailored specifically to the user requirements with minimum hardware and software redundancy. The steering file defines, for instance, store size, CCU or CMX, buffer sizes, amount of workspace, peripheral devices, communications lines and their characteristics, device drivers and so on. User generation is the most detailed and specific method of generating a 7906 operating system to the user's exact requirements.

3.3.3 User communications programs

As described in Section 3.1 the user may decide that it is necessary to write his own communications programs rather than using the ICL turnkey program. By doing so, the full facilities of the 7906 are made available such as:

1. Connection to remote mainframes and up to four Local Processor Links
2. Complex networks and high data throughput rates can be catered for
3. Commands and messages for the system operator may be devised and incorporated
4. The use of local peripheral devices is possible, particularly the Fixed/Exchangeable Disc Storage (FEDS) device and the Line Printer.
5. The programmer has a choice of multiple buffering arrangements for different terminal and message types
6. Priorities can be allocated to different message types or different remote locations
7. Timing mechanisms are available, through the use of the 7906 interval timer

3.3.4 System generation by ICL

If the user so decides, or a 7906/24 is not available, he can use one of the system generation services offered by ICL. This is the usual method of generating the base operating system to support the ICL non-dedicated turnkey program ND TKEY.

There are two ICL generation services:

1. Service 1 is for the initial hardware configuration, and for new ICL hardware or software that requires regeneration for its support. Service 1 is free to the user
2. Service 2 is for all other purposes and would normally be in response to changed user needs. Service 2 is charged to the user

A steering file is specified by the user in the same manner as for user-generation, and is then passed to ICL for checking and generation of the required base system. The new system is returned to the user tailored as specified, with unwanted modules excluded. If the generated system is to support the ND TKEY program, then only the sub-set of the system generation commands that is applicable to ND TKEY can be specified. The user has more control over the tailoring of the operating system than is available to the SOS user. For example the amount of work space and control block space can be specified.

Details of the use of the ICL generation services are given in the ICL publications *1900 Series: 7906 Operation* and *VME/B: 7906 Operation*.

3.3.5 ND TKEY

Whether the base system is generated by the user or ICL, the user may decide that the non-dedicated turnkey program ND TKEY meets his requirements for using the 7906 as a front end processor. ND TKEY is provided by ICL on paper tape ready to load into the 7906.

ND TKEY interfaces with the base operating system to provide buffering and routing of messages. It supports all the device types and protocols supported by SOS (see Section 2.3.2).

Although ND TKEY does not allow changes to the 7906 configuration as all devices must be declared when the system is generated by SYSGEN, it does allow the system operator to specify which devices are to be used by the system at any time. Commands may also be given to ND TKEY at run time to create files to handle particular devices and streams.

The operator can specify buffers for tracing routines to use for monitoring parts of the operating system.

It is possible to dump the ND TKEY program to the mainframe, after initilisation but before running the program. In this way, subsequent loadings from the mainframe will load ND TKEY in an already-initialised state; the program can also start running automatically after loading if required. This procedure provides a fast *load and go* facility.

Details of ND TKEY are given in the ICL publications *1900 Series: 7906 Operation* and *VME/B: 7906 Operation*.

3.4 Mainframe environments supporting XOS

XOS is supported by 1900 series processors operating under GEORGE 3, GEORGE 4 or MPOE; by 2960 processors under DME/1900; and by 2900 systems under VME/B old KOMMFY. Details are the same as those given for SOS environments in Sections 2.4.2, 2.4.3, 2.4.4 and 2.4.5.

The Mainframe Support Environment (MSE) shown in Figure 3.2 is necessary for users of ND TKEY, but is optional for user-developed XOS. There are three versions of the MSE to match the three types of mainframe system quoted above. The versions are the same as those required by SOS users, see Section 2.4.1.

Figure 3.2 shows the following features of the MSE supporting an XOS employing ND TKEY:

1 Specification of the system configuration to the ICL SYSGEN Service, followed by patch header generation, patching and assembling of the operating system base returned to the user

2 Loading of the generated operating system base and ND TKEY into the 7906 to form XOS

3 Error logging and dumping are as described for the SOS in Section 2.4

3.5 User considerations

Following the preliminary analysis described in Section 2.5, the following decisions are necessary in order to achieve an extended operating system:

1 The Standard Operating System does not satisfy all the requirements, and therefore an XOS is required

2 Does ND TKEY provide all the communications facilities required? If not, specify, write and test in-house programs. (This entails the use of a 7906/24 for compilation and testing)

3 Is a 7906/24 available on which SYSGEN can be run? If not, use the ICL SYSGEN service

The next step is the preparation of the SYSGEN steering file that defines the requirements of the operating system base. A brief description of SYSGEN, with references to other publications, is given in Section 4.3.4. The operating system base is then generated.

Finally, the generated system, plus ND TKEY or the in-house programs, are loaded, any necessary commands are input, and the system is then ready to run.

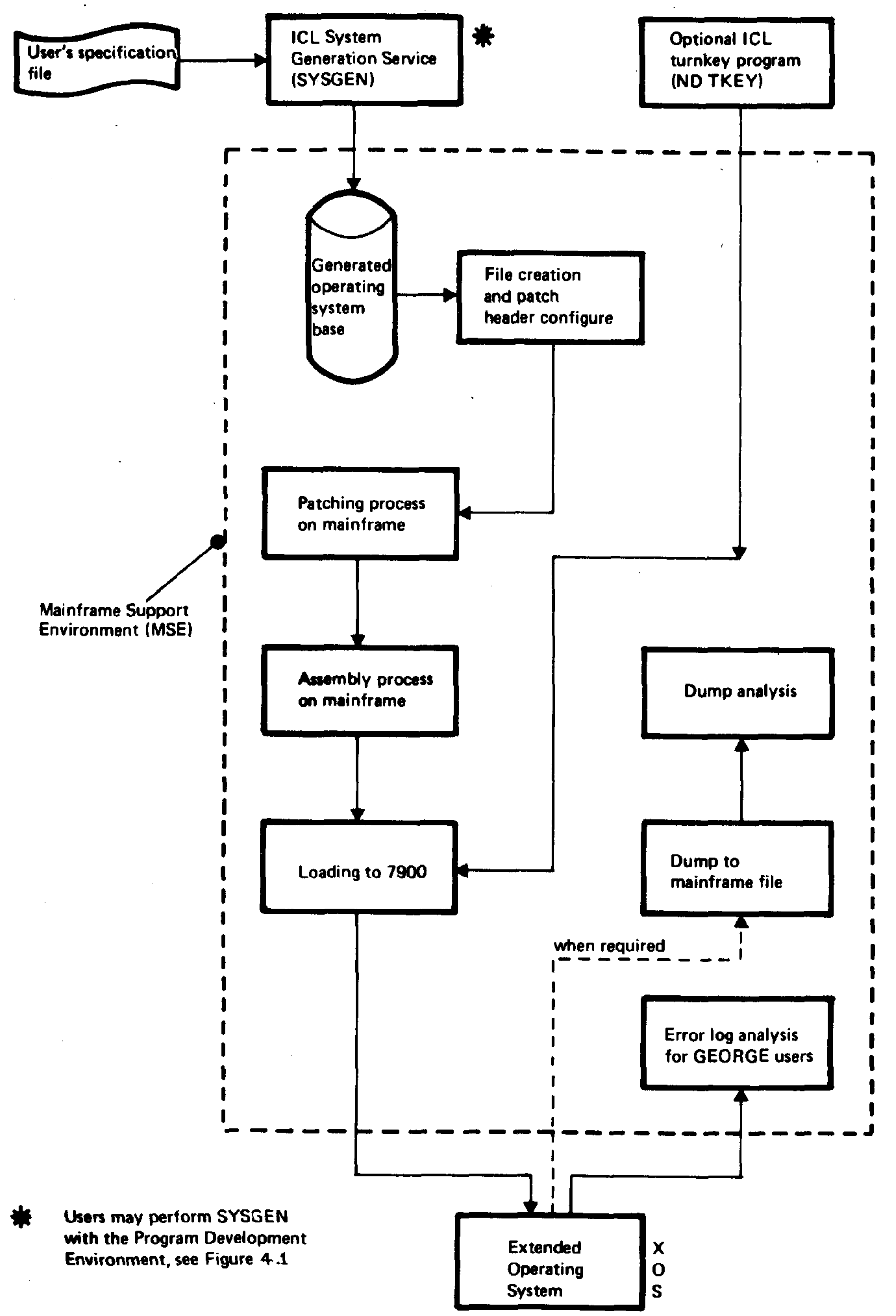

Figure 3.2 Building an Extended Operating System

4.1 Introduction

The Program Development Environment (PDE) is used with a 7906/24 system to develop Extended Operating Systems (XOS) in which the communications programs are written by the user. As described in Chapter 3, an XOS comprises one or more communications programs run under a communications operating system (COS) whose facilities must match those required by the communications programs. The PDE therefore also enables a user to generate his own versions of the communications operating system, by using the model 7906/24.

An operating system developed by PDE gives the user full control over all the communications operating system functions supplied by ICL, by means of the user-written communications program. This means that the user can provide all the benefits and facilities of XOS systems described in Chapter 3.

The PDE comprises software available from ICL, which is run on a 7906/24 as a free standing computer system; no connection to a mainframe is necessary. An XOS can be developed, by using PDE, to control any model of 7906, acting either as a front end processor or as a remote communications processor.

The PDE software is run on the 7906/24 under the control of an operating system called the Program Development Operating System (PDOS). This operating system is not a communications operating system like SOS or XOS; it is used to control the use of a free standing 7906/24 for program development. A basic version of PDOS is provided as part of PDE.

4.2 Facilities of PDE

The way in which the Program Development Environment is used to create an Extended Operating System (XOS) is shown in Figure 4.1. PDE provides facilities in three main areas:

1 An operating system (PDOS) and associated software to support the efficient use of all software within PDE. The facilities provided in this area are:

- (a) Control of the operation of the system through the Executive console
- (b) Use of all local peripherals available on a 7906 system
- (c) Support of local user teletypes for program development work
- (d) Control of batch and interactive jobs by a job control language provided by the Advanced Operating Facilities (AOF)
- (e) Creation and control of disc files by the Standard Disc Filing System (SDFS)

2 A suite of software for developing communications programs written by the user. The facilities provided in this area are:

- (a) Assembly of programs written in the 7900 assembly language called NAL
- (b) Compilation of programs written in the high-level language CORAL 66
- (c) Program editing, linking and loading
- (d) Testing and debugging programs
- (e) Dumping and printing programs
- (f) Data management, such as disc volume and file managers, disc and file copying, and user Dictionary control

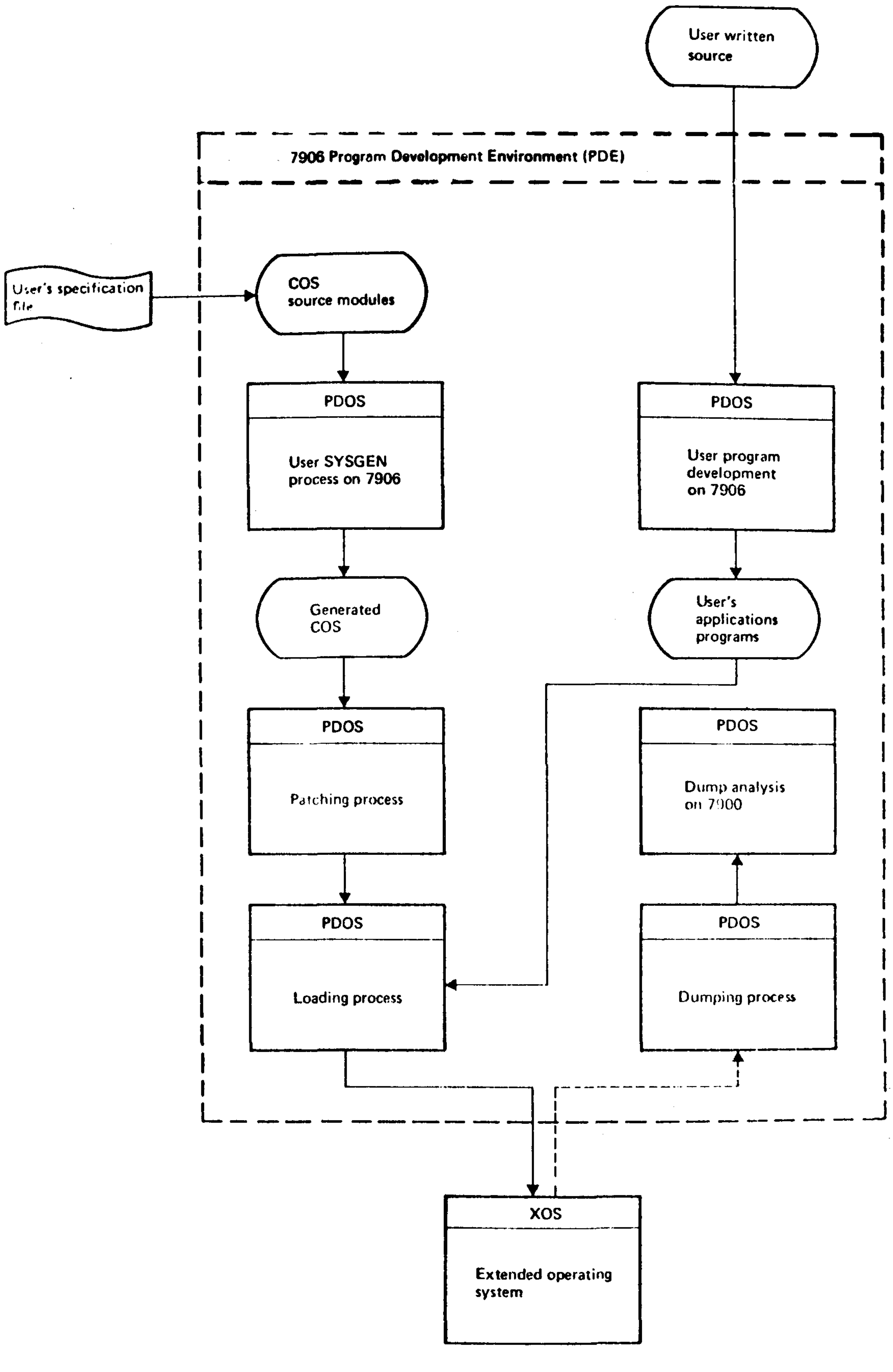

Figure 4.1 The Program Development Environment

3 A suite of programs called SYSGEN for generating an extended operating system base from the full set of software modules provided by ICL. The facilities provided in this area are:

   (a) Selection only of those modules required to support the user's particular hardware configuration

   (b) Incorporation only of the operating system facilities required by the communications programs written by the user

   (c) Validation of the steering file created by the user for input to the SYSGEN process

   (d) Generation of the operating system base for any XOS system

   (e) Generation of a version of PDOS specifically tailored to the user's program development requirements. The PDOS generated by the user can replace the basic version of PDOS issued by ICL for running the Program Development Environment

   (f) Programs for tidying, clearing, and copying files

4.3 Using the PDE

The major software elements of the Program Development Environment (PDE) are shown diagrammatically in Figure 4.2. The individual software items are described in the following sections.

4.3.1 PDOS

PDOS is the initial operating system provided by ICL, enabling a user to undertake general maintenance work, program development work, and system generation. PDOS supports the Standard Disc Filing System (SDFS) and Advanced Operating Facilities (AOF), but no communications software. PDOS runs on, and supports, the following configuration:

56 Kwords of store
Interval Timer
2 Fixed/Exchangeable Discs (4 x 2.5 = 10 Mbyte)
Line printer
Executive console
User teletypewriter locally connected
Paper tape reader

PDOS has a multiprogramming capability, so if store size permits several PDE programs can be used concurrently.

In order to make use of extra hardware, for example more store, a second printer, or the paper tape punch, the user should generate an appropriate program development operating system, using the system generation suite (see section 4.3.8)

4.3.2 Advanced Operating Facilities (AOF)

The Advanced Operating Facilities (AOF) provide the following features:

1 A simple but comprehensive job control language (JCL), for interactive use at user teletypewriters or for input to a batch job control file. A monitor file is automatically printed for all jobs

2 Multi-access to shared system resources. The AOF organises the use of hardware by scheduling concurrent jobs for optimum throughput. Input and output files are spooled to disc, priority being given to interactive work

3 Commands for creating, amending, deleting, copying and printing disc files and records, in conjunction with the Standard Disc Filing System (SDFS) and associated disc utilities

4 Global and local macro commands which call macro definitions stored on disc

5 Setting-up sub-systems to run programs that are frequently used, such as the compiler, the editor or other utilities

6 An editor (EDITAOF) for modification of program and text files

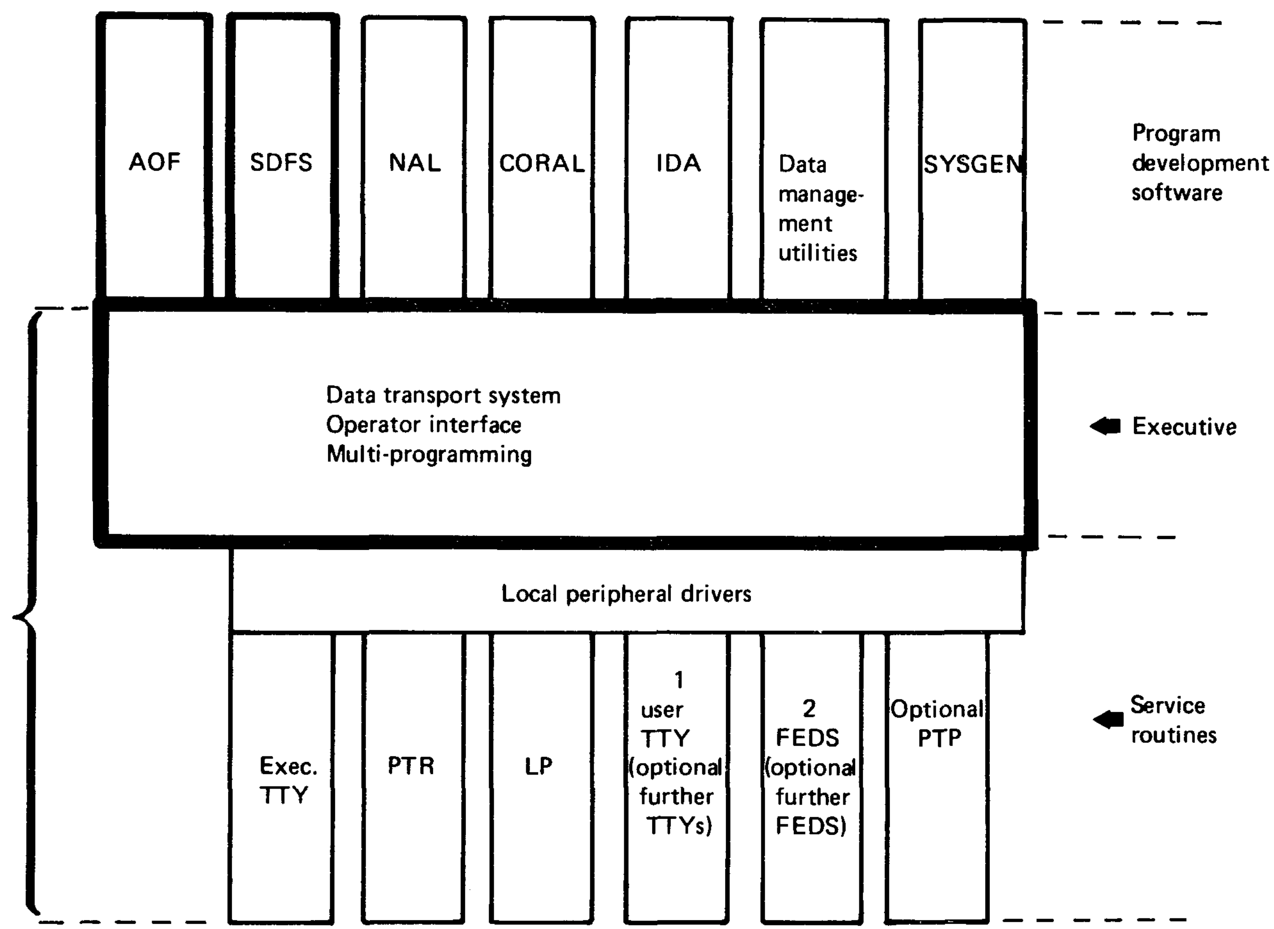

Figure 4.2 Components of the Program Development Environment (PDE)

4.3.3 Standard Disc Filing System (SDFS)

The Standard Disc Filing System (SDFS) is a disc file management system. It executes disc commands issued by 7900 programs, for example, a command to read a record. Facilities are provided to create, amend, delete and copy disc files and records, and to enable many users to share discs.

4.3.4 NAL (New Assembly Language)

NAL is a flexible block-structured assembly language that is particularly suited for programming the 7906. A source program may be developed as a series of modules to be assembled separately and subsequently combined by a link editor program.

NAL is described in the following ICL publications:

LIMPID Link Editor
7905: Introduction to NAL Programming
NAL (New Assembly Language)
NALA (E4) User

4.3.5 CORAL 66

CORAL 66 is a general purpose high-level langauge designed specifically for real time applications. CORAL 66 is block-structured, and fast, compact object programs can be produced using this language.

A text-replacement macro processor is included in the compiler, allowing the incorporation of NAL statements in the CORAL source program.

CORAL 66 is described in the following ICL publications:

7905 CORAL 66 Programming
7905: CORAL 66 Language
7905: CORAL 66 User

4.3.6 IDA (Interactive Development Aid)

An interactive development aid (IDA) is provided, to permit the development and debugging of programs in interactive mode. IDA enables snapshots of current activity to be initiated within the program under development, and also enables commands to be issued from the user teletype to control program areas and current activities.

IDA is described in the ICL publication

7905: IDA - Interactive Development Aid

4.3.7 Utility programs

Several utility programs are supplied to aid the user; they may be run interactively or in batch mode and fall into the following broad categories:

1 Assemblers for NAL and a link editor (LIMPID)

2 Compiler for CORAL, and CORAL formatter

3 Debugging aids such as IDA, post-mortem dump and post-mortem print

4 Editors for disc data (DEDIT), system source (MEDIT), and source text (EDITAOF)

5 Loaders such as the initial bootstrap program

6 Copying programs for discs, dictionaries, disc partitions, and files

7 Disc organisation programs such as volume manager, initialisation of partitions, dictionary manager and file manager

4.3.8 System generation (SYSGEN)

The user system generation process consists of running a suite of programs (the SYSGEN suite) on a 7906/24 system. The process selects and configures modules from master files of source modules supplied by ICL, and produces a loadable, binary operating system base. The operating system base generated by SYSGEN is held as a 7906 disc file, and can be one of several types:

1 An alternative program development operating system (PDOS) to the basic PDOS issued by ICL

2 An operating system base to be used with the ICL program ND TKEY to form an extended operating system (XOS)

3 An operating system base to be used with one or more user-written communications programs to form an XOS

4 A combination of any of the above types

The modules to be selected and configured by SYSGEN are specified by commands in a steering file created by the user. The main stages of SYSGEN are:

1 Validation of the user's steering file and insertion of default values into steering file commands (program SYSVAL)

2 Selection of source code modules and control block modules from the COS master files supplied by ICL (program SYSCODE)

3 Macros in the selected control block modules are expanded to pure code by program ML/1, and program SHUFL is used to set up data structures within the control blocks

4 The code and control block modules are assembled into binary modules by a variant of the NAL assembler (NALG)

5 The binary modules are linked together by the link editor LIMPID

6 The resultant binary operating system is written to 7900 disc by program CDLOAD

An overview of the SYSGEN process is given in *7905 Introduction to the Operating System*; guidance on how to plan a system generation is also given. The definition of all steering file commands available is given in the current version of a computer listing titled *COS Options and Device Macros*.

As an alternative to system generation by the user, ICL offers two system generation services. These are described in Chapter 3, and full details of their use are given in *1900 Series: 7906 Operation* and *VME/B: 7906 Operation*.

4.4 User considerations

PDE allows the user to develop versions of the extended operating system which are tailored precisely to his own requirements, and to provide facilities which are not available with the ICL turnkey programs.

This is achieved by the user designing, writing and testing his communications programs, which use the operating system functions by issuing commands from a set of commands called the Communications User Interface.

The work involved in developing a communications program should not be underestimated. The requirements of the communications system must be established by detailed systems analysis, with help and advice from ICL support staff. If, as a result of the systems analysis, the user decides to proceed with program development, then the following major activities are involved:

1 Systems design, which requires a thorough understanding of the Communications User Interface and the structure of the ICL communications operating system (COS)

2 Systems programming by staff experienced in 7906 systems, and in CORAL 66 and NAL programming languages

3 Management of a model 7906/24 system to the standard expected for any small computer installation

4 Initialisation of the Program Development Operating System and its associated sub-systems such as Advanced Operating Facilities and Standard Disc Filing System

5 Operating the model 7906/24 during program development as a free standing computer system

6 Generation of base operating systems by using the SYSGEN suite of programs

7 Testing and maintenance of a real time system

A user who is not developing his own communications programs may still use the program development environment for system generation. This allows the user to generate his own operating system base to support the ICL non-dedicated turnkey program ND TKEY.

5.1 Introduction

Communications systems help the terminal users to do their jobs more effectively, and thereby increase the efficiency or competitiveness of their company. Any prolonged failure of the communications system may therefore have a serious effect on the day to day operation of the company. ICL recognises this situation, and strives to provide hardware and software communications products of high reliability. However even in the most reliable systems a failure may occur, so facilities must exist to minimise the effect of the fault and to correct it as soon as possible.

This chapter is concerned with the facilities provided in 7904 and 7906 systems for:

1. Safeguarding data and programs in the event of failure of one of the elements of the communications system
2. Maintaining a service to the terminal users and minimising disruption to the system
3. Providing sufficient information for rapid identification and rectification of errors

Detailed information on recovery procedures can be found in whichever of the following ICL publications is appropriate:

1900 Series: Using 7900 Standard Operating Systems
VME/B: Using 7900 Standard Operating Systems
1900 Series: 7906 Operation
VME/B: 7906 Operation

5.2 Safeguarding data and programs

5.2.1 The protection system

A protection system that safeguards both programs and data is built into the heart of the 7904 and 7906 operating system software. Details of the protection system are of interest to the user developing his own communications programs. Once a program is fully tested and reliable the protection system checks can be removed from the program, so that the operational communications program can run at maximum efficiency.

Within the protection system of the 7906 running under XOS, each job (a program for instance) operates within a *sphere* which comprises all the resources used by the job. This technique ensures that a job that fails does not affect other jobs, in spite of the fact that some resources may be deliberately shared between jobs; this is significant for the development of complex 7906 systems by the user.

Within a sphere, an activity may be nominated as the *error collector* for the sphere or a number of spheres. User-written routines may be included in the error collector to safeguard data, report errors, and recover from certain error situations as necessary.

5.2.2 Power failure

In the event of a power failure, an early warning system enables the processor to close down in an orderly manner, so as to preserve data and programs in store and to ensure that data stored on disc is not corrupted.

5.3 Maintaining a service

5.3.1 Transient hardware errors

Hardware faults of a transient nature are most commonly associated with communications lines. 7904 and 7906 systems ensure that a faulty communications line does not prevent the use of the other communications lines connected to the system.

Several attempts are made by the 7904 and 7906 software to receive or transmit a message down a poor-quality communications line, in order to overcome transient errors that result in corruption of the message.

This technique frequently overcomes a transient error automatically, and no operator action is required. If Error Manager is being used (see section 5.4.3), then the fact that retransmission on that line was required is notified to the mainframe for subsequent analysis.

If the line error cannot be overcome then the failure is reported at the Executive console, and the operator can initiate line changing procedures.

5.3.2 Changing communications lines

Depending on the facilities included in the network, the operator can change line connections so as to by-pass a component (device or line) which has developed a fault. For instance, leased lines may be switched to stand-by operation at a lower speed; the identifiers (devices) associated with two lines may be exchanged; double dial-up (using two lines) of the Post Office telephone network may be substituted for a high speed link.

5.3.3 Failure of a store link

In the event of a failure in the direct store link between a fast unit (for example, disc) and the store, the peripheral will, in some circumstances, automatically switch to the processor hesitation method (see section 6.3). Processing may then continue, though with diminished throughput.

5.3.4 Use of discs in 7906 systems

In the event of a fault developing in the mainframe or the LPL, user-written software can route incoming data to 7906 disc backing store for transmission when normal operations are resumed. This same method can be used to *store and forward* bulk data messages, that is, to store them during the day and transmit them during the night when PTT charges are lower.

5.3.5 Broadcast

A broadcast command (BC command) is available to the computer operator that allows him to specify a message of up to 70 characters.

This message is sent as the reply to any input message from a terminal that is not on-line to the mainframe.

The message can be changed by the operator as frequently as required, and can give the terminal users information about the current or impending state of the communications service.

5.3.6 Warm restart

Following a dump of the 7900 operating system (see section 5.4.2), the system can be quickly re-loaded from the mainframe and re-started with minimum operator involvement. This method has the advantage that the system can be reloaded and in use again while the dump is analysed, printed and examined for the cause of the failure. In this way comprehensive diagnostic information can be obtained without a long break in the communications service. The warm restart facility is not available for 7900 systems connected to 2900 systems under VME/B.

5.4 Providing error information

Malfunctions are indicated to the system operator by messages at the Executive console, and by indicator lights on the equipment.

Error messages concerning mainframe activities are output at the mainframe.

Full lists of error messages and their meanings are given in the relevant publications. Some of the more significant messages are described below as illustrations of the facilities available.

5.4.1 Types of error message

The state of the mainframe and LPL is indicated at the Executive console by the message DEV LL followed by the station number of the LPL and a description such as: Opened, Closed, Inop (inoperable), Fail, Time out.

Communications multiplexer and Command Chain Unit errors are displayed on the Excutive console in the form DEV MX INOP for the CMX and DEV CL INOP for the CCU. The relevant channel number is also given.

A system message is output to the Executive console whenever the 7900 software stops in a pst-mortem state. The message gives the current program address, the contents of the hardware registers, and a code that indicates the cause of the failure. This information is useful for the initial identification of the type of error.

During the configuration process, any errors in the specification commands are printed out and, whenever possible, attempts are made to recover from error conditions and to continue processing to the end of the run. There is a comprehensive set of configuration error messages covering environment and syntax.

5.4.2 Dumping and analysing the 7900 operating system

In the event of a system software error a message is output to the Executive console giving the type of error and the address where it occurred, and the system stops. A mainframe program is then run which dumps the contents of the 7904 and 7906 store to a mainframe disc file. This can be followed by a mainframe utility program to analyse the dump and print the result on a line printer.

A command is available which allows the 7900 operator to cause the same effect deliberately, that is, to freeze the operating system, dump it and print it. This can be useful during the development stage of an operating system.

For SOS and ND TKEY users the analysed dump is usually sent to ICL as the main part of the diagnostic information required with a software error report.

Dumping to the local disc is available to users of the Program Development Operating System.

5.4.3 Error Manager

A facility called Error Manager is provided for 7904 and 7906 SOS or ND TKEY users with a mainframe operating under GEORGE 3 or GEORGE 4 operating systems.

The Error Manager sends to the mainframe details of the occurrence of hardware or transient line errors, to assist engineers in maintaining the communications equipment. If a data transfer fails or requires one or more retries, then the Error Manager software sends information about the transfer as a record to the mainframe. The transient error record contains the date and time, the CCU address and channel identifier or the CMX address, the retry limit and the number of retries actually performed. The hardware error information contains the date and time, the CCU address and channel identifier or the CMX address and CCU simulator channel identifier, the retry limit, the device type and address, and the Channel Status Register contents. For LPL errors other status register values are included.

The error information can be processed at the mainframe for archiving, analysis and inspection. Faulty hardware and unacceptably noisy lines can be readily identified, easing the task of maintenance and providing evidence to the PTT authorities on the quality of lines.

Detailed information on the use of Error Manager can be found in whichever of the following ICL publications is appropriate:

1900 Series: Using 7900 Standard Operating Systems
1900 Series: 7906 Operation

5.4.4 The Trace facility

Every 7900 system has a TRACE facility available that runs while the communications operating system is in use.

The TRACE facility monitors the actions taken by each type of line driver and the LPL handler, and records information about these actions in areas of 7900 store. These areas of store are nominated by the user of XOS; for SOS users the 2 Kword Engineer Test slot is used for tracing. Each store area is re-used in a cyclic manner, such that the most recent actions taken by each type of driver are continuously recorded.

If required, the TRACE facility can be set to monitor actions associated with a specific communications line, or associated with a specific device.

Information gathered by the TRACE facility is written to the dump file at the mainframe when a post-mortem dump is taken. Information on the last few minutes of the system's operation before a failure is thus readily available in the print of the analysed dump (see section 5.4.2).

This information is of great assistance in the rapid identification and resolution of fault conditions.

5.4.5 Engineer's test programs

A suite of engineering test programs is available to test communications lines and hardware components of the system. These test programs are run in the 7900, and are interactive with the Executive console.

They may be run simultaneously with live use of other lines and hardware.

Tests are available for multiple LPL systems, CMX ports, scanners, and various types of remote terminal.

The tests enable the engineer to identify the part of the system that appears to be at fault, so that the PTT authority or modem supplier can be asked to investigate the fault when necessary.

Note: LPL tests require the link to be quiescent.

5.5 User considerations

5.5.1 Designing resilient systems

Resilience begins at the design stage, by allowing spare capacity in all elements of the system. A weakness in any link in the chain from terminal via lines, controllers and FEP to the mainframe can render part or all of the system inoperative. It is necessary to decide on crisis levels, that is, degrees of urgency in correcting the different types of failure that may occur in the system. For instance, it is necessary to ask whether it is essential to maintain a link to a particular terminal at all times at any cost, or whether a message could be held at the communications processor on a store and forward basis. Other possibilities are that the message could be held over until the following day, or a personal telephone call or telex may be adequate.

If a link must be maintained, it is possible to have at least one spare line to each site, so that the high priority terminals can use the spare line when necessary. For maximum resilience lines from a site should be connected to different communication processors, so that should a communications processor or line become unusable the system operator can switch to another.

If transmission faults are experienced with a leased telephone line then switching to the lower (standby) speed often enables transmission to continue. If the fault is more serious and suitable modems are in use, then the fault can be by-passed by using the public switched network instead.

With suitable modems high speed lines can be switched to double dial-up on the public switched network.

The final decision must be an appropriate balance between built-in redundancy and cost on the one hand, and deterioration of the service on the other.

5.5.2 Sizing

A sizing operation is essential to ensure that the proposed hardware will be able to support the expected traffic rate. The speeds of lines, the sizes of store buffers and the ability of the whole system to handle any expected growth in traffic are important criteria. ICL support services should be called on during this stage of the design process.

5.5.3 Error Manager

Where the mainframe system allows, the network manager should use the facilities provided by Error Manager on a regular basis for reporting and analysing errors. In this way the user can identify incipient hardware faults and correct them before they cause serious disruption of the communications service. For example, a communications line showing an abnormally high level of retransmissions can be reported to the PTT authority for investigation.

6.1 Basic modules

7904 and 7906 systems are supplied as a set of inter-connecting modules, each with its own power pack. The modules available are:

1 Processor

2 Store

3 Local Processor Link (LPL)

4 Communications Multiplexer (CMX), or Command Chain Unit (CCU) plus scanner system(s)

5 Local peripherals

These modules are further described in sections 6.2 to 6.6.

To facilitate the ordering of the correct combination of hardware modules, several standard models of 7904 and 7906 are available. These are described in sections 6.8 and 6.9.

Details of the operation of all 7904 and 7906 hardware units are given in the publication *7904 and 7906 Hardware Operating*.

Processing is performed by a set of modular sub-systems, illustrated in Figure 1.2, operating in conjunction with each other:

1 The *processing sub-system* is a 16-bit word mini-computer supplied with an interval timer. The processor queues and services interrupts from activities in the other sub-systems. The hardware architecture facilitates the use of modular programming which improves throughput for multi-task applications

2 The *memory sub-system* uses store with a cycle time of 800 nanoseconds, supplied on plug-in cards of 8 Kwords each. For hardware reasons store is divided into *books* of up to seven cards (56 Kwords), with a maximum of two books known as book 0 and book 1. A feature of the system is that book 1 store is accessed directly by the other sub-systems without interrupting the processor. Book 0 store may also be accessed in this way. These *direct store links* enable input and output transfers to continue concurrently with communications processing. Store is used for data buffering, accumulating messages and storing line control data

3 The Local Processor Link (LPL) is a hardware module which handles data interchange and code conversion from 24-bit words to 7900 Series 16-bit words and vice versa. When the LPL is connected to a 2900 Series mainframe a 1900 application module must be used for code conversion. The LPL operates sufficiently fast that it imposes no limitation on the throughput of the system. At least one LPL must be present if the 7900 is acting as an FEP

4 The *Communications sub-system* handles code conversion over the mainframe link, and provides interfaces to most PTT circuits and modem types to which standard ICL and compatible terminals can be connected. The sub-system is further described in section 6.5. Two alternative communications modules are available to perform basic line control functions autonomously, synchronising the data flow and transforming bit serial transmissions to byte parallel words in store:

(a) The Communications Multiplexer (CMX) is designed for small communications systems with overall throughput up to 4.1 Kcharacters per second. It is a bit multiplexer which carries a CCU simulator package (see below). It can handle asynchronous lines at speeds of 75 to 1200 bits per second (bps) or synchronous lines at speeds up to 9600 bps. A CMX having 12 ports is suitable only for smaller configurations where a maximum of two different speeds for asynchronous lines can be used. A CMX with 54 ports can have up to three asynchronous line speed variants. The number of lines connected to a port depends on the speed of the line itself

(b) The Command Chain Unit (CCU) is used to control one or more of the following scanner systems:

(i) The Low Speed Scanner (LSS) handling up to 64 asynchronous lines at speeds up to 300 bps. This scanner performs character and block parity checks on received data

(ii) The Medium Speed Scanner (MSS) handling up to 16 synchronous or asynchronous lines at speeds up to 9600 bps for synchronous lines or 1200 bps for asynchronous lines. This scanner performs character parity checks, block checks, cyclic redundancy checks and special character checks on received data

(iii) The High Speed Scanner (HSS) handling up to eight telephone lines designed to link remote procesors. The maximum speed for single channel operation is 300,000 bps, while for all eight channels the maximum speed is 48,000 bps. This scanner checks the received data in the same way as the MSS

(iv) The High Level Scanner (HLS) handling up to 16 telephone lines designed to link remote processors. The maximum speed for single channel operation is 384,000 bps, while for all 16 channels the maximum speed is 48,000 bps. This scanner checks the received data to the High Level Data Link Control (HDLC) format

The HSS and HLS scanners are not available with 7904 systems. HLS scanners require specialised software, and their use should be discussed with the ICL representative.

The CCU controls the scanner by accessing a Command Chain, constructed for the channels by the operating software and held in the 7900 system store, on a fixed priority basis

5 The *peripheral sub-system* handles the local peripherals such as the paper tape reader and the Executive console. In addition to these peripherals, any 7906 with a user written communications program or with the Program Development Operating System may have any of the other local peripherals detailed in section 6.6

6.2 Processing sub-system

At the heart of the *processing sub-system* in all 7904 and 7906 systems is the 7906/00 communications processor. This is a minicomputer specifically designed for multiprogramming, and capable of operating as a stand-alone processor with its own local peripherals.

A paper-tape reader, an Executive console teletypewriter and an interval timer are supplied as standard with the processor.

The processor has 7 ports available for High Speed Interface Connections (HSIC) to store modules and eight to peripherals, as shown in Figure 6.1. Each HSIC connection can address up to 32 Kwords, so that two processor ports must be connected to address a book of store larger than 32 Kwords. In practice there are never more than four store ports connected at the processor unless retained 7905 store modules are being used from a replaced 7905 system, see section 7.2.4. Of the eight peripheral ports, three are reserved for connecting the Executive console, the interval timer and the free-standing paper tape reader. The remaining five are for the Local Processor Link, if required, a Communications Multiplexer (CMX) or Command Chain Unit (CCU) as appropriate, and any of the optional local peripherals described in section 6.6. The Peripheral Multiplexer (PMX) can be used to connect up to 11 additional peripherals (except paper tape reader and discs) to one port, although, with only one channel shared between them, their transfer rates are obviously slower.

6.3 Memory sub-system

In the *memory sub-system* each book of store has a controller with the first 8 Kwords, together with up to six additional increments of 8 Kwords, each supplied as a plug-in card. A book of over 32 Kwords requires an expansion unit to address modules up to the 56 Kwords maximum. Access cycle time for the store is 0.8 microseconds and each controller incorporates an automatic parity check.

7904 systems are supplied with 48 or 56 Kwords of store in a single book, known as Book 0. A further book of up to 56 Kwords, Book 1, can be connected on 7906 systems to give a maximum of 112 Kwords.

The paper tape reader has its associated Read Only Memory (ROM) for bootstrap loading connected to the first port on Book 0.

The processor accesses store locations in book 0 via its own Interface Control Unit (ICU). The fast peripherals (CMX or CCU, LPL and discs) can also use the ICU for store accesses, without involving the processor itself, by means of *store hesitations*. The processor is momentarily halted but its internal registers are not affected.

For book 1 and optionally for book 0 the fast peripherals can access store locations via a *direct store link* (DSL) to a store book controller. Such transfers operate completely independently of the processing unit until a full data block has been transferred. The complete block can then be serviced on a single interrupt to the processor. Direct store links permit partial or complete overlap of operation of the CCU and the processor, so that input/output transfers and other processing are performed in parallel. This removes a significant load from the processor and thus increases overall throughput.

The slow peripherals interrupt the processor itself on a character by character basis.

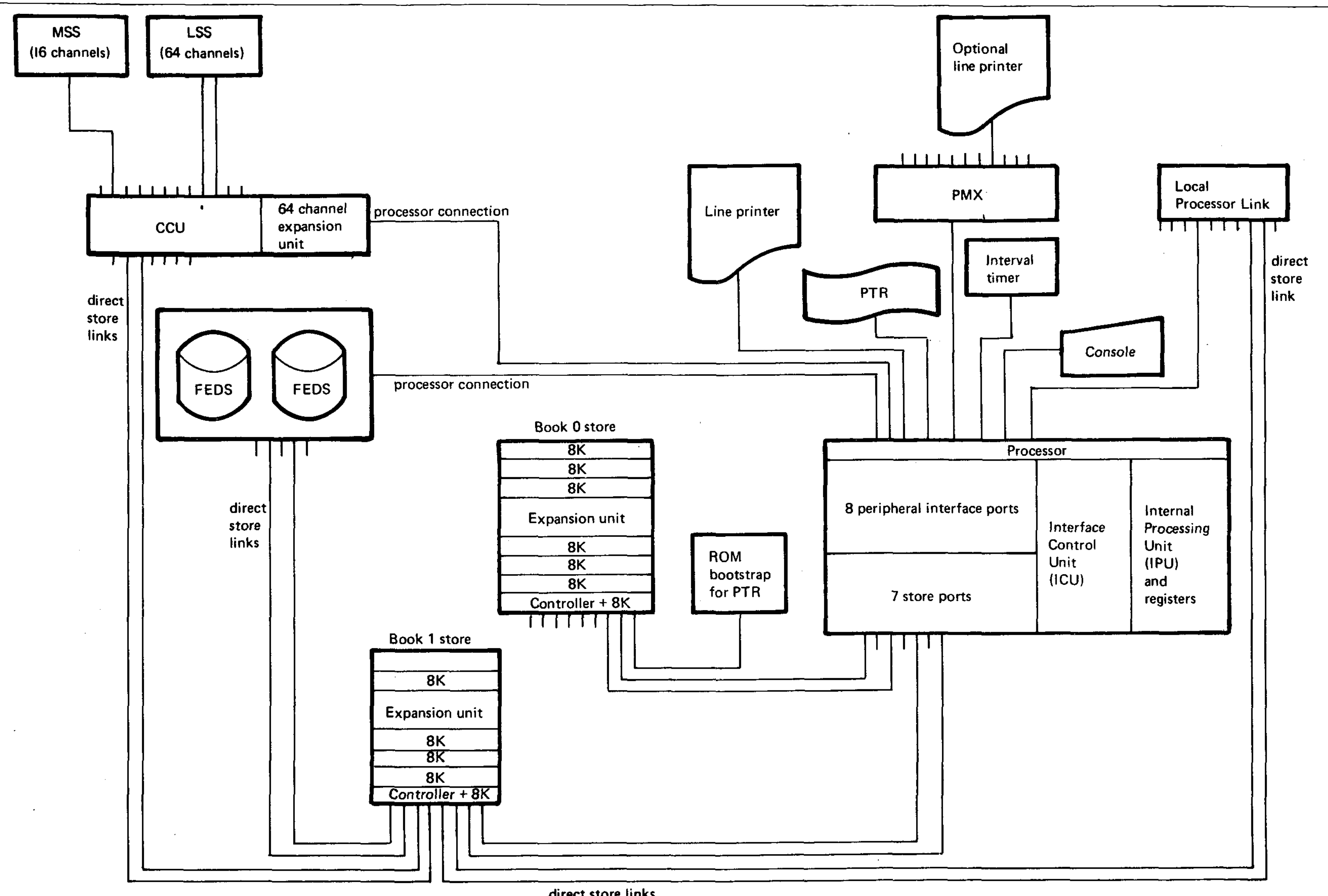

Figure 6.1 Processor and store configuration for a 96K 7906 system

6.4 Mainframe links

7904 and 7906 systems can be connected by the 7987/11 Local Processor Link (LPL) to 1900 Series and 2900 Series systems as indicated below; 7906 systems can also be connected to a mainframe by remote link.

1 *1900 Series mainframes*

(a) Connection as a local front end processor via a 7987/03 LPLN (Local Processor Link Nineteen hundred) to a Standard Interface channel

(b) Remote connection, for 7906 processors only, to a local communications processor (7903, 7904, 7905 or 7906) or local scanner-only system (7920 or 7930) via a communications line at speeds up to 9600 bit/s

2 *2960 processor running under the 1900 Direct Machine Environment (DME/1900)*

(a) Connection as a local front end processor via a 7987/03 LPLN to the 1900 Standard Interface Application Module (AM10), which is in turn connected to a 2900 General Peripheral Controller (GPC)

(b) Remote connection, for 7906 processors only, are as for 1900 Series mainframes

3 *2900 Series processors running under VME/B*

(a) Connection as a local front end processor via a 7987/03 LPLN to AM10 and a General Peripheral Controller (GPC), to interface with VME/B old KOMMFY systems from Release 5X32 onwards

(b) Connection as a local front end processor to VME/B new KOMMFY will be available in a future release of 7900 software

6.4.1 Local Processor Links

A Local Processor Link (LPL) handles the interface between the 7904 or 7906 system and a host mainframe, carrying out code conversion and bit manipulation as it transfers data between the two units.

The Standard Operating System (SOS) on 7904 and 7906 supports one LPL connected to one mainframe at a time.

An XOS system that contains the ICL non-dedicated turnkey (ND TKEY) program, supports the connection and simultaneous operation of up to 4 LPLs.

The LPL itself has three parts:

1 Interface to the 7904 or 7906 processor and memory sub-systems

2 Transfer control module. This central module controls data transfer between the two processor interfaces in the LPL. There are four transfer modes, two for transparent binary (used when loading and dumping) and two for code conversion (1900 internal code to ISO-7 or EBCDIC code). For conversion to 2900 internal code a 1900 application module must be used

3 The mainframe interface. The LPL acts on mainframe commands to control data transers in both directions, and passes interrupts and status change commands to the mainframe software. When two transmissions arrive simultaneously, one in either direction, their priorities are determined by the status of the mainframe interface

The LPL has a nominal transfer rate governed by the attached mainframe, but a maximum average rate would be about 150 Kbps. Data is usually transferred to a 1900 processor in bursts of complete words, but a single character interface can be used if required. Data is transferred to 2900 processors in bytes.

6.5 The communications sub-system

This section gives further details of terminal connections to the two alternative terminal and line control modules described in section 6.1. The Communications Multiplexer (CMX) is supplied with smaller 7904 and 7906 systems. The Command Chain Unit (CCU) is supplied with a Medium Speed Scanner (MSS) for model 7904/23, 7906/23 and 7906/24 systems. Further scanners can be added, to provide for the connection of up to 256 lines.

A 7904 or 7906 system cannot support both a Communications Multiplexer and a Command Chain Unit.

6.5.1 Communications Multiplexer (CMX)

The Communications Multiplexer (CMX) is a bit multiplexer which scans the communications lines to take single bits from them or send single bits along them. Characters are assembled and dis-assembled via a direct store link. The CMX can support telegraph, telephone and directly connected lines.

Basic line control functions, such as synchronising data transmission and transferring characters, are handled autonomously by the CMX. Software overheads cause the CMX to be suitable only for small configurations with a maximum overall throughput of up to 4.1 Kcharacters per second. Only one CMX can be connected to a 7904 or 7906.

The smaller version of the CMX has 12 connection ports in two rows of six. Each row can be set for a different line speed, with a maximum of 9600 bps. The larger version of CMX has 54 ports in three rows of 18, permitting three different speeds to be set. Each port has two port numbers corresponding to channels, as shown in Table 6.1.

The number, type and speed of the possible line connections to these ports are as follows:

Line type	*Port positions*
Asynchronous up to 300 bps	Connected in pairs to each port
Asynchronous up to 600/1200 bps	One line per port
Synchronous up to 2400 bps	One line per port
Synchronous 4800/9600 bps	One line occupies two consecutive ports in a vertical plane; that is, it occupies ports in each of two adjacent rows

PORT

0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34
1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35
36	38	40	42	44	46	48	50	52	54	56	58	60	62	64	66	68	70
37	39	41	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71
72	74	76	78	80	82	84	86	88	90	92	94	96	98	100	102	104	106
73	75	77	79	81	83	85	87	89	91	93	95	97	99	101	103	105	107

4800 and 9600 bps lines	4 port numbers
600 to 2400 bps lines	2 port numbers
100 to 300 bps lines	1 port numbers

Table 6.1 CMX port layout

Lines requiring two channels (one port) must use two consecutive port numbers, where the lowest number is even (for instance, 0 and 1 or 42 and 43 but not 1 and 36 or 0 and 2).

Lines requiring four channels (two ports) must use four port numbers in a vertical plane where the lowest numbered port is even (such as 0, 1, 36, 37 or 58, 59, 94, 95).

Port numbers are important because it is these that the operator uses when giving specific trace or exchange line commands. A line connection which requires more than one port number is referenced by the lowest port number that it uses.

The maximum notional throughput of the CMX is 80 Kbps, and the sum of the transmission speeds of all the lines connected should not exceed this figure.

6.5.2 Command Chain Unit (CCU)

The Command Chain Unit (CCU) is a device that accesses store directly to assemble and dis-assemble messages. It actions chains of commands in store to determine which protocol is used to handle each line on the scanners attached to it (command chaining). The CCU performs polling and message handling autonomously, thus reducing the interrupt load on the processor.

The CCU is supplied with 12 ports for scanners. Maximum notional throughput for the CCU is 150 Kbytes per second.This means that the sum of the transmission speeds of all the lines connected must never exceed this figure, and must sometimes be less, depending on the types of link between CCU and scanners.

The basic CCU controls up to 64 scanner lines; expansion units can be added to raise this to 128 or 256 lines. The number of scanners that can be attached to the CCU depends not on their capacity but on the number of lines actually connected to them. Within these limits the number of lines that are viable for a system depends on their maximum concurrent throughput. Up to two CCUs can be connected.

6.5.3 Scanner sub-systems

Scanner sub-systems are available in four ranges:

1 The Low Speed Scanner (LSS) handles up to 64 half-duplex asynchronous (start-stop) lines operating at speeds of 50,100,110,150,200 or 300 bps. The LSS can support telegraph, telephone, and directly connected lines. The LSS can accept data as 5, 6 7 or 8 bits per character. The facilities of echo printing and automatic answering of Datel 200 lines are provided

2 The Medium Speed Scanner (MSS) provides 16 half-duplex channels for connecting up to 16 direct connection lines, telephone or telegraph lines as follows:

   (a) For asynchronous (start-stop) operation at 50,100,110,150,200, 300,600 and 1200 bps

   (b) For synchronous operation from 600 to,9600 bps using clocked modems. Clock sources can be fitted in the scanner for operation at 600/1200 bps without modem source clocks

   As well as those facilities provided by the LSS, the MSS and can recognise a much larger repertoire of transmission control characters.

3 The High Speed Scanner (HSS) handles connections to remote terminals over telephone circuits operating from 1200 bps to 300 Kbps and using only self-clocking modems. Typically it is used to connect a local 7906 to a remote processor at line speeds over 9600 bps. The HSS supports only synchronous operation; other facilities are as for the MSS. The HSS does not support low speed facilities associated with devices such as teletypewriters, nor does it support automatic answering.

   The HSS provides eight half-duplex channels. The fewer channels that are used, the faster the possible line speeds, as follows:

Number of channels	*Line speeds (bps)*
8	Up to 48K
4	Up to 75K
2	Up to 150K
1	Up to 300K

   The HSS is available only by user programming of the 7906

4 The High Level Scanner (HLS) utilises HDLC line control procedures for the connection of local 7906 processors to remote 7906 processors. The HLS has 16 half duplex channels for use with telephone lines. The fewer channels that are used, the faster the possible line speeds, as follows:

Number of channels	*Line speeds (bps)*
16	Up to 48 K
8	Up to 96 K
4	Up to 192 K
2	Up to 384 K

The scanner logic can detect and classify control characters in 7-bit ISO/USASCII, 8-bit EBCDIC and IBM 6-bit codes, as well as storing up to 16 optional characters loaded from the processor

6.5.4 Line connectors

A 7904 or 7906 system can support a wide variety of communications lines. Telegraph lines, telephone lines, and the direct connction of terminals are supported by a Communications Multiplexer or Command Chain Unit sub-system. A variety of different transmission speeds can be used on the communications lines, and many different types of modem can be used with telephone lines.

To handle the wide variety of communications lines, ICL provides a range of *line connectors* that interface the different types of line to a CMX or the scanner of a CCU sub-system. Line connectors must be ordered separately for each model of 7904 and 7906, and this can only be done when the user has full details of the types of line, their speed, and the types of modem to be used. The user should involve his ICL support representative, who can consult the ICL internal publication *Communications Hardware Guide* for details of the types of modem that can be used, and the line connectors available.

Directly connected lines, and most telephone lines, require two connectors to interface the line to the communications sub-system of the 7900. Telegraph lines require, in addition, ICL model 7922 or 7923 line termination equipment. The user should note, when upgrading his communications sub-system, that a given type of line requires different line connections when moving from a CMX to a CCU sub-system, or from a low speed scanner to another type of scanner.

6.6 The peripheral sub-system

All 7904 and 7906 systems are supplied with an Executive console teletypewriter and a paper tape reader with associated ROM for bootstrap loading. On 7906 systems with user written communications programs or the Program Development Operating System, various local peripherals can also be connected for input and printing, and to provide local disc storage.

6.6.1 Executive console

An ASR teletypewriter is used as the operator's console to issue commands to, and print messages from, the Communications Operating System. It is supplied with its own pedestal and 15 feet of cable to connect it to the processor cabinet; special cable extensions up to 100 feet are available. The teletypewriter prints at 10 characters per second using a roll of paper 8.5 inches (21.6cm) wide.

6.6.2 Paper tape reader

The paper tape reader supplied with every system is a free-standing, table-top unit fitted with a cable up to four metres long, and it is usually placed on top of the processor cabinet. It can read standard one inch 8-track or 11/16 inch 5-track paper tape at 500 characters per second. A paper tape dispenser is also available.

6.6.3 Paper tape punch

The paper tape punch is available on 7906 only, and is fitted into a standard rack module and supplied with a cable up to six metres long. It punches standard one inch 8-track tape at up to 110 characters per second.

6.6.4 User teletypewriter

One or more user teletypewriters can be connected to 7906 systems for program development (see Chapter 4). The teletypewriter itself has the same features as the Executive console (see section 6.6.1); the only difference in operation is that the user teletypewriter can be switched off-line while the operating software is running. A standard cable 15 feet long is supplied with the user teletype, but this may be extended up to 100 feet if required.

6.6.5 Line printer

The line printer is available on 7906 only, and is a floor-standing unit supplied with a cable up to six metres long. There are 80 print positions spaced at 10 characters per inch. Using a 64 character repertoire the printer can output up to 600 lines per minute. Faster speeds are possible with smaller character sets and with line widths shorter than 80 characters.

The print barrel supplied has either the 1900 Series or the 2900 Series code repertoire. A separate 7900 barrel must be ordered for printing post-mortem dumps and for local program development. Both the print barrel and the ribbon cassette are changed by the operator without an engineer being involved.

The form feed mechanism is controlled by a format loop made of one inch 8-track Mylar tape with hole spacing for 6 or 8 lines per inch.

The printer uses standard fanfold, continuous listing paper. Up to four copies can be printed simultaneously.

6.6.6 Discs

The 7980 disc controller is available on 7906 only, and can drive up to four disc transports each accessing one or two discs. FEDS (Fixed and Exchangeable Disc System) transports contain two discs on a common spindle and sharing the same actuator for the read/write heads; the lower disc is permanently fixed to the spindle, while the upper disc is held in a removable cartridge.

Data is transferred in 256 word blocks, although each transfer may involve any number of fixed length blocks in the disc cylinder currently being accessed. Blocks on disc may hold data in binary or character form. The 7900 disc format is not compatible with that on other ICL disc systems.

Maximum storage capacity for one controller with four FEDS transports is 19.2 megabytes. Up to three 7980 disc controllers can be connected to a 7906 system, reduced to two if two Local Processor Links and two Command Chain Units are also connected. Disc characteristics are as follows:

Recording format	512 bytes per block, 12 blocks per track, 200 tracks per surface, 2 surfaces per disc
Disc capacity	2,457,600 bytes
Seek time	12 to 70 milliseconds
Average random seek time	38 milliseconds
Average latency	12.5 milliseconds
Average access time	50.5 milliseconds
Maximum transfer rate	246K bytes per second

6.6.7 Peripheral Multiplexer (PMX)

The Peripheral Multiplexer (PMX) can be used on 7906 only, to connect up to 12 of the following slow peripherals to a single processor channel : user teletypewriter, paper tape punch, line printer. These are all peripherals that interrupt the processor on a character by character basis. Obviously, when sharing a single channel in this way their transfer rates are slower.

The processor's interval timer can also be transferred to the PMX, thereby freeing another processor channel for a fast peripheral.

6.7 Installation information

Installation of 7904 or 7906 hardware generally involves up to four suppliers:

1 ICL, who supply the processor, interfaces to both host mainframes and communications lines, and, usually, the terminals that are connected at the other end of those lines

2 The local Postal, Telegraph and Telephone (PTT) authority who supplies the communications lines to remote terminals, and approves the modems to be used at each end of those lines

3 The modem supplier, who may be the PTT or an independent company

4 The user, who arranges the power supply for the processor and cabling to connect local peripherals

False flooring is recommended for cabling to CCU or LPL equipment, and is required where more than four HSIC links or over 64 communications lines are connected. 1200mm clearance is required around all units for maintenance purposes (1220mm for printers).

The majority of the equipment is designed for use in a controlled office environment with a normal urban atmosphere. Local ICL representatives can advise whether these requirements can be met without special air conditioning. The ICL representative can also consult the ICL internal publication *Communications Hardware Guide*, to provide details of weight, power consumption, and heat dissipation of 7900 units.

Table 6.2 details the dimensions of each of the units in millimetres.

Table 6.2
Equipment dimensions in millimetres

Unit	*Height*	*Width*	*Depth*
Racking for 4 standard modules: processor, up to 2 store books, CMX or disc controller (as fitted)	1443	1154	438
Racking for 2 extra standard modules	1443	592	438
Rack for LPL	1443	700	438
Rack for CCU and scanners	1719	698	749
Rack for disc controller	1111	586	860
Disc transport	1111	586	860
Paper tape reader	273	247	229
Console or user teletypewriter	840	560	480
Line printer	1170	760	1040

6.8 7904 systems

6.8.1 7904/21 Basic Turnkey System

The system comprises:

1 7906/00 communications processor
ASR Executive console teletypewriter
Interval timer
48 Kwords store (Book 0)
7961/01 paper tape reader
F1448 ROM boostrap loader for use with the paper tape reader

2 Racking for 4 standard modules:

Processor, Book 0 store, Communications Multiplexer and one spare rack

3 7987 Local Processor Link (LPL) to 1900 or 2900 Series host mainframe; supplied with its own racking

4 7955/07 Communications Multiplexer (CMX) with 12 ports

Line connections to the CMX are ordered separately.

An extra 8 Kwords can be added to bring Book 0 store to a maximum 56 Kwords.

6.8.2 7904/22 High Connectivity Turnkey System

The system comprises:

1 7906/00 communications processor
ASR executive console teletypewriter
Interval timer
48 Kwords store (Book 0)
7961/01 paper tape reader
F1448 ROM bootstrap loader for use with the paper tape reader

2 Racks for 4 standard modules

Processor, Book 0 store, Communications Multiplexer, and one spare rack

3 7987 Local Processor Link (LPL) to 1900 or 2900 Series host mainframe; supplied with its own racking

4 7955/09 Communications Multiplexer (CMX) with 54 ports

Line connections to the CMX are ordered separately.

An extra 8 Kwords can be added to bring Book 0 store to a maximum of 56 Kwords.

6.8.3 7904/23 High Throughput Turnkey System

The system comprises:

1 7906/00 communications processor
ASR Executive console teletypewriter
Interval timer
48 Kwords store (Book 0)
7961/01 paper tape reader
F1448 ROM bootstrap loader for use with the paper tape reader

2 Racking for 4 standard modules:

Processor, Book 0 store, and two spare racks

3 7987 Local Processor Link (LPL) to 1900 or 2900 Series host mainframe; supplied with its own racking

4 7951/03 Command Chain Unit (CCU) to control 64 scanner lines
7950/02 Medium Speed Scanner (MSS) with 16 channels
Racking for CCU and MSS (with room for another two scanner units)

7950/01 Low Speed Scanners (LSS) and extra 7950/02 MSS can be added up to the CCU limit of 64 connected scanner lines. Line connections to the scanners are ordered separately.

6.9 7906 systems

6.9.1 7906/21 Basic Programmable System

The system comprises:

1 7906/00 communications processor
ASR Executive console teletypewriter
Interval timer
48 Kwords store (Book 0)
7961/01 paper tape reader
F1448 ROM boostrap loader for use with the paper tape reader

2 Racking for 4 standard modules:

Processor, Book 0 store, Communications Multiplexer, and one spare rack

The SOS is configured, patched and assembled on the mainframe by using programs provided by the Mainframe Support Environment. The resultant SOS is loaded as required from the mainframe to the 7904 or 7906, using another program from the Mainframe Support Environment. MSE also provides dump and dump analysis facilities on the mainframe, see Figure 1.3. SOS is further described in Chapter 2.

1.3.3 Extended Operating System (XOS)

XOS is the alternative operating system to SOS for a 7906 communications processor. XOS is designed for a 7906 acting as a sophisticated FEP, or as a remote network node. XOS comprises an operating system base whose generation is specified by the user, plus at least one message buffering application program controlled by the operating system base.

The operating system base can be generated by the user on his 7906, and the message buffering application program can be developed by the user on his 7906. Both user system generation and application program development require the Program Development Environment described in the next section.

There is one use of XOS which does not require user programming. This is when XOS is used on a 7906 acting as a FEP, and employs a message buffering program provided by ICL. This optional message buffering program is called the *non-dedicated turnkey* (NDTKEY), and it provides all the facilities of SOS plus the facilities listed below.

1 Connection of more than one Local Processor Link

2 Connection to up to four mainframes

3 Selection of the devices required in the communications network from the full set of devices declared to the system generation process. The selection is performed by the system operator at the start of each days operation

Any use of XOS on a 7900 used as an FEP requires the Mainframe Support Environment (MSE). This environment provides facilities for patching, assembling, loading, dumping and dump analysis on a mainframe. ICL also provides an optional system generation service, which generates the XOS base to the user's specification, and the optional ND TKEY. These are not part of the MSE.

Figure 1.4 shows how the ICL software described above is used to build an Extended Operating System (XOS).

XOS is further described in Chapter 3.

1.3.4 Program Development Operating System (PDOS)

PDOS is designed for a free-standing model 7906/24, with discs and a line printer, dedicated to user system generation and program development. Communications facilities are not included in this operating system. It is intended for a 7906 used to develop communications programs written in CORAL 66 or 7900 Series Assembly Language (NAL).

PDOS incorporates software to handle the 7906 local peripherals and the operator interface. PDOS is used to run programs provided by the Program Development Environment (PDE). PDE provides the following facilities on a 7906:

1 CORAL 66 compiling system

2 NAL Assembler and link editor

3 Interactive Development Aid and editors

4 Advanced Operating Facilities (AOF)

5 Standard Disc Filing System (SDFS)

6 Utilities for loading, printing, dumping between the 7906 and its local peripherals

7 An operating system generation suite known as SYSGEN

PDE is shown in Figure 1.5. PDOS is further described in Chapter 4.

1.4 User considerations

1.4.1 Assessing requirements

The following considerations need to be borne in mind when formulating the requirements for a communication system:

1. The number of messages per day, or hour, paying close attention to peak conditions

2. The number and type of terminal devices

3. Line speeds and types, for instance multi-drop, dial-up; for a heavy workload it may be advisable to use private wire circuits for greater reliability. Spare lines may be configured to enable swapping of lines when a fault develops on a line

4. The number and size of store buffers required to hold the messages

5. The processor, communications equipment and modems required

It is good practice to design a network map showing remote devices, lines, and line equipment such as modems, connected to the central processing equipment.

ICL strongly urges users to call on its support services for assistance and advice early in the planning stage. It is important that requirements are accurately known, specified and sized, in order to arrive at a configuration which will perform the required tasks and also allow for expansion and modification as future needs arise.

A brief outline of system resilience and recovery considerations is given in Chapter 5.

1.4.2 Transition and growth

The ICL 7904 or 7906 provides a transition route from a 1900 environment to DME/1900, or to VME/B on 2960, 2970, 2972, 2976 or 2980. This approach safeguards the user's investment in the network.

As traffic volume increases and the user's requirements grow, there arises a demand for more terminals. A 7904 may be expanded from 7904/21 to models 22 or 23, to handle the higher number of lines, or the increased message rate.

If special requirements arise, such as spooling (the storage of messages on disc), use of a remote network node, or connection to more than one mainframe, a 7904 may be upgraded to a 7906. The 7906 SOS caters for larger systems, the 7906 XOS caters for more than one parent mainframe and more than one Local Processor Link, while the 7906 with user written communications programs can cater for spooling and remote network nodes.

The 7906 also has three models for use as a communications processor, numbers 21, 22 and 23, to enable the configuration to expand in step with increasing line activity or number of links. There is also a considerable range in the size of communications network that can be supported by each model, because further modules of store and line connection equipment can be added to each model.

The software is also modular and capable of modification to match the user's changing requirements. The Standard Operating System (SOS) is intended for FEPs and can support straightforward networks of considerable size. Changes in the number and type of terminals connected, and in the mix of line protocols or line speeds used in the network, can often be handled by a simple reconfiguration of the SOS by the user.

When the user's communications network becomes more complex the 7906 can be retained by switching to the Extended Operating System (XOS). Because the message handling part of the XOS can be written by the user, additional facilities such as remote connection to several mainframes, spooling, handling of non-standard terminals, or local connection of the 7906 as an FEP to several mainframes, can be provided. The generalised nature of the 7906 hardware and operating system, combined with access to the operating system routines from a user written program, means that many varied facilities can be developed by user programming.

Details of 7904 and 7906 upgrading and transition facilities are given in Chapter 7.

2.1 Introduction

The Standard Operating System (SOS) runs a 7904 or 7906 system as a front end processor to a single mainframe. It is designed for potentially large scale but straightforward communications networks.

The SOS can be run on 7904 systems with up to 56 Kwords of store and on 7906 systems with up to 112 Kwords of store.

The SOS is configured by the user at the mainframe (1900, DME/1900 or 2900 VME/B) using ICL-supplied software. Only the software modules required for the specified network are consolidated into the SOS which is therefore a compact, efficient, dedicated system. The SOS is loaded across The Local Processor Link (LPL) from the mainframe to the FEP.

The remainder of this chapter describes the main features of the SOS. Full details are in the SOS manuals *1900 Series: Using 7900 Standard Operating Systems*, and its equivalent publication *VME/B: Using 7900 Standard Operating Systems*.

2.2 Functions of SOS

The functions of the SOS are:

1 To interface, via Communications Multiplexer or Command Chain Unit and scanners, with most ICL communications devices (see section 2.3.2)

2 To interface with one mainframe via a Local Processor Link (LPL), using code translation appropriate to the parent mainframe

3 To control the Executive console teletypewriter and local paper tape reader

4 To provide software control of line protocol, polling, clocking and interrupt timing

5 To assemble the characters received from the terminals into messages, and to pass on these messages to the parent mainframe

6 To transmit messages received from the mainframe to the destination terminals

7 To provide the TRACE facility to check that all units are functioning normally while the SOS is controlling the communications network

8 To allow the ICL engineer to run a diagnostic program while the SOS is controlling the communictions network

9 To provide an Error Manager facility, in conjunction with a mainframe with GEORGE 3 or GEORGE 4 operating system, to record errors and transmit them to the mainframe for later analysis

2.3 Structure of SOS

Figure 2.1 shows the major components of the SOS. The central portion, the *Executive*, is common to all versions of SOS. The lower area, labelled *service routines*, shows the communication modules from which a selection is made at the configuration stage. The upper area, labelled *application programs*, shows the two operational programs which can be run concurrrently by Executive.

2.3.1 Executive

The major components within the Executive are:

1. Data Transport System
   Updates the hierarchy of controls within the system and passes control to the appropriate driver routine

2. Operator interface
   The system operator issues commands to the operating system at the Executive Console and receives replies and other messages. The part of the Executive that is responsible for this communication is the Operator Interface. The system operator can control the following functions:

   (a) Switching a communications line to its standby speed of transmission, or reverting to normal speed

   (b) Exchanging the terminals on two communications lines, or moving the terminals on a failed line to an alternative line

   (c) Broadcast of a user specified message to a terminal if it inputs a message when the mainframe is not available

   (d) Control of the Trace facility

   (e) Use of the Error Manager facility by George 3 and George 4 users

3. Error Manager
   This module is available only on systems with mainframe operating under GEORGE 3 or GEORGE 4. It reports errors, such as re-tries on a faulty communications line, to the mainframe for subsequent analysis

4. Tracing is carried out by the system, to provide a detailed record of the activities performed by the main software components of SOS, and therefore helps with the diagnosis of errors. The engineers on-line test program slot is used to store the trace information. Tracing is switched off when a test program is loaded

2.3.2 Service routines

The modules in this section are:

1. The LPL Handler which controls the connection to a local mainframe.

   The line control procedures used for LPL connections to 1900 Series and 2900 Series (DME/1900 or VME/B) systems are based on those used to communicate with a Dedicated Control Program (DCP) running a 7903 FEP.

2. The CCU or CMX Handler which contains routines for receiving data from, and transmitting data to, a Command Chain Unit (CCU) or Communications Multiplexer (CMX)

3. Line drivers which service the devices at the ends of the communications links. Different types of device need different contol procedures (protocols). The four types provided, from which a selection is made at the configuration stage, are:

   (a) Asynchronous (TTY) devices:
   teletypewriters, termiprinters, keyboard printers

   (b) Synchronous basic mode bulk (7020 type) devices:
   ICL 7020 and 7503 (using T3Ax) Remote Job Entry (RJE) clusters with bulk peripherals; remote 2903 Range and 1500 Series computers emulating basic mode bulk protocol

   (c) Synchronous basic mode interactive (7181 type) devices:
   Line Sharing Adaptor (TLSA and QLSA) controlling 7181 or 7184 videos; video clusters, with hard-copy printers for direct output, controlled by 7501 (using T2B5), 7502 (using T2Bx) and 7503 (using T3Bx); remote 1500 Series computers emulating basic mode interactive protocol

(d) Extended Basic mode (XBM) synchronous devices:
clusters, with bulk and interactive peripherals controlled by 7501 (using T2C5, T2SAxI, T2AP01, T2ASx) and 7502 (using T2Cx, T2SAxI, T2RBxx, T2AP01, T2IPxx or T2ASx), and 7503 (using T3Cxx or T3D1); remote 2903 Range andd 1500 Series processors emulating XBM protocol

All the above devices can be connected to a 7900 SOS, but the facilities of some of these devices may not be supported by the particular mainframe system to which the 7900 is connected, see section 2.4

2.3.3 Application programs

The ICL dedicated turnkey program is the software element which initiates the service routines as required, and is responsible for routing and buffering messages between terminals and mainframe.

The turnkey program also actions commands input by the console operator to initiate the facilities mention in section 2.3.1.

2.4 Mainframe environments

2.4.1 Mainframe Support Environments

The concept of the Mainframe Support Environment (MSE) is illustrated in Figure 2.2. The MSE represents all the facilities supplied by ICL to support a 7900 operating system; the following aspects of this support for SOS are illustrated:

1 Creation of the SOS by specifying the user's configuration to a Network Configuring Program (NCP), followed by a patching process to incorporate the latest amendments, followed in turn by assembly of the SOS.

2 Loading the complete SOS from mainframe disc file to the 7900

3 In the event of a system failure, the contents of the FEP store can be dumped to the mainframe disc backing store for later analysis. The system can be re-loaded quickly into the FEP, and can therefore be back in use while the dump is being analysed, printed and examined

4 An error log maintained by George 3 or George 4 in conjunction with the 7906 module Error Manager described in Chapter 5. The log can be analysed to produce statistics on the performance of lines and devices so that repeated failures can be high-lighted and corrective action take.

Note that for George 3 and 4 users, and for VME/B users, a set of macros is provided by ICL to run the mainframe utility programs included in the MSE.

The user must employ the version of Mainframe Support Environment appropriate to the type of parent mainframe system in use. The mainframe systems, and the names of the software sets that provide the associated Mainframe Support Environment (MSE), are given in the following table.

Mainframe System	*Name of software set for MSE*
1900 Series or DME/1900 using Multiprogramming Overlaid Executive (MPOE)	MME
1900 Series or DME/1900 using GEORGE 3 or 4	MG3
2900 Series using VME/B old KOMMFY	MVB

The communications facilities of the above mainframe systems that can be used with devices connected through 7900 SOS are given in the following sections.

2.4.2 1900 Series MPOE (Multi-programming Overlaid Executive) and GEORGE 2 systems

1 Under both MPOE and GEORGE 2, when used with Communications Manager, mainframe application programs written to the appropriate standards (but with buffer size restricted to 597 words maximum) can communicate with devices using asynchronous, basic mode interactive or XBM protocols.

Further details are given in *Communications Manager* (Edition 1, TP4420)

2 Under MPOE the MAXIMOP system can be used with Communications Manager to support data input and output and interactive program control from devices using asynchronous, basic mode interactive or XBM protocols. The MAXIMOP text editor may be used, and the same terminals can be used to access the GEORGE 2 RJE system (see below).

Further details of 1900 MAXIMOP are given in *Maximop System* (Edition 1, TP4399) and *Using Maximop* (Edition 1, TP4398)

3 Using GEORGE 2, #XKVB and Communications Manager, Remote Job Entry facilities are supported for 7020, 7503 and 7502 terminal clusters using basic mode bulk or XBM protocols. Job descriptions and data are input off-line, processed using the GEORGE 2 central module and the results printed off-line.

Further details of GEORGE 2 RJE are given in *GEORGE 2 Disc Based Operating* (Edition 1, TP4432)

2.4.3 1900 Series GEORGE 3 or 4 systems with integral MOP and RJE facilities

1 GEORGE 3 or 4 MOP facilities support data input and output, and interactive job and program control from devices using asynchronous basic mode interactive or XBM protocols. The GEORGE 3 or 4 text editor can also be used

2 GEORGE 3 or 4 Remote Job Entry (RJE) facilities support 7020, 7503 and 7502 terminal clusters using basic mode bulk or XBM protocols. Job descriptions and data files are input off-line to filestore, processed under control from the cluster console, and the results printed off-line

3 Mainframe application programs written to the appropriate standards can communicate via Communications Manager with devices using asynchronous, basic mode interactive or XBM protocols. Further details are given in *Communications Manager* (Edition 1, TP4420).

Any SOS running in a processor that is MOPPED OFF can be connected on-line to an application program. Mainframe identifiers known to a SOS may be attached (AH command) to a GEORGE message-buffering conceptual; the conceptual can then be connected on-line (OL command) to an application program.

Note: When GEORGE 3 or 4 requests a configuration report at load-time, a cluster of devices connected to a terminal processor is represented by the SOS as though the devices are multidropped along the line (see figure 2.3).

Further details of GEORGE 3 or 4 commands and facilities are given in *GEORGE 3 and 4 Operation Management* (Edition 6, TP4438) and *Operating Systems GEORGE 3 and 4* (Edition 5, TP4345)

2.4.4 2960 Processors with DME/1900

The facilities provided are as described in sections 2.4.2 and 2.4.3

2.4.5 2900 Series VME/B old KOMMFY systems

1 Under VME/B TP option, mainframe application programs written to the appropriate standards can communicate with interactive devices using asynchronous, basic mode interactive, or XBM protocols

2 Under VME/B the MAC service can be used to support data input and output, and interactive job and program control from interactive devices using asynchronous, basic mode interactive or XBM protocols

3 Under VME/B the RJE service can be used to support Remote Job Entry facilities for 7020 or 7503 systems using basic mode bulk protocol, for 7503 systems using PCT2 protocol, and for 7502 and 7503 systems using XBM protocol. Job descriptions and data files can be input to the mainframe, jobs processed under control from the terminal system console, and results output

Note that support for XBM protocol (and its sub-set PCT2 protocol) is not a standard facility of VME/B old KOMMFY when the lines are connected via a 7900. The user should consult his ICL suppport representative if he wishes to use these protocols.

2.5 User considerations

The preliminary work leading up to the final specification of the network to be handled by the operating system should include, as a minimum, the following steps. These would usually follow a systems analysis and design function.

1 Definition of the terminal devices required

2 Definition of any clusters of devices (that is, using the same communications line to the FEP)

3 In the case of ICL 7500 Series terminal systems, the hardware model and terminal executive in use should be defined

4 Definitions of lines required and their characteristics, such as telegraph or telephone, private wire or dial-up, modems, protocols and speeds

5 Definition of CCU channels or CMX ports

6 A network map should be drawn up. An example is given in Figure 2.3

7 A sizing operation should be performed on the resultant configuration. This should be done in conjunction with ICL support staff. The performance of a communications system is dependent on many factors, including:

   (a) Mainframe hardware, software and loading

   (b) 7904 or 7906 hardware and software

   (c) The number and types of terminal, their message sizes and polling rates

   The importance of sizing therefore, cannot be over-stressed.

8 Bearing in mind possible future requirements of the network, a decision can be taken as to whether the SOS is suited to the task or whether the capabilities of the XOS should be examined. XOS is described in Chapter 3

3.1 Introduction

The Extended Operating System (XOS) is a communications operating system used in front-end or remote 7906 processors. XOS is intended for the customer with a large-scale or complex communications network whose needs cannot be fully satisfied by the Standard Operating System described in Chapter 2. XOS provides precise tailoring of software to the user's requirements.

The XOS has two major divisions:

1 An operating system base, generated by means of the SYSGEN suite of programs, as described in Chapter 4.

2 One or more communications applications programs.

The SYSGEN procedure may be carried out by the user or by ICL. Similarly, the applications programs may be user written, in NAL or CORAL, or the ICL non-dedicated turnkey program ND TKEY may be used. ND TKEY is only applicable to the use of a 7906 as a front-end processor.

There are therefore two methods of producing an XOS which can be summarised as:

1 User-developed, where the user generates the XOS base by means of SYSGEN on a 7906, and develops his application programs, also on a 7906. For this, the Program Development Environment (PDE) described in Chapter 4 is required. The final system is either dumped to the mainframe disc store and re-loaded to the FEP as required, or it is loaded from 7906 disc to store.

   This method of producing an XOS is described further in Sections 3.3.2 and 3.3.3.

2 Specification of the SYSGEN steering file, followed subsequently by the loading of the generated system into the 7906, together with the ICL non-dedicated turnkey program ND TKEY. The communications configuration can then be specified to ND TKEY. This method has the advantage that the user need not concern himself with programming an FEP system

   This method of producing an XOS is described further in Sections 3.3.4 and 3.3.5.

Installing, running and maintaining an XOS are described in the publications *1900 Series: 7906 Operation* and *VME/B: 7906 Operation*.

3.2 Facilities

The facilities provided by an individual XOS depend on which method is used to produce the XOS. The facilities are described in the following two sub-sections.

3.2.1 User-developed XOS facilities

For the user who develops his own communications programs, XOS can provide all that SOS provides, plus the following features:

1 Precise tailoring of the operating system to user requirements at load time

2 User choice of buffer sizes and priorities for different terminals and message types

3 Up to four Local Processor Links can be handled

4 Multi-programming is supported

5 The programs can take full advantage of all the facilities offered by a 7906 processor, such as the use of local peripherals (discs and line printer), or operation as a remote communications processor, by means of the Communications User Interface (CUI). The CUI is described in the publication *7905: Communications User Interface*

6 The use of a timing mechanism known as the Interval Timer

These facilities mean that an extensive range of communications requirements can be satisfied, including: local or remote intelligent concentrator; a network node; spooling of data received during a time of non-availability of the mainframe or a time of peak activity; and routing of messages by type or message content.

3.2.2 Facilities available to the ND TKEY user

The user who elects to use the program ND TKEY has the facilities offered by SOS (see Section 2.3) plus the following features:

1 Connection of more than one Local Processor Link

2 Connection to more than one parent mainframe

3 Selection of devices to be used in the communications network by means of commands issued to ND TKEY at run time, provided these devices have been declared to SYSGEN

ND TKEY is only available for a 7906 used as a front-end processor. The ND TKEY user has no involvement in the complexities of user program development.

3.3 Structure of XOS

3.3.1 Components of XOS

Figure 3.1 shows the major components of XOS. The *Executive* is very similar to the Executive used in Standard Operating Systems, but can also provide a general multi-programming facility. The *service routines* required are selected by the system generation process, and the resultant combination of Executive and service routines is called the *operating system base* or *Communications Operating System* (COS). Note that in addition to the service routines available with SOS, XOS can also support local peripherals, several Local Processor Links, and connection to remote mainframes.

The operating system base can support several *application programs* provided that sufficient store is available in the 7906. If Fixed/Exchangeable Discs are available on the 7906, then the virtual store facility can be used to increase the effective store available to application programs.

The ICL non-dedicated turnkey program ND TKEY can be used, and user-written programs can be used. Note that usually either ND TKEY or user programs are used, but if store size permits both can be used simultaneously.

User-written programs make use of the facilities offered by the operating system base by issuing commands from the user-written program. There is a very wide range of commands available, giving detailed control over the operating system facilities. The commands form a set called the Communications User Interface (CUI), described in the publication *7905: Communications User Interface*.

3.3.2 System generation by the user

The user can generate an XOS on a 7906/24 with a Program Development Environment (PDE), as described in Chapter 4.

Generation is performed by specifying the communications system in a steering file to the SYSGEN suite of programs. By this means the generated system may be tailored specifically to the user requirements with minimum hardware and software redundancy. The steering file defines, for instance, store size, CCU or CMX, buffer sizes, amount of workspace, peripheral devices, communications lines and their characteristics, device drivers and so on. User generation is the most detailed and specific method of generating a 7906 operating system to the user's exact requirements.

3.3.3 User communications programs

As described in Section 3.1 the user may decide that it is necessary to write his own communications programs rather than using the ICL turnkey program. By doing so, the full facilities of the 7906 are made available such as:

1. Connection to remote mainframes and up to four Local Processor Links
2. Complex networks and high data throughput rates can be catered for
3. Commands and messages for the system operator may be devised and incorporated
4. The use of local peripheral devices is possible, particularly the Fixed/Exchangeable Disc Storage (FEDS) device and the Line Printer.
5. The programmer has a choice of multiple buffering arrangements for different terminal and message types
6. Priorities can be allocated to different message types or different remote locations
7. Timing mechanisms are available, through the use of the 7906 interval timer

3.3.4 System generation by ICL

If the user so decides, or a 7906/24 is not available, he can use one of the system generation services offered by ICL. This is the usual method of generating the base operating system to support the ICL non-dedicated turnkey program ND TKEY.

There are two ICL generation services:

1. Service 1 is for the initial hardware configuration, and for new ICL hardware or software that requires regeneration for its support. Service 1 is free to the user
2. Service 2 is for all other purposes and would normally be in response to changed user needs. Service 2 is charged to the user

A steering file is specified by the user in the same manner as for user-generation, and is then passed to ICL for checking and generation of the required base system. The new system is returned to the user tailored as specified, with unwanted modules excluded. If the generated system is to support the ND TKEY program, then only the sub-set of the system generation commands that is applicable to ND TKEY can be specified. The user has more control over the tailoring of the operating system than is available to the SOS user. For example the amount of work space and control block space can be specified.

Details of the use of the ICL generation services are given in the ICL publications *1900 Series: 7906 Operation* and *VME/B: 7906 Operation*.

3.3.5 ND TKEY

Whether the base system is generated by the user or ICL, the user may decide that the non-dedicated turnkey program ND TKEY meets his requirements for using the 7906 as a front end processor. ND TKEY is provided by ICL on paper tape ready to load into the 7906.

ND TKEY interfaces with the base operating system to provide buffering and routing of messages. It supports all the device types and protocols supported by SOS (see Section 2.3.2).

Although ND TKEY does not allow changes to the 7906 configuration as all devices must be declared when the system is generated by SYSGEN, it does allow the system operator to specify which devices are to be used by the system at any time. Commands may also be given to ND TKEY at run time to create files to handle particular devices and streams.

The operator can specify buffers for tracing routines to use for monitoring parts of the operating system.

It is possible to dump the ND TKEY program to the mainframe, after initilisation but before running the program. In this way, subsequent loadings from the mainframe will load ND TKEY in an already-initialised state; the program can also start running automatically after loading if required. This procedure provides a fast *load and go* facility.

Details of ND TKEY are given in the ICL publications *1900 Series: 7906 Operation* and *VME/B: 7906 Operation*.

3.4 Mainframe environments supporting XOS

XOS is supported by 1900 series processors operating under GEORGE 3, GEORGE 4 or MPOE; by 2960 processors under DME/1900; and by 2900 systems under VME/B old KOMMFY. Details are the same as those given for SOS environments in Sections 2.4.2, 2.4.3, 2.4.4 and 2.4.5.

The Mainframe Support Environment (MSE) shown in Figure 3.2 is necessary for users of ND TKEY, but is optional for user-developed XOS. There are three versions of the MSE to match the three types of mainframe system quoted above. The versions are the same as those required by SOS users, see Section 2.4.1.

Figure 3.2 shows the following features of the MSE supporting an XOS employing ND TKEY:

1 Specification of the system configuration to the ICL SYSGEN Service, followed by patch header generation, patching and assembling of the operating system base returned to the user

2 Loading of the generated operating system base and ND TKEY into the 7906 to form XOS

3 Error logging and dumping are as described for the SOS in Section 2.4

3.5 User considerations

Following the preliminary analysis described in Section 2.5, the following decisions are necessary in order to achieve an extended operating system:

1 The Standard Operating System does not satisfy all the requirements, and therefore an XOS is required

2 Does ND TKEY provide all the communications facilities required? If not, specify, write and test in-house programs. (This entails the use of a 7906/24 for compilation and testing)

3 Is a 7906/24 available on which SYSGEN can be run? If not, use the ICL SYSGEN service

The next step is the preparation of the SYSGEN steering file that defines the requirements of the operating system base. A brief description of SYSGEN, with references to other publications, is given in Section 4.3.4. The operating system base is then generated.

Finally, the generated system, plus ND TKEY or the in-house programs, are loaded, any necessary commands are input, and the system is then ready to run.

4.1 Introduction

The Program Development Environment (PDE) is used with a 7906/24 system to develop Extended Operating Systems (XOS) in which the communications programs are written by the user. As described in Chapter 3, an XOS comprises one or more communications programs run under a communications operating system (COS) whose facilities must match those required by the communications programs. The PDE therefore also enables a user to generate his own versions of the communications operating system, by using the model 7906/24.

An operating system developed by PDE gives the user full control over all the communications operating system functions supplied by ICL, by means of the user-written communications program. This means that the user can provide all the benefits and facilities of XOS systems described in Chapter 3.

The PDE comprises software available from ICL, which is run on a 7906/24 as a free standing computer system; no connection to a mainframe is necessary. An XOS can be developed, by using PDE, to control any model of 7906, acting either as a front end processor or as a remote communications processor.

The PDE software is run on the 7906/24 under the control of an operating system called the Program Development Operating System (PDOS). This operating system is not a communications operating system like SOS or XOS; it is used to control the use of a free standing 7906/24 for program development. A basic version of PDOS is provided as part of PDE.

4.2 Facilities of PDE

The way in which the Program Development Environment is used to create an Extended Operating System (XOS) is shown in Figure 4.1. PDE provides facilities in three main areas:

1 An operating system (PDOS) and associated software to support the efficient use of all software within PDE. The facilities provided in this area are:

  (a) Control of the operation of the system through the Executive console

  (b) Use of all local peripherals available on a 7906 system

  (c) Support of local user teletypes for program development work

  (d) Control of batch and interactive jobs by a job control language provided by the Advanced Operating Facilities (AOF)

  (e) Creation and control of disc files by the Standard Disc Filing System (SDFS)

2 A suite of software for developing communications programs written by the user. The facilities provided in this area are:

  (a) Assembly of programs written in the 7900 assembly language called NAL

  (b) Compilation of programs written in the high-level language CORAL 66

  (c) Program editing, linking and loading

  (d) Testing and debugging programs

  (e) Dumping and printing programs

  (f) Data management, such as disc volume and file managers, disc and file copying, and user Dictionary control

3 A suite of programs called SYSGEN for generating an extended operating system base from the full set of software modules provided by ICL. The facilities provided in this area are:

(a) Selection only of those modules required to support the user's particular hardware configuration

(b) Incorporation only of the operating system facilities required by the communications programs written by the user

(c) Validation of the steering file created by the user for input to the SYSGEN process

(d) Generation of the operating system base for any XOS system

(e) Generation of a version of PDOS specifically tailored to the user's program development requirements. The PDOS generated by the user can replace the basic version of PDOS issued by ICL for running the Program Development Environment

(f) Programs for tidying, clearing, and copying files

4.3 Using the PDE

The major software elements of the Program Development Environment (PDE) are shown diagrammatically in Figure 4.2. The individual software items are described in the following sections.

4.3.1 PDOS

PDOS is the initial operating system provided by ICL, enabling a user to undertake general maintenance work, program development work, and system generation. PDOS supports the Standard Disc Filing System (SDFS) and Advanced Operating Facilities (AOF), but no communications software. PDOS runs on, and supports, the following configuration:

56 Kwords of store
Interval Timer
2 Fixed/Exchangeable Discs (4 x 2.5 = 10 Mbyte)
Line printer
Executive console
User teletypewriter locally connected
Paper tape reader

PDOS has a multiprogramming capability, so if store size permits several PDE programs can be used concurrently.

In order to make use of extra hardware, for example more store, a second printer, or the paper tape punch, the user should generate an appropriate program development operating system, using the system generation suite (see section 4.3.8)

4.3.2 Advanced Operating Facilities (AOF)

The Advanced Operating Facilities (AOF) provide the following features:

1 A simple but comprehensive job control language (JCL), for interactive use at user teletypewriters or for input to a batch job control file. A monitor file is automatically printed for all jobs

2 Multi-access to shared system resources. The AOF organises the use of hardware by scheduling concurrent jobs for optimum throughput. Input and output files are spooled to disc, priority being given to interactive work

3 Commands for creating, amending, deleting, copying and printing disc files and records, in conjunction with the Standard Disc Filing System (SDFS) and associated disc utilities

4 Global and local macro commands which call macro definitions stored on disc

5 Setting-up sub-systems to run programs that are frequently used, such as the compiler, the editor or other utilities

6 An editor (EDITAOF) for modification of program and text files

4.3.3 Standard Disc Filing System (SDFS)

The Standard Disc Filing System (SDFS) is a disc file management system. It executes disc commands issued by 7900 programs, for example, a command to read a record. Facilities are provided to create, amend, delete and copy disc files and records, and to enable many users to share discs.

4.3.4 NAL (New Assembly Language)

NAL is a flexible block-structured assembly language that is particularly suited for programming the 7906. A source program may be developed as a series of modules to be assembled separately and subsequently combined by a link editor program.

NAL is described in the following ICL publications:

LIMPID Link Editor
7905: Introduction to NAL Programming
NAL (New Assembly Language)
NALA (E4) User

4.3.5 CORAL 66

CORAL 66 is a general purpose high-level langauge designed specifically for real time applications. CORAL 66 is block-structured, and fast, compact object programs can be produced using this language.

A text-replacement macro processor is included in the compiler, allowing the incorporation of NAL statements in the CORAL source program.

CORAL 66 is described in the following ICL publications:

7905 CORAL 66 Programming
7905: CORAL 66 Language
7905: CORAL 66 User

4.3.6 IDA (Interactive Development Aid)

An interactive development aid (IDA) is provided, to permit the development and debugging of programs in interactive mode. IDA enables snapshots of current activity to be initiated within the program under development, and also enables commands to be issued from the user teletype to control program areas and current activities.

IDA is described in the ICL publication

7905: IDA - Interactive Development Aid

4.3.7 Utility programs

Several utility programs are supplied to aid the user; they may be run interactively or in batch mode and fall into the following broad categories:

1 Assemblers for NAL and a link editor (LIMPID)

2 Compiler for CORAL, and CORAL formatter

3 Debugging aids such as IDA, post-mortem dump and post-mortem print

4 Editors for disc data (DEDIT), system source (MEDIT), and source text (EDITAOF)

5 Loaders such as the initial bootstrap program

6 Copying programs for discs, dictionaries, disc partitions, and files

7 Disc organisation programs such as volume manager, initialisation of partitions, dictionary manager and file manager

4.3.8 System generation (SYSGEN)

The user system generation process consists of running a suite of programs (the SYSGEN suite) on a 7906/24 system. The process selects and configures modules from master files of source modules supplied by ICL, and produces a loadable, binary operating system base. The operating system base generated by SYSGEN is held as a 7906 disc file, and can be one of several types:

1 An alternative program development operating system (PDOS) to the basic PDOS issued by ICL

2 An operating system base to be used with the ICL program ND TKEY to form an extended operating system (XOS)

3 An operating system base to be used with one or more user-written communications programs to form an XOS

4 A combination of any of the above types

The modules to be selected and configured by SYSGEN are specified by commands in a steering file created by the user. The main stages of SYSGEN are:

1 Validation of the user's steering file and insertion of default values into steering file commands (program SYSVAL)

2 Selection of source code modules and control block modules from the COS master files supplied by ICL (program SYSCODE)

3 Macros in the selected control block modules are expanded to pure code by program ML/1, and program SHUFL is used to set up data structures within the control blocks

4 The code and control block modules are assembled into binary modules by a variant of the NAL assembler (NALG)

5 The binary modules are linked together by the link editor LIMPID

6 The resultant binary operating system is written to 7900 disc by program CDLOAD

An overview of the SYSGEN process is given in *7905 Introduction to the Operating System*; guidance on how to plan a system generation is also given. The definition of all steering file commands available is given in the current version of a computer listing titled *COS Options and Device Macros*.

As an alternative to system generation by the user, ICL offers two system generation services. These are described in Chapter 3, and full details of their use are given in *1900 Series: 7906 Operation* and *VME/B: 7906 Operation*.

4.4 User considerations

PDE allows the user to develop versions of the extended operating system which are tailored precisely to his own requirements, and to provide facilities which are not available with the ICL turnkey programs.

This is achieved by the user designing, writing and testing his communications programs, which use the operating system functions by issuing commands from a set of commands called the Communications User Interface.

The work involved in developing a communications program should not be underestimated. The requirements of the communications system must be established by detailed systems analysis, with help and advice from ICL support staff. If, as a result of the systems analysis, the user decides to proceed with program development, then the following major activities are involved:

1 Systems design, which requires a thorough understanding of the Communications User Interface and the structure of the ICL communications operating system (COS)

2 Systems programming by staff experienced in 7906 systems, and in CORAL 66 and NAL programming languages

3 Management of a model 7906/24 system to the standard expected for any small computer installation

4 Initialisation of the Program Development Operating System and its associated sub-systems such as Advanced Operating Facilities and Standard Disc Filing System

5 Operating the model 7906/24 during program development as a free standing computer system

6 Generation of base operating systems by using the SYSGEN suite of programs

7 Testing and maintenance of a real time system

A user who is not developing his own communications programs may still use the program development environment for system generation. This allows the user to generate his own operating system base to support the ICL non-dedicated turnkey program ND TKEY.

5.1 Introduction

Communications systems help the terminal users to do their jobs more effectively, and thereby increase the efficiency or competitiveness of their company. Any prolonged failure of the communications system may therefore have a serious effect on the day to day operation of the company. ICL recognises this situation, and strives to provide hardware and software communications products of high reliability. However even in the most reliable systems a failure may occur, so facilities must exist to minimise the effect of the fault and to correct it as soon as possible.

This chapter is concerned with the facilities provided in 7904 and 7906 systems for:

1. Safeguarding data and programs in the event of failure of one of the elements of the communications system
2. Maintaining a service to the terminal users and minimising disruption to the system
3. Providing sufficient information for rapid identification and rectification of errors

Detailed information on recovery procedures can be found in whichever of the following ICL publications is appropriate:

1900 Series: Using 7900 Standard Operating Systems
VME/B: Using 7900 Standard Operating Systems
1900 Series: 7906 Operation
VME/B: 7906 Operation

5.2 Safeguarding data and programs

5.2.1 The protection system

A protection system that safeguards both programs and data is built into the heart of the 7904 and 7906 operating system software. Details of the protection system are of interest to the user developing his own communications programs. Once a program is fully tested and reliable the protection system checks can be removed from the program, so that the operational communications program can run at maximum efficiency.

Within the protection system of the 7906 running under XOS, each job (a program for instance) operates within a *sphere* which comprises all the resources used by the job. This technique ensures that a job that fails does not affect other jobs, in spite of the fact that some resources may be deliberately shared between jobs; this is significant for the development of complex 7906 systems by the user.

Within a sphere, an activity may be nominated as the *error collector* for the sphere or a number of spheres. User-written routines may be included in the error collector to safeguard data, report errors, and recover from certain error situations as necessary.

5.2.2 Power failure

In the event of a power failure, an early warning system enables the processor to close down in an orderly manner, so as to preserve data and programs in store and to ensure that data stored on disc is not corrupted.

5.3 Maintaining a service

5.3.1 Transient hardware errors

Hardware faults of a transient nature are most commonly associated with communications lines. 7904 and 7906 systems ensure that a faulty communications line does not prevent the use of the other communications lines connected to the system.

Several attempts are made by the 7904 and 7906 software to receive or transmit a message down a poor-quality communications line, in order to overcome transient errors that result in corruption of the message.

This technique frequently overcomes a transient error automatically, and no operator action is required. If Error Manager is being used (see section 5.4.3), then the fact that retransmission on that line was required is notified to the mainframe for subsequent analysis.

If the line error cannot be overcome then the failure is reported at the Executive console, and the operator can initiate line changing procedures.

5.3.2 Changing communications lines

Depending on the facilities included in the network, the operator can change line connections so as to by-pass a component (device or line) which has developed a fault. For instance, leased lines may be switched to stand-by operation at a lower speed; the identifiers (devices) associated with two lines may be exchanged; double dial-up (using two lines) of the Post Office telephone network may be substituted for a high speed link.

5.3.3 Failure of a store link

In the event of a failure in the direct store link between a fast unit (for example, disc) and the store, the peripheral will, in some circumstances, automatically switch to the processor hesitation method (see section 6.3). Processing may then continue, though with diminished throughput.

5.3.4 Use of discs in 7906 systems

In the event of a fault developing in the mainframe or the LPL, user-written software can route incoming data to 7906 disc backing store for transmission when normal operations are resumed. This same method can be used to *store and forward* bulk data messages, that is, to store them during the day and transmit them during the night when PTT charges are lower.

5.3.5 Broadcast

A broadcast command (BC command) is available to the computer operator that allows him to specify a message of up to 70 characters.

This message is sent as the reply to any input message from a terminal that is not on-line to the mainframe.

The message can be changed by the operator as frequently as required, and can give the terminal users information about the current or impending state of the communications service.

5.3.6 Warm restart

Following a dump of the 7900 operating system (see section 5.4.2), the system can be quickly re-loaded from the mainframe and re-started with minimum operator involvement. This method has the advantage that the system can be reloaded and in use again while the dump is analysed, printed and examined for the cause of the failure. In this way comprehensive diagnostic information can be obtained without a long break in the communications service. The warm restart facility is not available for 7900 systems connected to 2900 systems under VME/B.

5.4 Providing error information

Malfunctions are indicated to the system operator by messages at the Executive console, and by indicator lights on the equipment.

Error messages concerning mainframe activities are output at the mainframe.

Full lists of error messages and their meanings are given in the relevant publications. Some of the more significant messages are described below as illustrations of the facilities available.

5.4.1 Types of error message

The state of the mainframe and LPL is indicated at the Executive console by the message DEV LL followed by the station number of the LPL and a description such as: Opened, Closed, Inop (inoperable), Fail, Time out.

Communications multiplexer and Command Chain Unit errors are displayed on the Excutive console in the form DEV MX INOP for the CMX and DEV CL INOP for the CCU. The relevant channel number is also given.

A system message is output to the Executive console whenever the 7900 software stops in a pst-mortem state. The message gives the current program address, the contents of the hardware registers, and a code that indicates the cause of the failure. This information is useful for the initial identification of the type of error.

During the configuration process, any errors in the specification commands are printed out and, whenever possible, attempts are made to recover from error conditions and to continue processing to the end of the run. There is a comprehensive set of configuration error messages covering environment and syntax.

5.4.2 Dumping and analysing the 7900 operating system

In the event of a system software error a message is output to the Executive console giving the type of error and the address where it occurred, and the system stops. A mainframe program is then run which dumps the contents of the 7904 and 7906 store to a mainframe disc file. This can be followed by a mainframe utility program to analyse the dump and print the result on a line printer.

A command is available which allows the 7900 operator to cause the same effect deliberately, that is, to freeze the operating system, dump it and print it. This can be useful during the development stage of an operating system.

For SOS and ND TKEY users the analysed dump is usually sent to ICL as the main part of the diagnostic information required with a software error report.

Dumping to the local disc is available to users of the Program Development Operating System.

5.4.3 Error Manager

A facility called Error Manager is provided for 7904 and 7906 SOS or ND TKEY users with a mainframe operating under GEORGE 3 or GEORGE 4 operating systems.

The Error Manager sends to the mainframe details of the occurrence of hardware or transient line errors, to assist engineers in maintaining the communications equipment. If a data transfer fails or requires one or more retries, then the Error Manager software sends information about the transfer as a record to the mainframe. The transient error record contains the date and time, the CCU address and channel identifier or the CMX address, the retry limit and the number of retries actually performed. The hardware error information contains the date and time, the CCU address and channel identifier or the CMX address and CCU simulator channel identifier, the retry limit, the device type and address, and the Channel Status Register contents. For LPL errors other status register values are included.

The error information can be processed at the mainframe for archiving, analysis and inspection. Faulty hardware and unacceptably noisy lines can be readily identified, easing the task of maintenance and providing evidence to the PTT authorities on the quality of lines.

Detailed information on the use of Error Manager can be found in whichever of the following ICL publications is appropriate:

1900 Series: Using 7900 Standard Operating Systems
1900 Series: 7906 Operation

5.4.4 The Trace facility

Every 7900 system has a TRACE facility available that runs while the communications operating system is in use.

The TRACE facility monitors the actions taken by each type of line driver and the LPL handler, and records information about these actions in areas of 7900 store. These areas of store are nominated by the user of XOS; for SOS users the 2 Kword Engineer Test slot is used for tracing. Each store area is re-used in a cyclic manner, such that the most recent actions taken by each type of driver are continuously recorded.

If required, the TRACE facility can be set to monitor actions associated with a specific communications line, or associated with a specific device.

Information gathered by the TRACE facility is written to the dump file at the mainframe when a post-mortem dump is taken. Information on the last few minutes of the system's operation before a failure is thus readily available in the print of the analysed dump (see section 5.4.2).

This information is of great assistance in the rapid identification and resolution of fault conditions.

5.4.5 Engineer's test programs

A suite of engineering test programs is available to test communications lines and hardware components of the system. These test programs are run in the 7900, and are interactive with the Executive console.

They may be run simultaneously with live use of other lines and hardware.

Tests are available for multiple LPL systems, CMX ports, scanners, and various types of remote terminal.

The tests enable the engineer to identify the part of the system that appears to be at fault, so that the PTT authority or modem supplier can be asked to investigate the fault when necessary.

Note: LPL tests require the link to be quiescent.

5.5 User considerations

5.5.1 Designing resilient systems

Resilience begins at the design stage, by allowing spare capacity in all elements of the system. A weakness in any link in the chain from terminal via lines, controllers and FEP to the mainframe can render part or all of the system inoperative. It is necessary to decide on crisis levels, that is, degrees of urgency in correcting the different types of failure that may occur in the system. For instance, it is necessary to ask whether it is essential to maintain a link to a particular terminal at all times at any cost, or whether a message could be held at the communications processor on a store and forward basis. Other possibilities are that the message could be held over until the following day, or a personal telephone call or telex may be adequate.

If a link must be maintained, it is possible to have at least one spare line to each site, so that the high priority terminals can use the spare line when necessary. For maximum resilience lines from a site should be connected to different communication processors, so that should a communications processor or line become unusable the system operator can switch to another.

If transmission faults are experienced with a leased telephone line then switching to the lower (standby) speed often enables transmission to continue. If the fault is more serious and suitable modems are in use, then the fault can be by-passed by using the public switched network instead.

With suitable modems high speed lines can be switched to double dial-up on the public switched network.

The final decision must be an appropriate balance between built-in redundancy and cost on the one hand, and deterioration of the service on the other.

5.5.2 Sizing

A sizing operation is essential to ensure that the proposed hardware will be able to support the expected traffic rate. The speeds of lines, the sizes of store buffers and the ability of the whole system to handle any expected growth in traffic are important criteria. ICL support services should be called on during this stage of the design process.

5.5.3 Error Manager

Where the mainframe system allows, the network manager should use the facilities provided by Error Manager on a regular basis for reporting and analysing errors. In this way the user can identify incipient hardware faults and correct them before they cause serious disruption of the communications service. For example, a communications line showing an abnormally high level of retransmissions can be reported to the PTT authority for investigation.

6.1 Basic modules

7904 and 7906 systems are supplied as a set of inter-connecting modules, each with its own power pack. The modules available are:

1 Processor

2 Store

3 Local Processor Link (LPL)

4 Communications Multiplexer (CMX), or Command Chain Unit (CCU) plus scanner system(s)

5 Local peripherals

These modules are further described in sections 6.2 to 6.6.

To facilitate the ordering of the correct combination of hardware modules, several standard models of 7904 and 7906 are available. These are described in sections 6.8 and 6.9.

Details of the operation of all 7904 and 7906 hardware units are given in the publication *7904 and 7906 Hardware Operating*.

Processing is performed by a set of modular sub-systems, illustrated in Figure 1.2, operating in conjunction with each other:

1 The *processing sub-system* is a 16-bit word mini-computer supplied with an interval timer. The processor queues and services interrupts from activities in the other sub-systems. The hardware architecture facilitates the use of modular programming which improves throughput for multi-task applications

2 The *memory sub-system* uses store with a cycle time of 800 nanoseconds, supplied on plug-in cards of 8 Kwords each. For hardware reasons store is divided into *books* of up to seven cards (56 Kwords), with a maximum of two books known as book 0 and book 1. A feature of the system is that book 1 store is accessed directly by the other sub-systems without interrupting the processor. Book 0 store may also be accessed in this way. These *direct store links* enable input and output transfers to continue concurrently with communications processing. Store is used for data buffering, accumulating messages and storing line control data

3 The Local Processor Link (LPL) is a hardware module which handles data interchange and code conversion from 24-bit words to 7900 Series 16-bit words and vice versa. When the LPL is connected to a 2900 Series mainframe a 1900 application module must be used for code conversion. The LPL operates sufficiently fast that it imposes no limitation on the throughput of the system. At least one LPL must be present if the 7900 is acting as an FEP

4 The *Communications sub-system* handles code conversion over the mainframe link, and provides interfaces to most PTT circuits and modem types to which standard ICL and compatible terminals can be connected. The sub-system is further described in section 6.5. Two alternative communications modules are available to perform basic line control functions autonomously, synchronising the data flow and transforming bit serial transmissions to byte parallel words in store:

(a) The Communications Multiplexer (CMX) is designed for small communications systems with overall throughput up to 4.1 Kcharacters per second. It is a bit multiplexer which carries a CCU simulator package (see below). It can handle asynchronous lines at speeds of 75 to 1200 bits per second (bps) or synchronous lines at speeds up to 9600 bps. A CMX having 12 ports is suitable only for smaller configurations where a maximum of two different speeds for asynchronous lines can be used. A CMX with 54 ports can have up to three asynchronous line speed variants. The number of lines connected to a port depends on the speed of the line itself

(b) The Command Chain Unit (CCU) is used to control one or more of the following scanner systems:

(i) The Low Speed Scanner (LSS) handling up to 64 asynchronous lines at speeds up to 300 bps. This scanner performs character and block parity checks on received data

(ii) The Medium Speed Scanner (MSS) handling up to 16 synchronous or asynchronous lines at speeds up to 9600 bps for synchronous lines or 1200 bps for asynchronous lines. This scanner performs character parity checks, block checks, cyclic redundancy checks and special character checks on received data

(iii) The High Speed Scanner (HSS) handling up to eight telephone lines designed to link remote procesors. The maximum speed for single channel operation is 300,000 bps, while for all eight channels the maximum speed is 48,000 bps. This scanner checks the received data in the same way as the MSS

(iv) The High Level Scanner (HLS) handling up to 16 telephone lines designed to link remote processors. The maximum speed for single channel operation is 384,000 bps, while for all 16 channels the maximum speed is 48,000 bps. This scanner checks the received data to the High Level Data Link Control (HDLC) format

The HSS and HLS scanners are not available with 7904 systems. HLS scanners require specialised software, and their use should be discussed with the ICL representative.

The CCU controls the scanner by accessing a Command Chain, constructed for the channels by the operating software and held in the 7900 system store, on a fixed priority basis

5 The *peripheral sub-system* handles the local peripherals such as the paper tape reader and the Executive console. In addition to these peripherals, any 7906 with a user written communications program or with the Program Development Operating System may have any of the other local peripherals detailed in section 6.6

6.2 Processing sub-system

At the heart of the *processing sub-system* in all 7904 and 7906 systems is the 7906/00 communications processor. This is a minicomputer specifically designed for multiprogramming, and capable of operating as a stand-alone processor with its own local peripherals.

A paper-tape reader, an Executive console teletypewriter and an interval timer are supplied as standard with the processor.

The processor has 7 ports available for High Speed Interface Connections (HSIC) to store modules and eight to peripherals, as shown in Figure 6.1. Each HSIC connection can address up to 32 Kwords, so that two processor ports must be connected to address a book of store larger than 32 Kwords. In practice there are never more than four store ports connected at the processor unless retained 7905 store modules are being used from a replaced 7905 system, see section 7.2.4. Of the eight peripheral ports, three are reserved for connecting the Executive console, the interval timer and the free-standing paper tape reader. The remaining five are for the Local Processor Link, if required, a Communications Multiplexer (CMX) or Command Chain Unit (CCU) as appropriate, and any of the optional local peripherals described in section 6.6. The Peripheral Multiplexer (PMX) can be used to connect up to 11 additional peripherals (except paper tape reader and discs) to one port, although, with only one channel shared between them, their transfer rates are obviously slower.

6.3 Memory sub-system

In the *memory sub-system* each book of store has a controller with the first 8 Kwords, together with up to six additional increments of 8 Kwords, each supplied as a plug-in card. A book of over 32 Kwords requires an expansion unit to address modules up to the 56 Kwords maximum. Access cycle time for the store is 0.8 microseconds and each controller incorporates an automatic parity check.

7904 systems are supplied with 48 or 56 Kwords of store in a single book, known as Book 0. A further book of up to 56 Kwords, Book 1, can be connected on 7906 systems to give a maximum of 112 Kwords.

The paper tape reader has its associated Read Only Memory (ROM) for bootstrap loading connected to the first port on Book 0.

6.4 Mainframe links

7904 and 7906 systems can be connected by the 7987/11 Local Processor Link (LPL) to 1900 Series and 2900 Series systems as indicated below; 7906 systems can also be connected to a mainframe by remote link.

1 *1900 Series mainframes*

(a) Connection as a local front end processor via a 7987/03 LPLN (Local Processor Link Nineteen hundred) to a Standard Interface channel

(b) Remote connection, for 7906 processors only, to a local communications processor (7903, 7904, 7905 or 7906) or local scanner-only system (7920 or 7930) via a communications line at speeds up to 9600 bit/s

2 *2960 processor running under the 1900 Direct Machine Environment (DME/1900)*

(a) Connection as a local front end processor via a 7987/03 LPLN to the 1900 Standard Interface Application Module (AM10), which is in turn connected to a 2900 General Peripheral Controller (GPC)

(b) Remote connection, for 7906 processors only, are as for 1900 Series mainframes

3 *2900 Series processors running under VME/B*

(a) Connection as a local front end processor via a 7987/03 LPLN to AM10 and a General Peripheral Controller (GPC), to interface with VME/B old KOMMFY systems from Release 5X32 onwards

(b) Connection as a local front end processor to VME/B new KOMMFY will be available in a future release of 7900 software

6.4.1 Local Processor Links

A Local Processor Link (LPL) handles the interface between the 7904 or 7906 system and a host mainframe, carrying out code conversion and bit manipulation as it transfers data between the two units.

The Standard Operating System (SOS) on 7904 and 7906 supports one LPL connected to one mainframe at a time.

An XOS system that contains the ICL non-dedicated turnkey (ND TKEY) program, supports the connection and simultaneous operation of up to 4 LPLs.

The LPL itself has three parts:

1 Interface to the 7904 or 7906 processor and memory sub-systems

2 Transfer control module. This central module controls data transfer between the two processor interfaces in the LPL. There are four transfer modes, two for transparent binary (used when loading and dumping) and two for code conversion (1900 internal code to ISO-7 or EBCDIC code). For conversion to 2900 internal code a 1900 application module must be used

3 The mainframe interface. The LPL acts on mainframe commands to control data transers in both directions, and passes interrupts and status change commands to the mainframe software. When two transmissions arrive simultaneously, one in either direction, their priorities are determined by the status of the mainframe interface

The LPL has a nominal transfer rate governed by the attached mainframe, but a maximum average rate would be about 150 Kbps. Data is usually transferred to a 1900 processor in bursts of complete words, but a single character interface can be used if required. Data is transferred to 2900 processors in bytes.

6.5 The communications sub-system

This section gives further details of terminal connections to the two alternative terminal and line control modules described in section 6.1. The Communications Multiplexer (CMX) is supplied with smaller 7904 and 7906 systems. The Command Chain Unit (CCU) is supplied with a Medium Speed Scanner (MSS) for model 7904/23, 7906/23 and 7906/24 systems. Further scanners can be added, to provide for the connection of up to 256 lines.

A 7904 or 7906 system cannot support both a Communications Multiplexer and a Command Chain Unit.

3 7955/07 Communications Multiplexer (CMX) with 12 ports

Line connections to the CMX are ordered separately.

An extra 8 Kwords can be added to bring Book 0 store to a maximum 56 Kwords.

Up to four 7987 Local Processor Links (LPL) can be connected to 1900 or 2900 Series host mainframes. Each LPL is supplied with its own racking.

Any of the optional local peripherals described in section 6.6 can be connected.

6.9.2 7906/22 High Connectivity Programmable System

The system comprises:

1 7906/00 communications processor
ASR Executive console teletypewriter
Interval timer
56 Kwords store (Book 0)
7961/01 paper tape reader
F1448 ROM bootstrap loader for use with the paper tape reader

2 Racking for 4 standard modules:

Processor, Book 0 store, Communications Multiplexer, and one spare rack (available for a disc controller)

3 7955/09 Communications Multiplexer (CMX) with 54 ports

Line connections to the CMX are ordered separately.

Up to four 7987 Local Processor Links (LPLs) can be connected to 1900 or 2900 Series host mainframes. Each LPL is supplied with its own racking.
Any of the optional local peripherals described in section 6.6 can be connected.

6.9.3 7906/23 High Throughput Programmable System

The system comprises:

1 7906/00 communications processor
ASR Executive console teletypewriter
Interval timer
56 Kwords store (Book 0)
7961/01 paper tape reader
F1448 ROM bootstrap loader for use with the paper tape reader

2 Racking for 4 standard modules:

Processor, Book 0 store, and one spare rack (available for a disc controller)

3 7951/03 Command Chain Unit (CCU) to control 64 scanner lines
7950/02 Medium Speed Scanner (MSS)
Racking for CCU and MSS (with room for another two scanner units)

Book 1 store can be added in increments of 8 Kwords to a maximum 56 Kwords, giving a system total (with Book 0) of 112 Kwords.

7950/01 Low Speed Scanners (LSS), extra MSS and 7950/03 High Speed Scanners (HSS) and 7950/04 High Level Scanners (HLS) can be added. Channel expansion units may be fitted to the CCU to raise the limit for connected scanner lines from 64 to 128 or 256. Line connections to the scanners are ordered separately.

Up to four 7987 Local Processor Links (LPLs) can be connected to 1900 or 2900 Series host mainframes. Each LPL is supplied with its own racking.

Any of the optional local peripherals described in section 6.6 can be connected.

6.9.4 7906/24 Program Development System

The system comprises:

1 7906/00 communications processor
ASR Executive console teletypewriter
Interval timer
64 Kwords store (56K in Book 0, 8K in Book 1)
7961/01 paper tape reader
F1448 ROM boostrap loader for use with the paper tape reader

2 Racking for 4 standard modules:

Processor, Book 0 store, Book 1 store, one 7980 disc controller

3 7951/03 Command Chain Unit (CCU) to control 64 scanner lines
7950/02 Medium Speed Scanner (MSS)
Racking for CCU and MSS (with room for another two scanner units)

4 7980/01 disc controller
One disc transport (FEDS)

More store can be added in increments of 8 Kwords to raise Book 1 store to a maximum 56 Kwords, giving a system total (with Book 0) of 112 Kwords.

7950/01 Low Speed Scanners (LSS), extra MSS, 7950/03 High Speed Scanners (HSS) and 7950/04 High Level Scanners (HLS) can be added. High level scanners can only be used with a system having user written communications programs, as ICL does not supply software for this unit. Channel expansion units may be fitted to the CCU to raise the limit on connected scanner lines from 64 to 128 or 256. Line connections to the scanners are ordered separately.

Up to four 7987 Local Processor Links (LPLs) can be connected to 1900 or 2900 Series host mainframes. Each LPL is supplied with its own racking.

Any of the optional local peripherals described in section 6.6 can be connected. If the user wishes to run the Sysgen suite on his 7906/24, then a second FEDS transport and a line printer must be ordered.

User teletypewriters are usually required for program development work on a 7906/24 and must be ordered separately.

7.1 Introduction

The 7904 and 7906 processors are the key communications processors in the ICL transition strategy. This means that a communications network employing 7904 or 7906 communications processors can be readily moved from a 1900 Series computer to a 2900 Series computer under DME/1900, or to a 2900 Series computer under VME/B old Kommfy. A future release of 7900 software will complete the transition path by allowing the use of 7900 under VME/B new Kommfy.

In order to start out on the transition path outlined above the 1900 user may wish to change from his existing 7930 scanner-only system or 7903 communications processor system to a 7904 or 7906 system. In addition, at any point in the transition path an increase in the size or complexity of the network may require the user to upgrade his 7904 or 7906 to a larger version.

The aspects of transition outlined above are covered in this chapter under the following headings:

1 COMPATIBILITY This is of interest to the 1900 Series user moving from a 7930, 7903 or 7905 system to a 7904 or 7906 system (see section 7.2)

2 UPGRADING This is of interest to the user moving to a larger 7904 or 7906 system without changing the host mainframe (see section 7.3)

3 TRANSITION This is of interest to the user moving his communications network from a 1900 Series to a 2900 Series mainframe (see section 7.4)

The possible transition paths available to the user are shown in Figure 7.1.

The main technical considerations in the areas of compatibility, upgrading and transition are covered in the following sections. Any change to a 7904 or 7906 system should be discussed with the user's ICL support staff, who may have access to advice from other users who have already made the same change. Whenever a change is made the requirements of the network should be re-evaluated, and a network map produced as described in section 1.4.

The user's operations management should be consulted so that they can schedule the releases of the dedicated machine time required to test the new system. Because detailed procedures may change, the user should involve, during the later stages of testing, the staff that make use of the communications service, in particular, terminal operators.

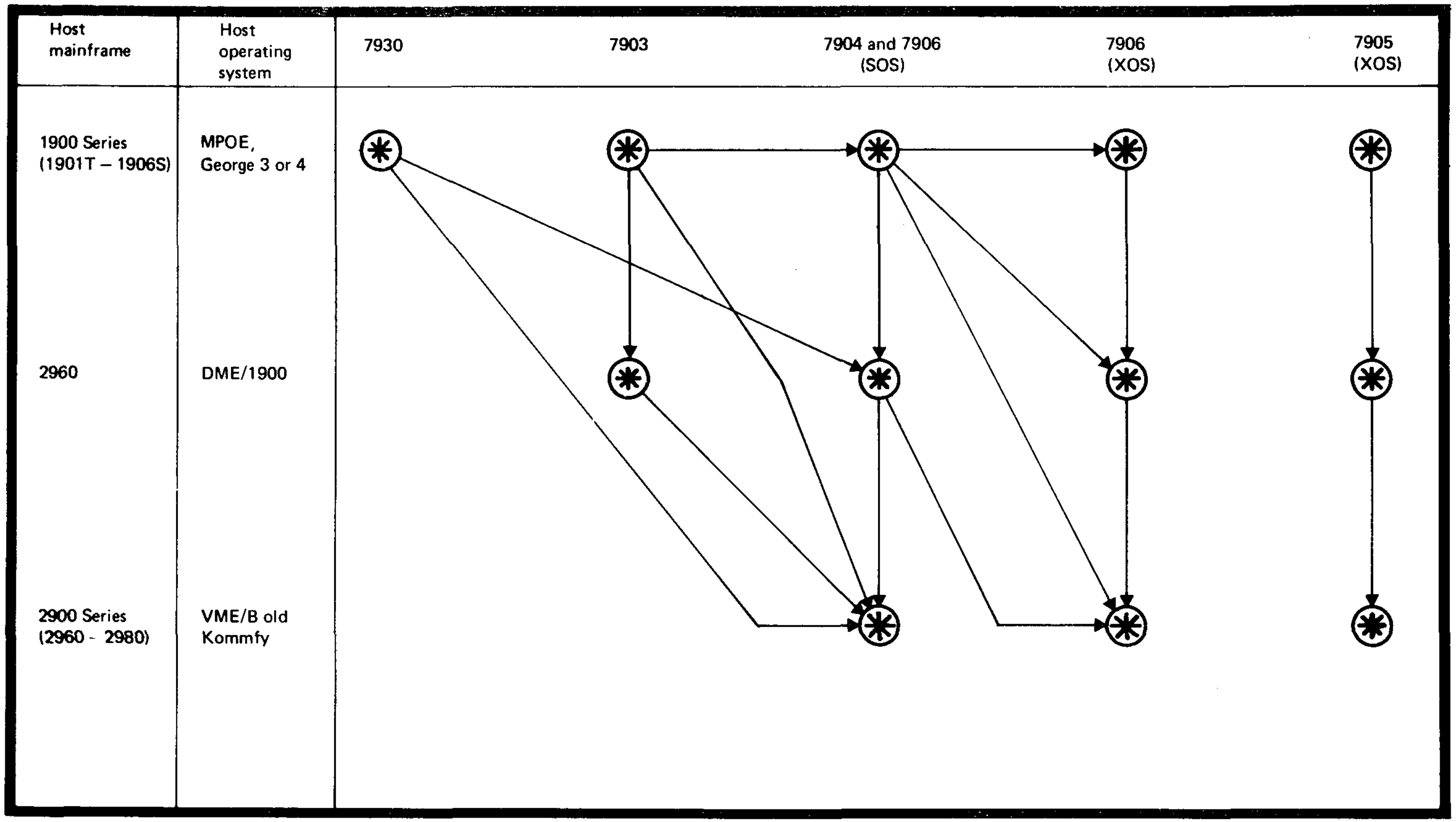

Figure 7.1 7900 transition paths

7.2 Compatibility

7.2.1 Moving from a 7930 scanner-only system to a 7904

The user of a 7930 scanner-only system would move to a 7904 running under the Standard Operating System. This is a relatively straightforward task provided that the scanner-only system is already controlled by the ICL standard communications package *Communications Manager*, or by George 3 or 4. A move from a 7930 scanner system to a 7904 is not hardware compatible; the 7930 equipment must be replaced by the equivalent 7904 system.

7.2.1.1 *Hardware*

All facilities of the 7930 are available on the 7904 except:

1 A directly connected 7181 VDU terminal cannot have more than 30 metres of cable

2 The 7153 video (predecessor of the ICL 7181 video) is not supported

3 7020 systems, or emulators of 7020 systems, cannot be multi-dropped on a single communications line

4 Protocols cannot be mixed on one line

5 The modem strapping used with a scanner is different from the strapping used with a 7904. The standard modem strapping for a 7904 is called T-strapping

6 Terminal systems using XBM protocol require some addresses to be changed by the ICL engineer in order to provide both teleload and video standby facilities

7.2.1.2 *Software*

In general the George 3 or 4 user does not have to change his application programs or user-written macros, provided the application philosophy and terminal hardware remain the same. ICL macros supplied with the 7904 software may need to be tailored to fit existing user standards.

Sites using Communications Manager that are not also using George 3 or 4, need to generate a new version of Communications Manager. Because the 7904 controls the terminals in a different way from a 7930 scanner-only system, some changes to the application programs may be necessary. The differences when changing from a character buffering to a message buffering system under Communications Manager are described in User Notice 3 to *Communications Manager*. When reading this User Notice the user should bear in mind the differences between a 7903 and 7904 system, as described in the next section.

The 7904 SOS handles all the connected lines, protocols and message buffering, which allows the application programs to perform only their true roles of processing transactions. This means that sites not using Communications Manager have to change their 7930 scanner routines. Considerable rewriting or restructuring of each application program addressing the scanner is required, and this activity should not be underestimated.

7.2.2 Moving from 7903 to 7904

The 7904 is designed as a replacement for the 7903, therefore most of the 7903 facilities and functions are included. However, the facilities offered by the 7904 should be checked against those currently provided by the 7903, to ensure that there are no incompatibilities which affect the user's applications. The user is recommended to seek the assistance of his ICL support staff in this area.

The 7904 SOS is similar to the 7903 DCP but offers greater throughput than the 7903, and requires less computer operator action when in use. This is achieved by automating some of the functions that required operator involvement with a 7903 system. For example, the 7904 operator does not: request a configuration report on the console; alter polling intervals; monitor the pool of free store cells at run time; read device statistics on the console (only system errors are reported); alter warning/continue levels for a device type; inhibit broadcast on videos; change the teletypewriter invitation-to-type character; or monitor the NOT NOW message on a specified 7181 line. There are no mainframe system commands to control the Standard Operating System.

The user's operations staff should be closely involved with the testing of the new system, in order to familiarise themselves with the changes in operational procedures.

7.2.2.1 *Hardware*

The 7904 is not hardware compatible with the 7903, so the 7903 and its associated line control equipment must be replaced by the 7904 and its equivalent communications sub-system. In general the terminal network supported by a 7903 can be supported by a 7904, but there are some exceptions for the rarer device sub-types and device modes.

The hardware exceptions specified in section 7.2.1.1 also apply to the move from a 7903 to a 7904 system.

7.2.2.2 *Software*

The 7904 SOS provides most of the facilities of the 7903 DCP, and the link with the mainframe uses the DCP protocol. In general the George 3 or 4 user does not have to change his application programs or user-written macros provided the application philosophy and terminal hardware remain the same. ICL macros may need to be tailored to fit existing user standards.

Sites using Communications Manager that are not also using George 3 or 4, do not need to generate a new version of Communications Manager unless the network is changed. Sites not using Communications Manager have to change user written software, see section 7.2.1.2.

For all types of user the method of configuring the 7904 is different from the 7903. The user must familiarise himself with the procedures provided by the Mainframe Support Environment.

7.2.3 Using 7904 and 7906 software with a 7905

The 7905 is the predecessor of the 7904 and 7906 processors, and SOS and XOS systems can run on an appropriate 7905 system. A 7905 is not physically converted to a 7904 or 7906 processor, because the software and facilities are compatible. Minor hardware and software changes may be required to the 7900 system, but no mainframe changes are required.

7.2.3.1 *Hardware*

To use ICL SOS or ND TKEY software on a 7905, the 7905 system must provide at least the 7905 processor, Executive console, real time clock, 48 Kwords of store, paper tape reader and ROM loader, an LPL, and either a 7955 Communications Multiplexer (CMX) or 7951 Command Chain Unit (CCU). Support for the CMX is the same as for the model 7904/22, and support for the CCU is the same as for the model 7904/23, see section 6.8.

The peripherals must be connected to specific processor channels, and have specific device location (dedloc) base addresses. ICL support staff can provide details on request.

7.2.3.2 *Software*

Given a suitable hardware configuration, ICL SOS or ND TKEY software can run on a 7905, provided that the module to control the interval timer is changed to the module to control the real time clock. This change is performed by the software provided by ICL for configuring or generating the operating system. All the user need do is include in the generation procedure the command that specifies that the operating system is to run on a 7905.

7.2.4 Converting from 7905 to 7906

The conversion of a 7905 to a 7906, to allow for the greater storage capabilities of the 7906 or to allow PDOS to be used, requires the 7905 processor to be replaced by a 7906 processor with an interval timer instead of a real time clock. However, some of the old 7905 store modules may be used in the new processor, together with the 7906 store modules. Some of the 7905 store modules may have to be displaced as they take up more ports than the 7906 modules do, and there may not be sufficient ports available.

A conversion is required for Command Chain Units, but all existing peripherals can be retained except for the F1489/00 ROM loader, the 7974 magnetic tape system, and the 7986/02 Local Processor Link on systems with more than 56 Kwords of store.

7.3 Upgrading

7904 and 7906 systems can be upgraded to handle growing terminal networks and enhanced to support different connections to local and remote mainframe systems. This section describes the possible methods of upgrading within 7904 and 7906 systems. It is possible to upgrade from the smallest 7904 model to the biggest 7906 model. This gives a very wide range of communications processor power. The 7906/24 is normally used when the user wishes to develop his own communications programs.

7.3.1 Hardware upgrading

7.3.1.1 *Local Processor Links*

The 7904 may only have one LPL. 7906 systems employing user programming may have up to four LPLs and they may be in simultaneous operation.

7.3.1.2 *Store*

An increase in line traffic usually requires more store to hold additional message buffers. The minimum store size of all 7904 models is 48 Kwords; this can be enhanced to 56 Kwords (1 book). The minimum store size of the model 7906/21 is 48 Kwords, of the models 7906/22 and 7906/23 is 56 Kwords, and of the 7906/24 is 64 Kwords. These four models can be enhanced to 88 Kwords for SOS or 112 Kwords (2 books) for XOS. Store can be added in units of 8 Kwords. A new version of the operating system should be produced in all cases, to make use of the additional store.

7.3.1.3 *Communications sub-system*

The basic 12-port CMX supports:

1 24 asynchronous lines at up to 300 bps

2 12 synchronous lines at up to 2400 bps

3 6 synchronous lines at up to 9600 bps

The 12 port CMX can be upgraded to the 54 port CMX, which supports:

1 108 lines at up to 300 bps

2 54 lines at up to 2400 bps

3 18 lines at up to 9600 bps

A CMX and its associated line connectors can be replaced by a CCU, scanners and their line connectors if high throughput is required.

The basic CCU supports 16 lines through one medium speed scanner. This can be enhanced to 64 lines on a 7904, and up to 256 lines on a 7906. A second CCU can be connected for additional flexibility and resilience when required.

7.3.1.4 *Peripheral sub-system*

All 7904 and 7906 systems are supplied with an Executive console and a paper tape reader. For a program development service a 7906/24 needs at least 2 Fixed/Exchangeable discs (FEDS) and a line printer. Further peripherals such as paper tape punch and user teletypewriters may be specified if required. If more than the eight peripheral interface ports are required, then a Peripheral Multiplexer (PMX) can be connected to one port. The PMX allows up to 11 slow peripherals to share a single processor port.

7.3.2 Software upgrading

This section summarises the three levels of software available for 7904/7906 systems, and the hardware required to run each set.

1 The *Standard Operating System* (SOS) runs on 7904 systems, and on 7906 systems with up to 88 Kwords of store. It incorporates a message-buffering turnkey program which enables the 7900 to control the more common network configurations as a front-end processor to 1900 and 2900 Series systems. The system is configured at the mainframe using the Network Configuring Program. The SOS cannot be used for remote mainframe connections, nor to drive local peripherals

2 *Extended Operating Systems* (XOS) for 7906 processors can be generated for particular (usually complex) configurations using the SYSGEN suite of programs. These systems run one or more user-written communications programs which control a terminal network, local or remote mainframe links, and local peripherals (including interactive user teletypewriters), as specified when the system is generated. ICL supplies a non-dedicated message-buffering turnkey program, ND TKEY, for the 7906 to control a terminal network as front-end processor to 1900 or 2900 Series systems. User-written programs must be developed using the model 7906/24 program development system

3 The *Program Development Environment* (PDE) runs on a 7906/24, and enables users to generate extended operating systems to their own requirements, and to develop communications programs written in NAL and CORAL. An extended operating system can be transferred on paper tape or disc to run on the other 7906 processor models.

In summary, 7904 systems run the Standard Operating System. 7906 systems with up to 88 Kwords of store can run the Standard Operating System or an extended operating system generated to user specifications for running communications programs. 7906 systems with more than 88 Kwords of store run extended operating systems. Upgrading a 7906 model by connecting two FEDS transports and a printer creates a 7906 system capable of running the software for program development and system generation.

7.4 Transition

The 7904 and 7906 communications processors have been designed to make transition from 1900 Series to 2900 Series computers as easy as possible. The transition route can be from any 1900 Series computer running under George 3, George 4, or Communications Manager, to the equivalent system running on an ICL 2960 computer under DME/1900, and then to any 2900 Series computer from the ICL 2960 upwards running under VME/B old Kommfy. Alternatively the user can move directly from a 1900 Series computer to a 2900 Series computer under VME/B old Kommfy. In all cases the user's investment in 7904 or 7906 hardware and software is protected, as detailed in the following sections.

A future release of 7900 software will extend the transition route to VME/B new Kommfy.

7.4.1 Transition from 1900 to DME/1900

The transition to DME/1900 is very easy provided the new mainframe is an ICL 2960, which can support a 7904 or 7906.

7.4.1.1 *Hardware*

No 7900 hardware or terminal equipment is redundant when moving from a 1900 Series computer to a 2960 under DME/1900. All types of ICL terminal supported by a 7900 when connected to a 1900 computer are also supported under DME/1900.

The only additional item of hardware required is an AM10 application module, which connects the Local Processor Link of the 7900 to the General Peripheral Controller of the 2960 system.

7.4.1.2 *Software*

No change to 7900 software is required when moving from a 1900 Series computer to a 2960 under DME. The same range of Terminal Executives, Terminal Control Programs, and therefore line protocols, are supported by both systems. The configuring or generation process produces a 7900 operating system that can be used when the 7900 is connected to a 1900 or 2960 mainframe. The same Mainframe Support Environments are used for both types of mainframe.

7.4.2 Transition from 1900 or DME/1900 to VME/B old Kommfy

7.4.2.1 *Hardware*

No 7900 hardware or terminal equipment is redundant when moving from a 1900 Series computer or a 2960 under DME/1900, to a 2900 Series computer under VME/B old Kommfy. No additional hardware is required for a move from DME/1900 to old Kommfy. If the move is from a 1900 Series computer to a 2900 Series computer, then the user requires an AM10 application module, which connects the Local Processor Link of the 7900 to the General Peripheral Controller of the 2900 Series computer.

7.4.2.2 *Software*

Some changes to software are required when moving to a 7900 system controlled by VME/B old Kommfy. The Terminal Executives and Terminal Control Programs that use XBM or PCT2 protocols are only supported if changes are made to VME/B old Kommfy. This is a special procedure, and the user must consult his ICL support representative if he wishes to use facilities that require XBM or PCT2 protocol.

The 7900 operating system must be reconfigured or regenerated in order to interface with old Kommfy, but this requires a change to only one command in the configuring procedure. User-written software for a 7906 system needs minor changes in the area of code conversion performed by the LPL handler.

A different Mainframe Support Environment is required for VME/B mainframes, but the facilities provided are the same as those of the Mainframe Support Environments for 1900 and DME/1900 mainframes.

The Error Manager facility is not currently available under VME/B old Kommfy, and the Teleload menu select facility for 7501 and 7502 systems is not available at VME/B release 5X32.

A1.1 Introduction

The major part of this appendix consists of descriptions of Technical Publications (TP) and Restricted Publications (RP) concerning 7904 and 7906 computers. The descriptions are in publication number order.

This first part of the Appendix shows the numbers and titles of the publications under subject headings; for example: system introduction, operating, programming and so on. Many 7905 publications are still relevant to the 7906 user who is developing his own communications programs.

A user of the 7900 Standard Operating System needs only the following publications: RP0365, RP0363, and RP0360 or RP0361 as appropriate; see sections A1.1.1, A1.1.2 and A1.1.3.

A user of the ND TKEY system needs only the following publications: RP0365, RP0363, and RP0362 or RP0364 as appropriate; see sections A1.1.1, A1.1.2 and A1.1.3.

The 7900 user developing his own programs needs publications from all the following sections.

At the end of this first part of the Appendix a list of related mainframe publications is given; these are not described in the main body of the Appendix.

A1.1.1 System introduction

RP0365 Introduction to 7904 and 7906 (this publication)

A1.1.2 Hardware

RP0363 7904 and 7906 Hardware Operating

A1.1.3 Operating

RP0360 1900 Series: Using 7900 Standard Operating Systems

RP0361 VME/B: Using 7900 Standard Operating Systems

RP0362 1900 Series: 7906 Operation

RP0364 VME/B: 7906 Operation

A1.1.4 Operating system

TP4874 Introduction to the Operating system

TP4888 E4 Executive

TP4889 E4 System Operator

TP4890 Running SYSGEN

A1.1.5 Programming

TP4858 E4 User Operating

TP4858 NAL (New Assembly Language)

TP4864 NALA (E4) User

TP4865 CORAL 66 Language

TP4866 CORAL 66 Programming

TP4867 CORAL 66 User

TP4875 Introduction to NAL programming

TP4876 Programming Aid

TP4879 Communications User Interface (CUI)

TP4887 E4 Programming

A1.1.6 Utility programs

TP4860 E4 Loading Language (E4SLB)

TP4869 IDA-Interactive Development Aid

TP4870 LIMPID Link Editor

TP4871 GERBIL Language

TP4872 Serial File Organisation

TP4877 System Debugging Aid - POKE

TP4882 Editor

TP4884 Disc Utilities

A1.2 Related mainframe publications

The following publications describe aspects of other ICL systems that are relevant to implementing 7900 Series systems as communications processors:

TP4432 GEORGE 2 Disc Based Operating

RP3006 2950, 2956 DME/3 Operations *(for GEORGE 2)*

TP4448 MAXIMOP system

TP4445 Using MAXIMOP

TP4438 GEORGE 3 and 4 Operation Management

TP4345 Operating systems GEORGE 3 and 4

TP3723 2960 DME/1900 Operations *(for GEORGE 3)*

TP4420 Communications Manager

TP6522 VME/B System Construction and Maintenance

A1.3 Description of each 7900 publication

There follows a description of each 7900 publication, in order of publication reference number.

1900 Series: Using 7900 Standard Operating Systems

RP0360 (Edition 1, 1980)

This is a reference publication for all operations staff at 1900 or 2960 DME/1900 installations. It details the procedures for configuring, loading and running a Standard Operating System, for fault diagnosis using Error Manager, and for taking postmortem dumps to the host mainframe. This publication is for use with *7904 and 7906 Hardware Operating*.

Topic list

1 Configurations that SOS can support

2 Mainframe environments

3 Installating the contents of the release tape at the mainframe

4 Preparing the specification file and running the Network Configuring Program (NCP)

5 Patching SOS: patching files and running the utilities at the mainframe

6 Loading and running SOS: the mainframe loading utility; console messages and operator commands; error conditions; standby lines; powering off

7 Fault diagnosis: taking postmortem dumps; the mainframe dump utility; warm restart; running Error Manager and interpreting the output; actions following console error messages; test procedures and test programs

VME/B: Using 7900 Standard Operating Systems

RP0361 (Edition 1, 1980)

This is a reference publication for all operations staff at VME/B installations. It details the procedures for configuring, loading and running a Standard Operating System, for fault diagnosis, and for taking postmortem dumps to the host mainframe. This publication is for use with *7904 and 7906 Hardware Operating*.

Topic list

1 Configurations that SOS can support

2 Mainframe environments

3 Installing the contents of the release tape at the mainframe

4 Preparing the specification file and running the Network Configuring Program (NCP)

5 Patching SOS: patching files and running the utilities at the mainframe

6 Loading and running SOS: the mainframe loading utility; console messages and operator commands; error conditions; standby lines; powering off

7 Fault diagnosis: taking postmortem dumps; the mainframe dump utility; actions following console error messages; test procedures and test programs

1900 Series: 7906 Operation

RP0362 (Edition 2, 1980)

This is a reference publication for all operations staff at 1900 or 2960 DME/1900 installations with a 7906 system as a front-end processor. It details the procedures for specifying the system to be generated, for loading and running it, for fault diagnosis using Error Manager, and for taking postmortem dumps to a host mainframe. It does not describe programming for 7906 systems, program development, or the actual process of system generation.

This publication is for use with *7904 and 7906 Hardware Operating*.

Topic list

1 Configurations that can be supported by generated systems

2 System generation: options that can be specified for the steering file; using the ICL SYSGEN service

3 Installing a generated system with the ND TKEY program or any other communications program ; dumping the system to a mainframe file

4 Loading and running the generated system; running a communications program; console messages and operator commands; error conditions; standby lines; powering off

5 The ICL ND TKEY communications program; configuring the network to be controlled

6 Patching a generated system: patching files and running the utilities at the mainframe

7 Fault diagnosis: taking postmortem dumps; the mainframe dump utility; warm restart; running Error Manager and interpreting the output; actions following console error messages; test procedures and test programs

8 The virtual store facility

7904 and 7906 Hardware Operating

RP0363 (Edition 1, 1980)

This is a reference publication for operators. It details all hardware components and available peripherals for 7904 and 7906 systems, user maintenance procedures and how to operate the switches, keys and buttons.

This publication complements *Using 7900 Standard Operating Systems* and *7906 Operation*.

Topic list

1 Introduction to 7904 and 7906 systems, their component sub-systems, available local peripherals and supported communications devices

2 A description of the switches and display lights; loading and closedown actions; error procedures for processor, store, Local Processor Link, Communications Multiplexer, Command Chain Unit and scanners

3 Operating and user maintenance procedures for all available local peripherals

4 Display lights, operation and error conditions for the ICL terminals supported on 7904 and 7906 systems, with cross-references to the appropriate reference manual for each device type

VME/B 7906 Operation

RP0364 (Edition 1, 1980)

This is a reference publication for all operations staff at VME/B installations with a 7906 system as a front-end processor. It details the procedures for specifying the system to be generated, for loading and running it, for fault diagnosis, and for taking postmortem dumps to a host mainframe. It does not describe programming for 7906 systems, program development, or the actual process of system generation.

This publication is for use with *7904 and 7906 Hardware Operating*.

Topic list

1 Configurations that can be supported by generated systems

2 System generation: options that can be specified for the steering file; using the ICL SYSGEN service

3 Installing a generated system with the ND TKEY program or any other communications program; dumping the system to a mainframe file

4 Loading and running the generated system; running a communications program; console messages and operator commands; error conditions; stand lines; powering off

by

5 Running the ICL ND TKEY communications program; configuring the network be controlled

6 Patching a generated system: patching files and running the utilities atto the mainframe

7 Fault diagnosis: taking postmortem dumps; the mainframe dump utility; actions following console error messages; test procedures and test programs

8 The virtual store facility

Introduction to 7904 and 7906

RP0365 (Edition 1, 1980)

(This publication)

E4 User Operating

TP4858 (Preliminary edition, 1973)

This reference publication is for programmers and operators who wish to load, run and control programs interactively under the E4 operating system. The publication gives the complete definition of the communication between a programmer at a user teletype and the E4 operator software. Chapter 6 should be ignored; the current information on engineer's test program control in a dedicated system is given in *Using 7900 Standard Operating Systems*.

This publication should be used with *E4 Executive* and *E4 Programming*.

Topic list

1 Introduction to the facilities, commands and error messages available

2 Definition of the interaction process between the user and the Operator Interface

3 The interactive resource control commands: LOGIN, LOGOUT, LOAD, FREE

4 Job control commands: JOB, EOJ, ABORT, ADD, SUSPEND, RESUME, LIST

E4 Loading Language

TP4860 (Preliminary edition, 1973)

This reference publication defines the E4 System Loading Binary (E4SLB) language which is used to request system resources when a program is loaded. This loader language is produced by the NAL Assembler as well as the LIMPID link editor. As a language specification this manual also contains details of the three control blocks required to run a program, and includes an example of their usage; further details of these and other control block structures are given in *Introduction to the Operating System*.

This publication should be used with *E4 Programming*.

Topic list

1 A summary of the load information required for each program file; loader directives, program segment presets and control block data for the Initial Information Area

2 A definition of the program file in terms of the permitted character set and file layout

3 The five available loader directives

4 How to define the control blocks required for each program: segment CB, workspace CB, activity CB; and an example of their usage

NAL (New Assembly Language)

TP4862 (Preliminary edition, 1973)

This reference publication defines the assembly language NAL in terms of the basic instruction set.

Topic list

1 Introduction to basic concepts and facilities

2 Elements of the syntax

3 Annotated examples of NAL programs

4 Definitions (in Backus Naur notation) of all NAL statements

NALA (E4) User

TP4864 (Preliminary edition, 1973)

This publication describes the NAL Assembler and the variants available for NRL, E4SLB and GERBIL binary code. It includes details of control file commands, how to prepare the source file, the output listings, and how to run the Assembler in batch or interactive mode.

This publication should be used with *NAL (New Assembly Language)*.

Topic list

1 A summary of the peripherals, store and operating environment for running NALA

2 An outline description of NAL and NALA facilities, the two NALA input streams (control file and source file), the five output streams (binary code, source and store addresses listings, error listing, symbolic names listing, monitor listing) and an outline of the operating sequence

3 A description of the use of input and output streams

4 A description of assembler variants, NALA operating instructions and assembler error codes

5 A brief description and summary of data formats and control file specification records

CORAL 66 Language

TP4865 (Preliminary edition, 1973)

This publication describes the version of CORAL 66 language accepted by the CORAL compiler in relation to the industry standard, the *Official Definition of CORAL 66* (HMSO; ISBN 11 470221 7; 1970). As a language specification it assumes the reader is an experienced programmer already familiar with the official definition and the NAL assembly language. Programmers without this knowledge should first read *Introduction to NAL Programming, CORAL 66 Programming* and the *Official Definition of CORAL 66*.

Topic list

1 A description of deviations from, and implemented extensions to, the official definition

2 Descriptions of store allocation, addressing procedures and the use of registers as implemented by the compiler

3 A description of the code generated by the compiler for declarations and statements

CORAL 66 Programming

TP4866 (Preliminary edition, 1973)

This publication provides an introduction to CORAL 66 for systems programmers already familiar with NAL. It should be used in conjunction with *CORAL 66 Language*.

Topic list

1 An introduction to the main features of CORAL 66 and the notation used

2 A description of the basic program structure in terms of blocks, declarations, statements, labels, expressions, arrays, conditions and compound statements

3 Descriptions of procedures, loops (in terms of FOR and WHILE statements), constants and preset declarations, the SWITCH statement

4 Methods of handling packed data in the form of bits and tables; fixed point arithmetic; string, literal and macro facilities

5 Use of procedure parameters

6 Extensions to the official definition such as LIBRARY and KEEP statements, XYZ segmentation, address manipulation and overlays

7 How to insert NAL code into CORAL modules to provide specialised facilities not available in the compiler

8 How to specify run-time OPTIONS (the map and data structure sizes) and to preset words in specified segments using CONTROL declarations

9 Programming strategy using source annotation (PAGE and COMMENT), page layout; programming techniques such as procedures, the use of FOR rather than GO TO; debugging considerations

CORAL 66 User

TP4867 (Preliminary edition, 1973)

This reference publication describes how to run the CORAL 66 compiling system in batch or interactive mode, how to link the compiled modules into a loadable program, and the compiler diagnostics and error messages.

This publication should be used with *CORAL 66 Programming* and *CORAL 66 Language*.

Topic list

1 The operating environment required for running the compiler

2 A brief description of the compiler in terms of extensions to the *Official Definition of CORAL 66* (HMSO, 1970), and of the language facilities and compiler diagnostics

3 The stages of compilation and the operating prompts needed for both interactive and batch mode

4 Operator facilities; hardware and software requirements; source input typing formats

5 How to load the compiler and specify the files correctly

6 Linking procedures and segment usage

7 Diagnostics format; error and informative messages; the store map

IDA-Interactive Development Aid

TP4869 (Preliminary edition, 1973)

This reference publication describes how to prepare and run a program to be monitored by IDA.

Topic list

1 The use of IDA as a debugging aid to control the progress of a program, monitor the contents of segments, output snapshots of predetermined store areas in batch or interactive mode, and to alter program locations in interactive mode

2 Operator facilities and instructions, and operational requirements

3 The instructions used for initialising and snapshots; the syntax of IDA commands and the variants of IDA

4 The format and effect of the commands, arranged in logical sequence

LIMPID Link Editor

TP4870 (Preliminary edition, 1973)

This reference publication describes how to prepare a set of program modules in relocatable binary (GERBIL code assembled by NALA or the CORAL compiler) to be linked together using LIMPID, and how to run LIMPID in batch or interactive mode.

This publication should be used with *GERBIL Language* (TP4871).

Topic list

1 A description of the main operations of LIMPID: allocating store and resolving both forward and inter-module references to produce a complete program of linked modules, and a printout

2 The constraints on link sequence

3 A description of the set of available outputs: local and external maps, segment table, overflow and failure messages

4 File specifications as used for both batch and interactive operation

5 The operating instructions to load, initiate and run LIMPID

6 The available variants of LIMPID for SLB and NRL output

GERBIL Language

TP4871 (Preliminary edition, 1973)

This specialist reference publication provides the formal definition of the General Relocatable Binary Language, GERBIL, as output by the NAL Assembler or the CORAL compiler. The use of GERBIL is described in *LIMPID Link Editor* (TP4870).

Topic list

1 An introduction to GERBIL and its syntax

2 How GERBIL source is loaded in the form of load items relative to a parameter-specified base

3 A description of the various types of parameters (numeric, dummy and name) and their respective attributes (base, length, segment, loading pointer and association)

4 The skip facility provided by the Boolean Test Register

5 A description of the directives used to set parameter attributes

Serial File Organisation

TP4872 (Preliminary edition, 1973)

This reference publication specifies the serial file organisation and standard formats for card and paper tape documents. It provides an introduction to the concepts of volumes, files, blocks and records, records of fixed, variable and unknown length, and terminator characters.

Topic list

1 A description of file organisation into volumes, files, blocks and records

2 A definition of data types: symbolic and binary texts and binary data

3 A description of related media-dependent formats

4 Tables to summarise media-dependent formats and to define the character codes (internal, punched card and paper tape)

Introduction to the Operating System

TP4874(Edition 1, 1975)

This publication gives a general description of the Communications Operating System (COS) for systems programmers who already have a general grasp of its structure. In particular the publication describes the Executive functions and control interfaces that are available to user-written programs.

Introduction to NAL Programming outlines how these functions and interfaces are used, and *Communications User Interface (CUI)* gives details of the communications handling functions.

Topic list

1 A general introduction to facilities supported by COS

2 COS concepts: programs; activities; registers; resource control blocks; the message system

3 A summary of hardware facilities for driving communications devices and local peripherals

4 Communications control software: program and device mode files and data streams; functional levels in a communications link; connection scheduler, system intelligence and error handling

5 The Data Transport System and CDTCs (Communications Data Transport Calls)

6 A guide to specifying the steering file for system generation (dedicated or non-dedicated system)

7 Details of control blocks and control block structures

Introduction to NAL Programming

TP4875 (Preliminary edition, 1974)

This publication provides a descriptive introduction to the New Assembly Language (NAL) for systems programmers, including its interface with the processor architecture and Executive routines, and how it can be used to write communications programs.

This publication complements *E4 Programming* and *NAL (New Assembly Language)*

Topic list

1 An outline of processor architecture: instruction formats; arithmetic and logical operations; addressing

2 A definition of the basic 7900 function set (32 functions)

3 A description of NAL program structure seen conceptually and from the systems and programming points of view

4 A summary of storage conventions for XYZ segments

5 A description of input/output handling routines for local peripheral files

6 A description of communications files and using CDTCs (Communications Data Transport Calls)

7 A description of some of the Function 30 Executive calls

8 A description of the NAL Assembler (NALA) including the use of assembler directives

9 Tables: the function set; addressing modes; shift functions; test functions; terminology and symbols

10 Programming examples

Programming Aid

TP4876 (Edition 1, 1974)

This reference booklet provides an aide-memoire for programming and presents all the most essential information in tables.

Topic list

1 Assembler instructions: address modes; the 32 basic functions; CPY logical and arithmetic functions. These instructions are detailed in *Introduction to NAL programming*

2 Executive operations: supervisor calls (SVCs); Special Communications Utility Routines (SCURTs); Input/output operations (IOPs); Communications Data Transport Commands (CDTCs). The non-communications instructions are detailed in *E4 Programming*; the communications handling instructions are detailed in *Communications User Interface (CUI)*

3 Command Chain Instructions for the Command Chain Unit (CCU): buffer; command; branch

4 Communications error reject qualifiers

5 Character sets for paper tape and 80 columns cards

System Debugging Aid - POKE

TP4877 (Preliminary edition, 1974)

This reference publication provides a formal specification of POKE and its facilities.

Topic list

1 A summary of the functions provided, followed by information on the quiescent and active modes of operation and on Page 2 DEDLOCS

2 Specifying configuration options

3 Details of the command interface, the comment facility and parameter formats

4 A description of each of the 15 commands

5 Examples of the use of POKE

Communications User Interface

TP4879 (Edition 1, 1976)

This reference publication provides specifications of the communications set of Function 30 Executive calls. It also defines the commands used to control the communications device handlers and the interface to the related system modules, Error Manager and System Intelligence.

This publication complements *E4 Programming*.

Topic list

1 A set of reference tables summarising the commands, reports, error messages and error operation codes with a cross-reference from each entry to the appropriate specification

2 Supervisor calls (SVCs) concerned mainly with manipulating communications files

3 Input-output operations (IOPs) for handling program mode connections to another processor

4 IOPs for handling device mode connections to terminals

5 IOPs and Communications Data Transport Calls (CDTCs) for data transfer over a program mode link

6 CDTCs for data transfer over a device mode link

7 IOPs for handling line mode connections to another processor or to a terminal in the absence of device driver routines

8 A description of error collectors and error management facilities

9 A description of, and commands to control, System Intelligence and Error Manager

10 Specifications of the device drivers: teletype; video; 7020; MSHD remote mainframe link; XBM2 remote mainframe link; LPL

Editor

TP4882 (Edition 1, 1975)

This reference publication describes the facilities provided by, and the operating instructions for, the editor, which can be used on any source file in standard serial format. Each of the 11 editing commands is specified in detail and examples are given throughout.

Serial File Organisation, TP4872, describes the standard serial format.

Disc Utilities

TP4884 (Edition 1, 1975)

This reference publication describes the Disc Filing System, its associated utilities and how to set up, maintain and use disc files.

Topic list

1 An introduction to disc storage, the facilities available and the various utility programs

2 A discussion of the factors and constraints to be taken into account when planning disc storage

3 The use of, and operator interface to, the Disc Filing System

4 The physical file structure and detailed layout of a disc partition

5 The use and operation of the disc initialisation utilities: VOLMAN, SETUP, DICTMAN

6 The maintenance utilities: DEDIT, FILEMAN, DCOPY

7 Operating instructions for all the programs described

8 The syntax of file specifications; the user interface to the Text Processing System (TPS); reject qualifiers and TPS failure codes

E4 Programming

TP4887 (Edition 1, 1975)

This is a reference publication providing definitions and details of how to use Function 30 Executive calls, particularly the non-communications calls. Details of the communications calls can be found in *Communications User Interface (CUI)*, TP4879.

This publication should be used with *NAL (New Assembly Language)*, TP4862.

Topic list

1 An introduction to the use of parameters, Run-Time Names (RTNs), error handling and the concept of trusted programs

2 Instructions for: segment control; synchronisation control; activity control

3 An introduction to the Supervisor Call (SVC)

4 Instructions for: input/output control; access control; controlling software clocks; internal job control; internal error collection and error handling

5 Summary lists of the Executive calls and error numbers, codes and classicifations

6 Loader information and descriptions of relevant control blocks. Control block structures are described in *Introduction to the Operating System*

E4 Executive

TP4888 (Edition 1, 1975)

This publication describes the Executive kernel around which the Communications Operating System (COS) is built. Executive drives the processor and the non-communications hardware and schedules multiprogramming for COS.

This publication complements *E4 Programming*.

Topic list

1 E4 concepts: resources and resource catalogues; machine, supervisor and operator interface modules; running work; activities; the message system; system principals

2 The protection system for access right to spheres

3 A summary of the input/output system; this is further described in *E4 Programming*

4 A summary of error handling and debugging facilities

5 A description of the internal structure of Executive and its organisation of workspace

6 The activity instruction set: a survey of the non-communications Function 30 EXEC calls; these are further reviewed in *Introduction to NAL Programming* and detailed in *E4 Programming*

7 The operator interface: a list of basic, non-communications operator actions and console commands

E4 System Operator

TP4689 (Edition 1, 1975)

This is a reference publication for operators. It describes the operator commands that can be issued at the Executive console to control the facilities of the E4 Executive. Chapter 7 and the Appendices should be ignored by the 7906 user.

This publication should be used with *E4 Executive* and *7906 Operation*.

Topic list

1 Loading the 7900 system from paper tape or disc, and across a remote link

2 Initialising the 7900 system; power on, power off, virtual memory

3 Advisory system messages and device error recovery for local peripherals

4 Peripheral control and device identification

5 Operator commands to the operating system; making peripheral devices available or unavailable; opening and closing disc files

Running SYSGEN

TP4890 (Edition 1, 1976)

This reference publication outlines the system generation procedure, which can be run by a user on a suitable 7900 installation. The E4 operating instructions for each program in the SYSGEN suite are given.

The recommended approach to defining the operating system that is to be generated is given in the *7906 Operation* publications.

Topic list

1 Introduction to the structure of the SYSGEN procedure

2 Operating instructions for the SYSGEN programs

3 Meaning of error messages and warning messages produced by SYSGEN

4 The syntax and use of code selection directives and SHUFL directives

This glossary gives the definitions of all the computer terms that are likely to be met when reading 7900 publications. Words that are not specific to 7900 systems, but apply to any computer system, are not covered. Terms that only apply to detailed programming of the 7900 are not covered here because they are defined in a glossary to the publication *Communications User Interface*.

In the following definitions, words in italics are also defined in the glossary.

Activity

An activity is the smallest processor executable unit for which the *Communications Operating System* can schedule processor utilisation. It is created by the initiating procedures of a program call and causes a *Job* to progress by executing its code.

It may be likened to a PL/1 Task, or a subprogram in the 1900 system.

AM10

Application Module 10 is the hardware link between the LPL and an ICL 2900 Series mainframe.

AOF

The Advanced Operating Facilities: provide a spooling and job control language interface to the *Executive* and *SDFS*; load programs and files; optimise hardware usage; provide macro commands facility

AP

The standard abbreviation for *application program*.

Application program

An application program is an aggregate of software existing as a loadable program file. It uses the operating system facilities via the *Communications User Interface*, in order to handle the buffering and routing of messages.

ASR

Automatic Send and Receive, the facility on teletypewriters of using a paper tape reader and paper tape punch.

Asynch

Asynchronous: method of communications (start/stop) for slow terminals such as teletypewriters.

Auto-answering

The auto-answer facility allows a modem on the telephone network at the mainframe end, to detect the ringing tone and automatically to connect the call to the mainframe.

Batch Mode

The mode of operation where a command string is prepared and submitted to the *operator interface* which allows a group of jobs to be run under the control of the *system operator*.

BCC

Block Control Count, a parity check character.

BISYNC

BInary SYNChronous method of communication.

Block

The physical unit of transfer of data between a device (*file*) and an area in direct store (*segment*).

Book

A book is a unit of capacity of the main store equal to 56 Kwords.

BPS or bps

Bits per second, the rate of transmission along a communications line.

Buffer cell

A buffer cell is a contiguous area of 7900 store (not more than 4095 bytes) used to hold input or output data.

Catalogue Name

A six character name under which a *resource* is retained in the catalogue.

CB

This is the standard abbreviation for *Control Blocks*, used by *COS* to control hardware and system resources.

CCS

This is the standard abbreviation for the Communications System (the 7900 and linked machines as seen by the parent processor).

CCU

Command Chain Unit, a 7900 hardware unit that controls communication lines.

CDBOOT

A program in the Program Development Environment (*PDE*) that loads into 7900 store a disc resident operating system or program

CDUMP

A program in the Program Development Environment (*PDE*) that dumps 7900 store contents to a specified disc *partition*.

CDLOAD

A program in the *SYSGEN* suite that writes a loadable 7900 operating system base to a 7900 disc partition.

CDPRINT

A program in the Program Development Environment (*PDE*) that prints, without analysis, a 7900 store dump held in a disc *partition*.

CLEARFILE

A program in the *SYSGEN* suite that clears a 7900 disc file by marking each page as the last page in the file.

Clock

A *resource* used either to mark the passage of time or stimulate an *activity* at fixed periods.

CM

Communications Manager, a standard 1900 Series system program, responsible for a communications network.

CMUX

See *CMX*

CMX

Communications Multiplexer, a 7900 hardware unit that controls communication lines.

CODESEL

A program in the *SYSGEN* suite that selects code from a master file, using directives from a steering file.

Command

A command is an *Executive call*.

Communications File

A communications file is the means by which an *application program* makes connections. It may be

1 A Line Mode file, representing a multiplexer channel

2 A Device Mode file, which is a table of up to 63 ports each capable of representing a connection to a terminal device

3 A Program Mode file, which is a table of up to 63 ports each capable of representing a connection to an *application program* in a *remote processor*

Communications Operating System

The Communications Operating System is the name given to the integration of CTL's E4 and ICL's 7900 Communications Software System. The two items are not distinguishable when integrated and are collectively called the 7900 Communications Operating System, or, in abbreviated form, 7900 COS or simply *COS*.

Concentrator

See *Remote concentrator*.

Configuring

The act of using the *Network Configuration Process (NCP)* to tailor the *Standard Operating System* to the user's requirements.

Console

See *TTY* and *Control console*.

Control Blocks

Control blocks, of varying sizes, are used by Executive to protect and control *Resources* and *Activities*. Space is taken from continuous areas of direct store.

Control Console

A console (usually a teletypewriter or visual display console) which has been allocated *job control* facilities. It is used interactively. One specific console, the *executive console*, provides system control and is under the control of the *system operator*.

COPYIT

A general utility in the *SYSGEN* suite that copies a file from any 7900 input device to any 7900 output device.

CORAL 66

CORAL 66 is a high-level programming language, intended specifically for real time applications.

COS

This is the standard abbreviation for 7900 *Communications Operating Sytem.*

CREATE (Resource)

Certain resources can be created at run-time by an *Activity*. These are *Activities, Files, Segments; Locks, Links* and *Clocks.*

Data Transport System (DTS)

The DTS is an *Executive* sub-routine which, acting on parameters in an *application program*, passes control to a Line or Device Driver.

DCM

Diagnostic Communications Manager, a 1900 Series communications software package.

DCOPY

A program in the Program Development Environment (*PDE*) that copies files belonging to a *principal* from one disc *partition* to another.

DCP

This is the standard abbreviation for the Dedicated Control Program, which is the name given to the control software in the 1900 Series Communications Processors (as seen by the mainframe).

DEDIT

A program in the Program Development Environment (*PDE*) that modifies or prints the contents of a disc *partition* or disc *volume.*

DEDLOC

DEDicated LOCation of an interrupt to *COS*.

Device Identification

Identifies a particular peripheral (and a *Logical Volume Number* in the case of a disc known to Executive). Consists of a *physical medium identifier* (PMI) followed by a *physical peripheral identifier* (PPI).

DICTMAN

A program in the Program Development Environment (*PDE*) that creates or deletes *principals* or user dictionaries. DICTMAN can also specify or adjust the space in a *partion* allocated to user files.

Directory

Part of any disc *partition* which is used for disc filing; holds organisation information about files located within that partition as well as allocation bit maps and *extent* entries for each *File.*

DSL

Direct Store Link (from a peripheral or communications device).

DSR

Data Set Ready (Modem on-line).

DTS

See *Data Transport System*.

Dynamic Store Scheduling

This is the ability to relocate dynamically segments between direct store and an area of disc; this process is performed without user intervention.

EDITAOF

A program in the Program Development Environment (*PDE*) that copies (or amends) text from one 7900 disc file.

EM

This is the standard abbreviation for *Error Manager*.

EOM

This is the standard abbreviation of End of Message.

Error Collector

Any *activity* within a *sphere* can nominate itself to be notified of all errors occurring in the sphere. Two error masks are specified, one for the activity itself and one for all other activities within the sphere.

Error Manager

Error Manager is an activity designated at system generation time, or at run time using the DECLARE EM command, to report transient line errors and re-tries to 1900 mainframes operating under GEORGE 3 or GEORGE 4.

Error mask

An error mask is a programmable mask separately specified for each *job* in the system indicating the error classes which are to result in reject action by *COS*.

Error operation code

This is an 8-bit number (2 hexadecimal characters) coded to indicate the Executive routine or event which detected the error.

Execution Procedure

When a program is initiated, its activities enter their execution procedures. This procedure involves the machine setting up housekeeping information within *Executive* and setting up the relocation and working contents for each activity. The processor then obeys the instructions, including Executive-simulated instructions.

Executive

The element of the operating system that controls the processor, store and local peripherals, and provides the *operator interface*.

Executive Console

One console is nominated at *system generation* to be the Executive Console. This console is used by the *system operator* to exercise overall control of the hardware and software systems, by means of a number of special operator commands.

Extent (disc file)

When space on a disc cannot be allocated in one continuous area to a *file*, the disc filing system allocates the space as two or more separate areas known as extents. A file may have a number of extents, to a limit set at disc initialisation time. The user treats his files as a logically continuous area.

FCOPY

A program in the Program Development Environment (*PDE*) that copies the contents of one 7900 disc file to another.

FEDCOPY

A program in the Program Development Environment (*PDE*) that copies the contents of one 7900 disc *volume* to another.

FEDS

This is the standard abbreviation of Fixed/Exchangeable Disc Storage.

FEP

See *Front End Processor*.

File

A file is the source and/or destination of information and is accessed through the input/output system. All input/output can be considered as the transfer of data between a file and a *segment* in direct store. Thus output to a teletype is as much an output to a file as output to a disc file.

File Identification

Identifies an actual file by a filename consisting of up to 16 graphical characters from the ISO-7-UK set. A *principal* number and *password* are associated with each file identification.

Filelock

Every *file* has associated with it a filelock which may be engaged by an *activity*, subject to that activity having lock access to the file. Filelock is provided to allow activities to protect against interleaved updating of data.

FILEMAN

A program in the Program Development Environment (*PDE*) that provides facilities for controlling disc files.

File Specification

Used to identify a file and the device holding the file. It consists of *device identification* and *file identification*. There are two formats for a file specification: type A and the more recent type B.

Fixed-active

Fixed-assigned

Fixed-configured

A stream or communications file is said to be fixed-active (or fixed-assigned or fixed-configured) if the corresponding stream (or file) is set-up at *system generation* time by a suitable macro as activated, assigned or configured respectively.

Front End Processor

A front end processor is the term used to describe a communications processor connected via a Local Processor Link (*LPL*) to a mainframe processor.

GEN

Generation (of file on disc)

GERBIL

GEneral Relocatable BInary Language that can be output from a *NAL* assembly or *CORAL* compilation.

GPC

General Peripheral Controller (2900 mainframe hardware unit).

HCP

Hard Copy Printer, usually a serial printer attached to a terminal.

HDLC

High-level Data Link Control - a line protocol.

HSIC

High Speed Interface Channel.

HSS

High Speed Scanner, which handles up to eight telephone lines at up to 48,000 bps.

IDA

See *Interactive Development Aid*.

Illegal Operation

Any operation which causes an error. Includes both hardware conditions (address bound violations, for example) and software conditions (attempting to write to a file while having read-only access to that file).

Interactive Development Aid (IDA)

This is a program debugging aid that is part of the Program Development Environment (*PDE*).

Interactive Mode

The mode of interaction between a user and the *operator interface* where the user from a control console has direct control over the running of a program. It is contrasted with *batch mode*.

Interrupt Service Routine

That piece of code which is entered when an external interrupt is accepted by the processor and which services that interrupt. Usually in *special state* though sometimes in *normal state* when an interrupt is handled by a user *activity*.

ISR

See *Interrupt Service Routine*.

JCL

Job Control Language - provided by the *Advanced Operator Facilities*.

Job

A unit of work submitted for execution on behalf of a *Principal* in order to create, delete, and modify *resources*.

Initiation of a job implies:

creation of a new *protection sphere*
initiation of execution of a nominated program

Job Control

The facilities available for controlling the progress of a *job*. Includes the ability to initiate, suspend and resume the job, alter its priority or terminate the whole job or individual *activities* within the job.

Job Identification

The name given to a job when it is initiated. It is up to four graphical characters from the ISO-7-UK set, excluding comma, semi-colon and full stop, but starts with a letter.

Job Specification

Used when initiating a program into a *job* and consists of *job identification* and a job priority (in a system with priority scheduling) in the range 1 to 127.

Labelled File

A labelled *file* is a file which has an associated header label which is checked whenever the file is opened. A labelled file is one sort of retained *resource* though this has no meaning for serial peripheral files, where the label is checked on input and generated on output.

LIMPID

A 7900 link editor for *GERBIL* modules; part of the Program Development Environment (*PDE*) and the *SYSGEN* suite.

Link

This is a *resource* providing a connection between a particular interrupt base address and an *activity*. It includes an *interrupt service routine*, entered when the interrupt is received. Links are of use where the user needs to control interrupting devices himself.

Link Level

This is the name used to refer to the control software responsible for transferring data from one end of a physical link to the other.

LISTING

A 7900 program that lists the control data of files within a user's dictionary; part of the Program Development Environment (*PDE*).

LL

This is the standard abbreviation for *Link Level*.

Locator Pages

The first two readable pages on a disc volume. Their contents define the bounds of *partitions* and the associated access restrictions.

Lock

May be used by *activities* to ensure that shared *resources* are protected from interleaved updates. A lock is not implicitly associated with a resource but represents a conscious effort by the programmer to protect resources. Activities attempting to engage locks are held in a queue for the lock, being suspended while in the queue.

Logical Volume Number (LVN)

See *Volume*.

Log In

The process by which a user declares the *principal* for whom he is to work and a valid *password* associated with the principal number. Used both in *interactive* and *batch* modes of operation.

Log Out

The process by which a user removes his right to use the system. At this time all programs loaded but not retained by him will be freed.

LPL

Local Processor Link. (See also *Front End Processor*)

LPLH

This is the standard abbreviation for local processor link handler, the software module responsible for controlling *local processor links*.

LPLN

This is the standard abbreviation for the 7967/1 local processor link, which connects a 7900 to a 1900 Series processor.

LSR

Line Service Routine is an alternative name to Device Driver.

LSS

Low Speed Scanner, which handles up to 64 lines at up to 300 bps.

LVN

See *Volume*.

Machine

That part of Executive which handles the hardware. It schedules and controls the use of the processor, store and peripherals, and services interrupt demands. Handles demands from user *activities* and commands from the Supervisor, returning error reports if necessary.

Mainframe Support Environment

A set of programs issued by ICL for use on 1900 and DME/1900, or VME/B mainframes, that enables the user to initialise, use and maintain a 7900 operating system (*SOS* or *XOS*)

Master Source File

Contains the source of the ICL *COS* used to generate all versions of the operating system base by the *System Generation* process.

MCDI

Multi-channel Device Interface.

MEDIT

A program in the Program Development Environment (*PDE*) that edits *COS* source modules.

ML1

A program in the *SYSGEN* suite that expands macros to produce *NAL* source code.

MPOE

Multi-programmed Overlaid Executive, a 1900 Series operating system.

MPX

Multiplexer. See CMX.

MSE

See *Mainframe Support Environment*

MSHD

Medium Speed Half Duplex, a transmission line mode, available to special order on a 7900 system.

MSS

Medium Speed Scanner, which handles up to 16 communications channels at up to 9600 bps (synchronous).

MTP

Modular Terminal Processor (7500 Range).

NAL

New Assembly Language. The assembly language of the 7900 Series computers.

NCP

Network Configuration Process. A mainframe program responsible for configuring a *SOS*.

ND TKEY

Non-Dedicated Turnkey. An ICL program which can optionally form part of an *XOS*, and permits the definition of the communications network to be modified at *run-time*.

Normal State

All user programs operate in Normal State in which interrupts can be accepted and in which there is restricted access to hardware *resources*. It is distinguished from *special state* which has privileged access. When the processor is in *Special State* interrupts can not be accepted.

NPU

This is the standard abbreviation for the *DCP* protocol term Normal Processing Unit.

NRL

New Relocatable Language, a version of *NAL*.

Operating Interface

The set of rules governing human intervention in the management of the operating system and the resources in its charge. Communication with Executive is carried out via the *operator interface* which is a part of Executive.

Operator Interface

Is the part of Executive which handles communications between the system operator, interactive and batch user, and Executive.

Page

The unit of segmentation of store and of transfer from disc. A page consists of 256 consecutive words (512 bytes).

Partition

Contiguous areas of space on 7900 disc within defined bounds. Each partition is independent of other partitions and will have its own associated access restrictions.

Password

This is associated with the *principal* number and used in validation of requests by *principals* to gain access to *resources*. Usually kept secret to maintain greater security.

PCOPY

A program in the Program Development Environment (*PDE*) that copies a program file from one disc *partition* to another.

PDE

Program Development Environment, software provided by ICL for use on a 7906 to develop user-written communication programs.

PDOS

Program Development Operating System, used to control the programs in the *Program Development Environment*.

PDUMP

A program in the Program Development Environment (*PDE*) that copies disc *partitions* or *volumes*.

Physical Medium Identifier (PMI)

This identifies a device type by a two character descriptor (externally) or a binary value (internally). Used in conjunction with a *PPI* to identify fully an actual peripheral.

Physical Peripheral Identifier (PPI)

This identifies an actual occurrence of a device type defined by a *PMI*. It is a number in the range 1 to 65535. Values up to 255 identify a physical device; values above 255 define a *Logical Volume Number* of an exchangeable medium.

Physical Volume Identifier (PVI)

See *Volume*.

PM

Post Mortem (state of the 7900 system prior to a dump).

PMI

See *Physical Medium Identifier*.

PPI

See *Physical Peripheral Identifier*.

Principal

A registered user of the system. Made known to Executive by the *system principal* and allocated a principal number which is used when logging in. Subsequent work for the logging-in user is associated with that principal number which is used for validation purposes. See also *System Principal* and *User*.

Principal 0

Otherwise known as the common principal. Used to make *resources* freely available to all *principals*.

Program Development Environment

See *PDE*.

Program File

Where programs are held in E4SLB (E4 Systems Loader Binary).

The *file* may reside on any device capable of being used for loading (specified at *system generation*).

Programming Interface

The set of instructions which may be obeyed by an activity. The instructions are written by using the languages NAL or Coral 66.

Program Mode

This is the name used to describe inter-processor communication using 7900 COS. In essence, Program Mode operation is communication between application programs in mutually remote processors.

Program Name

A name of up to six characters from the graphical set of ISO-7-UK contained on the *program file*. It is used to identify the program when loading and may be used subsequently by a user to initiate the loaded program into *jobs*.

Program Specification

This consists of a *program name* and a *segment specification*. It must be specified when loading a program.

Protection System

The system by which Executive ensures that *jobs* can use and manipulate *resources* only in accordance with defined access rights. Basically, access to resources is gained by jobs. This gives rise to the concept of *spheres* encompassing this access, access itself being governed by the *principal* number and *password* which are used to specify the access available to the resource when it was created.

PSN or PSTN

The Public Switched (Telephone) Network.

PTT

Post, Telephone and Telegraph authority (the Post Office in the UK).

PVDU

Pseudo Visual Display Unit.

PVI

Physical Volume Identifier. See *Volume*.

QLSA

Queueing Line Sharing Adaptor, a hardware unit that allows several 7181 videos to share one telephone line.

RC

This is the standard abbreviation for *remote concentrator*.

Real Time Clock

This is a high-precision clock providing general purpose timing facilities, which can be monitored by a processor and also generate an interrupt after a variable time interval set by program.

Record

The logical unit of a *file*. Physically, records must be wholly contained within the blocks of transfer of files so that only complete records are moved during physical transfers of blocks.

Re-entrant Program

A program written so that it can be used a number of times with concurrent execution. Distinguished from *re-usable programs* which must terminate before they can be re-initiated.

Remote Concentrator

This term is used to describe a communications processor connected to a mainframe communications link (for example, via the PTT network).

Resource

Facilities which may be required to aid the execution of a *job*. They may be created, deleted and modified in the execution of a job, and they are owned by a particular *principal* (some system resources are owned by the *system principal*). Each resource has access restrictions associated with it and these are defined by the owning principal.

Re-usable Program

This is any program which is only capable of one initiation at a time, unlike *re-entrant programs* which can have many concurrent initiations.

RJE

Remote Job Entry, the entry of data from a remote device.

ROM

Read Only Memory, write-protected store containing a bootstrap routine.

RTS

Ready to Send.

Run-Time

The time during which a *job* is active, i.e. between its initiation and its termination.

SDFS

Standard Disc Filing System, a system program that provides disc file management facilities to *application programs*.

Segment

A Segment consists of a number of consecutive 256 word *pages* of store.

Segment Specification

A list of the sizes of *segments* of a program to be loaded. The sizes are determined by the characteristics of the program and their order of presentation is that of the index number used when the program was written.

SETUP

A program in the Program Development Environment (*PDE*) that prepares an empty disc *partition* for use by *SDFS*.

SHUFL

A program in the *SYSGEN* suite that resolves the data structures arising from *system generation*, and produces two *NAL* source files.

SI

This is the standard abbreviation for *System Intelligence*.

SIZER

A program in the *SYSGEN* suite that reports the size of a file to the monitor file. SIZER can also truncate the file beyond the last used page if required.

SOM

This is the standard abbreviation for Start Of Message.

SOS

This is the standard abbreviation for the ICL *Standard Operating System*.

Special State

Every time the processor accepts an interrupt, either hardware or software, it switches automatically into special state. In that state it cannot process further interrupts, although the interrupts may be queued on the processor interface. See also *Normal State* and *Suspend State*.

Specific Terminal Identifier

The Specific Terminal Identifier is the unique name by which a terminal device is known in a network. The values are assigned to terminal devices at system generation time and are installation dependent. It is constructed by concatenating two values; the terminal group identifier and the number of the device on the terminal group. The terminal group identifier is a unique number within the communications network.

SPHDUMP

A program in the Program Development Environment (*PDE*) that produces a dump of the user's *sphere*.

Sphere

A sphere can be regarded as the internal representation of a *job*. It comprises a collection of *activities* and the access rights the job has to *resources*. It also has an associated *error mask* determining what happens when its activities commit certain types of errors. Optionally includes an *error collector*.

Standard Operating System

An ICL operating system for use with a 7904 or 7906 as a front end processor.

Station Level

This is the term used to describe the control software responsible for handling messages to a particular type of device over a single physical link, that is, multiplexing.

Steering File

A string of parameters (at least one) input to a program to control its execution. The same term is used for the set of commands that control the *Network Configuration Process*, and for the set of commands that control the *SYSGEN* process, even though the sets differ.

STI

This is the standard abbreviation for *Specific Terminal Identifier*.

Supervisor

Supervisor is the part of Executive responsible for those operations involving decisions on protection and the granting of access rights. It services requests both from user *activities* and the *operator interface*, and is also responsible for system start-up and shut-down.

Suspend State

A status register for each *activity* contains information on the *activity's* suspend state; that is, whether the *activity* is suspended or not and the reason it is suspended (for example waiting for an I/O transfer or queuing for a *lock*). See also *Normal State* and *Special State*.

SYN

Synchronising characters sent to establish and maintain timing on telephone lines.

SYNC

Synchronous method of communications.

SYSCODE

A program in the *SYSGEN* suite that selects code from an ICL master file, using directives from a *steering file*.

SYSFAST

A program in the Program Development Environment (*PDE*) that writes an object program or macro to disc in a loadable format.

SYSGEN

The SYStem GENeration suite of programs. See *System Generation*.

SYSPAR

A program in the Program Development Environment (*PDE*) that can display and change the *SYSFAST* parameters on a disc file.

System Control

The control of the hardware resources which is exercised by the *system operator*.

System Generation

The process by which the base operating system is tailored to the user's application. It is driven by the selection of hardware and software options, and results in a source listing, a store map and a binary version of the tailored operating system base, (see *SYSGEN*).

System Operator

The system operator has overall control of the 7900 system. His responsibilities range from the control of the hardware to reloading and running the batch stream.

System Principal

A special *principal* who is trusted and therefore permitted to carry out operations not available to normal *principals*. Carries out operations such as initialising discs and altering principal/password lists.

SYSVAL

A program in the *SYSGEN* suite that validates and expands the user's *steering file* commands.

TCP

Terminal Control Program, that controls the operation of the ICL 7503.

TE

Terminal Executive, that controls the operation of the ICL 7501 or 7502.

TLSA

Transparent Line Sharing Adaptor, a hardware unit that allows several 7181 videos to share one telephone line.

Trusted

Some *activities* have a special privilege bit which is used to enable *resources* to be acquired with certain checks on access rights being omitted. Such *activities* are termed trusted.

TTY

TTY is the abbreviation for teletypewriter. It may be the Executive Console TTY or a user (programmer) terminal for interactive program development.

User

A user is a person who uses the system either in an *interactive* or *batch mode*. This is different from a *principal* because any number of users may work for the same *principal* and each user may work for a number of *principals*.

VDU

Visual Display Unit, video terminal.

Virtual Store

A program development system including a 7980/07 or a 7980/04 *FEDS* can in some circumstances include the swapping (dynamic relocation of *segments*) capability. This area contains fixed places for all mobile *segments* currently loaded into the system.

VOLMAN

A program in the Program Development Environment (*PDE*) that sets up and controls disc *partitions*.

Volume

There are two terms relating to volume; they are physical volume and logical volume.

The term physical volume refers to the physical unit of exchangeable medium; that is, the actual piece of hardware, the exchangeable disc pack, the paper tape reader. The physical volume identifier (PVI) is a number written on, or in some way appended to, the physical volume for purposes of identification.

The term logical volume refers to the contents of the physical volume. A logical volume number (LVN) is a number recorded with and identifying the contents. It is the number by which programmers refer to the contents. The LVN is a *physical peripheral identifier* in the range 256 to 65535.

VT

Video Terminal.

Workspace Segment

When a program is initiated it is necessary to nominate a *segment* called a workspace segment into which Executive may preset certain information. This includes literal data or a *steering file* nominated on initiation.

XBAD

There are two different programs with the same name XBAD as follows:

1 A program in the 1900 *Mainframe Support Environment* that analyses and prints the contents of a 7900 store dump held in a mainframe disc file

2 A program in the Program Development Environment (*PDE*) that analyses and prints a dump of a 7900 operating system held in a 7900 disc partition

XBAE

A program in the 1900 *Mainframe Support Environment* that is used to patch a 7900 operating system base.

XBAF

A program in the 1900 *Mainframe Support Environment* that assembles a 7900 operating system base.

XBAG

A program in the 1900 *Mainframe Support Environment* that performs the Network Configuring Process (*NCP*) on a 7900 *Standard Operating System*.

XBAH

A program in the 1900 *Mainframe Support Environment* that loads and dumps a 7900 operating system between the host mainframe and the 7900 front end processor.

XBM

Extended Basic Mode. An ICL line protocol also called ICLC-02.

XOS

Extended Operating System, an operating system for the 7906, which can include user-written *application programs* or the ICL program *ND TKEY*.